The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-9 June 2022, Oslo, Norway

DNS AND LES ON UNSTRUCTURED GRIDS: PLAYING WITH MATRICES TO PRESERVE SYMMETRIES USING A MINIMAL SET OF ALGEBRAIC KERNELS

F.X.Trias[1], X.Álvarez-Farré[1], A.Alsalti-Baldellou[1], A.Gorobets[2] and A.Oliva[1]

[1] Heat and Mass Transfer Technological Center, Technical University of Catalonia, C/Colom 11, 08222 Terrassa (Barcelona), francesc.xavier.trias@upc.edu

[2] Keldysh Institute of Applied Mathematics, 4A, Miusskaya Sq., Moscow 125047, Russia

Key words: Time-integration, eigenbounds, DNS, LES, symmetry-preserving, unstructured

Abstract. The essence of turbulence are the smallest scales of motion. They result from a subtle balance between two differential operators differing in symmetry: the convective operator is skew-symmetric, whereas the diffusive is symmetric and negative-definite. On the other hand, accuracy and stability need to be reconciled for numerical simulations of turbulent flows in complex configurations. With this in mind, a fully-conservative discretization method for collocated unstructured grids was proposed [Trias *et al.*, J.Comp.Phys. 258, 246-267, 2014]: it preserves the symmetries of the differential operators and it has shown to be a very suitable approach for DNS and LES. On the other hand, an efficient cross-platform portability is nowadays one of the greatest challenges for CFD codes. In this regard, our *leitmotiv* reads: *relying on a minimal set of (algebraic) kernels is crucial for code portability and maintenance!* In this context, this work focuses on the computation of eigenbounds for the above-mentioned convection and diffusion matrices which are needed to determine the time-step *à la* CFL. A new inexpensive method that allows this, without explicitly constructing these time-dependent matrices is proposed and tested. It only requires a sparse-matrix vector product where only the vector changes on time. Hence, apart from being significantly more efficient than the standard CFL condition, cross-platform portability is straightforward.

1 INTRODUCTION

In the last decades, CFD has become a standard design tool in many fields such as automotive, aeronautical and wind power industries. The driven force behind this, is the development of numerical techniques in conjunction with the progress of high performance computing (HPC) systems. However, nowadays we can say that its legacy from the 90-2000s is hindering its progress. The reasons are two-fold: (i) codes designed for CPUs cannot be easily ported and optimized to new architectures (GPUs, ARM...) and (ii) legacy numerical methods chosen to solve (quasi)steady problems using RANS models are not appropriate for more accurate (and more expensive) techniques such as LES or DNS. This work aims to interlace these two pillars with the final goal to enable LES/DNS of industrial applications to be efficiently carried out on modern HPC systems while keeping codes easy to port and maintain. In this regard, a fully-conservative discretization for collocated unstructured grids was proposed [1]. It exactly preserves the symmetries of the underlying differential operators and is based on only five discrete operators (*i.e.* matrices): the cell-centered and staggered control volumes (diagonal matrices), Ω_c and Ω_s, the face normal vectors, N_s, the cell-to-face interpolation, $\Pi_{c\rightarrow s}$ and the cell-to-face divergence operator, M. Therefore, it constitutes a robust approach that can be easily implemented in already existing codes such as OpenFOAM® [2]. The benefits of symmetry-preserving discretizations for DNS/LES have been extensively reported in many publications in the last decades [2–9]. Hereafter, we follow the same operator-based notation as in Ref. [1].

On the other hand, for the sake of cross-platform portability and optimization, CFD algorithms must rely on a very reduced set of (algebraic) kernels [10–12] (*e.g.* sparse-matrix vector product, `SpMV`; dot product; linear combination of vectors) in order to keep a good balance between code portability and performance. This imposes restrictions and challenges that need to be addressed such as the inherent low arithmetic intensity of the `SpMV`, the reformulation of flux limiters [13] or the efficient computation of eigenbounds to determine the time-step, Δt. This work focuses in the latter problem and aims to answer the following research question: *Can we avoid to explicitly construct both convective,* $\mathsf{C}(\boldsymbol{u}_s)$*, and diffusive,* $\mathsf{D}(\boldsymbol{\alpha}_s)$*, matrices while still being able to compute proper eigenbounds in an inexpensive manner?* Read on...

2 RETHINKING CFL CONDITION: EIGENBOUNDS OF CONVECTIVE AND DIFFUSIVE OPERATORS

2.1 Gershgorin-based linear stability analysis

Explicit (and semi-explicit) time-integration schemes impose severe restrictions on the time-step, Δt, due to the fact that eigenvalues of the amplification matrix must lie inside the stability region of the time-integration method. Namely, linearizing (if needed) the dynamical system (*e.g.* momentum equation on a 3D collocated mesh with n volumes and m faces) leads to

$$\frac{d\boldsymbol{u}_c}{dt} = \mathsf{R}_{\boldsymbol{u}}\boldsymbol{u}_c \qquad \text{where} \quad \mathsf{R}_{\boldsymbol{u}} = (\mathsf{I}_3 \otimes \mathsf{R}) \in \mathbb{R}^{3n\times 3n}, \tag{1}$$

where the matrix $\mathsf{R} \equiv -\Omega_c^{-1}(\mathsf{C}(\boldsymbol{u}_s) + \mathsf{D}) \in \mathbb{R}^{n\times n}$ accounts for the effects of convection and diffusion and $\boldsymbol{u}_c \in (\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)^T \in \mathbb{R}^{3n}$. Then, different time-integration schemes lead to different stability regions [14]. The simplest example thereof in the first-order Euler explicit scheme:

$$\frac{\boldsymbol{u}_c^{n+1} - \boldsymbol{u}_c^{n}}{\Delta t} = \mathsf{R}_{\boldsymbol{u}}\boldsymbol{u}_s^n \quad \Longrightarrow \quad \boldsymbol{u}_c^{n+1} = (\mathsf{I}_3 \otimes \mathsf{A})\,\boldsymbol{u}_c^n \;\; \text{where} \;\; \mathsf{A} \equiv (\mathsf{I} + \Delta t \mathsf{R}). \tag{2}$$

The A-stability is guaranteed if the spectral radius of the amplification matrix, A, is smaller than one, *i.e.* $\rho(\mathsf{A}) < 1$. This leads to the stability region in terms of the eigenvalues of $\tilde{\mathsf{R}} \equiv \Delta t \mathsf{R}$ shown in Figure 1 (left). Similar analysis can be done for other temporal schemes [14]. An example thereof is shown in the same figure for the one-parameter second-order explicit method

$$\frac{\boldsymbol{u}_c^{n+\kappa+1/2} - \boldsymbol{u}_c^{n+\kappa-1/2}}{\Delta t} = \mathsf{R}_{\boldsymbol{u}}\boldsymbol{u}_c^{n+\kappa}, \tag{3}$$

where the off-step velocities are given by

$$\boldsymbol{u}_c^{n+\kappa+1/2} = (\kappa+1/2)\boldsymbol{u}_c^{n+1} - (\kappa-1/2)\boldsymbol{u}_c^{n} \qquad \text{and} \qquad \boldsymbol{u}_c^{n+\kappa} = (1+\kappa)\boldsymbol{u}_c^{n} - \kappa\boldsymbol{u}_c^{n-1}. \tag{4}$$

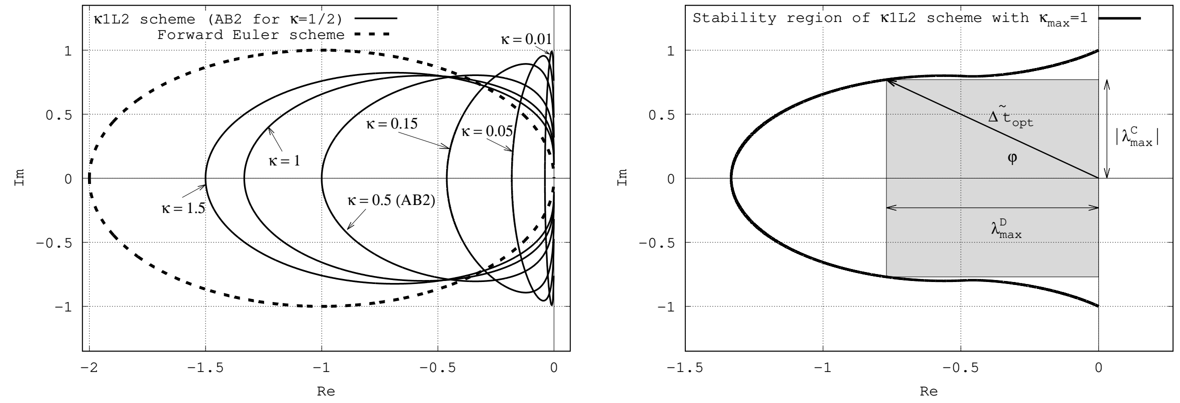

Figure 1: Stability region of the first-order forward Euler scheme (Eq. 2) together with the family of κ-dependent second-order κ1L2 time-integration scheme (Eqs. 3 and 4) (left) and their envelope (right).

This time-integration scheme named κ1L2 can be viewed as a generalization of the classical second-order Adams-Bashforth (AB2) scheme ($\kappa = 1/2$). This was used in Refs. [3, 15] for DNS of incompressible flows keeping the parameter κ constant during the simulation. Then, in Ref. [16], a self-adaptive strategy was proposed: the parameter κ is being re-computed to adapt the linear stability domain to the instantaneous flow conditions in order to maximize Δt. The idea of the method is depicted in Figure 1 (right). Hence, at the end, this or any other method necessarily relies on bounding the eigenvalues of the dynamical system, *i.e.* in our case finding eigenbounds of the matrix R given in Eq.(1). In the original work [16], this was done by applying the Gershgorin circle theorem to $\Omega_c^{-1}\mathsf{C}(\boldsymbol{u}_s)$ and $\Omega_c^{-1}\mathsf{D}$ together with the Bendixson theorem.

Theorem 1 (Bendixson [17])**.** *Given two square matrices of equal size,* X *and* Y*, one with real-valued eigenvalues,* $\lambda^{\mathsf{X}} \in \mathbb{R}$*, and the other with imaginary ones,* $\lambda^{\mathsf{Y}} \in i\mathbb{R}$*, then every eigenvalue of the sum,* $\mathsf{X}+\mathsf{Y}$*, is contained in the rectangle*

$$\lambda^{\mathsf{X}}_{\min} \le \mathsf{Re}(\lambda^{\mathsf{X}+\mathsf{Y}}) \le \lambda^{\mathsf{X}}_{\max} \qquad \mathsf{Im}(\lambda^{\mathsf{Y}}_{\min}) \le \mathsf{Im}(\lambda^{\mathsf{X}+\mathsf{Y}}) \le \mathsf{Im}(\lambda^{\mathsf{Y}}_{\max}). \tag{5}$$

This can be easily applied to matrix $\mathsf{R} = -\Omega_c^{-1}\mathsf{C}(\boldsymbol{u}_s) + \Omega_c^{-1}\mathsf{D}$ recalling that $\mathsf{C}(\boldsymbol{u}_s) = -\mathsf{C}^T(\boldsymbol{u}_s)$, *i.e.* $\lambda^{\mathsf{C}} \in i\mathbb{R}$, and $\mathsf{D} = \mathsf{D}^T$ negative semi-definite, *i.e.* $\lambda^{\mathsf{D}} \in \mathbb{R}^-_0$. At this point, there are a couple of technical issues that worth mentioning. Although the (skew-)symmetry is lost when matrices $\mathsf{C}(\boldsymbol{u}_s)$ and D are left-multiplied by Ω_c^{-1}, their eigenvalues are still imaginary and real-valued, respectively. They actually have the same spectrum as the (skew-)symmetric matrices $\Omega_c^{-1/2}\mathsf{C}(\boldsymbol{u}_s)\Omega_c^{-1/2}$ and $\Omega_c^{-1/2}\mathsf{D}\Omega_c^{-1/2}$,

$$\Omega_c^{-1}\mathsf{D}\boldsymbol{v} = \lambda^{\mathsf{D}}\boldsymbol{v} \implies \Omega_c^{1/2}(\Omega_c^{-1}\mathsf{D})\Omega_c^{-1/2}\Omega_c^{1/2}\boldsymbol{v} = \lambda^{\mathsf{D}}\Omega_c^{1/2}\boldsymbol{v} \implies (\Omega_c^{-1/2}\mathsf{D}\Omega_c^{-1/2})\boldsymbol{w} = \lambda^{\mathsf{D}}\boldsymbol{w}, \tag{6}$$

where $\boldsymbol{w} = \Omega_c^{1/2}\boldsymbol{v}$. Notice that the matrix Ω_c has strictly positive diagonal elements. This method to bound the eigenvalues of R was originally proposed and referred as *EigenCD* in Ref. [16]. Later it was successfully used for a large variety of DNS and LES simulations on both structured and unstructured meshes (see Refs. [4–7, 9, 16] among others).

2.2 CFL condition: brief historical review

Nevertheless, this is not the standard way to bound the eigenvalues of R. In the CFD literature, and in virtually all popular CFD packages, stability constraints for Δt are usually expressed in terms of the so-called *CFL* condition originally proposed in the seminal paper [18] by R.*C*ourant, K.*F*riedrichs, and H.*L*ewy in 1928! They derived the following stability condition

$$C = \frac{u \Delta t}{\Delta x} < C_{\max}, \tag{7}$$

for a 1D transport equation

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = 0, \tag{8}$$

discretized in a uniform mesh with spacing equal to Δx where u is the advection velocity. The intuitive idea or "physical interpretation" of this formulae can be found, for instance, in the OpenFOAM® documentation as *"a measure of the rate at which information is transported under the influence of a flux field"* [19]. This, or very similar formulae can be found in NEK5000 [20], COMSOL® [21] or Basilisk [22] codes, among many others. An alternative definition is used in ANSYS-Fluent [23]

$$CFL = \frac{\Delta t \sum_{faces} \lambda_f^{\max} A_f}{2V}, \tag{9}$$

where A_f are the face areas, V is the cell volume and $\lambda_f^{\max}$ is the maximum of the local eigenvalues. For incompressible (also compressible at low speed) flows, $\lambda_f = U_f$ (here, U_f is the face velocity); therefore, this CFL condition becomes identical to the definition used in OpenFOAM® [19], SU2 code [24] or Code Saturne [25] and slightly different than the definition used in the DLR-TAU code (see Eq.18 in [26]). Nevertheless, the original idea of the formula given in Eq.(9) is not completely clear (at least, not for the authors) and according to [27], it goes back to Eq.(22) in Ref. [28] where the following definition of the CFL condition is given

$$CFL = \frac{\Delta t \lambda_{max}}{V}, \tag{10}$$

where λ_{max} is the maximum eigenvalue of the system given by $|v|$ for incompressible (also compressible at low speed) flows. It must be noted that a multiplication by the face area, A_f, is missing in Eq.(10). Moreover, no summation by faces is specified here. Going back to previous works by the same authors, we find the same definition in Ref. [29] without specifying how the eigenvalues are being computed. Moreover, in Ref. [30] (see Eq.16) they used the following formula for bounding the Δt,

$$\Delta t = \min \left(\frac{CFL \Delta x}{u' + c'}, \frac{\sigma \Delta x^2}{\nu} \right), \tag{11}$$

where σ is referred as von Neumann number, u' and c' are respectively the velocity and the speed of sound for the non-preconditioned system and Δx is defined as the intercell length scale over which diffusion occurs. Furthermore, in Ref. [31] (Eq. 4) we find the following formula

$$CFL = \Delta t \lambda_{max}(D), \tag{12}$$

where $\lambda_{max}(D)$ is the maximum eigenvalue of the chemical Jacobian. The time integration method is a first-order implicit Euler scheme and the condition (12) is used to keep the system positive definite, *i.e.* $(I-\Delta t D)$ is a positive definite matrix. The eigenvalue of D were determined numerically using the LAPACK library [32]. In any case, CFL condition became soon very popular among all the CFD community. To celebrate the article's 40th anniversary, in 1967 the IBM Journal published a special issue, that included the English translation of the original paper [33]. In 2010, the meeting "CFL-condition, 80 years gone" was held in Rio de Janeiro [34].

2.3 Two sides of the same coin

The CFL condition given in Eq.(7) can be easily related to the above explained constraints related with the eigenvalues of the matrix R given in Eq.(1). Let's consider a 1D uniformly spaced mesh with constant advective velocity, u. In this case, the convective and diffusive terms in the NS equations simplify to

$$\frac{\partial \phi}{\partial t} = -u\frac{\partial \phi}{\partial x} + \nu \frac{\partial^2 \phi}{\partial x^2}. \tag{13}$$

Then, a second-order semi-discrete finite-difference (also finite-volume) discretization of Eq.(13) leads to

$$\frac{\partial \phi_i}{\partial t} = -u\frac{\phi_{i+1}-\phi_{i-1}}{2\Delta x} + \nu \frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}. \tag{14}$$

This can be re-arranged in a matrix-vector form as follows

$$\frac{\partial \boldsymbol{\phi}_h}{\partial t} = \begin{bmatrix} 0 & \ddots & & & \\ \ddots & 0 & \ddots & & \\ & \frac{u}{2\Delta x} & 0 & -\frac{u}{2\Delta x} & \\ & & \ddots & 0 & \ddots \\ & & & \ddots & 0 \end{bmatrix} \boldsymbol{\phi}_h + \begin{bmatrix} 0 & \ddots & & & \\ \ddots & 0 & \ddots & & \\ & \frac{\nu}{\Delta x^2} & -\frac{2\nu}{\Delta x^2} & \frac{\nu}{\Delta x^2} & \\ & & \ddots & 0 & \ddots \\ & & & \ddots & 0 \end{bmatrix} \boldsymbol{\phi}_h, \tag{15}$$

where $\boldsymbol{\phi}_h = (\phi_1, \cdots, \phi_n)^T \in \mathbb{R}^n$ is a column vector containing all the components of the scalar field $\boldsymbol{\phi}$. Hence, eigenvalues of the convective and diffusive part can be bounded using the Gershgorin circle theorem as follows

$$|\lambda^{\mathsf{C}}| \le \frac{u}{\Delta x} \qquad |\lambda^{\mathsf{D}}| \le \frac{4\nu}{\Delta x^2}, \tag{16}$$

which leads to the classical CFL definition proposed almost a century ago [18].

At this point, we expect that it becomes clear that it is probably more appropriate (and more accurate) to get rid of generalizations of the classical CFL definition given in Eq.(7) for general cases (*i.e.* multi-dimensional, non-uniform, non-constant velocity, unstructured meshes...). Instead, the Gershgorin circle theorem can be applied assuming that the coefficients of the discrete convective, $\mathsf{C}(\boldsymbol{u}_s)$, and diffusive, D, operators are available. This was the main idea of the paper published one decade ago [1]. Hence, at this stage, we are replacing inexact approximations combined with heuristic, sometimes even trial-and-error, approaches to prescribe the so-called

CFL value. In general, for the sake of robustness, these approaches tend to underestimate Δt leading to an increase in the overall computational cost of the simulations. In practice, CPU cost reductions up to more than 4 were measured for unstructured grids [16].

3 A NEW EFFICIENT APPROACH TO COMPUTE EIGENBOUNDS OF CONVECTION AND DIFFUSION MATRICES AVOIDING ITS CONSTRUCTION

3.1 Deconstructing convection and diffusion matrices

The application of the Gershgorin circle theorem requires to explicitly evaluate the coefficients of the matrix/ces that arise from the spatial discretization. However, in many ocasions this may be impractical (or inneficient). Namely, let us consider the convective operator

$$\mathsf{C}\left(\boldsymbol{u}_s\right) \equiv \mathsf{M}\mathsf{U}_s\Pi_{c\rightarrow s} \in \mathbb{R}^{n\times n}, \tag{17}$$

$\mathsf{M} \in \mathbb{R}^{n\times m}$ is the face-to-cell divergence operator, $\Pi_{c\rightarrow s} \in \mathbb{R}^{m\times n}$ is cell-to-face interpolation and $\mathsf{U}_s = \mathrm{diag}(\boldsymbol{u}_s) \in \mathbb{R}^{m\times m}$ is a diagonal matrix that contains the face velocities, $\boldsymbol{u}_s \in \mathbb{R}^m$. It is obvious that the coeffients of the matrix $\mathsf{C}\left(\boldsymbol{u}_s\right)$ are changing every time-step. In a code based in mesh-loop functions, these coefficients can be recomputed "on the fly" and then call some specific function to compute eigenbounds based on those coefficients of the matrix. This implies the construction of the matrix every time-step (more RAM memory requirements), makes the code more complex and difficult to maintain, hindering its efficient cross-platform portability.

A similar problem exists for the diffusive term with non-constant diffusivity

$$\mathsf{D}(\boldsymbol{\alpha}_s) \equiv \mathsf{M}\Lambda_s\mathsf{G} \in \mathbb{R}^{n\times n}, \tag{18}$$

where $\Lambda_s = \mathrm{diag}(\boldsymbol{\alpha}_s) \in \mathbb{R}^{m\times m}$ is a diagonal matrix that contains the values of the diffusivities defined at the faces, $\boldsymbol{\alpha}_s \in \mathbb{R}^m$. Notice that this is also (very) relevant for eddy-viscosity turbulence models. For details about the discretization the reader is referred to [1].

At this point, we aim to answer the following research question: *can we avoid to explicitly construct both convective, $\mathsf{C}\left(\boldsymbol{u}_s\right)$, and diffusive, $\mathsf{D}(\boldsymbol{\alpha}_s)$, matrices while still being able to compute proper eigenbounds in an inexpensive manner?* To do so, we firstly write the divergence operator, M, in terms of the cell-to-face, $\mathsf{T}_{cs} \in \mathbb{R}^{m\times n}$ and face-to-cell, $\mathsf{T}_{sc} \in \mathbb{R}^{n\times m}$, incidence matrices

$$\mathsf{M} \equiv \mathsf{T}_{sc}\mathsf{A}_s \in \mathbb{R}^{n\times m}, \tag{19}$$

where $\mathsf{A}_s \in \mathbb{R}^{m\times m}$ is a diagonal matrix containing the face surfaces. Moreover, recalling the duality between the divergence and the gradient operators [1]

$$\mathsf{M} = -(\Omega_s\mathsf{G})^T \quad \Longrightarrow \quad \mathsf{G} = -\Omega_s^{-1}\mathsf{M}^T \in \mathbb{R}^{m\times n}, \tag{20}$$

together with the relation $\mathsf{T}_{sc} = \mathsf{T}_{cs}^T$ leads to

$$\mathsf{G} \equiv -\Omega_s^{-1}\mathsf{A}_s\mathsf{T}_{sc}^T = -\Delta_s^{-1}\mathsf{T}_{cs}, \tag{21}$$

where $\Delta_s \in \mathbb{R}^{m\times m}$ is a diagonal matrix containing the projected distances, $\delta n_f = |\boldsymbol{n}_f \cdot \overrightarrow{c1c2}|$, between the centroids, $c1$ and $c2$, of the two cells adjacent to a face, f (see Figure 2). Plugging all this into the definition of the diffusive operator (18) leads to

$$\mathsf{D}(\boldsymbol{\alpha}_s) = -\mathsf{T}_{sc}\mathsf{A}_s\Lambda_s\Delta_s^{-1}\mathsf{T}_{cs} = -\mathsf{T}_{sc}\tilde{\Lambda}_s\mathsf{T}_{cs} = -\mathsf{T}_{cs}^T\tilde{\Lambda}_s\mathsf{T}_{cs}, \tag{22}$$

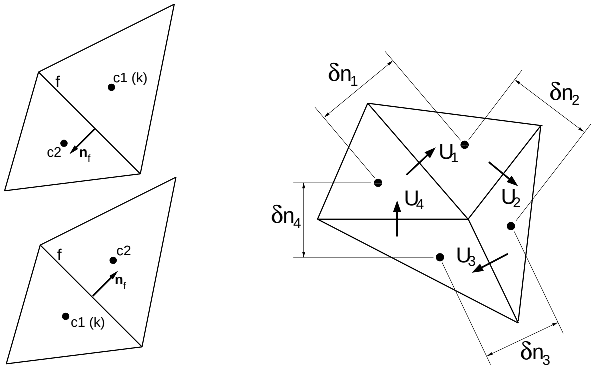

Figure 2: Left: face normal and neighbor labeling criterion. Right: definition of the volumes of the face-normal velocity cell.

where the diagonal matrix $\tilde{\mathsf{\Lambda}}_s = \mathsf{A}_s \mathsf{\Lambda}_s \mathsf{\Delta}_s^{-1} \in \mathbb{R}^{m \times m}$ has strictly positive diagonal coefficients. Therefore, the diffusive operator is symmetric and negative semidefinite likewise the continuous Laplacian, ∇^2.

Similarly, the convective term given in Eq.(17) can be written as follows

$$\mathsf{C}\left(\boldsymbol{u}_s\right) = \mathsf{T}_{sc} \mathsf{U}_s \mathsf{A}_s \Pi_{c \rightarrow s}, \tag{23}$$

where the cell-to-face interpolation, $\Pi_{c \rightarrow s}$, defines the numerical scheme we are using. For instance, taking

$$\Pi_{c \rightarrow s} = \frac{1}{2} |\mathsf{T}_{cs}|, \tag{24}$$

leads to a skew-symmetric matrix, *i.e.* $\mathsf{C}\left(\boldsymbol{u}_s\right) = -\mathsf{C}^T\left(\boldsymbol{u}_s\right)$; *i.e.* a second-order symmetry-preserving discretization [1, 3]. In summary, convective and diffusive operators read

$$\mathsf{D}(\boldsymbol{\alpha}_s) = -\mathsf{T}_{cs}^T \tilde{\mathsf{\Lambda}}_s \, \mathsf{T}_{cs} \qquad \text{where } \tilde{\mathsf{\Lambda}}_s \text{ is a diagonal matrix with } [\tilde{\mathsf{\Lambda}}_s]_{ii} > 0, \tag{25}$$

$$2\mathsf{C}\left(\boldsymbol{u}_s\right) = \ \mathsf{T}_{cs}^T \mathsf{F}_s |\mathsf{T}_{cs}| \qquad \text{where } \mathsf{F}_s \equiv \mathsf{A}_s \mathsf{U}_s \text{ is diagonal and } \mathrm{diag}(\mathsf{F}_s) \in ker(\mathsf{T}_{cs}^T). \tag{26}$$

where, in general, both $\tilde{\mathsf{\Lambda}}_s$ and F_s change on time.

3.2 Eigenbounds for the diffusion matrix

The idea at this point is to construct other matrices that have the same spectrum (except for the zero-valued eigenvalues). To do so, we will use the following well-known property:

Theorem 2. *Let* $\mathsf{A} \in \mathbb{R}^{n \times m}$ *and* $\mathsf{B} \in \mathbb{R}^{m \times n}$ *be two rectangular matrices with the appropriate dimensions and* $m \geq n$*, then the square matrices* $\mathsf{AB} \in \mathbb{R}^{n \times n}$ *and* $\mathsf{A}^T \mathsf{B}^T \in \mathbb{R}^{m \times m}$ *have the same eigenvalues except for the zero-valued ones.*

Proof. A square matrix Q and its transpose, Q^T, have the same characteristic polynomial, *i.e.* $\det(\lambda \mathsf{I} - \mathsf{Q}) = \det(\lambda \mathsf{I} - \mathsf{Q}^T) = 0$; therefore, they also have the same spectrum. Then, both $\mathsf{A}^T \mathsf{B}^T$ and BA have the same spectrum

$$\mathsf{A}^T \mathsf{B}^T \boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i \quad \rightarrow \quad \mathsf{BA} \boldsymbol{z}_i = \lambda_i \boldsymbol{z}_i \qquad \forall i = 1, \ldots, m \tag{27}$$

Then, let $\lambda \neq 0$ be an eigenvalue of AB with an associated eigenvector $\boldsymbol{v}$,

$$\mathsf{AB}\boldsymbol{v} = \lambda \boldsymbol{v} \quad \rightarrow \quad \mathsf{BA}(\mathsf{B}\boldsymbol{v}) = \lambda(\mathsf{B}\boldsymbol{v}) \quad \rightarrow \quad \mathsf{BA}\boldsymbol{z} = \lambda \boldsymbol{z}. \tag{28}$$

Notice that $\mathsf{B}\boldsymbol{v} \neq 0$ since $\lambda \neq 0$. Hence, λ is a non-zero eigenvalue of BA and subsequently also an eigenvalue of $\mathsf{A}^T\mathsf{B}^T$. □

Therefore, a family of matrices with the same spectrum (except for the zero-valued eigenvalues) as those given in Eqs.(25) and (26) can be constructed using Theorem 2. Namely, for the diffusive term all matrices with the form $\tilde{\Lambda}_s^{\alpha}\mathsf{T}_{cs}\mathsf{T}_{cs}^T\tilde{\Lambda}_s^{1-\alpha}$ have the same spectrum as $\mathsf{T}_{cs}^T\tilde{\Lambda}_s\mathsf{T}_{cs}$. Hence, the following four matrices (the last three correspond to values of α equal to $1/2$, 0 and 1, respectively) have the same spectrum (except for the zero-valued eigenvalues)

$$\left\{ -\mathsf{T}_{cs}^T\tilde{\Lambda}_s\mathsf{T}_{cs}, -\mathsf{T}_{cs}\mathsf{T}_{cs}^T\tilde{\Lambda}_s, -\tilde{\Lambda}_s^{1/2}\mathsf{T}_{cs}\mathsf{T}_{cs}^T\tilde{\Lambda}_s^{1/2}, -\tilde{\Lambda}_s\mathsf{T}_{cs}\mathsf{T}_{cs}^T \right\}. \tag{29}$$

The advantatge of the new forms is that only matrix $-\mathsf{T}_{cs}\mathsf{T}_{cs}^T$ have to be computed (once) and stored. Note that this matrix has -2 in the diagonal and $+1$ in the non-zero off-diagonal elements. Then, to apply the Gershgorin circle theorem to find an upper bound (in absolute value) of the eigenvalues we can simply compute

$$\rho(\mathsf{D}(\boldsymbol{\alpha}_s)) = \rho(\mathsf{T}_{cs}\mathsf{T}_{cs}^T\tilde{\Lambda}_s) \leq \max\{|\mathsf{T}_{cs}\mathsf{T}_{cs}^T|\,\mathrm{diag}(\tilde{\Lambda}_s)\}, \tag{30}$$

$$\rho(\mathsf{D}(\boldsymbol{\alpha}_s)) = \rho(\tilde{\Lambda}_s^{1/2}\mathsf{T}_{cs}\mathsf{T}_{cs}^T\tilde{\Lambda}_s^{1/2}) \leq \max\{\mathrm{diag}(\tilde{\Lambda}_s^{1/2}) \circ |\mathsf{T}_{cs}\mathsf{T}_{cs}^T|\,\mathrm{diag}(\tilde{\Lambda}_s^{1/2})\}, \tag{31}$$

$$\rho(\mathsf{D}(\boldsymbol{\alpha}_s)) = \rho(\tilde{\Lambda}_s\mathsf{T}_{cs}\mathsf{T}_{cs}^T) \leq \max\{\mathrm{diag}(\tilde{\Lambda}_s) \circ |\mathsf{T}_{cs}\mathsf{T}_{cs}^T|\mathbf{1}_s\}, \tag{32}$$

where $\circ$ denotes the Hadamard product (element-wise product).

Remark 1. *In practice, we need estimations of the spectral radius of* $\Omega_c^{-1}\mathsf{D}(\boldsymbol{\alpha}_s)$ *and not* $\mathsf{D}(\boldsymbol{\alpha}_s)$*. This can be easily done by replacing* $|\mathsf{T}_{cs}\mathsf{T}_{cs}^T|$ *by* $|\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T|$ *in Eqs.(30), (31) and (32). Equivalent remark can be made for the forthcoming discussion about the convetive matrix,* $\mathsf{C}\,(\boldsymbol{u}_s)$*.*

3.3 Eigenbounds for the convection matrix

Convective term given in Eq.(26) can be treated in a similar manner. However, in this case, the diagonal matrix F_s (mass fluxes across the faces) can take both positive and negative values depending on the flow direction. Similar to Eq.(29), all these matrices have the same spectrum (except for the zero-valued eigenvalues)

$$\left\{ \mathsf{T}_{cs}^T\mathsf{F}_s|\mathsf{T}_{cs}|, \mathsf{T}_{cs}|\mathsf{T}_{cs}^T|\mathsf{F}_s, |\mathsf{F}_s|^{1/2}\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\mathsf{F}_s|^{-1/2}\mathsf{F}_s, |\mathsf{F}_s|^{-1/2}\mathsf{F}_s\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\mathsf{F}_s|, \mathsf{F}_s\mathsf{T}_{cs}|\mathsf{T}_{cs}^T| \right\}. \tag{33}$$

In the last four splittings, only the matrix $\mathsf{T}_{cs}|\mathsf{T}_{cs}^T|$ has to be pre-computed and stored. This matrix is skew-symmetric with ±1 in the non-zero off-diagonal elements. Then, the Gershgorin circle theorem can be applied as follows

$$2\rho(\mathsf{C}\,(\boldsymbol{u}_s)) = \rho(\mathsf{T}_{cs}|\mathsf{T}_{cs}^T|\mathsf{F}_s) \leq \max\{|\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\,|\,\mathrm{diag}(|\mathsf{F}_s|)\}, \tag{34a}$$

$$2\rho(\mathsf{C}\,(\boldsymbol{u}_s)) = \rho(|\mathsf{F}_s|^{\frac{1}{2}}\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\mathsf{F}_s|^{-\frac{1}{2}}\mathsf{F}_s) \leq \max\{\mathrm{diag}(|\mathsf{F}_s|^{\frac{1}{2}}) \circ |\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\,\mathrm{diag}(|\mathsf{F}_s|^{-\frac{1}{2}}\mathsf{F}_s)\}, \tag{34b}$$

$$2\rho(\mathsf{C}\,(\boldsymbol{u}_s)) = \rho(\mathsf{F}_s\mathsf{T}_{cs}|\mathsf{T}_{cs}^T|) \leq \max\{\mathrm{diag}(|\mathsf{F}_s|) \circ |\mathsf{T}_{cs}|\mathsf{T}_{cs}^T|\mathbf{1}_s\}, \tag{34c}$$

to find an upper bound of their eigenvalues (in this case they lie on the imaginary axis). However, in practical flows, none of these approaches is able to provide better (or, at least, similar estimates) as using the Gershgorin circle theorem to the matrix $\mathsf{C}\left(\boldsymbol{u}_s\right)$. A simple explanation for this is the following: matrix $|\mathsf{T}_{cs}|\mathsf{T}_{cs}^T||\mathsf{F}_s$ has more non-zero off-diagonal coefficients per row than matrix $\mathsf{T}_{cs}^T\mathsf{F}_s|\mathsf{T}_{cs}|$, *e.g.* for a structured Cartesian mesh in d-dimensions the former has $2(2d-1)$ whereas the latter has only $2d$. Therefore, more mass fluxes (in absolute value) are contributing to the calculation of the Gershgorin circle radii.

Theorem 3 (Perron-Frobenius theorem [35, 36])**.** *Given a real positive square matrix, i.e.* $\mathsf{A} \in \mathbb{R}^{n\times n}$ *and* $[\mathsf{A}]_{ij} > 0 \;\forall i,j$ *, it has a unique largest (in magnitude) real eigenvalue,* $r \in \mathbb{R}^+$*, with a corresponding eigenvector,* $\boldsymbol{v} \in \mathbb{R}^n$*, with strictly positive components, i.e.*

$$A\boldsymbol{v} = r\boldsymbol{v} \qquad \Longrightarrow \qquad |\lambda| < r \quad \textit{and} \quad \boldsymbol{v}_i > 0 \quad \forall i = 1, \cdots, n, \tag{35}$$

where λ *denotes any eigenvalue of* A *except* r*, and* r *is the so-called Perron-Frobenius eigenvalue.*

Theorem 4 (Wielandt's theorem [37])**.** *Given a matrix* $\mathsf{A} \in \mathbb{R}^{n\times n}$ *that satisfies the conditions of the Perron-Frobenius theorem (see Theorem 3) and a matrix* $\mathsf{C} \in \mathbb{R}^{n\times n}$ *such as*

$$|c_{ij}| \leq a_{ij} \qquad \forall i,j, \tag{36}$$

where $c_{ij} = [\mathsf{C}]_{ij}$ *and* $a_{ij} = [\mathsf{A}]_{ij}$*. Then, any eigenvalue* λ^{C} *of* C *satisfies the inequality* $|\lambda^{\mathsf{C}}| \leq r$ *where* r *is the Perron-Frobenius eigenvalue of* A*.*

Theorem 5 (Lemma 2 in Nikiforov [38])**.** *Let* $\mathsf{A} \in \mathbb{R}^{n\times n}$ *be an irreducible non-negative symmetric matrix and* $\mathsf{R} \in \mathbb{R}^{n\times n}$ *be the diagonal matrix of its rowsums,* $[\mathsf{R}]_{ii} = \sum_{j=1}^{n}[\mathsf{A}]_{ij}$*. Then*

$$\rho\left(\mathsf{R} + \frac{1}{r-1}\mathsf{A}\right) \geq \frac{r}{r-1}\rho(\mathsf{A}), \tag{37}$$

with equality holding if and only if all rowsums of A *are equal.*

To circunvent this problem with the boundings of the spectral radius of the convective term, $\mathsf{C}\left(\boldsymbol{u}_s\right)$, we can use the Wielandt's theorem (see Theorem 4) to relate the spectral radius of the matrices

$$2\mathsf{C}\left(\boldsymbol{u}_s\right) \equiv \mathsf{T}_{cs}^T\mathsf{F}_s|\mathsf{T}_{cs}| \qquad \text{and} \qquad \mathsf{D}^{\mathsf{C}} \equiv -\mathsf{T}_{cs}^T|\mathsf{F}_s|\mathsf{T}_{cs}, \tag{38}$$

where $\mathsf{C}\left(\boldsymbol{u}_s\right)$ is the same convective operator defined in Eq.(26) and $\mathsf{D}^{\mathsf{C}} \in \mathbb{R}^{n\times n}$ is a diffusive-like operator where the face diffusivities are replaced by the magnitude of the mass fluxes, $|\mathsf{F}_s|$. The matrix $\mathsf{C}\left(\boldsymbol{u}_s\right)$ is zero-diagonal whereas the matrix D^{C} has strictly negative diagonal coefficients. At this point, it is worth noticing that the off-diagonal elements of $2\mathsf{C}\left(\boldsymbol{u}_s\right)$ (in absolute value) and D^{C} are equal. Hence, the zero-diagonal matrix

$$\mathsf{D}^{\mathsf{C},\text{off}} \equiv \mathsf{D}^{\mathsf{C}} - \text{diag}(\text{diag}(\mathsf{D}^{\mathsf{C}})), \tag{39}$$

satisfies the conditions of the Perron-Frobenius theorem (see Theorem 3). Then, we can apply Wielandt's theorem (see Theorem 4) since

$$2|[\mathsf{C}\left(\boldsymbol{u}_s\right)]_{ij}| \leq [\mathsf{D}^{\mathsf{C},\text{off}}]_{ij} \quad \forall i,j \qquad \Longrightarrow \qquad 2|\lambda^{\mathsf{C}}| \leq \rho(\mathsf{D}^{\mathsf{C},\text{off}}). \tag{40}$$

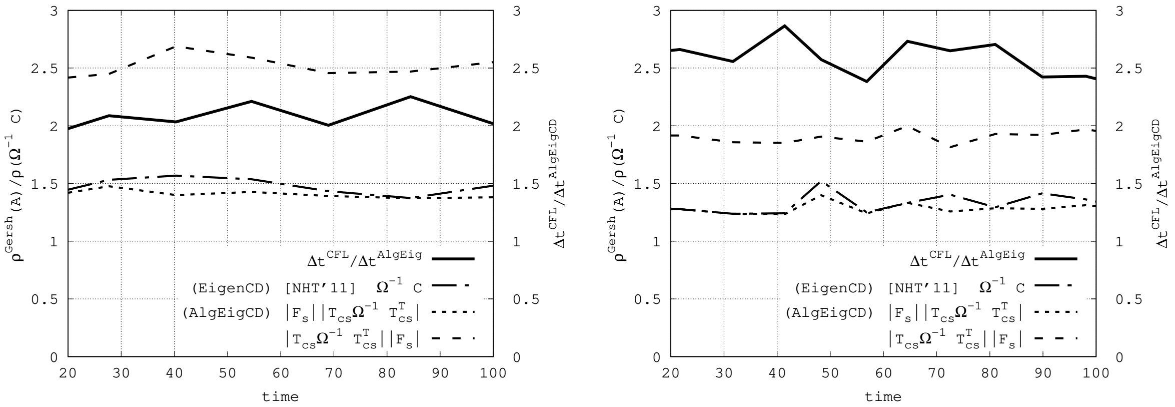

Figure 3: Numerical results obtained for a differentially heated cavity using a Cartesian stretched mesh (left) and an unstructured mesh composed of triangular elements (right).

In our case, taking $\mathsf{R} = -\operatorname{diag}(\operatorname{diag}(\mathsf{D}^{\mathsf{C}}))$, $\mathsf{A} = \mathsf{D}^{\mathsf{C},\mathrm{off}}$ and $r = 2$ in Eq.(37) together with the inequality (40) leads to

$$\rho(|\mathsf{D}^{\mathsf{C}}|) \geq 2\rho(\mathsf{D}^{\mathsf{C},\mathrm{off}}) = 4\rho(|\mathsf{C}\,(\boldsymbol{u}_s)\,|) \geq 4\rho(\mathsf{C}\,(\boldsymbol{u}_s)). \tag{41}$$

Recalling that the *leitmotiv* for all this analysis was to avoid constructing the matrix $\mathsf{C}\,(\boldsymbol{u}_s)$, it is obvious that relying of the contruction of another (similar in structure) matrix such as $|\mathsf{D}^{\mathsf{C}}|$ would not have much sense. At this point, we can use of the following properties of incidence and adjacency matrices

$$|\mathsf{T}_{cs}^T\mathsf{T}_{cs}| = |\mathsf{T}_{cs}^T||\mathsf{T}_{cs}|, \tag{42}$$

$$\big|\mathsf{T}_{cs}^T|\mathsf{F}_s|\mathsf{T}_{cs}\big| = |\mathsf{T}_{cs}^T||\mathsf{F}_s||\mathsf{T}_{cs}|, \tag{43}$$

to show that

$$\rho(|\mathsf{D}^{\mathsf{C}}|) = \rho\left(\big|\mathsf{T}_{cs}^T|\mathsf{F}_s|\mathsf{T}_{cs}\big|\right) \overset{(43)}{=} \rho(|\mathsf{T}_{cs}^T||\mathsf{F}_s||\mathsf{T}_{cs}|) \overset{Thm\ 2}{=} \rho(|\mathsf{T}_{cs}||\mathsf{T}_{cs}^T||\mathsf{F}_s|) \overset{(42)}{=} \rho(|\mathsf{T}_{cs}\mathsf{T}_{cs}^T||\mathsf{F}_s|). \tag{44}$$

Finally showing that $\rho(\mathsf{C}\,(\boldsymbol{u}_s))$ can be bounded either with $\rho(|\mathsf{T}_{cs}\mathsf{T}_{cs}^T||\mathsf{F}_s|)$ or $\rho(|\mathsf{F}_s||\mathsf{T}_{cs}\mathsf{T}_{cs}^T|)$

$$\rho(|\mathsf{T}_{cs}\mathsf{T}_{cs}^T||\mathsf{F}_s|) = \rho(|\mathsf{F}_s||\mathsf{T}_{cs}\mathsf{T}_{cs}^T|) = \rho(|\mathsf{D}^{\mathsf{C}}|) \geq 2\rho(\mathsf{D}^{\mathsf{C},\mathrm{off}}) = 4\rho(|\mathsf{C}\,(\boldsymbol{u}_s)\,|) \geq 4\rho(\mathsf{C}\,(\boldsymbol{u}_s)). \tag{45}$$

4 NUMERICAL RESULTS AND CONCLUDING REMARKS

In summary, the newly proposed *AlgEigCD* method simply relies on the contruction of the matrix $|\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T|$ which can be done in a pre-processing stage. Then, this matrix is used to find eigenbounds of matrices $\Omega_c^{-1}\mathsf{D}$ and $\Omega_c^{-1}\mathsf{C}\,(\boldsymbol{u}_s)$ as follows

$$\begin{aligned} \rho(\Omega_c^{-1}\mathsf{C}\,(\boldsymbol{u}_s)) \leq &\quad 1/4\rho(|\mathsf{F}_s||\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T|) &&\leq 1/4\max\{\operatorname{diag}(|\mathsf{F}_s|)\circ|\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T|\mathbf{1}_s\}, &&(46)\\ \rho(\Omega_c^{-1}\mathsf{D}(\boldsymbol{\alpha}_s)) = &\quad \rho(\tilde{\mathsf{\Lambda}}_s\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T) &&\leq \max\{\operatorname{diag}(\ \tilde{\mathsf{\Lambda}}_s\ )\circ|\mathsf{T}_{cs}\Omega_c^{-1}\mathsf{T}_{cs}^T|\mathbf{1}_s\}, &&(47) \end{aligned}$$

where the former inequality follows from Eq.(45) and application of the Gershgorin circle theorem to the matrix $|\mathsf{F}_s||\mathsf{T}_{cs}\mathsf{T}_{cs}^T|$. Similarly, the second one follows from Eq.(32). Notice that in these cases, the diagonal matrix Ω_c^{-1} has been introduced (see Remark 1). First prelimary results are displayed in Figure 3. They correspond to a square air-filled ($Pr = 0.71$) differentially heated cavity at Rayleigh number equal to $Ra = 10^9$ (for details of this flow configuration see, for instance, Refs. [4, 5]) computed with two different meshes: a structured Cartesian mesh with a stretching towards the walls similar to those used in Refs. [4, 5] and an unstructured mesh composed of triangular elements with a similar stretching. There is a significant gain (see ratio $\Delta t^{CFL}/\Delta t^{AlgEigCD}$) respect to the classical CFL condition. This is an observation that was already done in Ref. [16] where the *EigenCD* method was proposed. The new method is slightly improving the former one. However, the key element of the newly proposed *AlgEigCD* is the fact that no new matrix have to the re-computed every time-step and that, in practice, only relies on very basic linear algebra kernels: an `SpMV` and a point-wise Hadamard product of vectors. Therefore, the cross-platform portability of the method is straightforward.

Acknowledgments. F.X.T., X.A-F., A.A-B. and A.O. are supported by the *Ministerio de Economía y Competitividad*, Spain, RETOtwin project (PDC2021-120970-I00). F.X.T. and A.O. are supported by the *Generalitat de Catalunya* RIS3CAT-FEDER, FusionCAT project (001-P-001722). A.A-B. is supported by the predoctoral grants DIN2018-010061 and 2019-DI-90, by MCIN/AEI/10.13039/501100011033 and the AGAUR. A.G. is supported by the RSF project 19-11-00299. Calculations were carried out on MareNostrum 4 supercomputer at BSC. The authors thankfully acknowledge these institutions.

References

[1] F. X. Trias, O. Lehmkuhl, A. Oliva, C.D. Pérez-Segarra, and R.W.C.P. Verstappen. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured meshes. *Journal of Computational Physics*, 258:246–267, 2014.

[2] E. Komen, J. A. Hopman, E. M. A. Frederix, F. X. Trias, and R. W. C. P. Verstappen. A symmetry-preserving second-order time-accurate PISO-based method. *Computers & Fluids*, 225:104979, 2021.

[3] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-Preserving Discretization of Turbulent Flow. *Journal of Computational Physics*, 187:343–368, 2003.

[4] F. X. Trias, A. Gorobets, M. Soria, and A. Oliva. Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Ra-number up to 10^{11} - Part I: Numerical methods and time-averaged flow. *International Journal of Heat and Mass Transfer*, 53:665–673, 2010.

[5] F. X. Trias, A. Gorobets, M. Soria, and A. Oliva. Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Ra-number up to 10^{11} - Part II: Heat transfer and flow dynamics. *International Journal of Heat and Mass Transfer*, 53:674–683, 2010.

[6] F. X. Trias, A. Gorobets, and A. Oliva. Turbulent flow around a square cylinder at Reynolds number 22000: a DNS study. *Computers & Fluids*, 123:87–98, 2015.

[7] A. Pont-Vílchez, F. X. Trias, A. Gorobets, and A. Oliva. Direct Numerical Simulation of Backward-Facing Step flow at $Re_\tau = 395$ and expansion ratio 2. *Journal of Fluid Mechanics*, 863:341–363, 2019.

[8] N. Valle, F. X. Trias, and J. Castro. An energy-preserving level set method for multiphase flows. *Journal of Computational Physics*, 400(1):108991, 2020.

[9] F. Dabbagh, F. X. Trias, A. Gorobets, and A. Oliva. Flow topology dynamics in a three-dimensional phase space for turbulent Rayleigh-Bénard convection. *Physical Review Fluids*, 5:024603, 2020.

[10] X. Álvarez, A. Gorobets, F. X. Trias, R. Borrell, and G. Oyarzun. HPC^2 - a fully portable algebra-dominant framework for heterogeneous computing. Application to CFD. *Computers & Fluids*, 173:285–292, 2018.

[11] X. Álvarez, A. Gorobets, and F. X. Trias. A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers. *Computers & Fluids*, 214:104768, 2021.

[12] À. Alsalti-Baldellou, X. Álvarez-Farré, A. Gorobets, A. Oliva, and F. X. Trias. Strategies to increase the arithmetic intensity of linear solvers. In *Parallel Computational Fluid Dynamics*, Alba, Italy, May 2022.

[13] N. Valle, X. Álvarez-Farré, A. Gorobets, J. Castro, A. Oliva, and F. X. Trias. On the implementation of flux limiters in algebraic frameworks. *Computer Physics Communications*, 271:108230, 2022.

[14] P. Wesseling. *Principles of Computational Fluid Dynamics.* Springer, 2001.

[15] R. W. C. P. Verstappen and A. E. P. Veldman. Spectro-consistent discretization of Navier-Stokes: a challenge to RANS and LES. *Journal of Engineering Mathematics*, 34:163–179, 1998.

[16] F. X. Trias and O. Lehmkuhl. A self-adaptive strategy for the time-integration of Navier-Stokes equations. *Numerical Heat Transfer, part B*, 60(2):116–134, 2011.

[17] I. Bendixson. Sur les racines d'une équation fonamentale. *Acta Mathematica*, 25:359–365, 1902.

[18] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[19] OpenFOAM® user guide. `https://www.openfoam.com/documentation/user-guide`.

[20] P. F. Fischer, J. W. Lottes, and S. G. Kerkemeier. Nek5000 Web page. `https://nek5000.mcs.anl.gov`.

[21] COMSOL Multiphysics®. `https://www.comsol.com`. COMSOL AB, Stockholm, Sweden.

[22] Basilisk Web page. `http://www.basilisk.fr`.

[23] ANSYS FLUENT documentation. `https://www.afs.enea.it/project/neptunius/docs/fluent`.

[24] SU2 code. `https://su2code.github.io`.

[25] Code_Saturne. `https://www.code-saturne.org`.

[26] S. Langer, A.Schwöppe, and N. Kroll. The DLR flow solver TAU - Status and recent algorithmic developments. In *52nd Aerospace Sciences Meeting*, National Harbor, MD, USA, January 2014.

[27] Florian Menter and Boris Makarov. Private communication, 2021.

[28] J. M. Weiss, J. P. Maruszewski, and W. A. Smith. Implicit Solution of Preconditioned NavierStokes Equations Using Algebraic Multigrid. *AIAA Journal*, 37, 1999.

[29] J. M. Weiss, J. P. Maruszewski, and W. A. Smith. Implicit solution of the Navier-Stokes equations on unstructured meshes. In *13th Computational Fluid Dynamics Conference AIAA-97-2103*, Snowmass Village, CO, USA, June 1997.

[30] J. M. Weiss and W. A. Smith. Preconditioning Applied to Variable and Constant Density Flows. *AIAA Journal*, 33, 1995.

[31] J. M. Weiss. Calculation of reacting flowfields involving stiff chemical kinetics. In *14th Computational Fluid Dynamics Conference AIAA-99-3369*, Norfolk, VA, USA, November 1999.

[32] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[33] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal*, pages 215–234, March 1967.

[34] C.A. De Moura and C. S. Kubrusly, editors. *The CourantFriedrichsLewy (CFL) Condition: 80 Years After Its Discovery.* Springer, 2013.

[35] O. Perron. Zur Theorie der Matrices. *Mathematische Annalen*, 64:248–263, 1907.

[36] G. Frobenius. Ueber Matrizen aus nicht negativen Elementen. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften*, page 456477, 1912.

[37] S. I. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and Products.* Elsevier, Academic Press, 7th edition, 2007.

[38] V. Nikiforov. Chromatic number and spectral radius. *Linear Algebra and its Applications*, 426:810–814, 2007.