Spectral recycling strategies for the solution of nonlinear eigenproblems in thermoacoustics

Pablo Salas * Luc Giraud† Yousef Saad‡ Stéphane Moreau§

April 28, 2014

Abstract

In this work we consider the numerical solution of large nonlinear eigenvalue problems that arise in thermoacoustic simulations involved in the stability analysis of large combustion devices. We briefly introduce the physical modelling that leads to a nonlinear eigenvalue problem that is solved using a nonlinear fixed point iteration scheme. Each step of this nonlinear method requires the solution of a complex non-Hermitian linear eigenvalue problem. We review a set of state of the art eigensolvers and discuss strategies to recycle spectral informations from one nonlinear step to the next. More precisely, we consider the Implicitly Restarted Arnoldi method, the Krylov-Schur solver and its block-variant as well as the subspace iteration method with Chebyshev acceleration. On a small test example we study the relevance of the different approaches and illustrate on a large industrial test case the performance of the parallel solvers best suited to recycle spectral information.

1 Introduction

The increasingly demanding modern pollutant regulations have led to the use of lean combustion in gas turbine combustion chambers [3, 10]. Although this technology allows the reduction of pollutant emissions such as NO_x, it is prone to develop thermoacoustic instabilities. This phenomenon results from the coupling between the combustion in the flame zone and the acoustic modes of the combustion chamber, leading to high pressure and heat release oscillations which can even provoke its destruction [6, 14]. Therefore, the study and prediction of combustion instabilities during the design stage of aeronautical or industrial gas turbine combustion chambers, is of first importance. The problem to be solved arises from the discretization of a Helmholtz equation, which must be solved in order to compute the thermoacoustic modes of 3D gas turbine combustion chambers that can their

*Inria-CERFACS, France; present address Sherbrooke University, Canada
†Inria, France
‡Dept of computer Science and Eng., University of Minnesota, USA
§Sherbrooke University, Canada

safe functioning. The modeling of the physics leads to the solution of a large nonlinear eigenvalue problems and a only few tens smallest magnitude eigenvalues have to be computed. The nonlinear solver based on a fixed point method requires to solve a large sparse non-Hermitian linear eigenvalue problem at each iteration. In the framework of a few state of the art linear eigensolvers we propose different techniques that aims at accelerating the solution of each linear problem of the sequence by recycling spectral information from one nonlinear iteration to the next.

The industrial and physical context of this work is described in the following. One appropriate approach for the study of combustion instabilities, is the use of the linear wave equation for the pressure fluctuations written for a non isothermal reacting flow [14]:

$$\frac{\partial^2 p_1(\vec{x},t)}{\partial t^2} - \nabla \cdot c_0^2(\vec{x}) \nabla p_1(\vec{x},t) = (\gamma - 1) \frac{\partial q_1(\vec{x},t)}{\partial t}, \tag{1}$$

where p_1 and q_1 are the fluctuating part of the pressure p and heat release q, respectively, c_0 is the mean sound speed and γ is the adiabatic index of the gas. Assuming an harmonic form for both the pressure and heat release fluctuations

$$p_1(\vec{x},t) = Re(\hat{p}(\vec{x})e^{-i\omega t}), \quad q_1(\vec{x},t) = Re(\hat{q}(\vec{x})e^{-i\omega t}),$$

leads to the non-homogeneous Helmholtz equation:

$$\nabla \cdot c_0^2(\vec{x}) \nabla \hat{p}(\vec{x}) + \omega^2 \hat{p}(\vec{x}) = i\omega(\gamma - 1)\hat{q}(\vec{x}). \tag{2}$$

The real part of the complex frequency $\omega = 2\pi f$ corresponds to the resonant frequency of the mode while its imaginary part corresponds to its growth rate. The heat release is modeled by means of Flame Transfer Functions (FTF), which allows to express the heat release $\hat{q}(\vec{x})$ in terms of the acoustic pressure $\hat{p}(\vec{x}_{ref})$ at a given reference point $\vec{x}_{ref}$ [4, 5, 12]. Since the FTF depends, in general, on the complex frequency ω, the resulting Helmholtz equation is a functional nonlinear eigenvalue problem, which can be written as [12]:

$$\nabla \cdot c_0^2(\vec{x}) \nabla \hat{p}(\vec{x}) + \omega^2 \hat{p}(\vec{x}) = \frac{(\gamma - 1)}{\rho_0(\vec{x})} \mathrm{FTF}(\omega) \nabla \hat{p}(\vec{x}_{ref}) \cdot \vec{n}_{ref}, \tag{3}$$

where $\rho_0(\vec{x})$ is the density and $\vec{n}_{ref}$ is an unitary vector normal to the inlet surface at the reference point $\vec{x}_{ref}$. The solution of Equation (3) is the objective of the acoustic code AVSP [12], developed at CERFACS. Its discretization on unstructured meshes using a finite volume method leads to a nonlinear complex eigenvalue problem whose size n is equal to the number of nodes in the mesh. Even when the combustion-acoustics interaction is not taken into account, i.e., FTF $= 0$, the discretization of the homogeneous Helmholtz equation leads, in general, to a nonlinear eigenvalue problem. Indeed, the boundary conditions accounting for a reduced boundary impedance Z are represented by a Robin condition of the form

$$c_0 Z \nabla \hat{p} \cdot \vec{n} - i\omega \hat{p} = 0,$$

where $\vec{n}$ is the outgoing unit normal vector to the boundary. The general nonlinear nature of the problem comes from the frequency-dependent value of the complex impedance $Z = Z(\omega)$. Hence, the discretization of these boundary conditions introduces terms that depend on the complex frequency ω [12]. The resulting discretized nonlinear eigenproblem reads as

$$\mathbf{A}\bar{p} + \omega \mathbf{B}(\omega)\bar{p} + \omega^2 \bar{p} = \mathbf{C}(\omega)\bar{p}, \tag{4}$$

where:

- The nonlinear complex eigenvalue ω corresponds to the resonant frequency (real part) and growth rate (imaginary part) of the mode. Experimental studies have revealed that the combustion instabilities rise at low frequencies, which means that the interest is in solving Equation (4) to obtain a few smallest magnitude nonlinear eigenvalues.
- The eigenvector $\bar{p}$ represents the acoustic pressure at every mesh node: it describes the mode structure.
- $\mathbf{A}$ is a $n \times n$ sparse real matrix, where n is the number of vertices of the unstructured mesh. This matrix arises form the discretization of the operator $\nabla c_0^2(\vec{x})\nabla$.
- $\mathbf{B}(\omega)$ is a $n \times n$ complex diagonal matrix. Its nonzero entries are associated with the vertices on the boundary where the Robin condition is applied.
- $\mathbf{C}$ is a low-rank sparse complex matrix that arises from the discretization of the right-hand side term of Equation (3).

To the best of our knowledge, there are no available nonlinear eigensolvers for the solution of general problems such as Equation (4). Therefore, in the present work, a fixed point iteration procedure is used in order to obtain the desired nonlinear eigenpairs. This means that the problem is linearized, obtaining a sequence of linear eigenproblems that have to be solved iteratively in order to obtain one nonlinear eigenpair of Equation (4). In this paper different strategies, depending on the chosen linear eigensolver, are considered in order to accelerate the solution of each linear eigenproblem during the nonlinear fixed point iteration process.

The paper is organized as follows. In Section 2 we describe the nonlinear scheme that has been implemented in the AVSP simulation code. In the next section we review the state of the art linear eigensolvers that have been considered in this study. The strategies to recycle spectral information from one nonlinear step to the next are introduced in Section 4. The numerical behaviors of the proposed approaches are investigated in Section 5, first on a small test case representative of a simple combustion instability problem, then on a large scale problem arising from the study of a complex tridimensional industrial combustor. Somme concluding remarks are reported in Section 6.

2 Linearization of the eigenproblem: fixed point iteration method

In this section we describe the numerical procedure considered for the solution of the nonlinear Equation (4), where the nonlinearity is introduced by both the $\omega\mathbf{B}(\omega)$ and $\mathbf{C}(\omega)$ terms. The nonlinear solution scheme is based on a fixed point procedure. It consists in choosing a linearization value referred to as $\tilde{\omega}^{(j)}$ for the nonlinear terms, so that these terms become linear and can be merged with the linear term $\mathbf{A}$. If we denote $\mathcal{A}^{(j)} = \mathbf{A}+\tilde{\omega}^{(j)}\mathbf{B}(\tilde{\omega}^{(j)})-\mathbf{C}(\tilde{\omega}^{(j)})$, the resulting linear eigenvalue problem that must be solved at the j^{th} nonlinear step reads

$$\mathcal{A}^{(j)}\bar{p} + {\omega^{(j)}}^2\bar{p} = 0. \tag{5}$$

Unless the linearization value $\tilde{\omega}^{(j)}$ is already a solution of Equation (4), all the linear eigenvalues $\omega_\ell^{(j)}$ solution of Equation (5) will differ from $\tilde{\omega}_j$. The procedure proceeds by choosing $\omega_i^{(j)}$ $(1 \leq i \leq \ell)$ among the linear eigenvalues $\omega_\ell^{(j)}$, so that $\omega_i^{(j)}$ is the closest to $\tilde{\omega}_j$. Therefore $\omega_i^{(j)}$ is the new linearization value, i.e., $\tilde{\omega}^{(j+1)} = \omega_i^{(j)}$. Provided that the procedure do not diverge, the sequence of linear solutions

$$\tilde{\omega}^{(1)},\ \tilde{\omega}^{(2)},\ \ldots,\ \tilde{\omega}^{(j)}$$

will converge towards one nonlinear solution of Equation (4). If the problem is stiff, a relaxation parameter can be introduced to ensure convergence. The nonlinear stopping criterion is based on the relative distance between two successive linearization values. For a prescribed threshold ϵ, the nonlinear iteration will be stopped when

$$\frac{|\tilde{\omega}^{(j+1)} - \tilde{\omega}^{(j)}|}{|\tilde{\omega}^{(j)}|} < \epsilon. \tag{6}$$

3 Numerical methods: linear eigensolvers

At each fixed point iteration, a linear non-Hermitian eigenproblem must be solved. In this section we briefly describe the eigensolvers we have considered for the solution of these linear eigenproblems. For large eigenproblems, the computation of the whole spectrum is out of question, and we can only expect to compute a few n_{ev} eigenvalues (several tens typically) lying on a certain part of the spectrum (largest magnitude eigenvalues, smallest magnitude eigenvalues, smallest real part, ...). Moreover, the methods considered for this purpose must necessarily be iterative, as a consequence of the Abel's famous theorem. The methods presented below for the calculation of approximate eigenvalues of a complex non-Hermitian $n \times n$ matrix $\mathcal{A}$ build iteratively a $n \times k$ (with $k \ll n$) search subspace spanned by the column of U_k. The successive subspaces U_k contains an increasingly accurate approximation to an invariant subspace of $\mathcal{A}$.

For the sake of simplicity, in the present paper the search subspace U is considered orthonormal, so that $U^H U = I$. In order to extract the spectral information contained in U, the Rayleigh-Ritz procedure is employed. It consists in computing the $k \times k$ matrix B, known as Rayleigh quotient, as

$$B = U^H \mathcal{A} U. \tag{7}$$

The method proceeds by computing the eigenpairs of the Rayleigh quotient B, so that we can write $BW = WD$, being W the eigenvectors of B corresponding to the eigenvalues that appear on the diagonal of the diagonal matrix D. Then, the pair (D, UW) are the approximate eigenpairs of the original matrix $\mathcal{A}$ and are called the Ritz pairs [20].

The Rayleigh-Ritz extraction is used in the subsequent methods for the calculation of the Ritz pairs. As for any iterative method, a criterion is needed in order to stop the procedure. In the present context, the methods iterate until the normalized residuals corresponding to the Ritz pairs are smaller than a certain threshold ε. In this paper the considered stopping criterion is:

$$\frac{\|\mathcal{A}\tilde{x} - \tilde{\lambda}\tilde{x}\|}{|\tilde{\lambda}|} \leq \varepsilon, \tag{8}$$

where $(\tilde{\lambda}, \tilde{x})$ is the approximate eigenpair.

3.1 Implicitly Restarted Arnoldi method

This method, often referred to as IRA, is based on Arnoldi decompositions of a $n \times n$ matrix $\mathcal{A}$ [1]. Starting from a single unitary vector v_1, it builds an orthonormal basis $V_k = [v_1 \ v_2 \ \dots \ v_k]$ of a Krylov subspace of dimension k [20]:

$$\mathcal{K}_k(\mathcal{A}, v_1) = span\{v_1, \ \mathcal{A}v_1, \dots, \ \mathcal{A}^{k-1}v_1\}, \tag{9}$$

for which the following Arnoldi decomposition is satisfied:

$$\mathcal{A}V_k = V_k H_k + \alpha v_{k+1} e_k^T \tag{10}$$

where H_k is a $k \times k$ upper Hessenberg matrix that corresponds to the Rayleigh quotient associated with $\mathcal{A}$ and V_k; e_k^T is the k^{th} canonical vector of $\mathbb{R}^k$.

When the Arnoldi decomposition proceeds, V_k will contain increasingly accurate spectral information of $\mathcal{A}$. Due to the limited amount of available memory, the maximum allowed size of the factorization will be typically of a few hundreds for large problem sizes. For this reason, in practice, the method is used prescribing a maximum decomposition size m. When this limit is attained without having reach the demanded level of accuracy for the Ritz pairs, the method has to be restarted, reducing the size of the decomposition to k_{min} ($n_{ev} \leq k_{min} < m$). The restart must be performed in a smart way, attempting to keep in the resulting reduced Arnoldi factorization of size k_{min} the most valuable spectral information contained in the original Arnoldi factorization of size m. This is the purpose of the Implicitly Restarted Arnoldi (IRA) algorithm, developed by Lehoucq and Sorensen in [9]. The parallel Fortran library (P)ARPACK [8], considered as the definitive implementation of this method, is used in the present work.

3.2 Krylov-Schur method and its block variant

The Krylov-Schur method [19], as the IRA algorithm, is based on Krylov subspaces. Under certain conditions (use of exact shifts), it is mathematically equivalent to the Implicitly Restarted Arnoldi method [20]. The restart part of the algorithm is much simpler, due to the fact that, contrary to the Arnoldi factorization, no particular structure of the Rayleigh quotient matrix must be preserved.

Starting from a single unitary vector u_1, this method builds the Krylov decomposition of size k of the matrix $\mathcal{A}$:

$$\mathcal{A}U_k = U_k B_k + u_{k+1} b_{k+1}^H, \tag{11}$$

where the columns of U_k form an orthonormal basis of the search space, u_{k+1} is orthonormal to U_k and the $k \times k$ matrix $B_k = U_k^H \mathcal{A} U_k$ is the Rayleigh quotient of $\mathcal{A}$ associated with U_k. An important property of Krylov decomposition is that they are invariant to similarity transformations. Indeed, let Q be nonsingular and post-multiplying Equation (11) we obtain

$$\mathcal{A}(UQ) = (UQ)Q^{-1}BQ + u(b^H Q) \text{ that reads } \mathcal{A}\tilde{U} = \tilde{U}\tilde{B} + u\tilde{b}^H. \tag{12}$$

Both Krylov decompositions (11) and (12) are similar. This offers a natural way for restarting the decomposition when the dimension of the search space reaches the maximum allowed size m, referred to by Stewart [20] as Krylov-Schur restart:

1. Compute a sorted Schur decomposition of the Rayleigh quotient $B_m Q = QT$, with the diagonal elements of T sorted conveniently (from smallest to largest magnitude in the present case). Then apply the similarity transformation based on Q, obtaining
$$\mathcal{A}\tilde{U}_m = \tilde{U}_m T + u_{k+1}\tilde{b}_{k+1}^H.$$

2. The resulting Krylov decomposition can be written as
$$\mathcal{A}(\tilde{U}_1\ \tilde{U}_2) = (\tilde{U}_1\ \tilde{U}_2)\begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} + u(\tilde{b}_1^H\ \tilde{b}_2^H),$$
where $(T_{11}, \tilde{U}_1)$ concentrates the most valuable spectral information. Then, $\mathcal{A}\tilde{U}_1 = \tilde{U}_1 T_{11} + u\tilde{b}_1^H$ is also a Krylov decomposition, that can be used to restart the procedure.

3.2.1 Block Krylov-Schur method

The extension of the Krylov-Schur method to its block variant is straightforward. In [21], the authors present a block Krylov-Schur method for symmetric eigenproblems. Inherent issues associated with block methods such as the treatment of the rank deficiency in the extending block are also discussed. The algorithm presented here is a generalization of the algorithm proposed in [21] for general non-Hermitian matrices. It relies on the block Arnoldi factorization of the matrix $\mathcal{A}$: let p be the block size, then at each step of the block Arnoldi procedure, the Arnoldi basis is extended with p new vectors. The block Arnoldi factorization reads as:
$$\mathcal{A}\mathscr{V}_k = \mathscr{V}_k\mathscr{H}_k + V_{k+1}H_{k+1,k}E_k^T = \mathscr{V}_{k+1}\hat{\mathscr{H}}_k, \tag{13}$$
where $[\mathscr{V}_k\ V_{k+1}]$ is an orthonormal basis of the block Krylov space of dimension $(k+1)\times p$; E_k is the matrix of the p last columns of the $(kp)\times(kp)$ identity matrix and $\mathscr{H}_k$ is a band upper Hessenberg matrix with bandwith p.

The block Krylov-Schur method is implemented as described in the following main steps:

1. Given an initial $n \times p$ matrix V_1 with orthonormal columns, Ruhe's variant of block Arnoldi block builds the decomposition
$$\mathcal{A}\mathscr{V}_m = \mathscr{V}_m H_m + V_{m+1}H_{m+1,m}E_m^T.$$

2. An unitary similarity transformation is used to compute the equivalent block Krylov-Schur decomposition. In the diagonal of the Schur form S_m of the Rayleigh quotient $\mathscr{H}$ appear the eigenvalues sorted according to the target part of the spectrum (from smallest to largest magnitude in our application). Then we have
$$\begin{aligned}&\mathcal{A}(\mathscr{V}_m Q) = (\mathscr{V}_m Q)S_m + V_p\mathscr{H}_{m+1,m}E_m^T Q \equiv \\ &\mathcal{A}\mathscr{U}_m = \mathscr{U}_m S_m + \mathscr{U}_p B^H,\end{aligned}$$

where $\mathscr{U}_m = \mathscr{V}_m Q$, $U_p = V_p$, $B^H = H_{m+1,m} E_m^T Q$ and S_m is the complex Schur form of the band Hessenberg matrix $\mathscr{H}_m$, with the n_{ev} wanted Ritz values in the leading block. The compact form reads

$$\mathcal{A}\mathscr{U}_m = \mathscr{U}_{m+1}\hat{S}_m,$$

being $\mathscr{U}_{m+1} = [\mathscr{U}_m \; U_{m+1}]$ and $\hat{S}_m = \begin{pmatrix} S_m \\ U_p B^H \end{pmatrix}$.

3. Rayleigh-Ritz extraction is used to compute approximate eigenpairs, whose accuracy is tested. Then the decomposition is reduced to size $\ell \geq n_{ev}$, just by truncating the decomposition and keeping only the first ℓ columns:

$$\mathcal{A}\mathscr{U}_\ell = \mathscr{U}_\ell S_\ell + U_{\ell+1} B^H = \mathscr{U}_{\ell+1}\hat{S}_\ell.$$

4. This decomposition is extended to size m using again Ruhe's variant of block Arnoldi.

Steps 2 to 4 are repeated until the wanted eigenvalues have converged or the maximum number of restarted iterations is reached. This procedure is completely equivalent to the single-vector version of the Krylov-Schur method. Only the algorithm for the computation of the block Krylov decomposition changes and ℓ and m have to be kept multiples of the block size p.

During the restart phase, a single-vector Krylov method applies a filter polynomial of degree $m - k$, i.e., the size of the extended Krylov decomposition minus the size of the restarted one. For the block variant, assuming $k \equiv \ell$ and m are kept the same as for the single-vector counterpart, the filter polynomial applied at restart is of lower degree $(m - \ell)/p$.

The concern of rank loss is treated in our implementation by using a vector-wise construction of the basis, known as Ruhe's variant. This choice gives up on the better use of the memory hierarchy (BLAS-3 efficiency) of block methods. We notice that our thermoacoustic code is based on a matrix free implementation where only single matrix-vector computational kernel is available, which prevents us to benefit from matrix-matrix calculation speed. Consequently Ruhe's variant does not incur any computational penalty.

3.3 Jacobi-Davidson method

The former methods extend the search basis by building Krylov subspaces. The Jacobi-Davidson method extends the search space by solving the so-called correction equation. The underlying idea is simple: starting from a given eigenpair approximation (μ, z), we must find corrections η and v so that $(\mu + \eta, z + v)$ is a better approximation to the actual eigenpair. Being $r = \mathcal{A}z - \mu z$ the residual corresponding to the approximate eigenpair (μ, z), the correction equation reads:

$$(I - zz^H)(\tilde{\mathcal{A}} - \mu I)(I - zz^H)v = -r, \qquad v \perp z. \tag{14}$$

The solution v of Equation (14) provides the new direction to append to the search space. Further details about the Jacobi-Davidson method and the correction equations can be

found in [7, 15, 17, 18, 20]. In this work, the selected implementation is the one presented in [7], known as Jacobi-Davidson style QR algorithm. The main steps of the method, in order to compute eigenpairs of $\mathcal{A}$ nearest to chosen target τ (in this case we take $\tau = 0$ to compute those of smallest magnitude) are:

1. Starting from an orthonormal basis of the search space V_k of dimension k, the Rayleigh-Ritz procedure is used to extract the eigenpair approximation (μ, z) nearest to the targeted value τ. The corresponding residual $r = \mathcal{A}z - \mu z$ is computed.

2. If the scaled residual is larger than the threshold ε, then the correction equation (14) is solved to obtain w, orthogonal to z. The new vector v is normalized and orthogonalized against V_k to produce v_{k+1}, so that $V_{k+1} = [V_k \; v_{k+1}]$.

3. If the scaled residual is larger than the threshold ε, then the correction equation (14) is solved to obtain w, orthogonal to z. The new vector v is normalized and orthogonalized against V_k to produce v_{k+1}, so that $V_{k+1} = [V_k \; v_{k+1}]$.

4. When the search basis reaches the maximal allowed size m, it is truncated keeping the most relevant spectral information: $V_m \to V_\ell$ and then the methods proceeds until the allowed number of Jacobi-Davidson steps is exceeded. In this work, the solution of the correction equation accounts for one Jacobi-Davidson step.

3.4 Subspace iteration with Chebyschev acceleration

The Subspace Iteration method is also considered in this work because of its ability to start from a set of vectors (as a block method), instead of from a single vector. The simplest version of the subspace iteration method is a block version of the power method, first introduced by Bauer under the name of *Treppeniteration* (staircase iteration) [2]. Starting with an initial block of m vectors arranged in the $n \times m$ matrix $X_0 = [x_1, \ldots, x_m]$ the block $X_k = \mathcal{A}^k X_0$ is computed for a certain power k. The columns in X_k will loose their linear independency for increasing values of k, so that the idea is to re-establish their linear independence using, for instance, the QR factorization. Under a certain number of assumptions [15], the columns of X_k will converge to the Schur vectors associated with the m dominant eigenvalues of $\mathcal{A}$: $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$. Instead of using the columns of X_k as approximations to the Schur vectors, using them in a Rayleigh-Ritz procedure will produce in general better approximations.

This method is well suited for the computation of the largest magnitude eigenpairs, but in the present case the interest is in those of smallest magnitude. Chebyshev polynomials of first kind are used as filter polynomials to overcome this issue: applying them at each iteration focuses the algorithm into a certain region of the spectrum. Details about Chebyshev polynomials and their properties are given in [15]. All that we need to know here is that a Chebychev polynomial has associated an ellipse in the complex plane. The part of the spectrum enclosed in this ellipse will be filtered. One realizes that, to build the filter polynomial, a certain amount of information on the spectrum is needed, which constitutes the main drawback of this method: we must be able to set an ellipse that encloses the unwanted part of the spectrum. In that respect, we need the following a priori information

on the spectrum: the largest magnitude eigenvalue, the largest imaginary part eigenvalue, and the first wanted eigenvalue that is not enclosed in the ellipse. These values define the ellipse eccentricity e and its center c. The good news is that, for our application, the spectra of the sequence of matrices $\mathcal{A}^{(j)}$ are very similar to each other, as shown for a small example in Figure 1. We can then compute the needed information on the spectrum of $\mathcal{A}^{(1)}$ using any of the available methods, and the ellipse fitted according to the spectrum of $\mathcal{A}^{(1)}$ can be used for the subsequent nonlinear iterations.

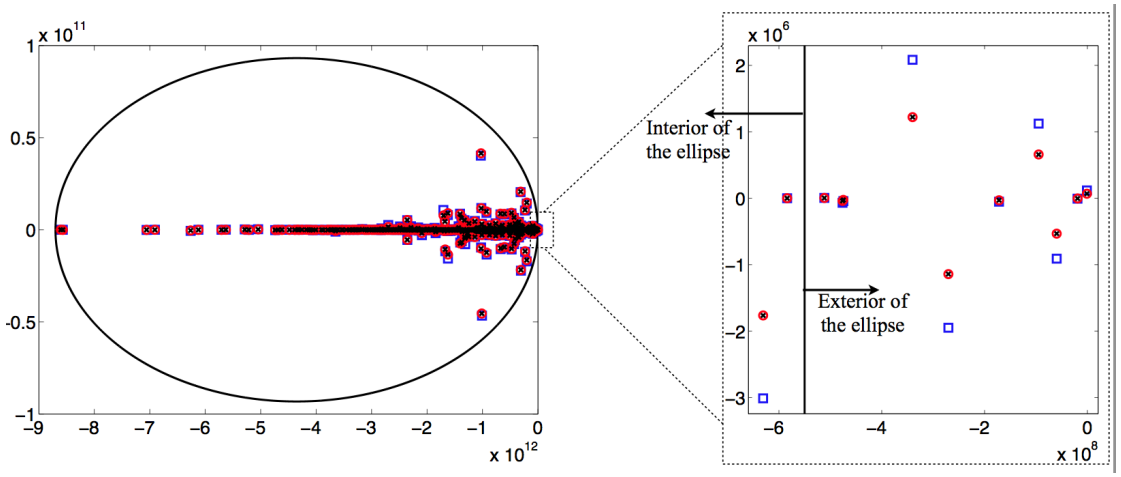

Figure 1: Spectrum of test matrices $\mathcal{A}^{(1)}$ (□), $\mathcal{A}^{(2)}$ (□), $\mathcal{A}^{(3)}$ (×) obtained from successive nonlinear iterations when converging the smallest nonlinear eigenvalue ω_1. In black, the ellipse used for the Chebyshev polynomial, containing the unwanted part of the successive spectra.

The filter polynomial has the form:

$$p_k(\lambda) = \frac{C_k[(\lambda - c)/e]}{C_k[(\lambda_1 - c)/e]}, \tag{15}$$

where C_k is the Chebyshev polynomial of degree k of the first kind and λ_1 is an approximation of the first wanted eigenvalue that is not enclosed in the ellipse E. Therefore, the successive applications of the polynomial defined by Equation (15) to a set of vectors U during the subspace iterations, will make U converge to an invariant subspace corresponding to the eigenvalues that lie out of the ellipse E.

The computation of $z_k = p_k(\mathcal{A})z_0$ is performed iteratively thanks to the three-term recurrence for Chebyshev polynomials [15] as:

1. Given the initial vector z_0, compute

$$\sigma_1 = \frac{e}{\lambda_1 - c},$$

$$z_1 = \frac{\sigma_1}{e}(\mathcal{A} - cI)z_0.$$

2. Iterate for $i = 1, \ldots, k-1$:

$$\sigma_{i+1} = \frac{1}{2/\sigma_1 - \sigma_i},$$

$$z_{i+1} = 2\frac{\sigma_{i+1}}{e}(\mathcal{A} - cI)z_i - \sigma_i\sigma_{i+1}z_{i-1}.$$

4 Recycling strategies

For the thermoacoustic calculations in complex geometries, it has been observed that the nonlinear fixed point iteration introduced in Section 2 often converges very quickly. An immediate consequence is that the successive solutions of the sequence of linear eigenproblem are close to each other so that the following quantities become smaller and smaller as the nonlinear scheme converges:

- The relative Frobenius norm of the difference between two consecutive matrices

$$\Delta_F = \frac{\|\mathcal{A}^{(j-1)} - \mathcal{A}^{(j)}\|_F}{\|\mathcal{A}^{(j)}\|_F}.$$

- The relative distance between the eigenvalues of two consecutive nonlinear iterations

$$\delta = \frac{|\tilde{\omega}^{(j-1)} - \tilde{\omega}^{(j)}|}{|\tilde{\omega}^{(j)}|}.$$

- The angle between the subspaces formed by the eigenvectors between two consecutive nonlinear iterations

$$\angle(P^{(j-1)}, P^{(j)}).$$

This suggests that the eigensolution of the $(j-1)^{th}$ iteration is, in fact, a good approximation of the j^{th} iteration, that should be used to define the initial guess to solve the j^{th} linear eigenproblem. In the following, depending on the eigensolver, different procedures are proposed to exploit this a priori information and recycle eigenvectors from one step to the next one in order to reduce the computational cost of the solution of the nonlinear eigenproblem.

4.1 Recycling with IRA and Krylov-Schur methods

These methods can only start from a single vector, whereas we have a set of n_{ev} eigenvectors that we would like to recycle from the previous nonlinear iteration. To take into account all the available eigenvectors, we propose to use a normalized linear combination of them as initial vector u_1. In exact arithmetic, it is well known that starting from a vector u_1 that is a linear combination of k eigenvectors, then the Krylov sequence based on u_1 terminates within k steps [20]. In terms of eigensolvers, it means that they would converge

to the solution within the first iteration. In the same circumstances with finite precision, these methods will not necessary converge within the first iteration but they will converge much faster than if started from a random vector. Therefore, based on this property of Krylov subspaces, by continuity one can expect that starting the Krylov solver from u_1 the normalized vector sum $\bar{p}_1^{(j-1)} + \bar{p}_2^{(j-1)} + \cdots + \bar{p}_{nev}^{(j-1)}$ will improve the convergence for the eigensolution of $\mathcal{A}^{(j)}$, compared to starting from a random vector. The following simple procedure is then considered:

1. The problem $\mathcal{A}^{(j-1)}\bar{p} = -{\omega^{(j-1)}}^2\bar{p}$ corresponding to the $(j-1)^{th}$ nonlinear iteration is solved and n_{ev} eigenpairs $(\omega^{(j-1)}, \bar{p}^{(j-1)})$ are computed.
2. Form the vector $\bar{p}_1^{(j-1)} + \bar{p}_2^{(j-1)} + \cdots + \bar{p}_{n_{ev}}^{(j-1)}$ and normalize it to define u_1.
3. Using ARPACK or the Krylov-Schur solver, solve the problem $\mathcal{A}^{(j)}\bar{p} = -{\omega^{(j)}}^2\bar{p}$ using u_1 as initial vector.

4.2 Recycling using the Jacobi-Davidson method

The Jacobi-Davidson method can be used to build the search space solving iteratively the correction equation from a single random vector. Nevertheless, if the initial vector does not contain any particular information on the solution (which is the case in general), the convergence of the method can be very erratic at the beginning, until the method gather enough spectral information related to the region of interest around the target τ. It is a general practice (adopted in this work as well) to build first an Arnoldi basis of a given size k (here $k = n_{ev}$) from a random vector, before the Jacobi-Davidson method takes over from the generated subspace and corresponding Rayleigh quotient. This is what we referred to as a random initialization.

But in fact, much better than an Arnoldi subspace built from a random vector, is the subspace formed by the eigenvectors computed at the previous nonlinear fixed point iteration, since it already contains useful information about the solution that has to be computed. Therefore, the proposed strategy simply reads:

1. The problem $\mathcal{A}^{(j-1)}\bar{p} = -{\omega^{(j-1)}}^2\bar{p}$ corresponding to the $(j-1)^{th}$ nonlinear iteration is solved and n_{ev} eigenpairs $(\omega^{(j-1)}, \bar{p}^{(j-1)})$ are computed.
2. An orthonormal basis $U_{n_{ev}}$ of $P^{(j-1)} = [\bar{p}_1^{(j-1)}, ..., \bar{p}_{n_{ev}}^{(j-1)}]$ is computed with its associated Rayleigh quotient $C_{n_{ev}} = U_{n_{ev}}^H \mathcal{A}^{(j)} U_{n_{ev}}$. Then the Rayleigh-Ritz procedure extracts an approximate eigenpair $(\tilde{\omega}, \tilde{p})$ closest to the target τ.
3. Then classical Jacobi-Davidson is used to go on with the computation of the desired eigenpairs.

4.3 Recycling using block methods

By definition, the distinctive feature of block methods, as block Krylov-Schur or Subspace Iteration schemes, is that they start the iterative process to compute the desired eigenpairs

from a set of p vectors. Therefore, we can use the n_{ev} available eigenvectors from a nonlinear step (actually an orthonormal basis of them), as initial block of vectors for the solution of the next linearized problem. If they are a good approximation of the invariant subspace that has to be computed, then the initial level of the residuals is expected to be much closer to the demanded accuracy than starting from a block of random vectors.

5 Numerical Results

In this section several numerical experiments are performed in order to investigate the effectiveness of the proposed recycling techniques. First, these techniques are used with the most classical eigensolvers, i.e., the Krylov-based and the Jacobi-Davidson solvers, demonstrating their efficiency. Then, the block Krylov-Schur and the Chebyshev subspace iteration methods are compared to the Krylov-Schur method. Although the different meaning of the numerical parameters of each method does not allow an objective comparison between the different techniques, this will not prevent us from extracting some general qualitative conclusions within the present context.

We start considering a test problem of size $n = 1480$, with boundary conditions such as $\mathbf{B}(\omega) = 0$. First, it is solved without combustion (i.e., $\mathbf{C}(\omega) = 0$), so that the associated linear eigenproblem $\mathbf{A}\bar{p}+\omega^2\bar{p} = 0$ is solved to obtain $n_{ev} = 5$ smallest magnitude eigenvalues. Then the nonlinear fixed point method is used to compute one eigensolution of the problem with combustion $\mathbf{A}\bar{p} + \omega^2\bar{p} = \mathbf{C}(\omega)\bar{p}$, using the smallest magnitude eigenvalue ω_1 obtained for the problem without combustion, as first linearization value. Note that the eigenvalues are in fact $\lambda = -\omega^2$, with $\omega = 2\pi f$. The results concerning the eigenvalues are given in Hertz in Table 1, i.e., the provided values correspond to f.

In Table 1 are reported the results associated with the first three nonlinear iterations. It can first be observed that the nonlinear scheme converges fairly fast. It can also be seen that the eigenspaces computed at each iteration becomes quickly collinear, as illustrated by the angle displayed in the last column of Table 1.. Consequently, the eigenspace computed at a given iteration is a good approximation to initialize the eigensolver at the next one.

# nonlinear it.	f (Hz)	Δ_F	$\delta(\%)$	$\angle(P^{(j-1)}, P^{(j)})$ (degrees)
0	272.3000	–	–	–
1	159.6988 - 9.2850 i	6.7317e-2	70.6	13.72
2	159.6703 - 5.4399 i	4.7616e-3	2.41	0.3368
3	159.6703 - 5.4390 i	1.2049e-6	6.08e-4	8.5048e-5

Table 1: Results obtained using the nonlinear procedure when computing the smallest magnitude nonlinear eigenfrequency with combustion.

5.1 Krylov-based solvers

In Figure 2 is displayed the convergence history of the scaled residual $\|\mathcal{A}\bar{p} - \lambda\bar{p}\|/|\lambda|$ associated with the five smallest magnitude eigenvalues of $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ as a function of the number of restarts, when the calculation is performed using ARPACK and Krylov-Schur with a demanded accuracy $\varepsilon = 10^{-4}$. The continuous lines (——) are the residuals when the eigensolvers start from a random vector, whereas the lines with circles (–○–) correspond to the situation where the starting vector is the normalized sum of the n_{ev} eigenvectors computed at the previous nonlinear iteration. As expected, we see that recycling the spectral informations from one step to the next one significantly improves the convergence rate of both eigensolvers. Furthermore, the benefit becomes larger as the nonlinear scheme converges since the starting vector contains increasingly accurate information on the invariant subspace to be computed. Finally, although mathematically equivalent, it can be seen that the convergence histories of ARPACK and Krylov-Schur slightly, which is due due to finite precision arithmetic effects.

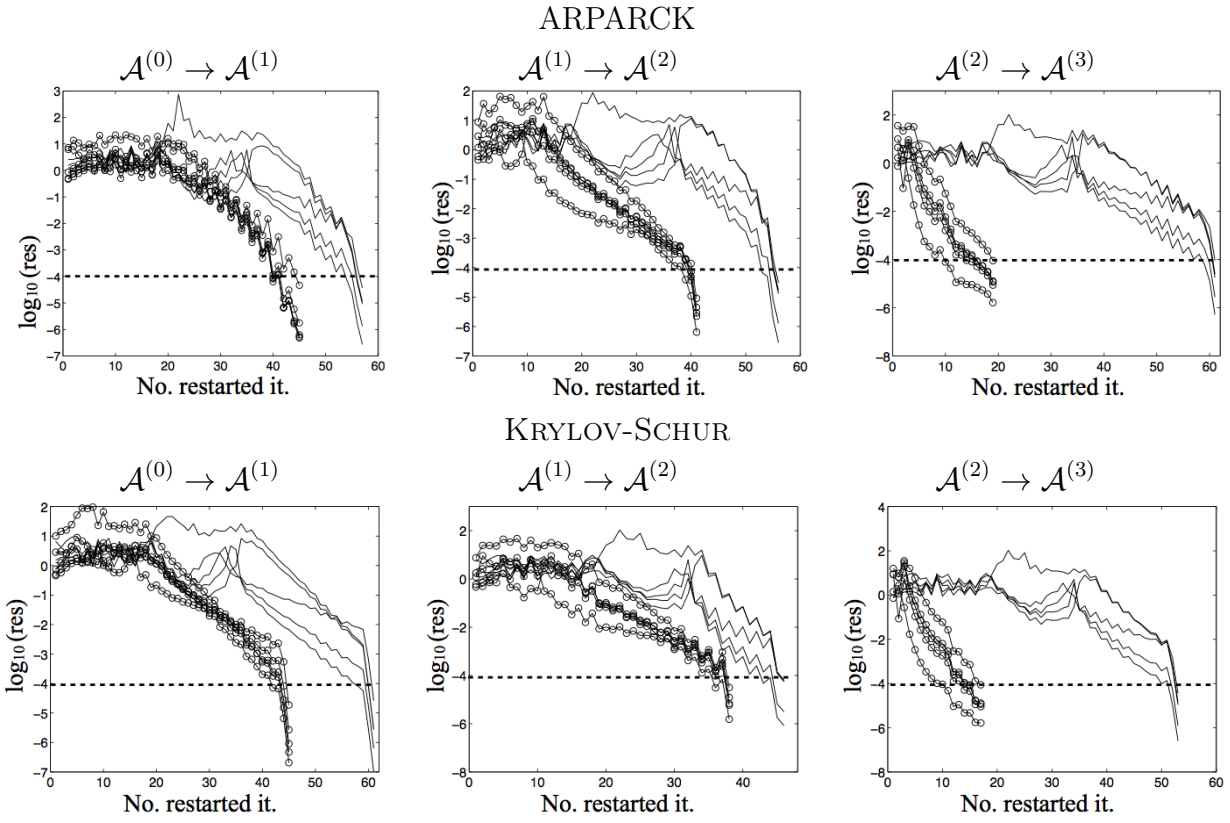

Figure 2: Convergence history of the scaled residuals for the $nev = 5$ smallest magnitude eigenpairs of the sequence of linear problems $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ solved with $\varepsilon = 10^{-4}$ for the computation of the smallest magnitude nonlinear eigenfrequency ω_1. (–○–): with recycling strategy; (——) without recycling strategy.

In Table 2 we report the sequential elapsed time required to perform the first three nonlinear steps with and without the recycling strategy. It can be seen that the recycling strategy allows to save about 40 % of overall computation time.

	without recycling	with recycling	Time savings (%)
ARPACK	18.24 s	11.09 s	39.2
KS	17.46 s	11.06 s	36.7

Table 2: Total elapsed time required by ARPACK and Krylov-Schur during the first three iterations corresponding to Figure 2 for the computation of the smallest magnitude nonlinear eigenfrequecny ω_1, with and without the recycling strategy.

5.2 Jacobi-Davidson solver

The same experiments are performed using the Jacobi-Davidson method with and without recycling of the $n_{ev} = 5$ eigenvectors associated with the smallest eigenvalues. Figure3 shows the convergence history using the same notation as for the Krylov solvers in the previous section. As for the Krylov solvers, these results show that recycling the spectral information yields a significant improvement in the convergence rate of the Jacobi-Davidson solver.

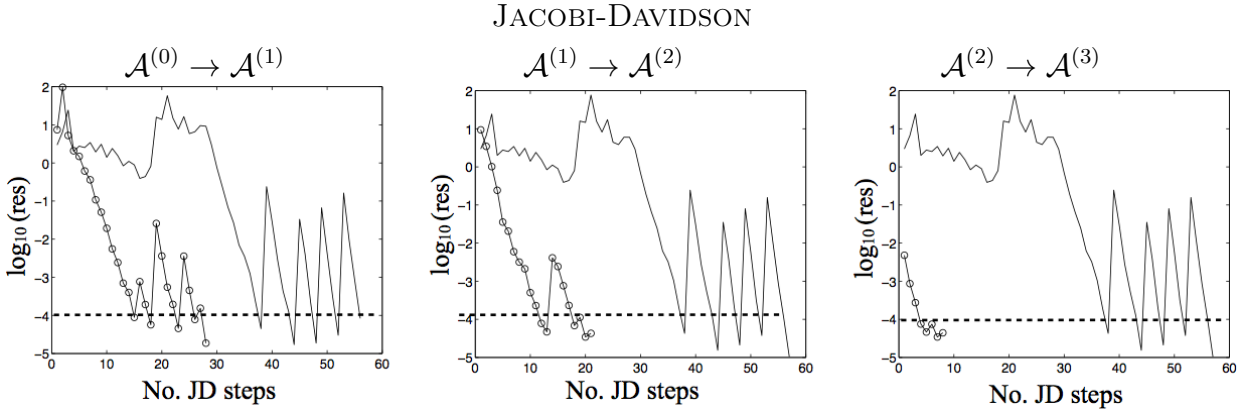

Figure 3: Convergence history of the scaled residuals for the $nev = 5$ smallest magnitude eigenpairs of the linear problems $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$ solved with $\varepsilon = 10^{-4}$. (–o–): with recycling strategy; (——) without recycling strategy.

The convergence speed improvement directly translates into a computational time decrease as it can be observed in Table 3. For this particular case, the saving in time is around 70%.

	without recycling	with recycling	Time savings (%)
Jacobi-Davison solver	13.0 s	3.9 s	70

Table 3: Elapsed for the first three nonlinear iterations for the calculation of the smallest magnitude nonlinear eigenvalue ω_1 with and withouth the recycling strategy implemented in the Jacobi-Davidson solver.

5.3 The block Krylov-Schur method

In this section we compare the convergence history of the block Krylov-Schur algorithm presented in Section 3.2.1 with its single vector counterpart when the spectral information from one nonlinear iteration to the next one is re-injected. For a fair comparison from a memory consumption viewpoint, we consider the same maximal dimension for the search space $m = 80$ for the two solvers. For this comparison only the first two nonlinear iterations are considered as they are enough to illustrate the main trends. For these experiments, we compute the $n_{ev} = 5$ smallest eigenvalues with a target threshold accuracy $\varepsilon = 10^{-5}$. As presented in Section 4.3 for the block solvers, the initial block is formed by an orthonormal basis of the n_{ev} eigenvectors computed at the previous nonlinear iteration (i.e., the ones from $\mathcal{A}^{(1)}$ to solve $\mathcal{A}^{(2)}$ and the ones of $\mathcal{A}^{(2)}$ to solve $\mathcal{A}^{(3)}$). To highlight the weakness of the resulting block solver, the classical single vector Krylov-Schur solver starts from a random vector, which can be considered as a penalty.

The convergence history of the two solvers is displayed in Figure 4. It can be seen that the convergence speed is much slower for the block version due to the lower degree of the equivalent filter polynomial applied at restart. This is particularly visible for the convergence history displayed in the right plot. Although the initial residuals are much smaller, thanks to the recycling mechanism implemented in the block version, this advantage quickly vanishes because of the slow convergence rate.

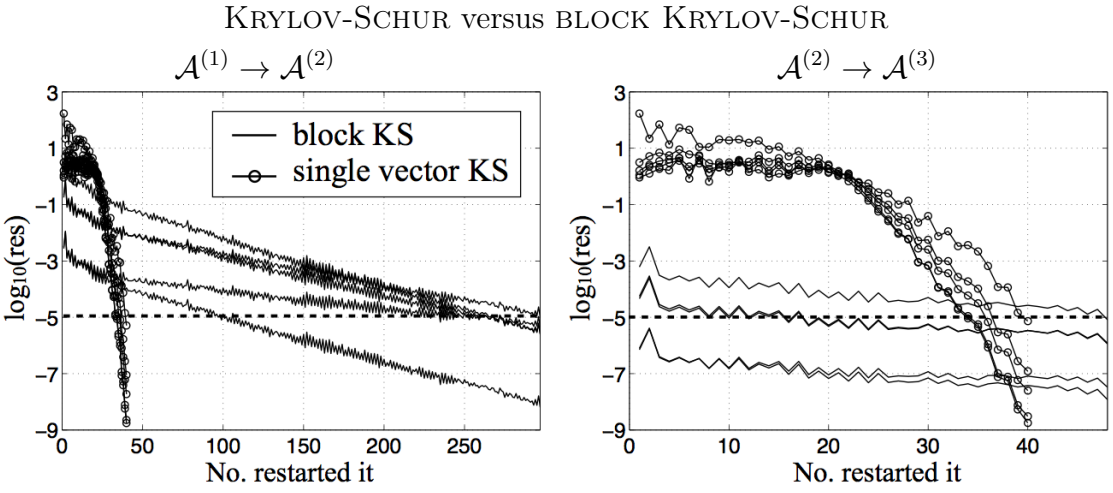

Figure 4: Convergence history of the scaled residuals for the $nev = 5$ smallest magnitude eigenpairs of the linear problems $\mathcal{A}^{(2)}$, and $\mathcal{A}^{(3)}$ solved with $\varepsilon = 10^{-5}$.

The ratios between the computation times as well as the number of restarted iterations are displayed in Table 4, they confirm the results suggested by the convergence histories in Figure 4. The block Krylov-Schur solver is not well suited in the context of this application. However, there are certain aspects that deserve to be mentioned:

1. The AVSP code does not enable to perform matrix-matrix product so that the Ruhe's variant of the block Arnoldi has been implemented. Such an implementation does not permit to benefit from fast calculation thanks to a better data locality (BLAS-3

# nonlinear iteration	$\frac{\text{Time block}}{\text{Time single}}$	$\frac{\text{\# restarts block}}{\text{\# restarts single}}$
2	291s/36s = 8.01	297/40
3	47s/37s = 1.28	48/40

Table 4: Computation time comparison of block and single vector Krylov-Schur methods when solving $\mathcal{A}^{(2)}$ and $\mathcal{A}^{(3)}$. The single vector solver uses a random initial vector, the block variant implement the spectral recycling strategy.

effect). However, the gap in convergence rate makes hopeless to expect that this computational benefit would compensate the poor numerical behavior.

2. The convergence rate is closely related to the block-size p (for a prescribed maximum search space dimension the larger the block-size, the slower the convergence). For the application considered in this work, the block-size can be large in general. Hence the block method is not well suited for the sought purpose. However, the block Krylov-Schur approach can be an interesting solver in other context where the block-size remains small (2 or 3), as for example, cases where the multiplicity of eigenvalues is known a priori [21].

5.4 Subspace Iteration with Chebyshev acceleration

We last investigate the idea of recycling eigenspaces in the framework of the Chebyshev subspace iteration method. For this solver, recycling the available eigenspace is fairly straightforward: the initial block used to start the iterations is an orthonormal basis of the eigenspace computed at the previous nonlinear step. Similarly to the block Krylov-Schur method, this approach is compared to the Krylov-Schur solver when computing the $n_{ev} = 10$ smallest magnitude eigenvalues of $\mathcal{A}^{(2)}$ with $\varepsilon = 10^{-4}$. In this case, both solvers recycle the 10 eigenvectors computed for $\mathcal{A}^{(1)}$. Given the different nature of the numerical parameters of the two solvers, the comparison cannot be performed keeping some of their common key parameters identical. In that context, their parameters have been tuned individually in an attempt to get their best performance. For the Krylov-Schur solver, the maximal size of the search subspace is set to $m = 60$. For block Chebyshev, we set the degree of the polynomial to $k = 150$ and the size of the search subspace is $m = 15 > n_{ev}$, which improves the convergence of the method, as recommended in [15]. Indeed, completing the initial subspace with a few additional random vectors yields a better convergence than just keeping $m = n_{ev}$. Therefore, the initial subspace formed by an orthonormal basis of the eigenvectors computed for $\mathcal{A}^{(1)}$, is extended with 5 linearly independent random vectors.

The comparative study has been performed using both a Matlab and a Fortran implementation of both solvers, that enable to highlight some computational features that we could not implement in the AVSP framework. The results are reported in Table 5. The row entitled “Iteration" has different meaning for each solver. For the block Chebyshev method, it represents the number of times the polynomial is applied; for the Krylov-Schur it corresponds to the number of restarts. The row “Time ratio" gives the ratio between the elapsed

time of the Krylov-Schur solver divided by the elapsed time of the block Chebyshev time. The different number of subspace iterations that can be observed between the Fortran and the Matlab implementation of the block Chebyshev solver are due to the different random number generators used in the two languages; this difference leads to slightly different convergence behaviors. The number of required matrix-vector products in the case of the block

	Fortran		Matlab	
	Krylov-Schur	block Chebyshev	Krylov-Schur	block Chebyshev
Iteration	45	11	45	13
# matvec	2237	24915	2237	29445
Time ratio	$\frac{3.50}{17.48} = 0.2$		$\frac{22.9}{4.32} = 5.3$	

Table 5: Computational cost for computing the ten smallest magnitude linear eigenpairs of $\mathcal{A}^{(2)}\bar{p} = -{\omega^{(2)}}^2\bar{p}$ with Krylov-Schur and block Chebyshev solvers for both the Fortran and the Matlab implementations. Both solvers implement the spectral recycling strategy.

Chebyshev method is more than 10 times the number of matrix-vector products required by the Krylov-Schur solver. This difference translates differently in terms of computation time between the two implementations, as the Chebyshev polynomial calculation is performed differently. The AVSP code is based on a matrix-free approach, so that the Fortran implementation, only one matrix-vector product can be performed at a time. For the Matlab implementation the matrices were extracted from the AVSP code and stored in a sparse format compatible with Matlab. Consequently, the Matlab implementation performs sparse matrix-matrix products so that the filter polynomial can be applied simultaneously to all the m vectors. In Matlab, the matrix-vector products are very effective and their relative cost compared to the other numerical kernels is lower; consequently the block Chebyshev solver is more efficient than the Krylov-Schur solver. On the other hand, the situation is the opposite in Fortran because the matrix-free matrix-vector product is much more expensive than in Matlab.

5.5 Recycling eigensolutions for a 3D industrial application with complex geometry

In this section we investigate the benefit of using recycling ideas for the parallel solution of a large problem arising from an industrial case. The geometry considered for the experiments corresponds to a full annular industrial gas turbine combustor, formed by 24 burners circumferentially arranged. The mesh, displayed in Figure 5, is composed of $n = 1,782,384$ vertices. For these experiments, only the solvers that have shown themselves as best suited for an efficient implementation in the matrix-free AVSP code are used, namely, P-ARPACK for the IRA method and in-house implementations of the Krylov-Schur and Jacobi-Davidson approaches. The $n_{ev} = 10$ smallest eigenvalues are computed with a demanded accuracy on the scaled residual of $\varepsilon = 10^{-4}$. The maximal size of the search subspace is set to $m = 120$ for the three eigensolvers, so that the study is roughly iso-memory. We mention that we do

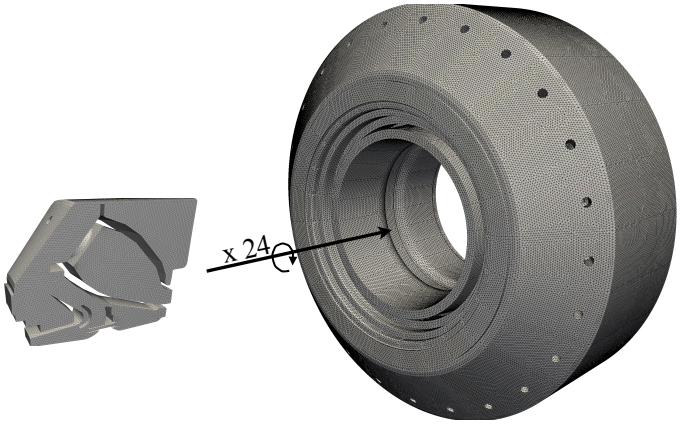

Figure 5: Mesh used for the discretization of an industrial gas turbine combustor. The number of nodes is $n = 1,782,384$.

not account for the extra memory used in the Jacobi-Davidson method where the correction equation is solved using a few full-GMRES [16] iterations.

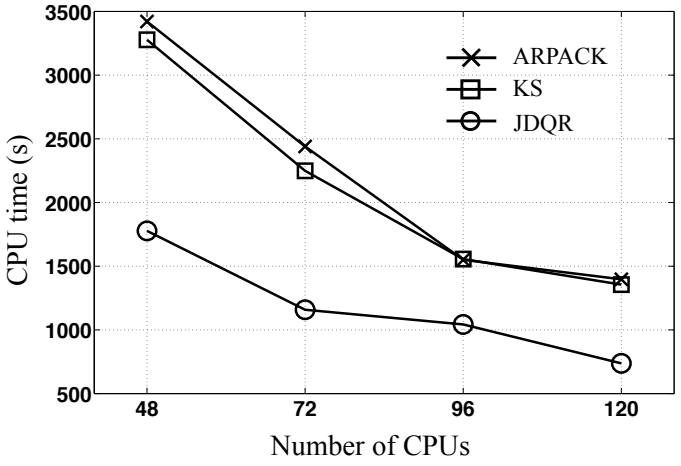

Figure 6: Scalability study: evolution of the computational cost as a function of the number of cores for (P)ARPACK, KS and JDQR solvers ($n_{ev} = 10$, $m = 120$ and $\varepsilon = 10^{-4}$ for the three eigensolvers).

We first illustrate the parallel efficiency of these solvers. For the three approaches the parallelism relies on a mesh-partitioning technique that allows to efficiently implement, on top of MPI, the most time consuming kernel, that is matrix-free matrix-vector calculation. All the numerical calculations performed on either the Hessenberg matrix or the Rayleigh quotient are performed redundantly to reduce the communication among the MPI processes. Figure 6 displays the strong scalability behavior of the three eigensolvers when the number of cores is varied from 48 to 120. For this experiment, the most simple case is solved (neither combustion nor complex boundary conditions are considered), leading to a linear eigenproblem. Table 6 displays the efficiency for each number of cores (taking 48 cores as the reference), the efficiency is computed as:

$$\text{Efficiency} = \frac{\text{time (48 cores)}}{\text{time (p cores)}} \frac{48}{p}.$$

In the ideal case where the algorithms would scale perfectly the efficiency should be constant and equal to one. The results displayed in Table 6 show that the parallel implementations of the three algorithms exhibit very strong scalability capabilities. We can even observe some super-linear effect with efficiency larger than one, that are most likely due to memory hierarchy effects.

	72 cores	96 cores	120 cores
PARPACK	0.93	1.1	0.98
Krylov-Schur	0.97	1.05	0.97
Jacobi-Davidson	1.02	0.85	0.96

Table 6: Strong scalability efficiency of the three algorithms when the number of MPI processes is varied.

For this example, the results of Figure 6 show that the implementations of the Krylov-Schur solver is slightly faster than ARPACK while the Jacobi-Davidson solver is noticeably more effective.

in what follows, we evaluate the benefit of recycling spectral information between the nonlinear steps on this large real life problem. Consequently the problem with combustion is considered, so that the nonlinear eigenproblem $\mathbf{A}\bar{p} + \omega^2\bar{p} = \mathbf{C}(\omega)\bar{p}$ has to be solved. Starting from the 6^{th} smallest frequency obtained for the problem without combustion, the linear eigenproblems corresponding to the first three nonlinear iterations are solved with and without recycling the eigensolutions obtained at previous iterations. The convergence history of the scaled residual obtained at each nonlinear step are displayed for the three eigensolvers in Figure 7. In red are plotted the convergence history of the scaled residuals obtained when the problem is solved starting from a random vector, while in blue appear the convergence history when the recycling strategy is used.

The gain due to the recycling of solutions is obvious, looking at the convergence history in Figure 7. For the sake of completeness, we report in Table 7 the parallel elapsed time required for solving each problem on 72 cores, with (rec) and without (rand) recycling. The amounts of time saved thanks to the recycling mechanism are remarkable. On this test case, the comparison between the three solvers ends up with a clear winner : the Jacobi-Davidson method is the fastest one. Furthermore, it is also the one that exploits the spectral recycling in the most efficient way.

6 Conclusion

In this work, we have considered the solution of nonlinear eigenproblem arising from the study of thermoacoustic instabilities using a Helmholtz solver is treated in the present work. A fixed point iterative scheme is used for its solution, which results in a sequence of linear eigenproblems that must be solved obtain one solution of the nonlinear one. For thermoacoustic simulations, the nonlinear iterations converge quickly so that the solutions obtained

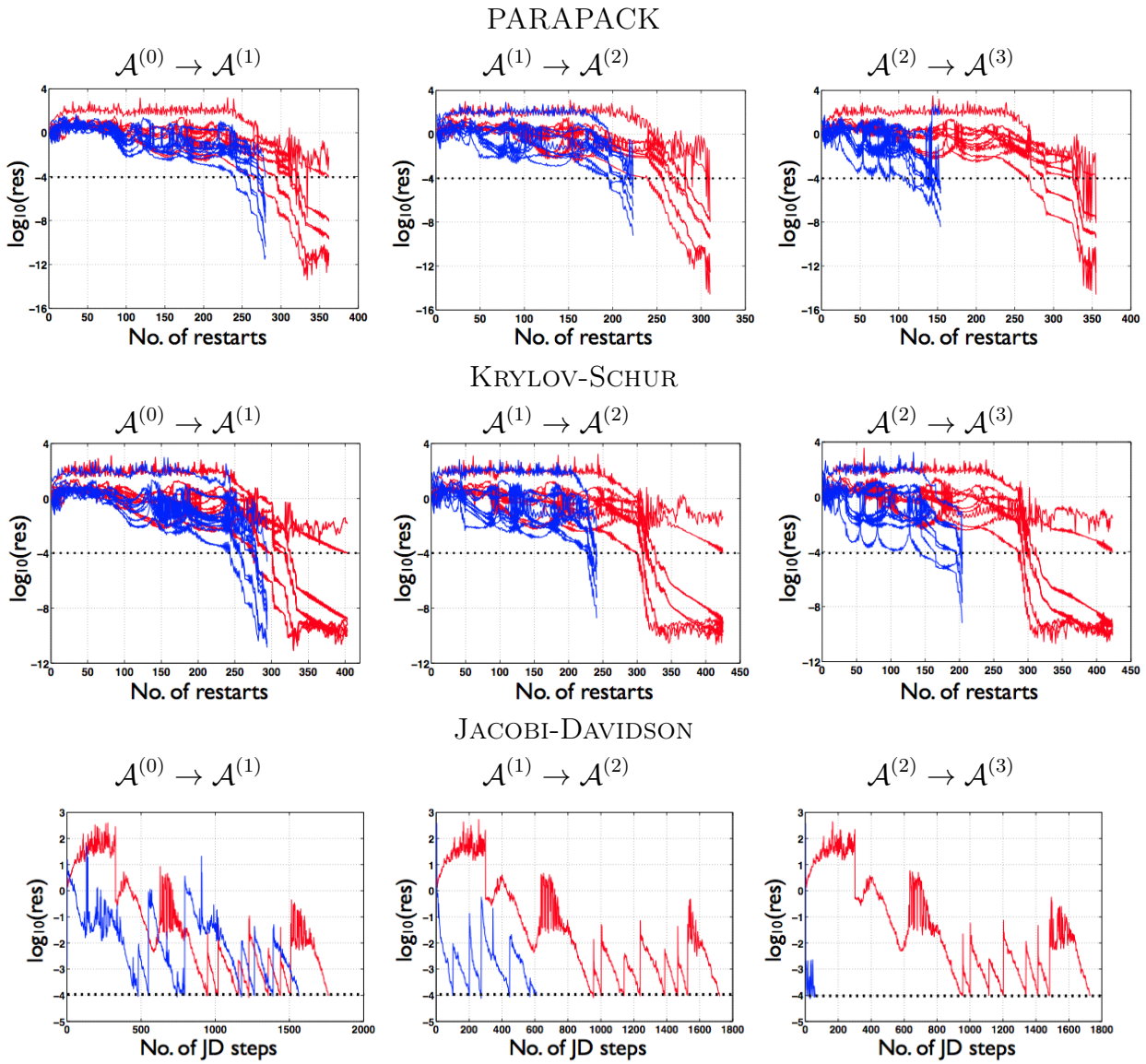

Figure 7: Convergence history of the scaled residuals for three nonlinear iterations with $n_{ev} = 10$, $m = 120$ and $\varepsilon = 10^{-4}$ - Blue witth recycling, Red without recycling.

	$\mathcal{A}^{(1)}$		$\mathcal{A}^{(2)}$		$\mathcal{A}^{(3)}$		Total	
	rand	rec	rand	rec	rand	rec	rand	rec
PARPACK	6040	4842	5162	3802	5988	2625	17190	11269
Krylov-Schur	6874	5122	7152	4057	7044	3852	21070	13031
Jacobi-Davidson	3150	2788	3067	1128	3079	130	9296	4046

Table 7: Parallel elapsed time on 72 cores to perform 3 nonlinear iterations for $n_{ev} = 10$, $m = 120$ and $\varepsilon = 10^{-4}$

for each linear problems are good approximations to the solution of the next one. This paper concentrates on recycling techniques allowing to reuse the eigensolutions obtained at previous nonlinear iterations to accelerate the solution of the next one, when using different state of-the-art eigensolvers: the Implicitly Restarted Arnoldi (IRA) method, the Krylov-Schur method and its block variant, the Jacobi-Davidson solver and the Subspace Iteration method with Chebyshev acceleration. The main features of these eigensolvers have been described, allowing to understand how eigensolutions are recycled depending on the chosen eigensolver.

A small eigenproblem has been used to illustrate which combinations of recycling technique and eigensolver are the best suited in the present numerical context. The retained eigensolvers, namely, the IRA method (implemented in ARPACK), the Krylov-Schur solver and Jacobi-Davidson are then used on a realistic industrial case to compute the thermoacoustic modes of a full annular gas turbine combustion chamber. The size of the associated eigenproblem is about $n = 2 \cdot 10^6$, which requires parallel implementations of the said eigensolvers, whose efficiency is also studied. The results concerning this industrial example are clear: the use of the simple recycling techniques here proposed allow reduced the computation time up to the to 50%, highlighting the computational savings that one can expect to attain by recycling the spectral information during the fixed point procedure. An exhaustive presentation of the numerical experiments and detailed description of the different algorithms can be found in [13]. Although the spectral recycling has been described in a nonlinear framework, the ideas introduced in this work can obviously be extended to other contexts. A good example may be the case of parametric studies where an important amount of eigenproblems close to each other has to be solved, as in [11].

References

[1] W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math*, 9(1):17–29, 1951. 5

[2] Friedrich L Bauer. Das verfahren der treppeniteration und verwandte verfahren zur lösung algebraischer eigenwertprobleme. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 8(3):214–235, 1957. 8

[3] Sébastien Candel. Combustion dynamics and control: progress and challenges. *Proceedings of the combustion institute*, 29(1):1–28, 2002. 1

[4] L. Crocco. Aspects of combustion instability in liquid propellant rocket motors. Part I. *J. American Rocket Society* , 21:163–178, 1951. 2

[5] L. Crocco. Aspects of combustion instability in liquid propellant rocket motors. part II. *J. American Rocket Society* , 22:7–16, 1952. 2

[6] F. E. C. Culick and P. Kuentzmann. *Unsteady Motions in Combustion Chambers for Propulsion Systems*. NATO Research and Technology Organization, 2006. 1

[7] D.R. Fokkema, G.L.G. Sleijpen, and H.A. Van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM Journal on Scientific Computing*, 20:94, 1998. 8

[8] R. Lehoucq and D. Sorensen. Arpack: Solution of large scale eigenvalue problems with implicitly restarted arnoldi methods. www.caam.rice.edu/software/arpack. User's guide, 1997. 5

[9] R.B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly re-started Arnoldi iteration. *SIAM J. Matrix Anal. Appl*, 17:789–821, 1996. 5

[10] T. Lieuwen and B. T. Zinn. The role of equivalence ratio oscillations in driving combustion instabilities in low nox gas turbines. *Proc. Combust. Inst.* , 27:1809–1816, 1998. 1

[11] F. Merz, C. Kowitz, E. Romero, J.E. Roman, and F. Jenko. Multi-dimensional gyrokinetic parameter studies based on eigenvalue computations. *Computer Physics Communications*, 2011. 21

[12] F. Nicoud, L. Benoit, C. Sensiau, and T. Poinsot. Acoustic modes in combustors with complex impedances and multidimensional active flames. *AIAA J.* , 45:426–441, 2007. 2

[13] Salas P. *Numerical and physical aspects of thermoacoustic instabilities in annular combustion chambers - TH/CFD/13/85*. PhD thesis, Université Bordeaux 1 - INRIA, 2013. phd. 21

[14] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. Third Edition (www.cerfacs.fr/elearning), 2011. 1, 2

[15] Y. Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992. 8, 9, 16

[16] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. 7:856–869, 1986. 18

[17] G.L.G. Sleijpen and H.A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Review*, pages 267–293, 2000. 8

[18] G.L.G. Sleijpen, H.A. van der Vorst, and E. Meijerink. Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems. *Electron. Trans. Numer. Anal*, 7:75–89, 1998. 8

[19] G. W. Stewart. A Krylov–Schur Algorithm for Large Eigenproblems. *SIAM J. Matrix Anal. Appl.*, 23(3):601–614, March 2001. 5

[20] G.W. Stewart. *Matrix Algorithms: Eigensystems*, volume 2. Society for Industrial Mathematics, 2001. 4, 5, 6, 8, 10

[21] Y. Zhou and Y. Saad. Block Krylov–Schur method for large symmetric eigenvalue problems. *Numerical Algorithms*, 47(4):341–359, 2008. 6, 16