Comparative study of inner–outer Krylov solvers for linear systems in structured and high-order unstructured CFD problems

Mehdi Jadoui, Christophe Blondeau, Emeric Martin, Florent Renac, François-Xavier Roux

Comparative study of inner-outer Krylov solvers for linear systems in structured and high-order unstructured CFD problems

Mehdi Jadoui[a], Christophe Blondeau[a,*], Emeric Martin[a], Florent Renac[a], François-Xavier Roux[b]

[a] *ONERA, Université Paris-Saclay, F-92322 Châtillon, France*
[b] *Laboratory of Jacques-Louis Lions, Sorbonne University, F-75005 Paris, France*

Abstract

Advanced Krylov subspace methods are investigated for the solution of large sparse linear systems arising from stiff adjoint-based aerodynamic shape optimization problems. A special attention is paid to the flexible inner-outer GMRES strategy combined with most relevant preconditioning and deflation techniques. The choice of this specific class of Krylov solvers for challenging problems is based on its outstanding convergence properties. Typically in our implementation the efficiency of the preconditioner is enhanced with a domain decomposition method with overlapping. However, maintaining the performance of the preconditioner may be challenging since scalability and efficiency of a preconditioning technique are properties often antagonistic to each other. In this paper we demonstrate how flexible inner-outer Krylov methods are able to overcome this critical issue. A numerical study is performed considering either a Finite Volume (FV), or a high-order Discontinuous Galerkin (DG) discretization which affect the arithmetic intensity and memory-bandwith of the algebraic operations. We consider test cases of transonic turbulent flows with RANS modelling over the two-dimensional supercritical OAT15A airfoil and the three-dimensional ONERA M6 wing. Benefits in terms of robustness and convergence compared to standard GMRES solvers are obtained. Strong scalability analysis shows satisfactory results. Based on these representative problems a discussion of the recommended numerical practices is proposed.

Keywords: Krylov solver, GMRES, adjoint problem, deflation, variable preconditioning

1. Introduction

One of the most important environmental challenge that aircraft manufacturers seek to overcome is the limitation of the carbon footprint in the atmosphere. Many technological improvements have been achieved to reduce harmful emissions. One of them concerns the improvement of the efficiency of the

*Corresponding author
Email addresses: Mehdi.Jadoui@onera.fr (Mehdi Jadoui), Christophe.Blondeau@onera.fr (Christophe Blondeau), Emeric.Martin@onera.fr (Emeric Martin), Florent.Renac@onera.fr (Florent Renac), roux@ann.jussieu.fr (François-Xavier Roux)

airfoil that is typically quantified by the lift to drag ratio. More specifically, one seeks to minimize an aerodynamic function with respect to a set of design variables that controls the shape of the airfoil. Gradient-based methods are the most suitable optimization algorithms to address such problems. The efficient computation of the sensitivities is consequently an essential step to obtain a relevant design. In aerodynamic design the adjoint formulation allows the computation of derivatives of several functions with respect to many design parameters at an affordable cost. However, high-fidelity analyses are now routinely applied even at the preliminary design stage and it becomes increasingly challenging to solve the resulting algebraic systems with regard to the size and the condition number of such problems. At this stage, it is obvious that the accuracy of the gradient computation depends on the accuracy of the solution of the linear system.

In spite of the rise of supercomputers and their high performance computing, solving linear systems is still a substantive issue in several scientific fields like computational fluid dynamics (CFD). Indeed, predicting the best design requires models of compressible turbulent flows in order to capture complex physical phenomena. From an algebraic point of view, this leads to large, sparse and ill-conditioned systems. Matrices are real, non symmetric, not positive definite, with a block-wise structure and a symmetric pattern. A robust way to solve this kind of systems relies on Krylov subspace methods, an iterative procedure which computes an approximation of the solution in a projection subspace called Krylov subspace. For non-symmetric problems, we focus on the Generalized Minimal Residual (GMRES) algorithm [1]. However, due to the prohibitive memory cost of the GMRES, the restarted version denoted as GMRES(m) is practically used, with m the maximum number of basis vectors allowed during a cycle. Once m iterations are carried out, the vectors are discarded and a new set of vectors is generated from scratch. The counterpart is the loose of the super-linearity property of the GMRES and therefore a convergence slowdown may be observed compared to the full GMRES without restart. Numerical ingredients as preconditioning techniques and spectral deflation have been therefore introduced to offset the restart effect. Generally speaking, preconditioners attempt to improve the spectral properties of the operator of the system by clustering as much as possible the eigenspectrum of the initial system. Benzi [2] surveyed different preconditioning techniques for the iterative solution of large linear systems. For instance, he mentioned the incomplete factorization preconditioner and its block variant denoted by ILU and BILU respectively. The ILU algorithm seems to be attractive thanks to its robustness and as a result is used as a preconditioner in most cases. But some limitations exist about stability and scalability. Stability is in fact affected by many parameters like the condition number, and the structure of the coefficient matrix. For our case, instability issues are avoided since the preconditioning operators have good algebraic properties. The BILU algorithm will be employed throughout this paper. Another class

of well-known iterative preconditioners is the Lower-Upper Symmetric Successive Over Relaxation (LU-SSOR) preconditioner combining the LU factorization with an SSOR relaxation [3]. However, convergence of iterative techniques may be hampered by the lack of diagonal dominance of the associated operator.

A seminal contribution to efficiency of preconditioned GMRES was the extension to variable preconditioning [4]. This gives GMRES a great flexibility to use advanced iterative solvers essentially as preconditioners which may themselves in turn consist in Krylov subspace methods. When it comes to use GMRES as a preconditioner, the method is called Flexible inner-outer GMRES or nested GMRES. However, it could be even difficult for the nested GMRES methods to address some extreme ill-conditioned systems. A solution consists in using the spectral deflation strategy. It aims at accelerating the convergence when preconditioned FGMRES fails to converge. This stagnation is essentially caused by the presence of eigenvalues in the neighborhood of zero. The idea is to find a way to remove their harmful effect from the eigenspectrum of the preconditioned system. Morgan [5] proposed to deflate the harmonic Ritz vectors from the next Krylov subspace since they are a good approximation of eigenvectors associated with the smallest eigenvalues. The harmonic Ritz vectors are formulated as a linear combination of Krylov vectors whose coefficients are solutions of a generalized eigenvalue problem [6]. It leads to a new algorithm denoted as GMRES-DR(m,k) (GMRES with Deflating Restarting) with k the number of selected harmonic Ritz vectors per cycle. Giraud et al. [7] proposed to generalize this algorithm to a flexible formulation, FGMRES-DR(m,k), where the preconditioner changes at each Arnoldi iteration. They found effective to deflate eigenvalues that are located at a distance away from specific clusters. In the field of aerodynamic shape optimization, few research has been conducted regarding the inner-outer GMRES method. In a previous work, Pinel and Montagnac [8] investigated the nested GMRES method as well as its block variant for the solution of a discrete adjoint problem. The latter problem results from the linearization of the discretized Navier-Stokes equations combined with the two-equation turbulence model of Wilcox. We point out that the eddy viscosity is kept constant during the linearization. They have proved that the block nested GMRES overcomes the scalar version with a factor of 3 in terms of CPU time. In addition to that, the same performance is reached with lower memory footprint for the block nested GMRES. Chen et al. [9] recently tested both GMRES-DR and GCRO-DR (Generalized Conjugate Residual with inner Orthogonalization and Deflated Restarting)[10] solvers by proposing an extension in terms of dynamic deflated restarting. The results demonstrate the effectiveness of this strategy on adjoint systems in terms of memory cost and CPU time.

This paper aims at investigating the flexible inner-outer GMRES solver capabilities to solve adjoint systems for turbulent flow by fully linearizing the turbulence model. Actually, only a handful of authors have examined this option due to the following reason [11, 12, 13, 14] (see the review [15] and references

therein). Linearizing the turbulence model leads generally to very ill-conditioned adjoint systems demanding the development of advanced linear solvers combined with others numerical artifices. However, considering the turbulence model during the linearization is crucial when it comes to solve a high-fidelity optimization problem. So, instead of making physical simplifications, a special focus on nested GMRES solvers associated with spectral deflation and preconditioning techniques has been achieved. In contrast to [8], a special attention to the block incomplete LU factorization (BILU) is conferred since it is proved to be more robust than the standard block Lower-Upper Symmetric Gauss-Seidel (LU-SGS) relaxation algorithm. LU-SGS is a particular case of LU-SSOR with a unity relaxation factor. The reader should refer to [16] for further detail about LU-SGS as an iterative solver for the implicit phase of Backward-Euler Newton steady solvers. Numerical experiments have shown that convergence is strongly affected depending on the choice of the preconditioner. We also point out that we recently implemented the GCRO-DR and a comparative study with GMRES-DR has shown that performances are strictly equivalent. For these reasons, the current work has been focused only on the nested GMRES solver. Considering two different discretization methods, such as FV and DG formulations, contributes to draw a better overview of the benefits of such a solver. Not only it affects conditioning of adjoint systems but also the arithmetic intensity and the memory-bandwidth pressure of basic linear algebra operations.

This paper is organized as follows. In section 2, we briefly present the adjoint problem for a sensitivity analysis. In section 3, we describe the minimal residual norm Krylov subspace methods in conjunction with deflation strategies and flexible preconditioning. Section 4 and 5 are devoted to numerical experiments where the robustness of flexible inner-outer Krylov solvers is demonstrated through representative test-cases for both FV [1] and DG [2] schemes. Strong scalability capability of such solvers is also assessed. To finis with, some numerical considerations are discussed in section 6.

2. Discrete adjoint method for sensitivity analysis

2.1. Steady-state problem

Let $\mathcal{D}$ be a bounded domain in $\mathbb{R}^3$. We denote by $\mathbf{W} \in \mathbb{R}^N$ the vector of conservative variables. The differential form of the governing equations for a viscous fluid governed by the RANS model is formulated as

$$\frac{d\mathbf{W}}{dt} + \nabla.\mathbf{F} = \mathbf{Q}, \tag{1}$$

[1] FV simulations were performed with the ONERA elsA software [17]

[2] DG simulations were performed with the ONERA Aghora software [18]

where $\mathbf{F}$ embeds the convective and diffusive fluxes and $\mathbf{Q}$ is a source term. We want to obtain the field of conservative variables $\mathbf{W}$ related to a steady-state equilibrium on domain $\mathcal{D}$. The system of equations is formulated in its discrete residual form as:

$$\mathbf{R}(\mathbf{W}, \mathbf{X}) = \mathbf{0}, \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^N$ is the vector of fluid grid coordinates.

2.2. Sensitivity analysis via the adjoint approach

Consider an aerodynamic shape optimization problem of the form $\mathcal{J}^* = \min_{\alpha} \mathcal{J}(\mathbf{W}(\alpha), \mathbf{X}(\alpha))$ under the constraints $\mathbf{R}(\mathbf{W}(\alpha), \mathbf{X}(\alpha)) = \mathbf{0}$, where α is the vector of design variables controlling the shape of the body. A gradient-based method is required to efficiently minimize $\mathcal{J}$. In aerodynamics, the adjoint technique is the method of choice to compute the sensitivities [19, 15]. The adjoint linear system is written as:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right]^T \lambda = -\frac{\partial \mathcal{J}^T}{\partial \mathbf{W}} \tag{3}$$

where the superscript ^{T} stands for the transpose operator. In the remainder of this paper, the transpose of the exact flux Jacobian matrix will be denoted $A = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{W}}\right]^T$.

3. Minimal residual Krylov subspace methods combined with spectral deflation

In this section we focus on a particular minimal residual norm Krylov subspace method for the solution of linear systems with a non-symmetric real coefficient matrix of type

$$Ax = b, \qquad A \in \mathbb{R}^{N \times N}; \quad b,\ x\ \in \mathbb{R}^N \tag{4}$$

A right-preconditioned system is considered so that system (4) becomes

$$A\mathcal{M}(t) = b, \tag{5}$$

$$x = \mathcal{M}(t) \tag{6}$$

with $t \in \mathbb{R}^N$ and $\mathcal{M} : \mathbb{R}^N \to \mathbb{R}^N$ the preconditioning operator which may be a nonlinear function.

3.1. Flexible GMRES algorithm with right preconditioning

Saad has proposed a minimal residual norm subspace method based on the standard GMRES approach [1] that allows a variable nonlinear preconditioning function $\mathcal{M}_j : \mathbb{R}^N \to \mathbb{R}^N$ at each iteration j, [4].

Starting from an initial guess x_0, the flexible Arnoldi relation is written as:

$$AZ_m = V_{m+1}\bar{H}_m, \tag{7}$$

where the matrices $V_{m+1} \in \mathbb{R}^{N\times(m+1)}$, $Z_m \in \mathbb{R}^{N\times m}$ and $\bar{H}_m \in \mathbb{R}^{(m+1)\times m}$ stand for the orthonormal basis of the Krylov space, the solution space and the upper Hessenberg matrix respectively. The approximate solution is written as $x_m = x_0 + Z_m y_m$ where y_m minimizes $||r_0 - AZ_m y||_2$ over $x_0 + span\{Z_m\}$, with $Z_m = \mathcal{M}V_m = [\mathcal{M}_1(v_1), ..., \mathcal{M}_m(v_m)]$ and both Z_m and V_m need to be stored. We point out that the operator $\mathcal{M}$ represents the action of the nonlinear operators $\mathcal{M}_j$ on the set of basis vectors v_j.

The restarted FGMRES(m,m_i) algorithm is presented in Algorithm 1. We denote by m_i the size of the Krylov subspace associated to the GMRES solver devoted to the inner linear system. We point out that the stopping criterion is essentially based on the true relative residual defined by $\frac{||b-Ax||}{||b||}$ (see [20] for several definitions of stopping criteria and some practical considerations for the implementation of the key points of the algorithm). Still from an implementation point of view, the relative least-squares residual is currently used as a cheap approximation of this quantity at each Arnoldi iteration.

Algorithm 1 FGMRES(m,m_i)

1: Choose an initial guess x_0 and a convergence threshold ϵ
2: Compute $r_0 = b - Ax_0$, $\beta = ||r_0||$ and $v_1 = r_0/\beta$;
3: c = $[\beta, 0...0]^T \in \mathbb{R}^{m+1}$
4: **for** $j = 1, ..., m$ **do**
5: $z_j = \mathcal{M}_j(v_j)$
6: $w = Az_j$
7: **for** $i = 1, ...j$ **do**
8: $h_{i,j} = v_i^T w$
9: $w = w - h_{i,j}v_i$
10: **end for**
11: $h_{j+1,j} = ||w||$ and $v_{j+1} = w/h_{j+1,j}$
12: Solve the least-squares problem $min_y||c - \bar{H}_j y||$ for y^*
13: Exit if $||c - \bar{H}_m y^*||/||b|| \leq \epsilon$
14: **end for**
15: Compute $x_m = x_0 + Z_m y^*$ where $Z_m = [z_1, ..., z_m]$
16: Set $x_0 = x_m$ and go to 2

The relative true residual is only computed at the end of each cycle and is used to construct the first

vector of the next Krylov subspace basis. In the case of large and ill-conditioned linear systems, least-squares and true residuals may differ due to loss of orthogonality during the construction of the Krylov basis. A standard way to tackle such a phenomenon is to ask for a second iteration of the Modified Gram-Schmidt algorithm (loop from line 7 to 10) in order to reinforce the orthogonality of the Krylov basis. In this paper, the nested GMRES strategy is adopted for the numerical experiments. Two preconditioning strategies are considered for the inner GMRES. The first one consists in a block version of a standard LU-SGS iterative solver. LU-SGS is applied to a first order diagonally dominant upwind approximation of the flux Jacobian matrix inspired by [3]. This operator is based on a first order spatial discretization of the convective and of the viscous fluxes using a simplifying thin layer assumption [16]. This strategy leads to a very compact stencil for the preconditioning matrix which will be denoted by $\mathbf{J}_{O1}^{APP}$ in the sequel of the paper. The second one is a Block Incomplete LU (BILU(k)) factorization applied to either an approximate or exact flux Jacobian matrix. For the so-called first-order exact Jacobian matrix $\mathbf{J}_{O1}^{EX}$ a first-order spatial Roe scheme is used for the discretization of the mean-flow convective fluxes and a 5-point corrected centered discretization scheme is used for the diffusive fluxes. More specifically, the spatial gradients at the cell interfaces are modified to avoid high frequency oscillations (see [21] or [22]). The BILU(k) preconditioner will be applied either to the first-order approximate Jacobian matrix $\mathbf{J}_{O1}^{APP}$, or to the first-order exact Jacobian matrix $\mathbf{J}_{O1}^{EX}$. $\mathbf{J}_{O1}^{EX}$ is different from $\mathbf{J}_{O1}^{APP}$ when it comes to memory footprint. More specifically, $\mathbf{J}_{O1}^{EX}$ has a 9-point stencil in 2D whereas a 5-point stencil is associated with $\mathbf{J}_{O1}^{APP}$. In 3D, we have a 7-point stencil for $\mathbf{J}_{O1}^{APP}$ and a stencil of 19 points for $\mathbf{J}_{O1}^{EX}$. Consequently, a better robustness is achieved but at the price of about twice the storage for $\mathbf{J}_{O1}^{EX}$ compared to $\mathbf{J}_{O1}^{APP}$. For the high-order DG formulation, an exact third-order Jacobian matrix $\mathbf{J}_{O3}^{EX}$ is built. Only the BILU(0) preconditioner is considered in the latter case. The size of the matrix reads $N_{elt}N_{eq}N_p$ where N_{elt}, N_{eq}, and N_p represent the numbers of mesh elements, of equations, and of degrees of freedom per equation respectively. We have $N_p = 1$ for the FV computations and either $N_p = 6$ in 2D, or $N_p = 10$ in 3D for the DG computations. Ignoring boundary conditions, the number of nonzero entries in the matrix can be estimated through $NNZ = (1 + 2sd)N_{elt}(N_{eq}N_p)^2$ where $1 + 2sd$ is the number of points in the stencil and d the space dimension. For the DG computations, we always have $s = 1$, but $N_p > 1$ which usually results in larger NNZ values compared to FV even when N_{elt} is lower. Both matrices are sparse and contain dense blocks of size $N_{eq}N_p$ which will have a strong impact on the performances of the linear solvers as shown in the numerical experiments.

We point out that the relevant numerical ingredients that characterize the GMRES algorithm are the matrix product (step 5 in Algorithm 1), the preconditioning step (step 4) and the scalar product (step 7). These algebraic operations are global in conjunction with a domain decomposition method.

More specifically, the globalization of the preconditioner (step 4) is achieved with a Restricted Additive Schwarz method [23]. In addition, the product by the operator A (step 5) is exact. We thus get a global and parallel FGMRES(m,m_i).

3.2. Domain decomposition-based preconditioners

We now describe two stationary block preconditioners that are used in this work and that have been enhanced with overlapping techniques. The first one is a combination of a Restrictive Additive Schwarz (RAS) domain decomposition method and of a LU-SGS relaxation strategy. It will be applied only to the FV problem. The second one a is a BILU(0) factorization over and extended partition of the fluid domain by some level of overlapping and will be applied to the DG problem.

3.2.1. Overlap Restrictive Additive Schwarz LU-SGS preconditioner

The LU-SSGS solver is used as a stationary preconditioner for the structured multi-block FV adjoint solution. It consists in an iterative technique combining a LU (Lower-Upper) factorization with a relaxation method [24]. In practice, we set the LU-SSOR relaxation parameter ω to 1, and the method is then referred to as LU-SGS. In the context of preconditioning, we are interested in the approximate solution of the linear system $Mx = v$. In the factorization step, the implicit operator M is decomposed into a lower triangular matrix L, a diagonal matrix D and an upper triangular matrix U: $M = L + D + U$.

The solution of the system is then approximated by relaxation [16, 3], performing forward and backward successive sweeps. Each relaxation cycle involves two stages:

$$(U + D)x^{p+\frac{1}{2}} = v - Lx^p \tag{8}$$

$$(L + D)x^{p+1} = v - Ux^{p+\frac{1}{2}} \tag{9}$$

where $p \in \mathbb{N}$ is index of the current cycle, starting with $x^0 = 0$. For one cycle of relaxation we can explicit the corresponding preconditioning operator as $M_{SGS} = (U + D)D^{-1}(L + D)$. It is worth mentioning that in the preconditioning context, typically 2 to 4 cycles (of 2 sweeps) are enough to provide a useful approximate solution. Besides, as we deal with block-structured problems, a block LU-SGS algorithm has been implemented, i.e., D is block-diagonal. As already mentioned, the matrix $\mathbf{J}_{O1}^{APP}$ is diagonally dominant by construction. However, when the turbulence model is linearized our numerical experiments have shown that it is beneficial to add a diagonal of the form $I/\Delta\tau$ where $\Delta\tau$ is consistent with a standard local pseudo-time step typically introduced in the implicit stage of a nonlinear inexact Newton steady solver. The local pseudo-time step is computed from a prescribed target Courant-Friedrichs-Lewy (CFL) number and from the steady-state flow field. This requires a trial and error strategy to choose the right

CFL number but, as we will see, an appropriate CFL can dramatically accelerate the convergence of the adjoint solver.

In a parallel multi-domain coupling context, the part of the right-hand side of (8, 9) associated with overlapping cells is updated before each stage in order to account for the information from connected blocks. For the current stage, depending on the ordering of the blocks, the right-hand side is built from the solution at the previous stage or from the blocks whose solution is already up-to-date. As we deal with first order flux Jacobian matrices, the size of the overlap is one. In the remainder of this paper the notation LUSGS(N,CFL) refers to N stages of sweep, that is, N is an even number.

3.2.2. Improved overlap BILU(0) preconditioner

For the unstructured DG adjoint solution, the BILU(0) preconditioner used in the inner GMRES solver is combined with the RAS method. Such a strategy aims to enhance the convergence of the inner system when the number of partitions increase and therefore the overall convergence of the outer system. Interactions between neighboring partitions are taken into account in order to enlarge the slice of the global implicit matrix that is carried out per partition. As the DG formulation requires a compact stencil of size one, we only need to exchange blocks from our direct neighbors by MPI. The BILU(0) algorithm is then applied on this slice to construct an enlarged local preconditioning matrix. To apply the preconditioning step on a global vector, a restriction operator is first used to obtain the vector corresponding to the enlarged slice of the current partition thanks to an extra MPI communication. At the end, no prolongation operator is played and we only retain entries of the vector associated with the original slice of the current partition. The matrix-vector product with the global implicit matrix is still performed in parallel in an exact way.

3.3. Deflation approach

The main drawback of the restarted GMRES(m) is the loss of spectral information contained into the current Krylov subspace during the restarting procedure. Let us recall the definition of a Ritz pair [6] as it plays an important role in the strategy of deflated restarting. The standard Rayleigh-Ritz problem specifies an orthogonality condition on the spectral residual:

$$(Ay - \lambda y) \perp \mathcal{K}_m(A, r_0) \quad \forall y \in \mathcal{K}_m(A, r_0) \tag{10}$$

Using $y = V_m g$ and the standard Arnoldi relation $AV_m = V_{m+1}\bar{H}_m$ in (10), we have

$$H_m g = \lambda g \tag{11}$$

Thus, the spectral residual norm in (10) of the Ritz pair $\{\lambda, y = V_m g\}$ satisfies:

$$\|A(V_m g) - \lambda(V_m g)\| = \|A(V_m g) - V_m H_m g\| = \|V_{m+1}\bar{H}_m g - V_m H_m g\| = |h_{m+1,m}||e_m^T g| \tag{12}$$

From a small value of $|h_{m+1,m}||e_m^T g|$, the Arnoldi method takes as an approximate eigenvalue-eigenvector pair of the operator A the Ritz pair [25, 26]. Indeed, neglecting the last row of the rectangular upper Hessenberg matrix $\bar{H}_m$ leads to:

$$H_m = V_m^T A V_m \tag{13}$$

Therefore, the spectrum of H_m naturally approximates a part of the spectrum of $A\mathcal{M}$. The idea of deflation techniques is to keep relevant spectral information from H_m in the search space of the next cycle to expect a better convergence of the Krylov iterative methods. In [7], Giraud et al. take into account both smallest and largest eigenvalues to maximize the deflation effect. In contrast, Morgan [5] only deflates the smallest ones. This last strategy will be adopted for our numerical experiments.

Actually, Ritz values of the operator A give a good approximation of its exterior eigenvalues. Unfortunately, interior eigenvalues are of greater interest because they are generally responsible for the convergence stagnation. The harmonic Ritz values of A are defined as the Ritz values of A^{-1} with respect to the subspace $A\mathcal{K}_m(A, r_0)$. The resulting orthogonality condition can be expressed as:

$$(A^{-1}y - \theta y) \perp A\mathcal{K}_m(A, r_0) \quad \forall y \in A\mathcal{K}_m(A, r_0) \tag{14}$$

The orthogonality condition for harmonic Ritz values leads to the generalized eigenvalue problem:

$$\theta \bar{H}_m^T \bar{H}_m g = H_m^T g, \tag{15}$$

After some algebraic manipulations, (15) can be reformulated as a standard eigenvalue problem:

$$(H_m + h_{m+1,m}^2 H_m^{-T} e_m e_m^T) g = \theta^{-1} g. \tag{16}$$

where θ^{-1} is the harmonic Ritz value and the corresponding harmonic Ritz vector is $y = AV_m g$. As harmonic Ritz values of A are Ritz values of A^{-1}, exterior eigenvalues θ of A^{-1} are now supposed to be well approximated. Therefore, theirs inverses θ^{-1} are expected to be good approximations of eigenvalues of A in the neighborhood of zero.

Obviously, in exact arithmetic solutions to (15) and (16) are identical. But it is not the case in finite precision since the operator $\bar{H}_m^T \bar{H}_m$ is usually ill-conditioned in the fully linearized turbulence case. Therefore, the accurate estimation of the eigenvectors could be strongly altered and lead to stagnation of the relative true residual of the GMRES process. In contrast, we get better results at the conditioning level with the standard eigenvalues problem overcoming the stagnation of the relative true residual.

As mentionned by Giraud et al. [7], the flexible Arnoldi relation obtained at each restart within the FGMRES with deflated restarting (FGMRES-DR) framework given by

$$AZ_k = V_{k+1}\bar{H}_k, \tag{17}$$

holds with $Z_k = Z_m P_k$, $V_{k+1} = V_{m+1}P_{k+1}$ and $\bar{H}_k = P_{k+1}\bar{H}_m P_k$ where $P_k \in \mathbb{R}^{m\times k}$ corresponds to the orthonormal matrix whose columns are spanned by the eigenvectors of (16).

Also, the right-hand side of the least-squares problem is computed at each restart as $c = V_{m+1}^T r_m$ which requires $2N(m+1)$ operations. Rollins and Fichtner [27] have proposed an efficient way to compute c so that we can save some inner products. Indeed, they demonstrate that the residual r_m is a linear combination of the columns of the deflation subspace V_{k+1}. Consequently, we have $c = V_{k+1}^T r_m$ with $2N(k+1)$ operations. Also, they improved the construction of P_k and in particular the last column of this matrix which is usually chosen as the vector $c - \bar{H}_m y_m$. More specifically, they demonstrated that the vector $c - \bar{H}_m y_m$ is colinear to the vector $[-\beta f^T\ 1]^T$ with $f = H_m^{-T} e_m$. The explicit relation is given below:

$$c - \bar{H}_m y_m = \begin{pmatrix} -\beta f \\ 1 \end{pmatrix} \left(\frac{\omega - \beta f^T v}{1 + \beta^2 f^T f} \right), \tag{18}$$

where v and ω are respectively the first m rows of c and the last element of c.

In exact arithmetic, using the vector $[\beta f^T\ 1]^T$ or the vector $c - \bar{H}_m y_m$ as the last column of P_k is equivalent due to the colinearity property. However, in finite precision arithmetic, the former vector is numerically preferable. More precisely, the use of the vector f both to compute the eigenvectors of $H_m + \beta^2 f e_m^T$ and to construct the last column of matrix P_k enables to reduce the rounding errors in the Arnoldi relation after a restart (see [27]). Consequently, maintaining the Arnoldi relation to a high accuracy leads to a better convergence of the GMRES solver. All these numerical considerations have been illustrated in the next section.

4. Numerical experiments

To evaluate this GMRES solver capabilities, two dedicated test cases have been defined. The first test case was conducted on an adjoint system of equations based on the turbulent transonic RANS flow over a two-dimensional ONERA OAT15A airfoil. The Mach number is 0.734, the Reynolds number based on the chord and freestream conditions is 6.5×10^6 and the angle of attack is 1.15°. The second test case is based on an adjoint system of equations for the three-dimensional turbulent transonic RANS flow over the ONERA M6 wing at Mach number $M = 0.84$, at Reynolds number, based on the mean chord and

freestream conditions, of 1.17×10^7 and with an angle of attack of 3.06°. In both computations the Spalart Allmaras one equation turbulence model has been selected. We focus on the solution of the adjoint system with the fully linearized turbulence model. For the FV solver, the numerical scheme is a second order upwind Roe spatial discretization associated with a MUSCL reconstruction and a van Albada limiter. For the preconditioning step $\mathbf{J}_{O1}^{EX}$ and $\mathbf{J}_{O1}^{APP}$ are considered. For the DG solver, the numerical scheme relies on the Roe flux for the convective term and on the Bassi and co-workers (BR2) [28] formulation for the diffusive term, and $\mathbf{J}_{O3}^{EX}$ is used as the preconditioner.

We emphasize that the FGMRES solvers have been implemented in different computing environments. For instance, the FGMRES solver dedicated to the FV problem has been developed in a modular Python framework. The characteristic algebraic operations (i.e scalar product, preconditioning step and matrix product) have been directly used by wrapping, via the open source software tool SWIG, the corresponding routines from the kernel of the CFD solver *elsA*. This organization is very flexible as it allows a rapid prototyping of promising numerical strategies in a Python environment while still achieving a good level of performance. In contrast, the FGMRES solver dedicated to the DG problem has been implemented in a Fortran environment with a different data structure. Therefore, no comparison can be drawn from a high-performance computing point of view.

4.1. ONERA OAT15A airfoil test case

The ONERA OAT15A airfoil has a thick trailing edge such that the initial structured mesh is made of two blocks. The structured mesh for the FV calculation is composed of 158 208 quadrilaterals and the unstructured mesh for the DG one of 64 416 quadrilaterals.

Figure 1a and Figure 1b depict the mesh partitioning obtained on 16 cores for the scalability study.

For the unstructured case, a general matrix-based algorithm [29] is implemented to construct lines by grouping strongly-connected degrees of freedom. The definition of these lines relies on a directional stiffness measure extracted from a stiffness matrix that represents the strength of couplings between the unknowns. Then, a multilevel k-way graph partitioning algorithm from the METIS library [30] is applied based on a strategy taking into account weights on both the vertices and the edges of the graph. Finally, four weights have been defined and calibrated to distinguish mesh cells belonging to a strong connection line but also mesh edges whose one of the parent cell is affected. The main interest is to preserve as much as possible certain critical areas of the original domain from being arbitrarily partitioned. For example in the case of a shock wave propagation or a turbulent boundary layer development. The resulting largest load imbalance for the unstructured case finally is of the same order as that obtained for the structured case and is about 8 %.

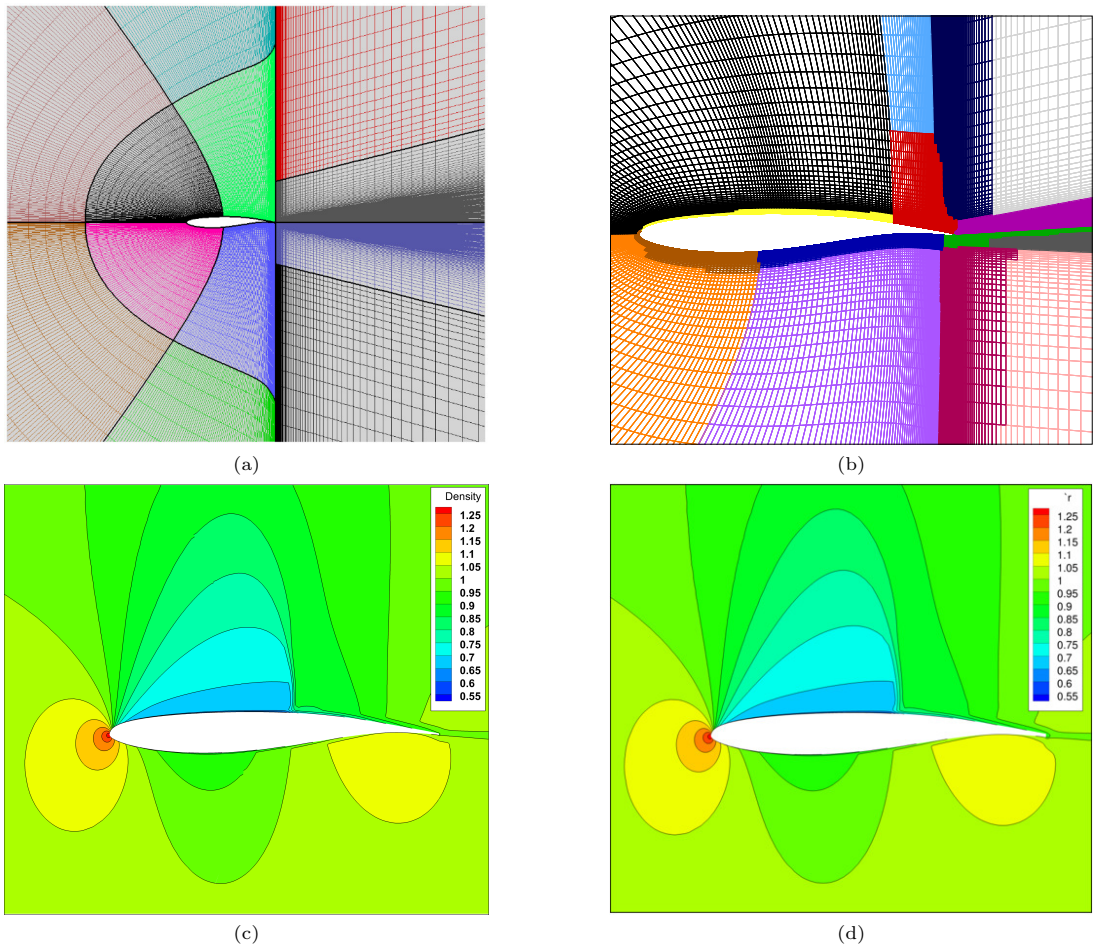


Figure 1: 2D transonic turbulent flow over the OAT15A airfoil: (a) and (b) illustrate the structured and unstructured meshes partitioned into 16 subdomains. The corresponding steady-state density field is illustrated in (c) and (d) for the FV and DG schemes respectively.

4.1.1. Adjoint sensitivity analysis

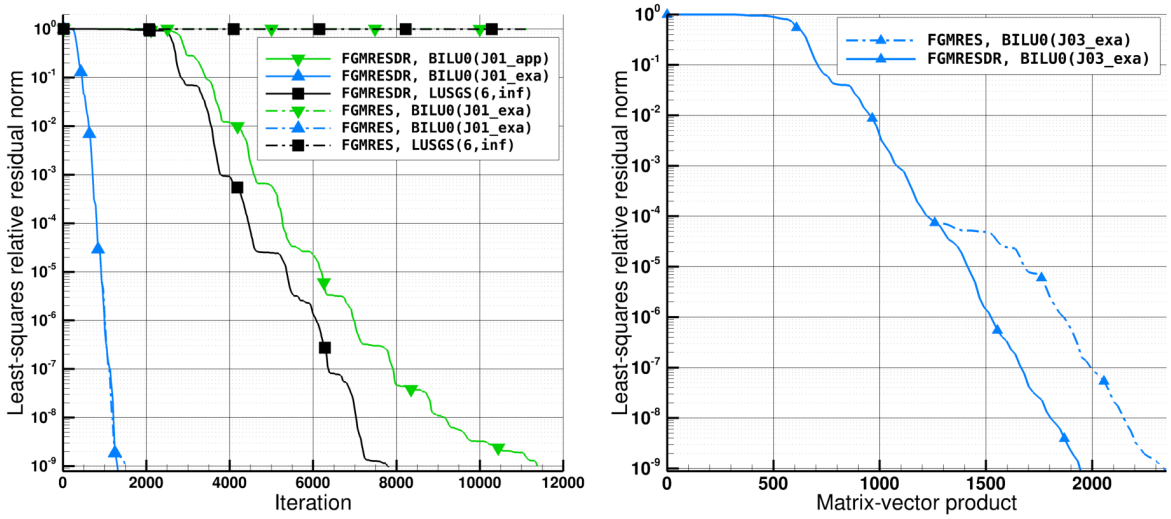
Recall that we want to solve the adjoint system (3) efficiently by using the inner-outer GMRES solver with deflated restarting denoted by FGMRES-DR(m,m_i,k). We adopt the relevant numerical parameters in Table 1. The inner GMRES solver will not be restarted as only small sizes of the Krylov inner subspace will be considered.

All computations are performed with 16 cores on 16-blocks structured and unstructured grid thus with one domain associated with each core. The underlying algebraic problem size is fixed to 1 million for the FV scheme and 2 million for the DG scheme. The number of entries is about 35 million and 288 million for the FV scheme and the DG scheme respectively. Figure 2 displays the relative least-squares

m	60	Size of the outer Krylov space
k	20	Number of deflated vectors
tol outer	1.e-9	Relative convergence threshold
m_i	20	Maximum size of the inner Krylov space
tol inner	0.5	Convergence threshold for the inner GMRES

Table 1: FGMRES-DR relevant numerical parameters

residual norm convergence histories of FGMRES-DR for both FV and DG schemes. In the FV case, we notice a stagnation of FGMRES(60,20) for both LU-SGS and BILU(0) algorithms applied to the first-order approximate Jacobian matrix $\mathbf{J}_{O1}^{APP}$. To overcome this stagnation it is natural to extend the approximation space $\mathcal{K}_m$ in order to capture as much as possible the eigenvalues of $A\mathcal{M}$. Since we cannot afford to extend the projection space due to memory limitation, the deflation strategy remains the best alternative to retrieve the residual convergence of a quasi-full FGMRES solver at low computational cost. A deflation of 30% of m is sufficient and the resulting FGMRES-DR(60,20,20) solver, preconditioned by LU-SGS(6) or BILU(0) now converges as can be seen in Figure 2a. However, when it comes to use the first order exact Jacobian matrix $\mathbf{J}_{O1}^{EX}$, FGMRES(60,20) already converges in a few iterations. In this case, deflating does not lead to any improvements which illustrates how robust $\mathbf{J}_{O1}^{EX}$ is. The downside of this is that we need approximately twice the storage of $\mathbf{J}_{O1}^{APP}$. Figure 2b shows convergence histories of both FGMRES(60,20,20) and FGMRES-DR(60,20,20) for the DG case. From a DG formulation, resulting Jacobian matrices are often ill-conditioned, especially when the spatial order of the scheme becomes high. A variant of the BILU(0) factorization applied to expanded domains (by some level of overlapping) has been implemented in order to globalize as much as possible the preconditioning effect on the whole fluid domain. In both cases, the residuals converge in a few iterations and deflation is shown to slightly improve the convergence rate. We have reported the performance details of the two solvers in Table 2 and Table 3 for both FV and DG problems.



(a) FV case

(b) DG case

Figure 2: Impact of deflation for various preconditioners. The relative residual norm convergence history is plotted with respect to iterations. The impact of deflation is dramatically effective both on the LUSGS and BILU preconditioners applied to the first-order approximate Jacobian matrix. We recall the numerical parameters of the FGMRES-DR solver as follows: $m = 60, m_i = 20$ and $k = 20$.

The number of nonzero entries (NNZ) of the preconditioner is reported in Table 2 and Table 3.

$\mathcal{M}$ = BILU(0)	FGMRES(60,20)		FGMRES-DR(60,20,20)	
Jacobian matrix	$\mathbf{J}_{O1}^{EX}$	$\mathbf{J}_{O1}^{APP}$	$\mathbf{J}_{O1}^{EX}$	$\mathbf{J}_{O1}^{APP}$
NNZ (million)	28	21	28	21
# its	74	-	74	546
# Mvps	1498	-	1321	11381

Table 2: Performance of FGMRES-DR for the FV method

$\mathcal{M}$ = BILU(0)+RAS	FGMRES(60,20)	FGMRES-DR(60,20,20)
Jacobian matrix	$\mathbf{J}_{O3}^{EX}$	$\mathbf{J}_{O3}^{EX}$
NNZ (million)	288	288
# its	111	93
# Mvps	2500	1940

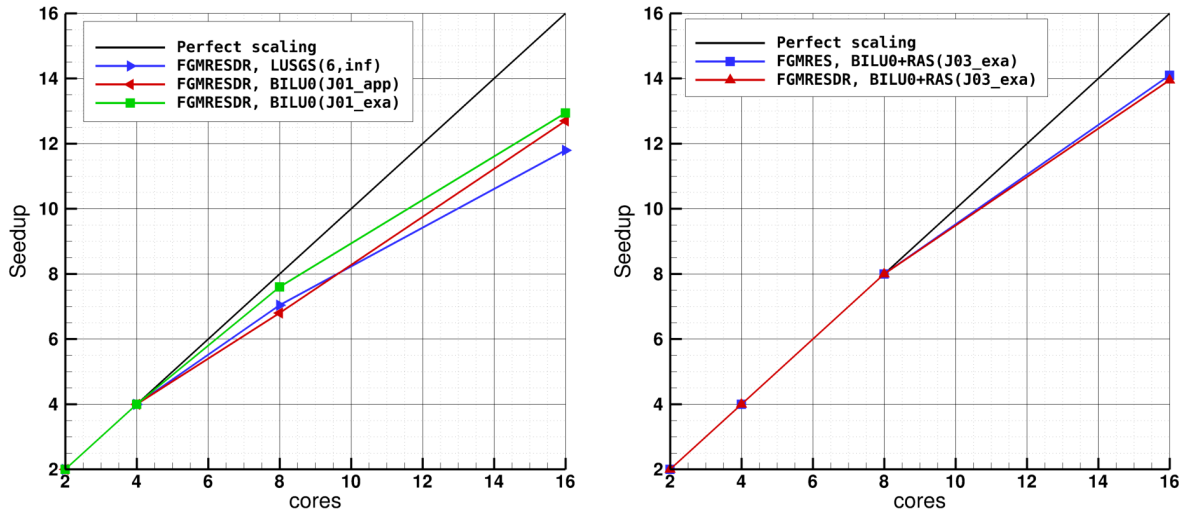
Table 3: Performance of FGMRES-DR for the DG method

4.1.2. Scalability

We have demonstrated in the previous section how competitive the deflated inner-outer GMRES is when applied to stiff problems. As already mentioned, the inner GMRES is by construction a global preconditioner and we are then interested in how its performances scale in a parallel environment. Here, we pay a special attention to the strong scalability in the sense that the problem size is fixed while varying the number of cores.

For the FV scheme, we notice in Figure 3a that the efficiency reaches up to 75 % for the LU-SGS(6) applied to $\mathbf{J}_{O1}^{APP}$ while the speedup for the BILU(0) applied to $\mathbf{J}_{O1}^{EX}$ reaches up to 80 %. The same results have been observed for the DG scheme with BILU(0) and RAS algorithms since an efficiency of 85 % for FGMRES(60,20) as well as for the FGMRES-DR(60,20,20) is measured (Figure 3b). We point out that the local nature of the BILU(0) preconditionner does not have a real impact on the robustness of the inner-outer GMRES methods unlike the standard GMRES [31]. Indeed, one of the main advantages of the inner-outer GMRES is the global nature of the inner GMRES preconditionner since it solves the same initial linear system (4).

For both structured and unstructured cases, the number of iterations to converge the adjoint problem shows a variation up to 20% compared to the two-subdomain reference configuration.



(a) FV scheme

(b) DG scheme

Figure 3: Strong scalability analysis for FGMRES-DR(60,20,20) solvers on the OAT15A adjoint system.

4.2. ONERA M6 wing test case

The second test case corresponds to the steady transonic turbulent flow around the ONERA M6 wing. For the FV method, the structured mesh is composed of 60 blocks for a total of 3.6 million of cells. For the DG case, the unstructured high-order mesh is composed of curved quadratic tetrahedra and wedges for a total of 168 956 cells. Figure 4 depicts both the mesh and the steady state pressure field. For the FV scheme, the algebraic problem size is fixed to 21 million with a number of entries around 2 billion against 10 million with a number of entries around 3.2 billion for the DG scheme.

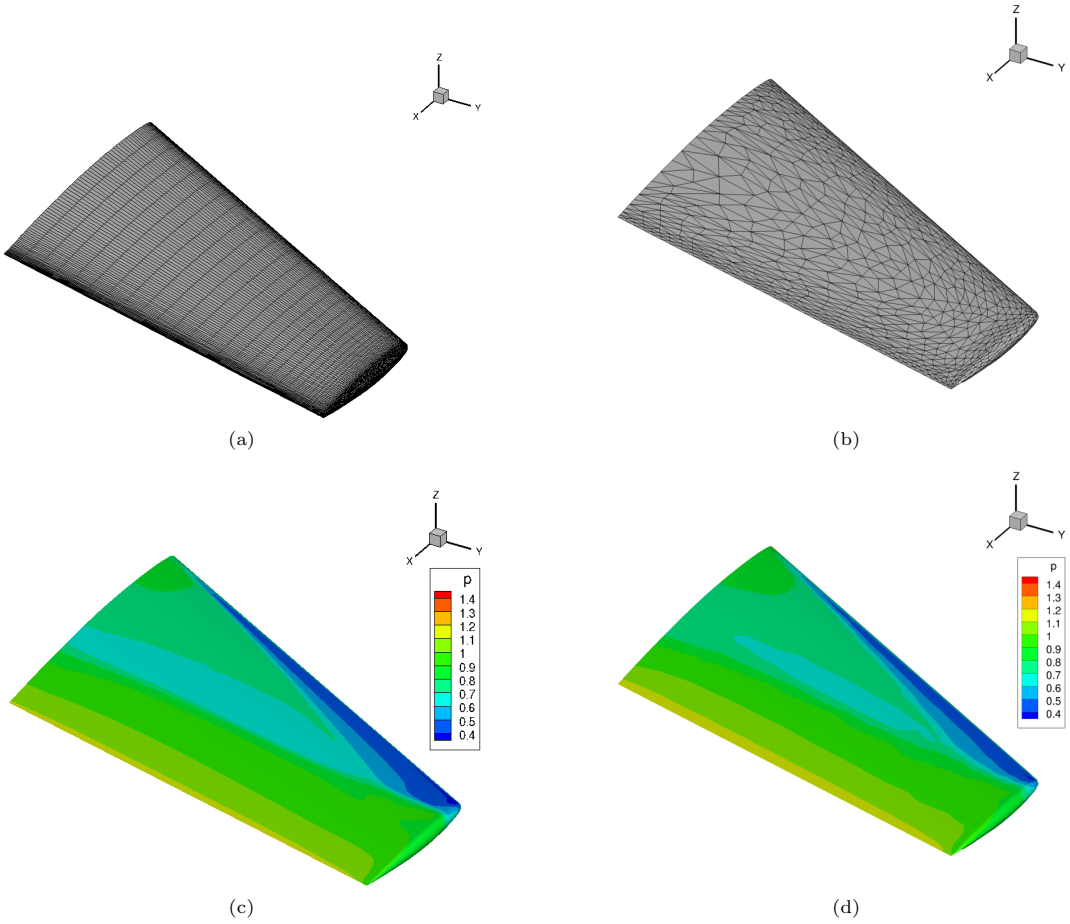


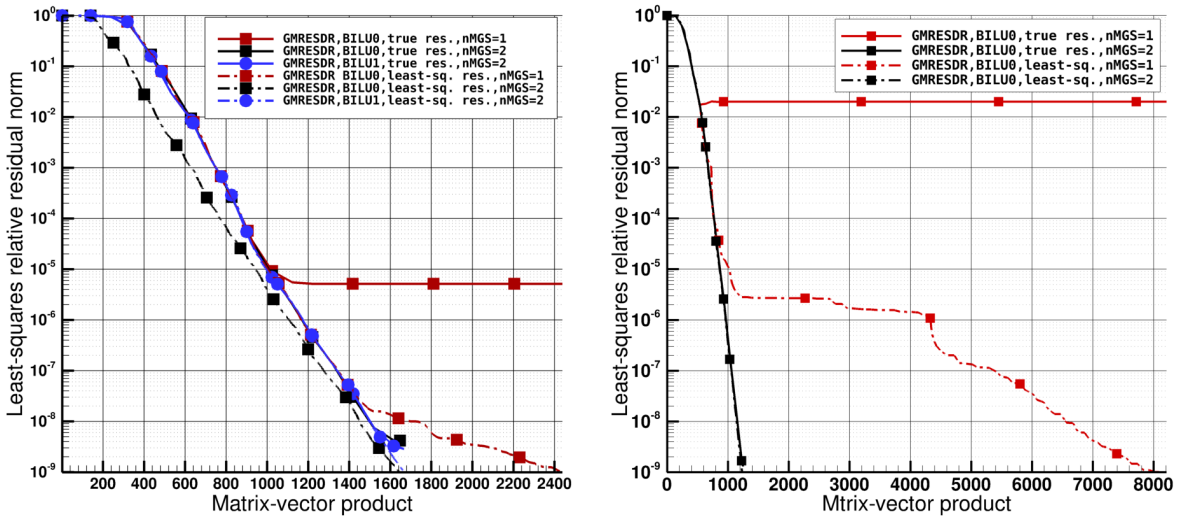
Figure 4: 3D transonic turbulent flow over the ONERA M6 Wing: (a) and (b) represent respectively the wing geometry for structured and unstructured meshes while the steady state pressure field is illustrated in (c) and (d) for both FV and DG computations.

4.2.1. Standard Krylov subspace methods

This section aims at demonstrating how the standard GMRES solver suffers from numerical limitations when tackling stiff problems. It is crucial to point out this aspect since many authors frequently employ the standard GMRES in their numerical experiments. Hence, it is worthwhile to assess the behavior of standard GMRES compared to its flexible counterpart for stiff problems. To perform a fair comparison, the size of the respective Krylov subspaces are chosen in order to get the same memory footprint. Based on a maximum size of 60 and 20 vectors for the outer and inner Krylov spaces respectively, the memory footprint of FGMRES-DR amounts to the storage of 140 vector fields.

The relative true residual norm is usually expected to be of the same order to that for the convergence of the stationary problem. But in several specific contexts, higher accurate solutions are required so we are looking for true residual norms as close as possible to machine epsilon.

For such requirements, the Modified Gram-Schmidt process (MGS) exhibits limitations because it is known to perform poorly when input vectors are almost colinear. In this case a standard remedy is to apply a re-orthogonalization step thus doubling the cost of the Arnoldi procedure. For computational cost reasons, the residual convergence history is assessed with the least-squares residual while the quality of the convergence is given by the true residual which is only computed at the end of each cycle. Both residuals are plotted in Figure 5. All operations with the GMRES-DR solver are performed in a double-precision arithmetic. When a single step of MGS orthogonalization is applied, the least-squares residual reaches the required tolerance while the corresponding true residual stagnates at 10^{-6} for BILU(0) applied to $\mathbf{J}_{O1}^{EX}$ in the FV method. For the DG case, true and least-squares residuals are quite similar up to 10^{-2}. At this level, the true residual stagnates while the least-squares residual decreases up to the prescribed tolerance. This phenomenon comes essentially from the rounding errors during the orthogonalization process since the size of the Krylov subspace is sufficiently large to give rise to numerical errors. The corresponding convergence histories for two steps of orthogonalization are given in Figure 5. For the FV problem, we get a gain of two orders of magnitude with a final relative true residual at 10^{-8}. For the DG problem, the gain is even better with a converged relative true residual at 10^{-9}. Another important aspect that explains why the GMRES-DR solver does not converge at the lower tolerance in the FV case comes from the quality of the preconditioner. We point out that the DG method uses essentially the third-order operator $\mathbf{J}_{O3}^{EX}$ as a preconditioner, which corresponds to the same order of accuracy as the discretization scheme, while the FV method simply relies on approximate first order preconditioning operators while a second order spatial discretization scheme is applied for the mean flow solution. In an attempt to increase the quality of the preconditioner we used a BILU(1) factorization of on the first-order exact Jacobian matrix $\mathbf{J}_{O1}^{EX}$. As can be seen in Figure 5a, the corresponding gain is however marginal.



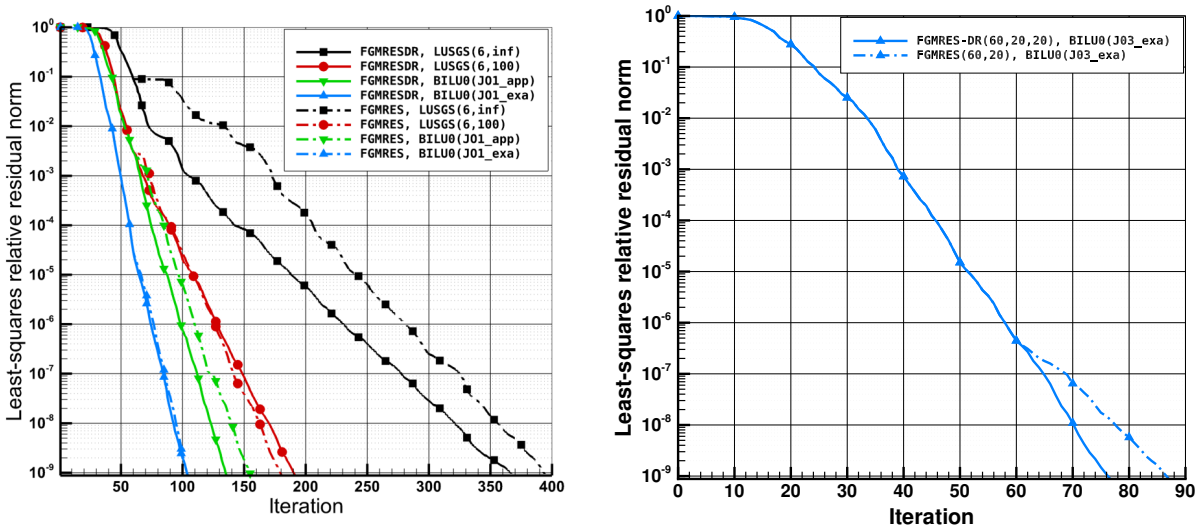
(a) FV case

(b) DG case

Figure 5: True and least-squares residual norm convergence histories by applying two steps of Gram-Schmidt orthogonalization procedure: *least-sq. res.* stands for least-squares residual while *true res.* stands for true residual. We recall the numerical parameters of the GMRES-DR solver as follows: $m = 140$ and $k = 42$

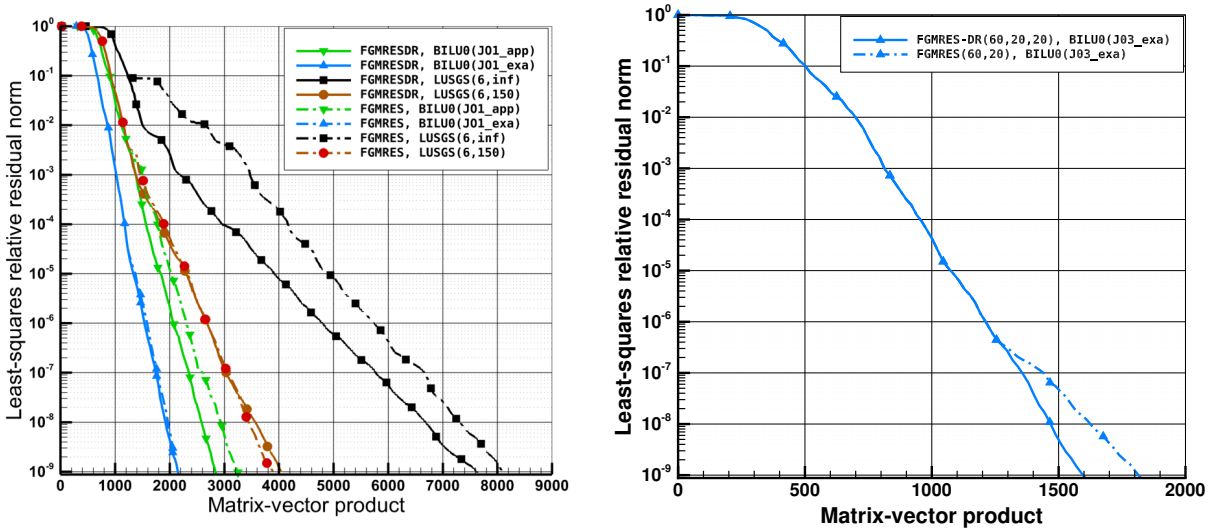
4.2.2. Inner-Outer Krylov subspace methods

For the nested strategy, we perform the same comparison study keeping the same numerical parameters detailed in Table 1 of subsection 4.1.1. Unlike the two-dimensional OAT15A test case, the FGMRES(60,20) solver preconditioned either by LU-SGS(6) or BILU(0) applied to $\mathbf{J}_{O1}^{APP}$ converges without the deflation strategy. In Figure 6a, FGMRES(60,20) exhibits a good convergence with 150 iterations for BILU(0) against 377 for LU-SGS(6). However, the convergence is a little sensitive to the deflation effect with a gain of 22 iterations for LU-SGS(6) and 15 iterations for BILU(0). Even though the preconditioning matrix $\mathbf{J}_{O1}^{APP}$ is less robust than $\mathbf{J}_{O1}^{EX}$, it still remains interesting from a memory point of view as we have explained for the 2D test case. Actually, it is more crucial to take into account the memory factor when it comes to solve three-dimensional stiff problems. Indeed, we recall that $\mathbf{J}_{O1}^{APP}$ has a stencil of 7-points in 3D while a stencil of 19-points is associated with $\mathbf{J}_{O1}^{EX}$, that is, a ratio of 2.7 against 1.8 for the two-dimensional case.



(a) FV case (b) DG case

Figure 6: Impact of deflation for various preconditioners. The relative residual norm convergence history is plotted with respect to iterations. The increase of the diagonal dominance of the LU-SGS preconditioner by addition of a scalar diagonal $I/\Delta\tau$ seems very promising but the strategy remains less effective than the one based on BILU. The impact of deflation is not very conclusive for the most efficient BILU preconditioner as only one restart is performed in this case. We recall the numerical parameters of the FGMRES-DR solver as follows: $m = 60, m_i = 20$ and $k = 20$.



(a) FV case (b) DG case

Figure 7: Impact of deflation for various preconditioners. The relative residual norm convergence history is plotted with respect to Jacobian-vector products. The hierarchy of the preconditioners remains the same as the one in Figure 6.

In order to better highlight the impact of deflation we have conducted additional numerical simulations in the FV case by varying the size of the inner Krylov space with 6, 10, 15 and 20 vectors for the BILU type preconditioners. Figure 8 shows that when at least two restarts are performed the deflation step is clearly of interest in terms of iterations. Interestingly, the benefit in terms of matrix-vector products is not similar except for the lowest size of the inner Krylov space. This suggests that for this specific computation the best compromise in terms of memory and efficiency is to choose an inner Krylov size of 15.

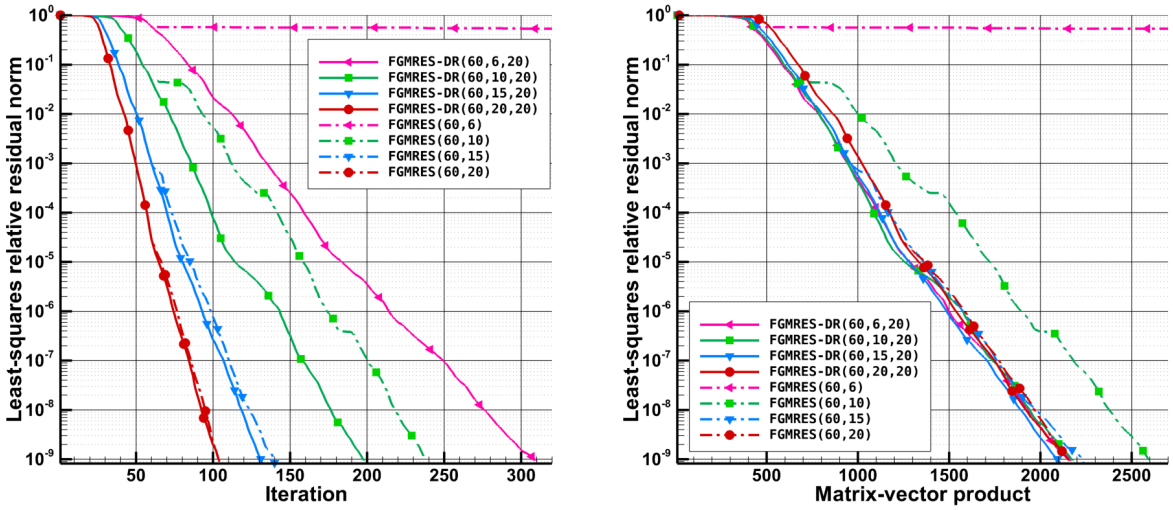


Figure 8: Impact of deflation for various sizes of the inner Krylov space for FV case. Adjoint relative residual norm convergence history with respect to iterations (left) and Jacobian-vector products (right) for FGMRES$(60, mi)$ and FGMRES-DR$(60, mi, 20)$ with $mi = 6, 10, 15, 20$.

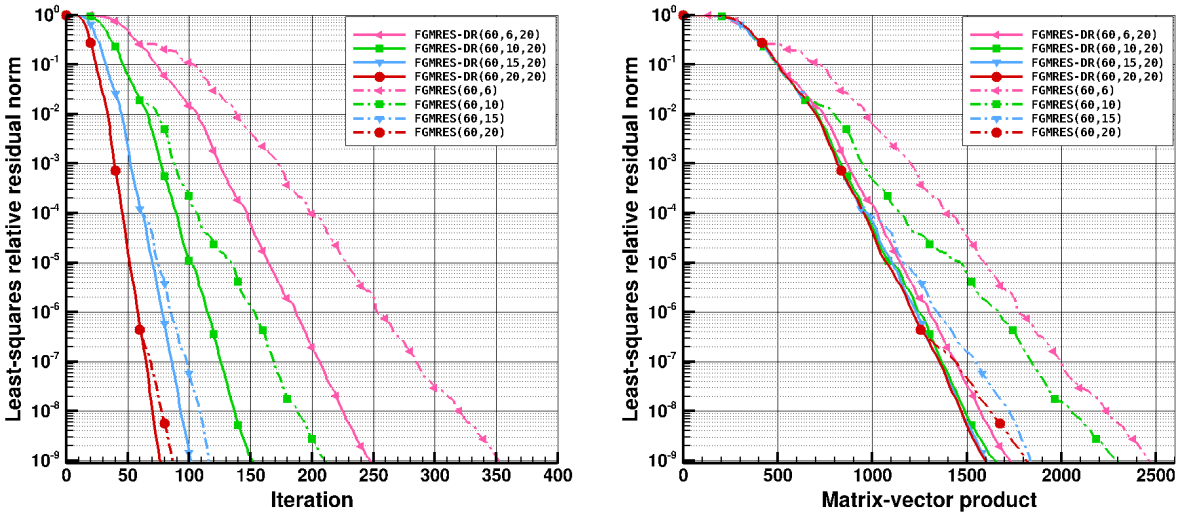


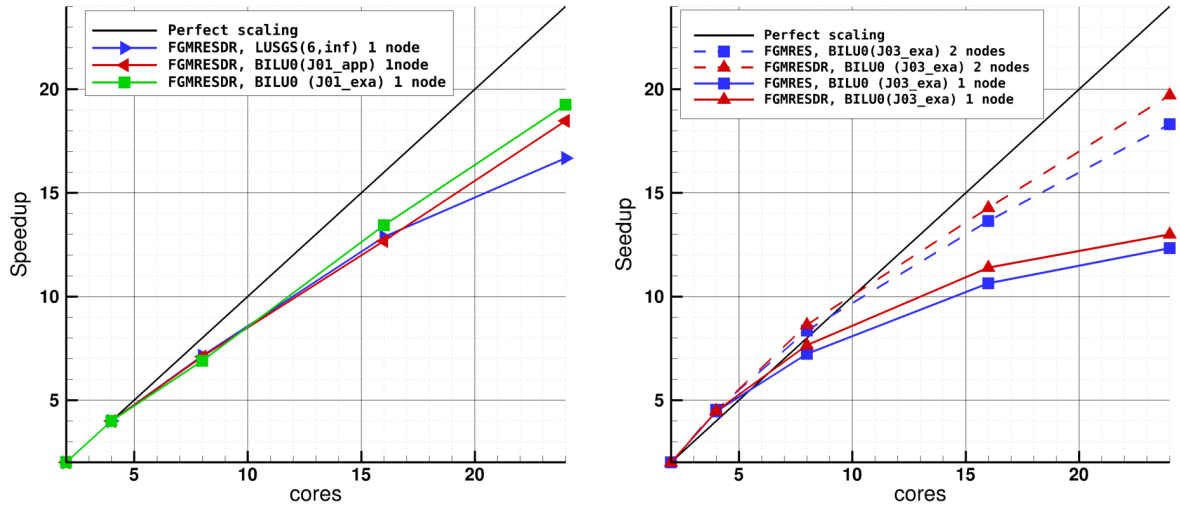
Figure 9: Impact of deflation for various sizes of the inner Krylov space for DG case. Adjoint relative residual norm convergence history with respect to iterations (left) and Jacobian-vector products (right) for FGMRES$(60, mi)$ and FGMRES-DR$(60, mi, 20)$ with $mi = 6, 10, 15, 20$.

4.2.3. Scalability

We have shown how the inner-outer GMRES solvers exhibit a satisfactory scalability with regard to the two-dimensional OAT15A airfoil. It would be advisable to retrieve the same scalability on the M6 wing configuration for both FV and DG formulations.

For the scalability analysis, we consider five configurations corresponding to the partition of the initial domain into 2, 4, 8, 16 and 24 sub-domains to be performed on 24-core compute nodes.

For the unstructured case, the line construction algorithm [29] previously used for the two-dimensional OAT15A test-case has been applied. Finding a relevant set of weights for the METIS algorithm was more difficult on this three-dimensional test-case. A good trade-off has finally been found with a largest load imbalance about 5 %. The number of iterations to converge the adjoint problem is sensitive to the number of cores with a variation up to 15 % compared to the two-subdomain reference configuration. As it could have a significant impact on such a challenging test-case, another strategy has been adopted and relies on the CHACO software [32] for partitioning graphs. Spectral methods partition the graph using the eigenvectors of a matrix constructed from the graph. We select the simplest spectral method which is a weighted version of spectral bisection. This method uses the second lowest eigenvector of the Laplacian matrix of the graph to divide it into two pieces [33]. This results invariably to a quite perfect partitioning in terms of number of cells with almost one cell per partition of difference. The number of iterations to converge the adjoint problem shows now a variation up to 6 %.



(a) FV case

(b) DG case

Figure 10: Strong scalability analysis for FGMRES-DR(60,20,20) solvers on M6 wing adjoint system: both 1 node and 2 nodes have been plotted in the DG case.

Figure 10 depicts parallel behavior of the FGMRES-DR solver. In Figure 10b, for the DG case, scalability analysis is performed on one node with one MPI process per core (solid line), and on two nodes (dashed line), with a balanced distribution of the MPI processes per node. This exercise exhibits a memory-bandwidth limitation for this test-case : the parallel efficiency is about 55% on one node and grows up to 85 % on two nodes. Dense linear algebra kernels (BLAS, LAPACK) are applied on blocks of size 60×60, corresponding to the number of degrees of freedom per mesh element, of the implicit matrix. But there are memory-bound operations with a low arithmetic intensity.

For the FV case, the same exercise has been performed with different scalability results. More specifically, the parallel efficiency is about 80 % regarding the BILU(0) preconditioner and 70 % for the LU-SGS one on one node. When it comes to use two nodes, the parallel efficiency grows up to 80 % for all preconditioners. The matrix operations now work on blocks of smaller size 6×6 and are therefore less prone to memory-bandwith limitations.

Since the preconditioner does depend on the number of sub-domains, we have reported in Table 4 the variation of the number of iterations of the FGMRES-DR according to the number of the sub-domains.

# its ($\mathcal{M}$) \ # sub-domains	2	4	8	16	24
# its (LUSGS(6,150))	190	189	190	190	190
# its (LUSGS(6,inf))	348	348	349	349	348
# its (BILU(0), $\mathbf{J}_{APP}^{O1}$)	134	135	134	134	135
# its (BILU(0), $\mathbf{J}_{EX}^{O1}$)	99	98	99	99	98

Table 4: Impact of the number of sub-domains on the number of iterations for FGMRES-DR(60,20,20). As the inner GMRES preconditionner is global by construction, we see no impact on the number of iterations as expected.

5. Some numerical considerations

In our numerical experiments we have considered low convergence thresholds because it is acknowledged that adjoint solutions may exhibit oscillations or still be locally under-resolved for moderate relative residual decrease (typically 3 orders of magnitude). This strong convergence is only attainable in practice if a number of precautions is taken in the implementation of the FGMRES-DR solver. First, the orthogonality of the Krylov basis at the outcome of the Arnoldi process or after a deflation step must be preserved as much as possible to ensure convergence in a reasonable number of iterations for stiff problems. This point was studied in [27] where the authors advocate the use of a re-orthogonalization during the MGS process. Morgan [5] also proposed an economical strategy where only the last Krylov vector of the deflated basis is re-orthogonalized with respect to other deflated basis vectors. The initial physical equations should always be solved in their non dimensionalized form. For large ill-conditioned linear systems, Saad [24] also recommends to apply a scaling to all the rows (or columns) e.g., so that their 1-norms are all equal to 1; then to apply a scaling of the columns (or rows). This practice is strongly recommended before any ILU type factorization. Additionally, recognizing that the linearization of the turbulence model dramatically deteriorates the condition number of the system matrix, Chisholm and Zingg [34] advice to apply a specific scaling coefficient to this particular equation. Additional numerical ingredients like the mixed-precision nested GMRES, i.e., single precision floating point arithmetic for the inner GMRES, and the application of the BILU factorization to the RAS partition of the fluid domain have proven beneficial. Last but not least, the availability of different physical approximations to build the candidate flux Jacobian matrix for the preconditioner is very interesting to keep a good compromise between efficiency and memory footprint as it has a strong impact on the size of the stencil of this operator. Finally, some memory saving improvements were proposed in [7] to perform inplace the matrix-matrix product leading to the new deflated Krylov vector basis at the starting of the next cycle. In fact in a nested FGMRES-DR framework this is not required anymore because the temporary memory

required by this product is already available for the storage of the inner Krylov subspace and is then naturally re-used during the outer level deflation step. It is also worth mentioning that in practice the inner GMRES solver is never restarted considering the small size of the inner Krylov subspace.

Another important aspect that should be highlighted is the numerical limitation of the Modified Gram-Schmidt process. Indeed, in terms of number of domains in the mesh partition, it requires asymptotically more messages and synchronization points in parallel, and asymptotically more data traffic between levels of the memory hierarchy.

6. Concluding remarks

Flexible GMRES solvers have been investigated for the solution of large sparse linear systems arising from discrete stiff adjoint problems when the standard GMRES fails. A FV and a high-order DG discretizations are considered to provide linear systems of different complexity from test-cases of optimization problems of turbulent flows with RANS modelling. Condition number and dimensions of the resulting matrices change but also performance of the underlying basic linear algebra operations. Varying the discretization methods contributes to a better overview of the solver capabilities. Relevant numerical ingredients have been developed to improve robustness and efficiency of the inner-outer GMRES strategy and concern preconditioning techniques and deflation strategy. Numerical experiments have been conducted on turbulent transonic flows over the two-dimensional supercritical ONERA OAT15A airfoil and over the three-dimensional ONERA M6 wing.

For the first ingredient, with the FV scheme, we have investigated the impact of using different approximations of the flux Jacobian matrix for the preconditioning step. The efficiency of the nested GMRES has been observed when deflation technique with the first order approximate Jacobian matrix is considered or by using the first order exact Jacobian matrix without deflation but at the price of twice the memory cost. The use of an exact linearization of the DG scheme to build the preconditioner offers very good performance of the linear solvers in terms of robustness and convergence.

For the second ingredient, significant gains has been obtained on the convergence speed. Not only eigen-information can be recovered at low cost using by-products of the Arnoldi iterative process, but matrix-vector products involving the Jacobian matrix can be saved at the beginning of each new cycle thanks to the recycling of eigenvectors into the new Krylov basis.

On top of that, a good parallel scalability of the FGMRES is achieved thanks to the inherent global effect of the inner GMRES. The interesting conclusion of this work is the capability of inner-outer GMRES strategy to solve stiff problems and exhibit the superlinear convergence of the full GMRES, but with a lower memory footprint. Strategies to select eigen-information and calibration rules for the numerical

parameters driving the solution of inner and outer systems might be worth taking into consideration in the future.

References

[1] Y. Saad, M. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Comput. 7 (3) (1986) 856–869. doi:10.1137/0907058.

[2] M. Benzi, Preconditioning Techniques for Large Linear Systems: A Survey, J. Comput. Phys. 182 (2) (2002) 418–477. doi:10.1006/jcph.2002.7176.

[3] S. Yoon, A. Jameson, Lower-upper symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations, AIAA journal 26 (9) (1988) 1025–1026. doi:10.2514/3.10007.

[4] Y. Saad, A Flexible Inner-Outer Preconditioned GMRES Algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469. doi:10.1137/0914028.

[5] R. Morgan, GMRES with Deflated Restarting, SIAM J. Sci. Comput. 24 (1) (2002) 20–37. doi:10.1137/s1064827599364659.

[6] R. Morgan, M. Zeng, Harmonic projection methods for large non-symmetric eigenvalue problems, Numerical Linear Algebra with Applications 5 (1) (1998) 33–55. doi:10.1002/(SICI)1099-1506(199801/02)5:1<33::AID-NLA125>3.0.CO;2-1.

[7] X. P. L. Giraud, S. Gratton, X. Vasseur, Flexible GMRES with Deflated Restarting, SIAM J. Sci. Comput. 32 (4) (2010) 1858–1878. doi:10.1137/080741847.

[8] X. Pinel, M. Montagnac, Block Krylov Methods to Solve Adjoint Problems in Aerodynamic Design Optimization, AIAA Journal 51 (9) (2013) 2183–2191. doi:10.2514/1.j052113.

[9] C.-H. Chen, S. Nadarajah, GCRO with Dynamic Deflated Restarting for Solving Adjoint Systems of Equations for Aerodynamic Shape Optimization, International Journal of Numerical Methods for Heat & Fluid Flow (2019). doi:10.2514/6.2019-3708.

[10] M. Parks, E. D. Sturler, G. Mackey, D. Johnson, S. Maiti, Recycling Krylov Subspaces for Sequences of Linear Systems, SIAM J. Sci. Comput. 28 (5) (2006) 1651–1674. doi:10.1137/040607277.

[11] R. Dwight, J. Brezillon, Effect of Various Approximations of the Discrete Adjoint on Gradient-Based Optimization, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2006. doi:10.2514/6.2006-690.

[12] M. Nemec, D. Zingg, Towards efficient aerodynamic shape optimization based on the Navier-Stokes equations, in: 15th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2001. doi:10.2514/6.2001-2532.

[13] F. Renac, C.-T. Pham, J. Peter, Sensitivity analysis for the RANS equations coupled with linearized turbulence model, in: 18th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2007. doi:10.2514/6.2007-3948.

[14] F. Renac, Improvement of the recursive projection method for linear iterative scheme stabilization based on an approximate eigenvalue problem, J. Comput. Phys. 230 (14) (2011) 5739–5752. doi:10.1016/j.jcp.2011.03.057.

[15] J. E. Peter, R. P. Dwight, Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches, Computers & Fluids 39 (3) (2010) 373–391. doi:10.1016/j.compfluid.2009.09.013.

[16] J. Peter, F. Drullion, Large stencil viscous flux linearization for the simulation of 3D compressible turbulent flows with backward-Euler schemes, Computers & Fluids 36 (6) (2007) 1005–1027. doi:10.1016/j.compfluid.2005.09.006.

[17] S. H. L. Cambier, S. Plot, The Onera elsA CFD software: input from research and feedback from industry, Mechanics & Industry 14 (3) (2013) 159–174. doi:10.1051/meca/2013056.

[18] F. Renac, M. de la Llave Plata, E. Martin, J. B. Chapelier, V. Couaillier, Aghora: A High-Order DG Solver for Turbulent Flow Simulations, in: Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer International Publishing, 2015, pp. 315–335. doi:10.1007/978-3-319-12886-3_15.

[19] A. Jameson, Optimum aerodynamic design using CFD and control theory, in: 12th Computational Fluid Dynamics Conference, 1995. doi:10.2514/6.1995-1729.

[20] L. G. V. Frayssé, S. Gratton, Algorithm 881: A set of flexible GMRES routines for real and complex arithmetics on high-performance computers, ACM Transactions on Mathematical Software (TOMS) 35 (2) (2008) 1–12. doi:10.1145/1377612.1377617.

[21] A. Resmini, Sensitivity analysis for numerical simulation of compressible flows in external aerodynamics., Ph.D. thesis, Université Pierre et Marie Curie Paris VI (2015).

[22] G. Puigt, CFD dedicated techniques applied to industrial problems, Accreditation to supervise research (HDR French diploma), Ph.D. thesis (2014).

[23] X.-C. Cai, M. Sarkis, A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems, SIAM J. Sci. Comput. 21 (2) (1999) 792–797. doi:10.1137/s106482759732678x.

[24] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, (2003). doi:10.1137/1.9780898718003.

[25] W. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quarterly of Applied Mathematics 9 (1) (1951) 17–29. doi:10.1090/qam/42792.

[26] Y. Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, laa 34 (1980) 269–295. doi:10.1016/0024-3795(80)90169-X.

[27] S. Röllin, W. Fichtner, Improving the accuracy of GMRES with deflated restarting, SIAM J. Sci. Comput. 30 (1) (2008) 232–245. doi:10.1137/060656127.

[28] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini, A High-Order accurate Discontinuous Finite Element Method for Inviscid and Viscous Turbomachinery Flows, in: Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Antwerpen, Belgium, 1997, pp. 99–109.

[29] B. Ahrabi, D. Mavriplis, An implicit block ILU smoother for preconditioning of Newton-Krylov solvers with application in high-order stabilized finite-element methods, Computer Methods in Applied Mechanics and Engineering 358 (2020). doi:10.1016/j.cma.2019.112637.

[30] G. K. V. Kumar, A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs, SIAM J. Sci. Comput. 20 (1) (1999) 359–392. doi:10.1137/S1064827595287997.

[31] V. Simoncini, D. Szyld, Flexible Inner-Outer Krylov Subspace Methods, SIAM J. on Numer. Anal. 40 (6) (2003) 2219–2239. doi:10.1137/s0036142902401074.

[32] B. H. R. Leland, The CHACO User's Guide, version 2.0, Technical Report SAND95-2344 (1995).

[33] H. Simon, Partitioning of unstructured problems for parallel processing, Computing Systems in Engineering 2 (2) (1991) 135–148. doi:10.1016/0956-0521(91)90014-V.

[34] T. Chisholm, D. Zingg, A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows, J.Comput.Phys. 228 (9) (2009) 3490–3507. doi:10.1016/j.jcp.2009.02.004.