The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/0961-5539.htm

GCRO with dynamic deflated restarting for solving adjoint systems of equations for aerodynamic shape optimization

Chih-Hao Chen and Siva Nadarajah
Department of Mechanical Engineering, McGill University, Montreal, Canada

Received 24 October 2018
Revised 5 April 2019
Accepted 9 May 2019

Abstract

Purpose – This paper aims to present a dynamically adjusted deflated restarting procedure for the generalized conjugate residual method with an inner orthogonalization (GCRO) method.

Design/methodology/approach – The proposed method uses a GCR solver for the outer iteration and the generalized minimal residual (GMRES) with deflated restarting in the inner iteration. Approximate eigenpairs are evaluated at the end of each inner GMRES restart cycle. The approach determines the number of vectors to be deflated from the spectrum based on the number of negative Ritz values, k^*.

Findings – The authors show that the approach restores convergence to cases where GMRES with restart failed and compare the approach against standard GMRES with restarts and deflated restarting. Efficiency is demonstrated for a 2D NACA 0012 airfoil and a 3D common research model wing. In addition, numerical experiments confirm the scalability of the solver.

Originality/value – This paper proposes an extension of dynamic deflated restarting into the traditional GCRO method to improve convergence performance with a significant reduction in the memory usage. The novel deflation strategy involves selecting the number of deflated vectors per restart cycle based on the number of negative harmonic Ritz eigenpairs and defaulting to standard restarted GMRES within the inner loop if none, and restricts the deflated vectors to the smallest eigenvalues present in the modified Hessenberg matrix.

Keywords GMRES, Adjoint method, Deflated restarting, GCRO-DR, Krylov solvers

Paper type Research paper

1. Introduction

The adjoint approach is the current standard for gradient-based aerodynamic shape optimization (ASO) (Lions, 1971; Jameson, 1994; Jameson and Reuther, 1994; Hicken and Zingg, 2010; Nadarajah and Jameson, 2007; Nadarajah *et al.*, 2006; Papadimitriou and Giannakoglou, 2007; Pinel and Montagnac, 2013; Tatossian *et al.*, 2011; Walther and Nadarajah, 2012; Lyu *et al.*, 2014). Its ubiquitous use has delivered the ability to evaluate sensitivities and the redesign of extremely complex geometries (Brezillon and Gauger, 2004; Kroll *et al.*, 2007; Cambier and Kroll, 2008; Xu and Timme, 2017). However, our quest to use the approach for larger computational grids and complex flow fields saturated with highly anisotropic features and separated flows have resulted in challenges in converging the linear adjoint system of equations (Kroll *et al.*, 2007; Kim *et al.*, 2004; Brezillon *et al.*, 2009). To

The authors gratefully acknowledge the comments and feedback raised by the reviewers as it greatly improved the article. They also acknowledge the generous support from the Natural Sciences and Engineering Research Council (NSERC). The authors also thank McGill HPC, Calcul Quebec for providing the computational facilities.

International Journal of Numerical Methods for Heat & Fluid Flow
Vol. 29 No. 7, 2019
pp. 2179-2205
© Emerald Publishing Limited
0961-5539
DOI 10.1108/HFF-10-2018-0586

recover accurate gradients through a discrete adjoint approach, the complete linearization of the discretized governing equations are required. A second-order finite-volume three-dimensional structured flow solver, leads to Jacobian matrices that are numerically stiff and contain larger off-diagonal terms compared to its first-order counterpart.

Solving such linear systems of equations with general iterative methods has become impractical if no effective preconditioning is offered (Dwight, 2006). Such an increase in computational cost impedes the ability to employ ASO for the redesign of aircraft geometries. To overcome these obstacles for both the efficacy and efficiency of ASO, more robust sparse linear system solvers within a parallel computing environment are required. One approach to seek the solution of such sparse systems is to use iterative Krylov solvers, such as GMRES (Heuveline and Strauß, 2009; Yu *et al.*, 2018; He *et al.*, 2018), BiCGStab (Amritkar *et al.*, 2015) and generalized conjugate residual method with an inner orthogonalization (GCRO) (Xu and Timme, 2017; Amritkar *et al.*, 2015). To improve the convergence of such Krylov subspace algorithms, the community uses effective preconditioners, as well as augmentation and deflated restarting approaches to enhance not only the convergence rate but also to reduce the number of matrix-vector products necessary to converge the solution to a pre-specified tolerance.

Restarted GMRES(m), where m is the maximum number of basis vectors generated during the iterative process is a common approach to reduce the memory overhead of the GMRES approach. However, for ill-conditioned systems, convergence is often stalled as the number of vectors m is insufficient to generate a complete subspace. Moreover, the loss of previous Krylov basis vectors leads to non-optimal searches during Arnoldi iterations. To circumvent this shortcoming, Morgan (Morgan, 1995; Morgan, 2000; Morgan, 2002; Morgan, 2005) proposed extracting Ritz values from the upper Hessenberg matrix that is typically available at the completion of the Arnoldi process and augment the Krylov subspace with the Ritz vectors for the next restart cycle. Termed as GMRES-DR, where DR denotes deflated restarting, the technique has been demonstrated to improve the convergence rate and most importantly avoid early stagnation (Pinel and Montagnac, 2013; Morgan and Zeng, 2006; Chen *et al.*, 2019). An alternate implementation of deflated restarting was introduced by Erhel *et al.* (Erhel *et al.*, 1996), where Schur vectors from the upper Hessenberg matrix instead of the Ritz pairs were used to update the preconditioner at the start of every cycle.

In parallel, de Sturler (1999) introduced the generalized conjugate residual with optimal truncation (GCROT) to subspace recycling. The author demonstrated an approach to evaluate critical subspaces that were essential for convergence. These subspaces were then used to orthogonalize future search directions. Hicken and Zingg (2010) extended the approach with an alternate truncation strategy and demonstrated the solver for the solution to the adjoint system of equations, where GCROT was shown to be more efficient and robust than restarted GMRES(m). Finally, the adoption of Morgan's (Morgan, 1995; Morgan, 2000; Morgan, 2002; Morgan, 2005) deflated restarting into GCRO (de Sturler, 1996), was initiated by Parks *et al.* (2006). The authors proposed a nested framework, where GMRES(m) is used within the inner loop of the GCR algorithm and Ritz values extracted at the end of the inner loop and applied towards Krylov subspace vectors of the outer. Results showed that the approach effectively avoided stagnation, and accelerated convergence without increasing the memory consumption.

In the stated approaches, the challenge is in selecting the correct number of vectors to deflate from the previous restart cycle. The choice is very much dependent on a multitude of parameters such as the chosen number of Krylov basis vectors, m to be used for the cycle as well as the type of Ritz vectors, where the latter is driven by the spectral distribution of the

Ritz values. Yamazaki *et al.* (2014) investigated deflating a combination of both the smallest and largest Ritz values and found that excluding the largest yielded higher convergence rates. Instead, Giraud *et al.* (2010) found it effective to deflate eigenvalues that are located at a distance away from clusters. Pinel and Montagnac (2013) used a strategy similar to Giraud *et al.* (2010); however, they choose to set the number of deflated vectors to half the size of the Krylov basis vectors. The authors introduced the deflation strategy for a block-GMRES solver, where the adjoint system was solved with multiple right-hand-side vectors.

Recent attempts to extend the use of deflated restarting can be seen in the work of Xu and Timme (2017). The authors used GCRO-DR and experimented with ILU of various fill-in levels and reported that GCRO-DR resulted in fewer matrix-vector products at a lower computational cost and equivalent memory requirements to GMRES. In Xu and Timme (2017), an extensive list of test cases was examined, ranging from 2D to 3D problems on both on- and off-design conditions. Although deflation has proven to be more effective than either increasing the number of restart vectors or the ILU fill-in level, comprehensive strategies on the number and type of eigenvalues to deflate have received less attention. We devised a novel strategy of using only Ritz vectors associated to k* of the smallest eigenvalues, where k^* was the number of negative eigenvalues (Chen *et al.*, 2019). The strategy was used within the GMRES-DR, deflation with restarting approach, where Ritz vectors associated with k^* Ritz values were orthogonalized with respect to the previous residual vector and included within the subsequent restart cycle. In Chen *et al.* (2019), we demonstrated the solver on the adjoint system for a number of test cases ranging from a two-dimensional NACA 0012 airfoil, to the 3D common research model (CRM) wing, and two three-dimensional complete aircraft configurations in viscous flow. Both lift and drag adjoints were attempted in addition to studies on the scalability of the solver, impact of the local and global preconditioners as well as the level of linearization of the discretized governing equation.

In this paper, we extend our deflation strategy (Chen *et al.*, 2019) to the GCRO-DR algorithm. The approach is based on de Sturler's (1999) algorithm, where GMRES is used within the inner loop. In Section 2, an introductory description of the adjoint approach, widely used in aerodynamic optimization, is presented. In Section 3, we briefly present both traditional deflation and dynamic deflation with restarting (Chen *et al.*, 2019). In Section 4, we present the novel aspect of this article, where dynamic deflation with restarting is introduced within GCRO. In Section 5, we demonstrate numerical results on two test cases and investigate the impact of the number of deflated vectors, the scalability and compare the difference between deflated and dynamic deflated with restarting for both GMRES(m) and GCRO(m).

2. Sensitivity analysis via the adjoint approach

Consider optimizing an objective function, $I = I(\mathbf{w}, \mathbf{x}_s)$, with respect to the surface geometry in shape optimization, where $\mathbf{w}$ is the conservative flow state, and $\mathbf{x}_s$ represents a vector of the boundary grid points. We can express the primal governing equation, as $\mathbf{R}(\mathbf{w}, \mathbf{x}_s)$, where we are interested in the state vector, $\mathbf{w}$, that satisfies the following condition:

$$\mathbf{R}(\mathbf{w}, \mathbf{x}_s) = 0, \tag{1}$$

for a given set of boundary grid points, $\mathbf{x}_s$. In the problems considered in this work, we are interested in minimizing an integral function, $I = I(\mathbf{w}, \mathbf{x}_s)$, such as the lift or drag coefficient. The differential equation constraining the state, $\mathbf{w}$ is incorporated to form a Lagrangian:

$$\mathbf{L}(\mathbf{w}, \mathbf{x}_s) = I(\mathbf{w}, \mathbf{x}_s) + \psi^T \mathbf{R}(\mathbf{w}, \mathbf{x}_s), \tag{2}$$

where the Lagrange multipliers, ψ will play the role of adjoint variables satisfying the dual problem. The Lagrangian can be made insensitive to perturbations in the state vector by requiring that $d\mathbf{L} = 0$ for any $d\mathbf{w}$. Enforcing this condition yields:

$$d\mathbf{L} = \frac{\partial \mathbf{I}}{\partial \mathbf{x}_s} d\mathbf{x}_s + \frac{\partial \mathbf{I}}{\partial \mathbf{w}} d\mathbf{w} + \psi^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}_s} d\mathbf{x}_s + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} d\mathbf{w} \right] \tag{3}$$

$$= \left[\frac{\partial \mathbf{I}}{\partial \mathbf{x}_s} + \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}_s} \right] d\mathbf{x}_s + \left[\frac{\partial \mathbf{I}}{\partial \mathbf{w}} + \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right] d\mathbf{w}, \tag{4}$$

where the partial derivatives are linearizations of the objective function, I and the residual $\mathbf{R}$ about the state, $\mathbf{w}$ and the boundary grid points, $\mathbf{x}_s$. For any arbitrary perturbation $d\mathbf{w}$, the stationary property of the Lagrangian function is achieved by choosing ψ such that:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right]^T \psi = - \left[\frac{\partial \mathbf{I}}{\partial \mathbf{w}} \right]^T, \tag{5}$$

and the sensitivity of the Lagrangian to the boundary $\mathbf{x}_s$ can be expressed as:

$$\frac{d\mathbf{L}}{d\mathbf{x}_s} = \frac{\partial \mathbf{I}}{\partial \mathbf{x}_s} + \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}_s}. \tag{6}$$

The adjoint approach effectively removes the need to compute a $\frac{d\mathbf{w}}{d\mathbf{x}_s}$, where an accurate estimate of this sensitivity requires a re-evaluation of the primal governing equation, $\mathbf{R}(\mathbf{w}, \mathbf{x}_s) = 0$, N times, where N represents the number of design variables and thus is computationally costly. Once the gradient of the objective function is acquired, it is given to an optimization algorithm which provides for a search direction that ensures a sufficient reduction in the objective function while ensuring that the constraints are satisfied.

3. Krylov subspace methods

Krylov subspace methods solve linear algebraic systems not by solving the system exactly, as in a Gaussian elimination approach for instance, but by approximating the problem of interest in a Krylov subspace. Given a good approximation, this method becomes an alternate method for solving linear systems of equations in a way which does not depend on the dimension of the system. Due to the property of mapping the linear systems into alternate subspaces which can represent the equations accurately with fewer dimensions, the large sparse system of equations can be solved in fewer iterative cycles when compared to conventional iterative methods (Saad, 2003). In most cases, the discrete adjoint equation in ASO is associated with a non-symmetric matrix, and hence the Krylov subspace method that is most suitable to solve such system of linear equations is the generalized minimal residual (GMRES) method.

3.1 A restarted generalized minimal residual method solver

Given an adjoint equation as described by equation (5), we need to solve a linear system of equations of the form, $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ denotes the transpose of the global Jacobian matrix,

$\mathbf{x}$ is the vector of the Lagrange multipliers, ψ, and $\mathbf{b}$ is the derivative of the objective with respect to the state. The entries of $\mathbf{A}$ and $\mathbf{b}$ are acquired through automatic differentiation.

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a square nonsingular $n \times n$ non-symmetric real matrix and $\mathbf{b} \in \mathbb{R}^n$ be a vector of length n. To curb the number of basis vectors that are generated during the iterative process due to memory restrictions, the restarted GMRES or GMRES(m), where m is the maximum number of Krylov basis vectors, is the preferred technique within the CFD community (Saad, 2003). If we employ a restarted GMRES(m) solver, then the Krylov subspace after m iterations can be denoted as:

$$Span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{m-2}\mathbf{r}_0, \mathbf{A}^{m-1}\mathbf{r}_0\}, \tag{7}$$

where $\mathbf{r}_0 \in \mathbb{R}^n$ is the residual vector. Then the approximate solution vector can be expressed as a linear combination of the basis vectors $\mathbf{V}_m$:

$$\mathbf{x}_m = \mathbf{x}_0 + \mathcal{K}_m = \mathbf{x}_0 + \mathbf{V}_m\mathbf{y}, \tag{8}$$

where $\mathbf{x}_m \in \mathbb{R}^n$ and $\mathbf{V}_m \in \mathbb{R}^{n \times m}$ is comprised of the Krylov subspace basis vectors:

$$\mathbf{V}_m = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{m-1}], \tag{9}$$

generated through the Arnoldi procedure and $\mathbf{y} \in \mathbb{R}^m$ is the coefficient vector for $\mathbf{V}_m$. The resulting Arnoldi relation can be presented as:

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m, \tag{10}$$

where $\overline{\mathbf{H}}_m$ is an upper Hessenberg matrix (Saad, 2003). If we define the m-th residual vector $\mathbf{r}_m \in \mathbb{R}^n$ with $\mathbf{r}_m \in Range(\mathbf{V}_m)$, as:

$$\begin{aligned} \mathbf{r}_m &= \mathbf{b} - \mathbf{A}(\mathbf{x}_0 + \mathbf{V}_m\mathbf{y}) \\ &= \mathbf{r}_0 - \mathbf{A}\mathbf{V}_m\mathbf{y} \\ &= \mathbf{V}_{m+1}\mathbf{c} - \mathbf{V}_{m+1}\overline{\mathbf{H}}_m\mathbf{y} \\ &= \mathbf{V}_{m+1}(\mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}), \end{aligned} \tag{11}$$

then there exists a unique coefficient vector, $\mathbf{y}$, that minimizes the L_2 norm of the residual vector, through the solution of the least squares problem $||\mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}||$. Here $\mathbf{c}$ is a vector formed through the product of the $m+1$ basis vectors and the residual from the previous restart cycle. When implementing a general restarted GMRES method, the vector $\mathbf{c}$ can be obtained through plane rotations, and does not need to be formed explicitly with the Krylov basis vectors, and the solution vector $\mathbf{y}$ for the least squares problem can be obtained with backward substitution. The solution $\mathbf{x}$ can then be formed through equation (8), and this concludes a typical series of steps for any base GMRES(m) solver.

3.2 Deflated restarting

Van der Vorst and Vuik (1993) analyzed the convergence behaviors of full GMRES and GMRES(m), and concluded that the full GMRES approaches a superlinear convergence rate as the number of iterative cycles increase. However, for ill-conditioned systems, the restarted GMRES method often results in slow or stalled convergence. Instead of discarding previous

Krylov subspace information which negatively impacts the superlinear convergence of GMRES(m), Morgan (1995) proposed including eigenvectors corresponding to eigenvalues with small or negative real parts of the Hessenberg matrix from the previous restart cycles. Deflation strategies can either be implemented through an augmentation of previous harmonic Ritz eigenvectors (Morgan, 2000) or through the addition of Schur vectors corresponding to the k smallest eigenvalues within a preconditioner (Erhel *et al.*, 1996). Both approaches have proven to improve the convergence rate when solving ill-conditioned linear systems; however, the later approach has demonstrated to increase memory consumption (Pashos *et al.*, 2009). For a large adjoint linear system of equations, the storage of the Jacobian matrices already requires a significant amount of memory, and hence the approach of using deflation strategy within a preconditioner further deteriorates the memory budget for iterative solvers.

The introduction of deflated restarting can be summarized through the following brief outline. Specific details of the implementation are expanded in the following subsection based on the approach presented by Morgan (2002). In essence, at the end of a cycle of standard GMRES(m), the m eigenpairs are evaluated and sorted to define the k smallest eigenpairs. The k harmonic Ritz vectors are then orthonormalized through QR factorization to form the first set of k columns of the Krylov subspace vectors of the next restart cycle. If the k harmonic Ritz vectors are $\mathbf{g}_i$, where $0 \leq i \leq k$, and $G_k = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}] \in \mathbb{R}^{m \times k}$, through QR factorization of:

$$\left[\begin{bmatrix} \mathbf{G}_k \\ \mathbf{0}_{1\times k} \end{bmatrix}, \mathbf{c} - \overline{\mathbf{H}}_m \mathbf{y}\right] = \mathbf{P}_{k+1}\mathbf{\Gamma}_{k+1}, \tag{12}$$

the resulting $\mathbf{P}_k$ are columns of orthogonal vectors with unit length. Thus, the span of the new Krylov subspace can be written as:

$$Span\{\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_{k-}1, \tilde{\mathbf{r}}_k, \mathbf{A}\tilde{\mathbf{r}}_k, \mathbf{A}^2\tilde{\mathbf{r}}_k, \ldots, \mathbf{A}^{m-k-1}\tilde{\mathbf{r}}_k\}, \tag{13}$$

where $\tilde{\mathbf{r}}_k = \mathbf{V}_m\mathbf{p}_k$, and $\mathbf{P}_k = [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{k-1}]$. In general, during Arnoldi iterations, the order in which the eigenvectors are factored out is based on the magnitudes of the eigenvalues. It is well-established that negative eigenvalues impair the convergence rate.

By augmenting the degenerate eigenvectors into the new Krylov subspace, the Gram–Schmidt process orthonormalizes the newly generated Krylov subspace basis vectors against those eigenvectors associated with Ritz pairs from the previous restart cycles. Once placed within the subspace for GMRES, the Ritz values are essentially deflated from the linear problem. Deflated restarting thus modifies the spectral distribution of the linear system and improves the convergence performance. In practical implementations, there exist difficulties in obtaining the exact eigenvectors during the process of solving a linear system of equations. Therefore, the accurate estimation of such eigenvectors is necessary for an effective deflated restarting procedure.

3.3 Estimation of eigenvectors for dynamic deflated restarting

For a large ill-conditioned linear system, the computational cost of extracting the eigenvalues and eigenvectors can be prohibitively large. Therefore, we have chosen to adopt the Rayleigh–Ritz method (Morgan and Zeng, 2006), which provides an accurate approximation in the vicinity of the origin at a lower cost. When solving a linear system of equations $\mathbf{Ax} = \mathbf{b}$ with a preconditioner $\mathbf{M}^{-1}$, there exists the Arnoldi relation that:

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m, \tag{14}$$

where the upper Hessenberg matrix $\overline{\mathbf{H}}_m \in \mathbb{R}^{(m+1)\times m}$. Due to the Petrov–Galerkin orthogonality, there exists a relation that:

$$\mathbf{V}_m^T\mathbf{M}^{-T}\mathbf{A}^T\left(\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m\mathbf{g}_k - \lambda_i\mathbf{V}_m\mathbf{g}_i\right) = 0, \tag{15}$$

where $0 \le i \le m$, and $(\lambda_i, \mathbf{g}_i)$ are defined as Ritz pairs, or even harmonic Ritz pairs when $\sigma = 0$ in Morgan and Zeng (2006). The Hermitian transpose of equation (14):

$$\mathbf{V}_m^T\mathbf{M}^{-T}\mathbf{A}^T = \overline{H}_m^T\mathbf{V}_{m+1}^T, \tag{16}$$

is then substituted into the orthogonality condition of equation (15) to yield:

$$\overline{\mathbf{H}}_m^T\overline{\mathbf{H}}_m\mathbf{g}_i = \lambda_i\overline{\mathbf{H}}_m^T\mathbf{V}_{m+1}^T\mathbf{V}_m\mathbf{g}_i, \tag{17}$$

where $\overline{\mathbf{H}}_m$ has the following form:

$$\overline{\mathbf{H}}_m = \begin{bmatrix} \mathbf{H}_m \\ h_{m+1,m}\mathbf{e}_m^T \end{bmatrix}, \text{ with } \mathbf{H}_m \in \mathbb{R}^{m\times m}, \tag{18}$$

where $h_{m+1,m}$ is the nonzero entry in the $(m+1)$-th row of the Hessenberg matrix $\overline{\mathbf{H}}_m$ before m plane rotations (Saad, 2003). Combining equation (17) and equation (18) results into the following eigenvalue problem as introduced by Morgan (Morgan, 2002) and appears as Proposition 1 in Giraud *et al.* (2010):

$$\left(\mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-T}\mathbf{e}_m\mathbf{e}_m^T\right)\mathbf{g}_i = \lambda_i\mathbf{g}_i \tag{19}$$

where $0 \le i \le m$. Equation (19) returns Ritz eigenpairs of matrix $\mathbf{A}$ from the Krylov subspace and provides for the best set of approximate eigenpairs (Paige *et al.*, 1995). In addition to the steps in the Arnoldi process for the general GMRES method, the GMRES-DR (m, k) solver requires a search for the k smallest harmonic Ritz pairs where k is the predetermined number of harmonic Ritz eigenvectors from the Hessenberg matrix for deflated restarting.

Due to lack of reliable guidelines to decide the number of deflated vectors, in our earlier work (Chen *et al.*, 2019), we proposed that the number of deflated vectors can be decided by the number of negative harmonic eigenpairs, k^* and subsequently, k^* of the smallest Ritz values by magnitude are chosen to be deflated from the spectrum. Once k^* is decided, the newly augmented Krylov subspace can be expressed as:

$$Span\{\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_{k^*} - 1, \tilde{\mathbf{r}}_{k^*}, \mathbf{A}\tilde{\mathbf{r}}_{k^*}, \mathbf{A}^2\tilde{\mathbf{r}}_{k^*}, \ldots, \mathbf{A}^{m-k-1}\tilde{\mathbf{r}}_{k^*}\}. \tag{20}$$

Its pseudocode is given in Algorithm 1 in the Appendix 1.

4. Recycling Krylov subspace solver

Recycling Krylov methods have been thoroughly discussed and presented in (Xu and Timme, 2017; Parks *et al.*, 2006). The purpose of the derivations in this section is to

demonstrate how our algorithm can be integrated into the current framework of recycling Krylov subspace solvers. Note that for more detailed explanations and derivations, please refer to the cited papers (Xu and Timme, 2017; Parks *et al.*, 2006).

4.1 Generalized conjugate residual (GCR) as outer iterations

Consider solving a linear system of equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{21}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{R}^{n\times 1}$ and $\mathbf{b} \in \mathbb{R}^{n\times 1}$, and the two matrices that span the outer subspaces are:

$$\begin{aligned} \mathbf{U}_k &= [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k], \\ \mathbf{C}_k &= [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k], \end{aligned} \tag{22}$$

where $\mathbf{U}_k \in \mathbb{R}^{n\times k}$, $\mathbf{C}_k \in \mathbb{R}^{n\times k}$, and $\mathbf{C}_k^T\mathbf{C}_k = \mathbf{I}_k$. The principal relation between these two matrices is:

$$\mathbf{C}_k = \mathbf{A}\mathbf{U}_k. \tag{23}$$

As the invariant subspace C_k is also a Krylov subspace with respect to the residual vector $\mathbf{r}_k$, then through the Arnoldi process, each basis vector is ensured to be orthogonal to other basis vectors, including $\mathbf{r}_k$. Thus, we can have the following relation:

$$\mathbf{C}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}_k) = 0, \tag{24}$$

which leads to:

$$\mathbf{A}\mathbf{x}_k = \mathbf{C}_k\mathbf{C}_k^T\mathbf{b}, \tag{25}$$

or

$$\mathbf{x}_k = \mathbf{U}_k\mathbf{C}_k^T\mathbf{b}. \tag{26}$$

Therefore, by subtracting $\mathbf{A}\mathbf{x} = \mathbf{b}$ from equation (25), the linear system of equations required to be solved becomes:

$$\mathbf{A}(\mathbf{x} - \mathbf{x}_k) = \mathbf{A}\boldsymbol{\varepsilon}_k = \mathbf{b} - \mathbf{A}\mathbf{U}_k\mathbf{C}_k^T\mathbf{b} = \left(\mathbf{I} - \mathbf{C}_k\mathbf{C}_k^T\right)\mathbf{b}. \tag{27}$$

In addition, from the following equation for the residual vector:

$$\mathbf{b} - \mathbf{A}\mathbf{x}_k = \mathbf{b} - \mathbf{A}\left(\mathbf{U_k}\mathbf{C}_k^T\mathbf{b}\right) = \mathbf{r}_k, \tag{28}$$

Equation (27) yields:

$$\mathbf{A}\boldsymbol{\varepsilon}_k = \mathbf{r}_k, \tag{29}$$

which is the primary linear system of equations to be solved in the inner iterative method.

4.2 Generalized minimal residual (GMRES) as inner iterations

Based on equation (29), the solution that needs to be approximated with the Krylov subspace basis vectors is $\boldsymbol{\varepsilon_k}$, rather than $\mathbf{x}$ in the linear system of equations, $\mathbf{Ax} = \mathbf{b}$. Note that when the Krylov subspace $\mathbf{V}_m$ is generated during Arnoldi iterations, the Gram–Schmidt process orthnormalizes the column vectors in both basis subspaces $\mathbf{V}_m$ and $\mathbf{C}_k$. In general, the Krylov subspace $\mathbf{V}_m \in \mathbb{R}^{n\times k}$ within a conventional restarted GMRES(m) solver can be written as:

$$\mathbf{V}_m = [\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_m]. \tag{30}$$

However, through orthogonalization of $\mathbf{V}_m$ with respect to $\mathbf{C}_k$, the resulting Krylov subspace $\mathbf{V}'_m$ ensures its orthogonality to $\mathbf{C}_k$ by excluding the orthogonal projections of the column vectors in $\mathbf{V}_m$ onto the subspace $\mathbf{C}_k$, and can be formed as:

$$\mathbf{V}'_m = \mathbf{V}_m - \mathbf{C}_k\mathbf{C}_k^T\mathbf{V}_m. \tag{31}$$

Thus, the solution vector $\boldsymbol{\varepsilon}_k$ can be presented as:

$$\boldsymbol{\varepsilon}_k = \mathbf{V}'_\mathbf{m}\mathbf{y} = \left[\mathbf{V}_m - \mathbf{C}_k\mathbf{C}_k^T\mathbf{V}_m\right]\mathbf{y}, \tag{32}$$

where $\mathbf{C}_k^T\mathbf{V}_m$ is the inner product between $\mathbf{V}_m$ and $\mathbf{C}_k$ for the length of a scalar vector projection. Therefore, the coefficient vector $\mathbf{y}$ for these subspace basis vectors can be acquired through:

$$y = argmin||\mathbf{r}_k - \mathbf{A}\left(\mathbf{I} - \mathbf{C}_k\mathbf{C}_k^T\right)\mathbf{V}_m\mathbf{y}|| = argmin||\mathbf{r}_k - \mathbf{A}\mathbf{e}_k||. \tag{33}$$

With $\mathbf{v}_0 = \mathbf{r}_k/||\mathbf{r}_k||$, the Arnoldi relation yields:

$$\mathbf{AV}_m = \mathbf{C}_k\mathbf{C}_k^T\mathbf{AV}_m + \mathbf{V}_{m+1}\overline{\mathbf{H}}_m. \tag{34}$$

By defining $\left(\mathbf{I} - \mathbf{C}_k\mathbf{C}_k^T\right)\mathbf{A}$ as $\mathbf{A}_{c_k}$, equation (33) and equation (34) can be rewritten as:

$$y = argmin||\mathbf{r}_k - \mathbf{A}_{c_k}\mathbf{V}_m\mathbf{y}||, \tag{35}$$

and

$$\mathbf{A}_{c_k}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m. \tag{36}$$

From equations (33) and (35), the solution vector $\boldsymbol{\varepsilon}_k$ can finally be represented as:

$$\boldsymbol{\varepsilon}_k = \mathbf{A}^{-1}\mathbf{A}_{ck}\mathbf{V}_m\mathbf{y}. \tag{37}$$

Once the coefficient vector $\mathbf{y}$ is obtained through the solution to the least squares problem in equation (35), it can be substituted into equation (32) to produce the final update algorithm for $\mathbf{x}_{k+1}$:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \boldsymbol{\varepsilon}_k \\ &= \mathbf{U}_k\mathbf{C}_k^T\mathbf{b} + \mathbf{A}^{-1}\mathbf{A}_{ck}\mathbf{V}_m\mathbf{y} \\ &= \mathbf{U}_k\mathbf{C}_k^T\mathbf{b} + \mathbf{A}^{-1}\left(\mathbf{I} - \mathbf{C}_k\mathbf{C}_k^T\right)\mathbf{A}\mathbf{V}_m\mathbf{y} \\ &= \mathbf{U}_k\mathbf{C}_k^T\mathbf{b} + \mathbf{V}_m\mathbf{y} - \mathbf{U}_k\mathbf{C}_k^T\mathbf{A}\mathbf{V}_m\mathbf{y} \\ &= \mathbf{U}_k\left[\mathbf{C}_k^T(\mathbf{b} - \mathbf{A}\mathbf{V}_m\mathbf{y})\right] + \mathbf{V}_m\mathbf{y},\end{aligned} \tag{38}$$

and $\mathbf{r}_{k+1}$:

$$\begin{aligned}\mathbf{r}_{k+1} &= \mathbf{b} - \mathbf{A}(\mathbf{x}_k + \boldsymbol{\varepsilon}_k) \\ &= \mathbf{b} - \mathbf{A}\left(\mathbf{x}_k + \mathbf{A}^{-1}\mathbf{A}_{c_k}\mathbf{V}_m\mathbf{y}\right) \\ &= \mathbf{r}_k - \mathbf{V}_{m+1}\overline{H}_m\mathbf{y}.\end{aligned} \tag{39}$$

Accordingly, equation (38) can be simplified to:

$$\mathbf{x}_{k+1} = \mathbf{U}_k\mathbf{z} + \mathbf{V}_m\mathbf{y}, \tag{40}$$

where $\mathbf{z} = \mathbf{C}_k^T(\mathbf{b} - \mathbf{A}\mathbf{V}_m\mathbf{y})$. Upon completion of the inner GMRES(m), by setting $\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{c}_{k+1}$, then $\mathbf{c}_{k+1}$ and $\mathbf{u}_{k+1}$ for the next outer GCR iteration can be formulated as:

$$\begin{aligned}\mathbf{c}_{k+1} &= \mathbf{V}_{m+1}\overline{H}_m = \mathbf{A}_{c_k}\mathbf{V}_m\mathbf{y} = \mathbf{A}\mathbf{V}_m\mathbf{y} - \mathbf{C}_k\mathbf{C}_k^T\mathbf{A}\mathbf{V}_m\mathbf{y}, \\ \mathbf{u}_{k+1} &= \mathbf{A}^{-1}\mathbf{A}_{c_k}\mathbf{V}_m\mathbf{y} = \mathbf{V}_m\mathbf{y} - \mathbf{U}_k\mathbf{C}_k^T\mathbf{A}\mathbf{V}_m\mathbf{y}.\end{aligned} \tag{41}$$

4.3 Deflated restarting within the GCRO framework

The nested Krylov subspace method GCRO uses a GCR solver for the outer iteration, and provides a choice for an alternate Krylov subspace iterative method for the inner loop. In outer GCR iterations, the *Range*($\mathbf{U}_k$) can be an arbitrary invariant subspace as long as the primary relation $\mathbf{C}_k = \mathbf{A}\mathbf{U}_k$ holds. One option to construct $\mathbf{U}_k$ is a Krylov subspace $K^k(\mathbf{A}, \mathbf{r}_0) = [\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0]$ (de Sturler, 1996). The option for GCRO-DR is to combine GCRO and GMRES-DR described in the previous section by using Ritz vectors as the base to span the outer subspace while preserving GCR orthogonality as shown in subsection 4.2.

Initially, when solving a linear system of equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ with a general restarted GMRES(m), we can obtain the Arnoldi relation $\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$, and subsequently acquire the Ritz pairs ($\mathbf{g}_k$, λ_k) by solving the eigenvalue problem $(\mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-T}\mathbf{e}_m\mathbf{e}_m^T)\mathbf{g}_i = \lambda_k\mathbf{g}_i$, where $0 \le i \le m$. $\mathbf{H}_m$ is derived from the upper Hessenberg matrix $\overline{\mathbf{H}}_m$ with the last nonzero row removed. Considering a fixed deflated restarting approach, the number of selected Ritz vectors k is predetermined beforehand. Through QR factorization of the columns of the selected Ritz vectors $\mathbf{G}_k = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1}] \in \mathbb{R}^{m\times k}$:

$$\mathbf{P}_k\boldsymbol{\Gamma}_k = \mathbf{G}_k, \tag{42}$$

The resulting $\mathbf{P}_k$ are columns of orthogonal vectors with unit length. Since $\mathbf{P}_k \in \mathbb{R}^{m\times k}$, it can be mapped to $\mathbb{R}^{n\times k}$ with the previous Krylov subspace $\mathbf{V}_m$, and we can thus evaluate the deflated vector, $\tilde{\mathbf{Y}}_k$

$$\tilde{\mathbf{Y}}_k = \mathbf{V}_m \mathbf{P}_k. \tag{43}$$

Through a QR factorization of $\overline{\mathbf{H}}_m \mathbf{P}_k$, where $\overline{\mathbf{H}}_m$ is an outcome of the last Arnoldi relation, $\mathbf{C}_k$ and $\mathbf{U}_k$ can be defined as:

$$\begin{aligned} \mathbf{C}_k &= \mathbf{V}_{m+1} \mathbf{Q}_k, \\ \mathbf{U}_k &= \tilde{\mathbf{Y}}_k \mathbf{R}^{-1}. \end{aligned} \tag{44}$$

From the primary relation of the outer GCR iteration, $\mathbf{C}_k$ can be obtained by $\mathbf{A}\mathbf{U}_k$, but in general it is more computationally expensive, since $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{U}_k \in \mathbb{R}^{n \times k}$, while $\mathbf{V}_{m+1} \in \mathbb{R}^{n \times (m+1)}$, and $\mathbf{Q}_k \in \mathbb{R}^{(m+1) \times k}$. Since the column vectors in $\mathbf{D}_k$ are not of unit length, to obtain an equivalent Arnoldi relation for GCRO-DR, we define $\mathbf{D}_k \in \mathbb{R}_k$ as a diagonal matrix:

$$\mathbf{D}_k = diag\left(||\mathbf{u}_1||^{-1}, ||\mathbf{u}_2||^{-1}, \ldots, ||\mathbf{u}_k||^{-1} \right), \tag{45}$$

such that $\tilde{\mathbf{U}}_k = \mathbf{U}_k \mathbf{D}_k$ where all vectors in $\tilde{\mathbf{U}}_k$ have unit length, since $\tilde{\mathbf{U}}_k$ is used as the basis vectors in the equivalent Arnoldi relation in equation (47). According to subsection 4.2, through $m - k$ steps of the Arnoldi process, similar to equation (34), we can obtain:

$$\mathbf{A}\mathbf{V}_{m-k} = \mathbf{C}_k \mathbf{C}_k^T \mathbf{A}\mathbf{V}_{m-k} + \mathbf{V}_{m-k+1} \overline{\mathbf{H}}_{m-k}. \tag{46}$$

If we define $\mathbf{C}_k^T \mathbf{A}\mathbf{V}_{m-k}$ as $\mathbf{B}_{k,m-k}$ in the above equation, it can then be combined with $\mathbf{C}_k = \mathbf{A}\mathbf{U}_k$, and the equivalent Arnoldi relation can be expressed as:

$$\mathbf{A}\widehat{\mathbf{V}}_m = \widehat{\mathbf{W}}_{m+1} \overline{\mathbf{G}}_m, \tag{47}$$

where:

$$\widehat{\mathbf{V}}_m = \begin{bmatrix} \tilde{\mathbf{U}}_k & \mathbf{V}_{m-k} \end{bmatrix}, \widehat{\mathbf{W}}_{m+1} = \begin{bmatrix} \mathbf{C}_k & \mathbf{V}_{m-k+1} \end{bmatrix}, \overline{\mathbf{G}}_m = \begin{bmatrix} \mathbf{D}_k & \mathbf{B}_{k,m-k} \\ 0 & \overline{\mathbf{H}}_{m-k} \end{bmatrix}. \tag{48}$$

The matrix $\overline{\mathbf{G}}_m$ can be regarded as an equivalent upper Hessenberg matrix. If a preconditioner $\mathbf{M}^{-1}$ is applied into the Arnoldi process while solving a linear system of equations with GCRO-DR, the Arnoldi relation in equation (47) can be expressed as:

$$\mathbf{A}\mathbf{M}^{-1}\widehat{\mathbf{V}}_m = \widehat{\mathbf{W}}_{m+1} \overline{\mathbf{G}}_m. \tag{49}$$

Accordingly, the solution and residual vectors can be obtained by:

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_i + \mathbf{M}^{-1}\widehat{\mathbf{V}}_m \mathbf{y}, \\ \mathbf{r}_{i+1} &= \mathbf{r}_i - \mathbf{A}\mathbf{M}^{-1}\widehat{\mathbf{V}}_m \mathbf{y} = \mathbf{r}_i - \widehat{\mathbf{W}}_{m+1} \overline{\mathbf{G}}_m \mathbf{y}. \end{aligned} \tag{50}$$

4.4 Estimation of eigenvectors for dynamic deflated restarting

Similar to finding Ritz vectors for GMRES-*dyn*DR in subsection 3.3, through Petrov–Galerkin orthogonality and the Arnoldi relation in subsection 4.3, the following relation can be obtained:

$$\widehat{\mathbf{V}}_m^T \mathbf{M}^{-T} \mathbf{A}^T \left(\mathbf{A} \mathbf{M}^{-1} \widehat{\mathbf{V}}_m \mathbf{g}_i - \lambda_i \widehat{\mathbf{V}}_m \mathbf{g}_i \right) = 0, \tag{51}$$

where $0 \leq i \leq m$. According to equation (49), and $\widehat{\mathbf{W}}_{m+1}^T \widehat{\mathbf{W}}_{m+1} = \mathbf{I}_{m+1}$, the above equation can be rewritten as:

$$\overline{\mathbf{G}}_m^T \overline{\mathbf{G}}_m \mathbf{g}_i - \lambda_i \left(\overline{\mathbf{G}}_m^T \widehat{\mathbf{W}}_{m+1}^T \widehat{\mathbf{V}}_m \right) \mathbf{g}_i = 0, \tag{52}$$

where $(\lambda_i, \mathbf{g}_i)$ are the approximate eigenpairs. By inspecting the spectral distribution of λ_k, the number of deflated vectors k^* in our development is based on the number of Ritz values λ_k with negative real parts. Subsequently, the Ritz vectors corresponding to the k^* of the smallest Ritz values by magnitude are selected as $\mathbf{g}_{k^*}$. Then, through a QR factorization of $\mathbf{g}_{k^*}$, the orthonormalized vectors $\mathbf{P}_{k^*}$, and the deflated vectors $\tilde{\mathbf{Y}}_{k^*} = \hat{\mathbf{V}}_m \mathbf{P}_{k^*}$ can be acquired. The dynamic k^* value allows unfavourable spectral distributions to be spectrally deflated. The Arnoldi relation in equations (34) and (49) remains unchanged in our approach, while the columns of $\widehat{\mathbf{V}}_m$, $\widehat{\mathbf{W}}_{m+1}$, and $\overline{\mathbf{G}}_m$ become:

$$\hat{\mathbf{V}}_m = \left[\widetilde{\mathbf{U}}_{k^*} \quad \mathbf{V}_{m-k^*} \right], \widehat{\mathbf{W}}_{m+1} = \left[\mathbf{C}_{k^*} \quad \mathbf{V}_{m-k^*+1} \right], \overline{\mathbf{G}}_m = \begin{bmatrix} \mathbf{D}_{k^*} & \mathbf{B}_{k^*,m-k^*} \\ 0 & \overline{\mathbf{H}}_{m-k^*} \end{bmatrix}, \tag{53}$$

so that the memory requirement for a GCRO- *dyn*DR solver for the subspace basis vectors is $m + k^*$, rather than $m + k$, while it is m for the developed GMRES- *dyn*DR solver. The pseudocode of the developed GCRO-*dyn*DR solver is given in Algorithm 2 in the Appendix 2.

5. Results

To validate the developed framework to solve ill-conditioned sparse linear systems, we subject the developed solver to two ASO problems; a two-dimensional NACA0012 airfoil and a three-dimensional isolated CRM wing in viscous flow. All flow solutions were acquired with a three-dimensional Reynolds-averaged Navier–Stokes solver with the SA turbulence model (Spalart and Allmaras, 1992). The governing equations were discretized with a second-order finite-volume scheme with matrix dissipation for the numerical flux. Both first- and second-order Jacobians were evaluated using automatic differentiation and includes the linearization of the turbulence model when noted. The first-order Jacobian was used to precondition the adjoint system. For two-dimensional viscous problems, the first-order Jacobian requires an equivalent of 36 Krylov vectors for storage while the second-order Jacobian warrants 52 vectors with a constant eddy viscosity. In three dimensions, these increase to 120 and 157 Krylov vectors respectively. We explored Xu and Timme (2017) linearly combining a weighted sum of the first- and second-order Jacobians and confirm that it does provide a speed-up in the convergence. However, in this work, we only considered the first-order Jacobian as the preconditioner and preferred to avoid the impact of the choice of weights on the convergence of the various solvers. We will investigate the convergence of drag adjoints for the NACA0012 and CRM cases, which includes the integration of pressure and shear-stress over the full surface of the geometry. Note that all investigations were conducted on a HPC platform with 16 cores per node.

5.1 Two-dimensional NACA0012 airfoil

The first demonstration test was conducted on an adjoint system of equations based on the flow over a two-dimensional NACA0012 airfoil at Mach $M = 0.8$, angle of attack of 0 degrees,

at a Reynolds number of 5.6 million with the SA turbulence model. The computational grid has an average y^+ of one over the surface of the airfoil. Although the size of the second-order Jacobian matrix (49152 × 49152) is relatively small, the extreme ill-conditioned nature of the matrix requires 656 Krylov subspace vectors with full GMRES.

The adjoint solvers considered here are PETSc GMRES, GMRES(m), and the developed GMRES-DR(m, k), GMRES-*dyn*DR(m), GCRO-DR(m, k) and GCRO-*dyn*DR(m). We adopt the notation where m denotes the total number of Krylov basis vectors, while k indicates the deflated vectors. All cases were preconditioned with ILU(0) for the local preconditioner and restrictive additive Schwarz with a single layer of overlapping for the global preconditioner with three outer Richardson iterations.

From the simulation results illustrated in Figure 1(a), full GMRES, in terms of both iterations and CPU time, shows the best convergence performance, while utilizing the most amount of memory at 656 Krylov basis vectors. With restarted GMRES, the minimum number of basis vectors required to ensure convergence is 15. At this level, PETSc GMRES (15) provides for a one-fortieth reduction of the memory required for Krylov subspace to that of full GMRES but requires relatively longer CPU time to reach the same residual level. In addition we compare the convergence behaviour of the developed GMRES(m) algorithm against PETSc's GMRES(m) solver. Since deflated restarting was developed within our in-house GMRES solver, a benchmark comparison of the base GMRES(m) solvers is warranted. In terms of the number of iterations both solvers are almost identical; however, in terms of CPU time, the developed solver is slightly faster. We speculate that the overhead that comes with a fully capable linear solver as it is the case with PETSc is contributing towards the additional computational cost.

Next, we investigate several variants of deflated GMRES and GCRO at equivalent total Krylov basis vectors, and study the impact of the number of deflated vectors to establish an optimal combination as well as to provide evidence to dynamically deflated restarting. The first case uses a total of 15 basis vectors where five are deflated vectors, while 10 are from performing Arnoldi iterations, GMRES-DR(15,5). As for GCRO-DR (15,5), 10 Arnoldi iterations are performed in the inner iterations. At which point, we obtain an upper Hessenberg matrix, $\overline{\mathbf{H}}_m$ of size 11 × 10. This is as opposed to a 16 × 15

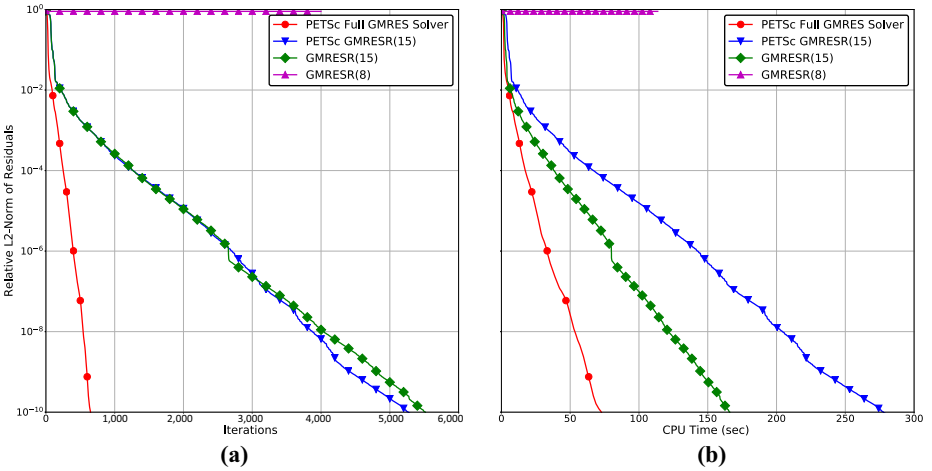

Figure 1. Convergence curves of solving an adjoint system of equations for the 2 D NACA0012 airfoil with PETSc full GMRES, PETSc GMRES(15), the developed GMRES (15), and GMRES(8) solver in terms of (a) iteration cycles and (b) CPU time

matrix in the GMRES-DR approach. To extract the Ritz vectors of length 16, an equivalent Hessenberg matrix is formed as described in equation (53) at the end of the Arnoldi iterations within the inner loop. QR factorization is then performed to produce the deflated vectors which are then employed to compute both $\mathbf{C}_k$ and $\mathbf{U}_k$, effectively doubling the amount of memory required to introduce deflation within GCR over GMRES. In summary, GCRO-DR(15,5) requires $m + k = 20$ vectors to be stored compared to only $m = 15$ for GMRES-DR. A comparison of their computational cost reveals both approaches are comparable, with a slight advantage to GMRES-DR. Again the additional computational cost for the GCRO-DR can be attributed to the difference in evaluating and deflating the Ritz values. In summary, the convergence rates demonstrate that removing k^* smallest eigenvalues at every restart does allow for an increase in the convergence rate, compared with restarted GMRES(15).

Finally, the dynamically deflated GCRO- *dyn*DR(15) and GMRES- *dyn*DR(15) are compared against their fixed deflated counterparts in Figure 2. We observe that both dynamically deflated solvers experience similar convergence rates in terms of the number of iterations and computational cost. However, when compared to their fixed variants with five deflated vectors, the dynamic versions are not as favourable. We conjecture that there are a larger number of small Ritz values than there are negative Ritz values. This can be further illustrated by examining the number of negative Ritz values identified during the execution of the *dyn*DR versions. As illustrated in Figure 4(a) only two deflated vectors were used in the second restart cycle and none were used in subsequent cycles for both the GMRES-*dyn*DR(15) and GCRO-*dyn*DR(15). Therefore, the fixed variants were deflating a larger number of small Ritz values and their inclusion allowed for a more rapid deflation of the associated eigenvalues from the linear system and thus improving the convergence. A further reduction to a total of eight Krylov vectors, as illustrated in Figure 3, demonstrates that selecting too few of vectors is insufficient to restore convergence. However, as shown in Figure 4(b), a judicious inclusion of the associated Ritz vectors into subsequent Krylov basis vectors does allow GCRO-*dyn*DR(8) to attain convergence with no a priori specification of the number of deflated vectors.

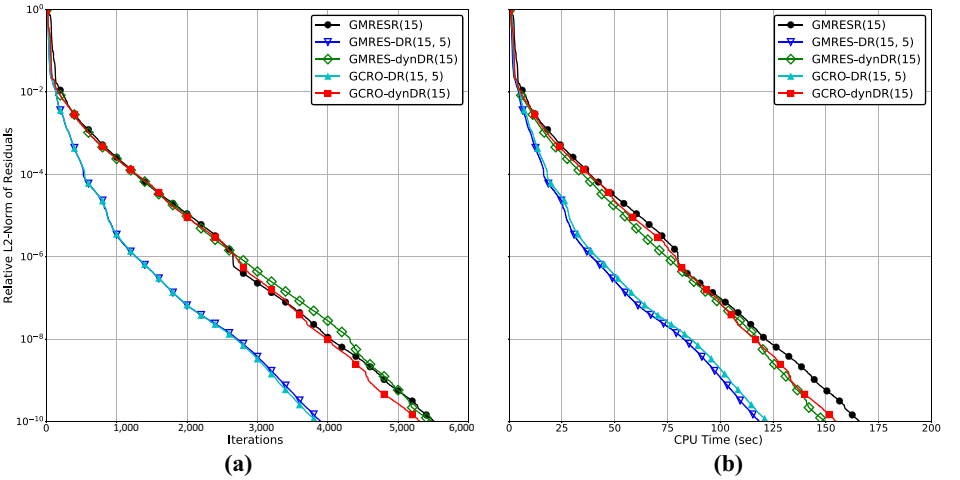

Figure 2.
Convergence curves of solving an adjoint system of equations for the 2 D NACA0012 airfoil with the developed GMRES(15), GMRES-DR(15, 5), GCRO-DR (15, 5), GMRES-*dyn*DR(15), and GCRO-*dyn*DR(15) solver in terms of (a) iteration cycles and (b) CPU time

5.2 Three-dimensional common research model (CRM) isolated wing

The second test case is based on an adjoint system of equations for the 3D RANS flow over an isolated CRM wing at Mach $M = 0.85$, angle of attack of 2.61 degrees and at a Reynolds number of 5 million with the SA turbulence model. The computational grid is of size 1.5 million with an average y^+ of one over the surface of the wing. The steady state nonlinear flow solution was obtained after a six order reduction in the density residual with a 3 W multigrid solver with an LU-SGS smoother at every level. Note that the intention to choose ten orders of magnitude for the residual errors in the following plots is to verify the developed adjoint solver. The second-order Jacobian for the adjoint equation demanded 162 GBs of storage (157 Krylov vectors), while the first-order Jacobian, used for the preconditioner required 133 GBs (120 Krylov vectors), and the eddy viscosities were held constant during the linearization. For this test case, we investigated a number of factors on

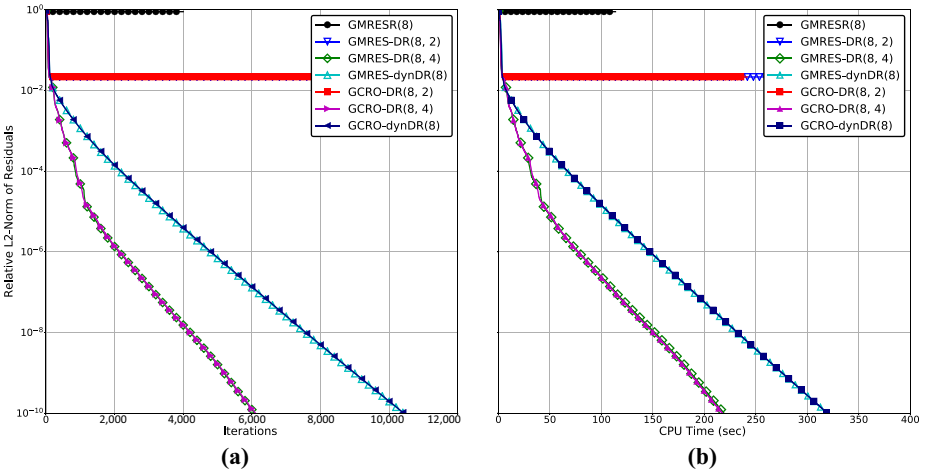

Figure 3. Convergence curves of solving an adjoint system of equations for the 2 D NACA0012 airfoil with the developed GMRES(8), GMRES-DR(8, 2), GCRO-DR(8, 2), GMRES-DR(8, 4), GCRO-DR(8, 4), GMRES-*dyn*DR(8) and GCRO-*dyn*DR(8) solver in terms of (a) iteration cycles, and (b) CPU time

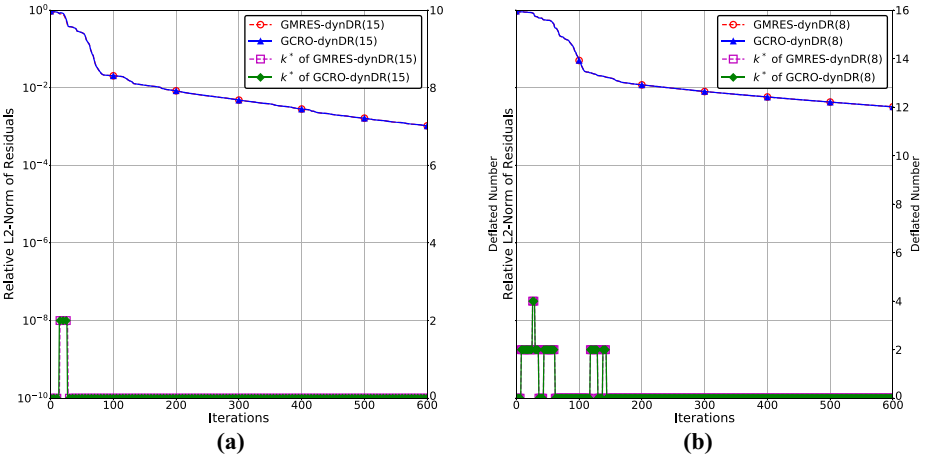

Figure 4. Changes of the number of deflated vector when solving an adjoint system of equations for the 2 D NACA0012 airfoil with the developed (a) GMRES-*dyn*DR(15), and GCRO-*dyn*DR (15), and (b) GMRES-*dyn*DR(8) and GCRO-*dyn*DR(8) in terms of iteration cycles

the convergence rate and the required computational expense and memory. First, we probed the impact of the number of fixed deflated vectors k; second, the total number of basis vectors for the proposed dynamic deflated variant; third, a comparison of GCRO-DR against GMRES-DR.

We begin the investigation by launching full GMRES with ILU(0) to gauge the required number of basis vectors to attain a ten-fold drop in the convergence. From Figure 5, we can observe that full GMRES requires just over 1,500 vectors. To provide a benchmark for the computational cost of the developed deflated restarting based linear solvers, we first compare PETSc's full GMRES and the developed GMRES solver. As shown in Figure 5(a), both solvers converge at exactly the same rate; however, similar to that reported for the NACA0012 case, the developed linear solver does converge twice faster for this particular test case. Having established the base solvers, we then investigate GMRES with restarts. From our earlier study (Chen *et al.*, 2019), by reducing the number of restart vectors for GMRES(m), we established that when the restart number is set to 75 or lower, the convergence of the restarted GMRES solver stalls immediately in the first few restart cycles. Hence we will employ 75 Krylov subspace vectors to investigate the performance of the developed GCRO-DR solver. From Figure 5(a), with just the inclusion of two deflated vectors, the convergence is re-established, at an equivalent computational cost to full GMRES with a factor 20 reduction of the memory required for storing the Krylov basis vectors. As we increase the number of deflated vectors, the convergence rate gradually increases and at 30 deflated vectors, GCRO-DR(75,30) is approximately equal to full GMRES in terms of the number of iterations. However, the gain in the convergence rate is marginal from increasing the number of deflated vectors from 15 to 30. In addition, the computational cost remains consistent throughout all deflated restarting cases.

Having established the convergence rate of GCRO-DR, we now compare the solver against GMRES-DR. To provide a thorough comparison, we executed each solver starting with four deflated vectors with an increase of two until 12 vectors. The choice is based on having established in Figure 5(b) that both 15 and 30 provided for identical convergence rates in terms of the computational cost. There are several observations we can draw from Figure 6(a). First, as we increase the number of deflated vectors, both GMRES-DR and GCRO-DR improve in terms of the convergence rate. The results demonstrate that optimal

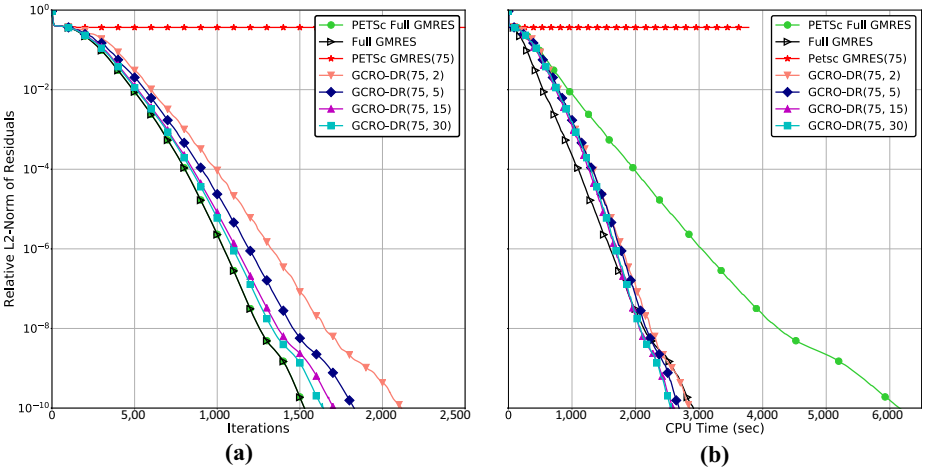

Figure 5.
Convergence curves of solving an adjoint system of equations for the 3 D CRM wing with PETSc full GMRES, the developed GMRES and GCRO-DR solvers with different deflation numbers in terms of (a) iteration cycles and (b) CPU time

convergence in terms of computational expense for this test case is achieved with 10 deflated vectors for both GMRES-DR and GCRO-DR. Second, both GCRO-DR and GMRES-DR demonstrate identical convergence rates in terms of iterations, except for the plateaus observed for GMRES-DR after a ten order reduction in magnitude. Since there exists algebraical equivalence between GCRO-DR and GMRES DR (Parks *et al.*, 2006), we can observe from Figure 6(a) that this is indeed the case, where both approaches have equivalent convergence profiles but depart as their residuals drop to around ten orders of magnitude. We attribute the difference to the dissimilarity between the Ritz values. In GMRES-DR(m, k) the first k columns of basis vectors are obtained by converting the k harmonic Ritz vectors from $\mathbb{R}^{m\times k}$ to $\mathbb{R}^{n\times k}$ with the subspace $\mathbf{V}_m$. Subsequently, a Gram–Schmidt process is initiated to ensure orthogonality among the k columns of basis vectors. However, as for GCRO-DR, the column vectors of $\mathbf{C}_k$ and $\mathbf{U}_k$ (see steps 11 and 12 of Algorithm 3) are ensured to hold orthogonality without an additional Gram-Schmidt process. We believe this difference leads to a positive impact on the accuracy of the approximated Ritz values, and therefore to the convergence performance. Third, GMRES-DR requires less computational cost, compared to its GCRO-DR counterpart. The additional computational cost can be attributed to a larger number of matrix computations to evaluate the Ritz pairs and the resulting Ritz vectors for GCRO-DR as detailed in step 8 of Algorithm 3 compared to step 8 of Algorithm 1 for GMRES-DR. In addition, to generate the subspace $\mathbf{V}_k^{new}$ in GMRES-DR (step 13: Algorithm 1) requires a single matrix computation of size $n \times k$; while, two are required for GCRO-DR (step 12: Algorithm 3). Moreover, GCRO-DR requires additional operations to compute the inverse of $\mathbf{R}$ (step 11: Algorithm 3) and to generate the subspace $\mathbf{U}_k$. Apart from an increase in computational cost, GCRO-DR requires additional memory copies of size $n \times k$ to update the subspace $\mathbf{C}_k$. In conclusion, deflated restarting within GMRES and GCRO do provide for a faster rate of convergence, primarily due to its ability to deflate small Ritz values from the linear system.

Next, we present in Figure 7, results of the dynamically adjusted deflation variants for both GMRES-DR and GCRO-DR. As stated earlier in subsection 4.4, we dynamically change the number of deflated vectors at every restart cycle such that only k^* of the smallest Ritz values are considered, where k^* is the number of negative harmonic Ritz values. We execute

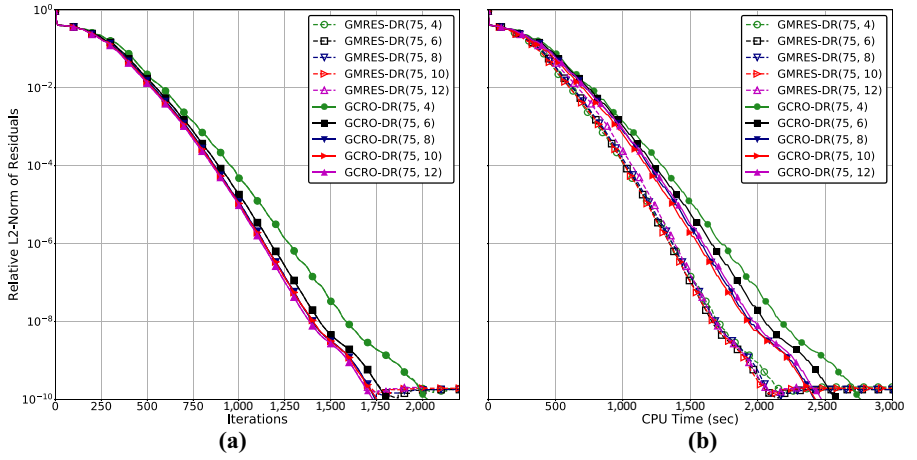

Figure 6. Convergence curves of solving an adjoint system of equations for the 3 D CRM wing with the developed fixed GMRES-DR, and GCRO-DR solver with different deflation numbers in terms of (a) iteration cycles and (b) CPU time

cases with 75 and 25 total Krylov vectors each and compare them against their fixed deflated cases with 10 vectors. Several observations can be drawn from Figure 7. First, as expected both dynamic versions have approximately equivalent convergence rates in terms of iterations. As for the computational cost, GCRO- *dyn*DR(75) is approximately equivalent to its fixed deflated counterpart and the same conclusion can be made of GMRES- *dyn*DR (75). Second, similar to the fixed versions, GMRES- *dyn*DR exhibits slightly faster rates in terms of computational cost over GCRO- *dyn*DR.

A further examination of the number of deflated vectors that are selected for each restart cycle is showcased in Figure 8. As clearly illustrated, both with 75 and 25 total Krylov vectors, GCRO- *dyn*DR and GMRES- *dyn*DR identified the same number of Ritz values. In Figure 8(a), at the start of the second and third cycles, six negative Ritz values were detected, while four in the fourth cycle. The solver then defaults to standard GMRES(*m*) from the fifth cycle, since no negative Ritz values are present and the convergence rate is held steady. As

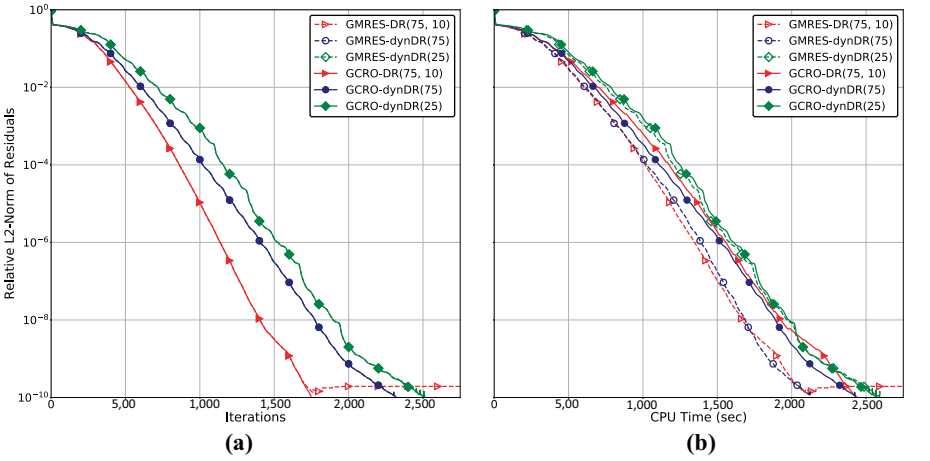

Figure 7.
Convergence curves of solving an adjoint system of equations for the 3 D CRM wing with the developed GCRO-DR(75, 10), GMRES-DR(75, 10), GMRES-*dyn*DR(75), GMRES-*dyn*DR(25), GCRO-*dyn*DR(75) and GCRO-*dyn*DR(25) solvers in terms of (a) iteration cycles and (b) CPU time

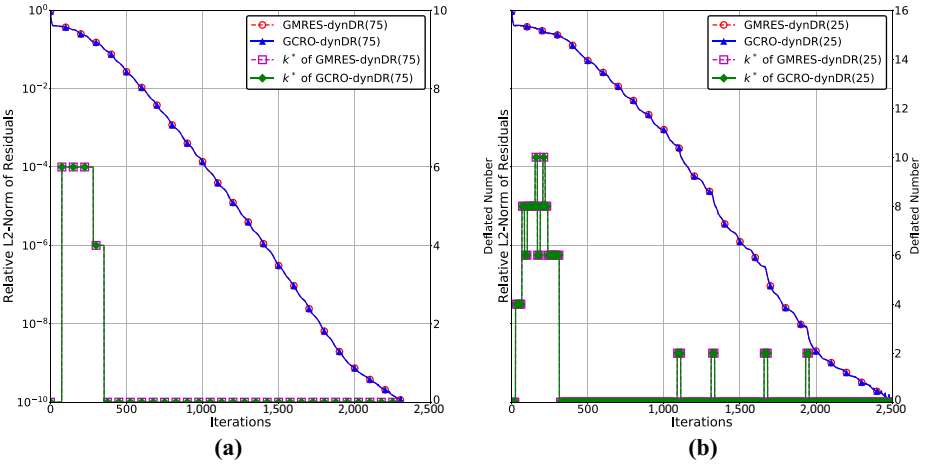

Figure 8.
Changes of the number of deflated vector when solving an adjoint system of equations for the 3 D CRM wing with the developed (a) GMRES-*dyn*DR(75), and GCRO-*dyn*DR(75) solvers, and (b) GMRES-*dyn*DR(25) and GCRO-*dyn*DR(25) solvers in terms of iteration cycles

for the case with a total of 25 vectors, as demonstrated in Figure 8(b), this value peaks at 10 with a minimum of four within the first 11 restart cycles. At which point none are identified and the solver resorts to standard GMRES(m) for much of the remaining; however, at four more instances two negative Ritz values are uncovered, and *dyn*DR is reinitiated for both GMRES and GCRO. We can thus conclude that the timely removal of the correct number of critical eigenvalues from the linear system improves the conditioning and further accelerates the convergence.

At this stage, we have established that *dyn*DR is able to effectively identify the minimum number of deflated vectors that are required for the subsequent cycles. In Figure 9, we compare GCRO-DR against its dynamic counterpart with a total of 25 Krylov vectors similar to that presented earlier where 75 was employed. We choose to compare between 4, 6, and 10 fixed deflated vectors since these number of Ritz values were identified in the dynamic version. At the initial stages of the convergence, we can observe that as we increase from 4 to 10 the convergence rate improves but eventually all fixed deflated versions converge at equivalent rates. However, a further increase to 20 fixed deflated vectors maintains the convergence rate but at the expense of increasing the computational cost (Figure 9(b)). The increase is primarily due to operations that is of order, $O(n \times k)$ as well as memory copies associated with the subspace $\mathbf{C}_k \in \mathbb{R}^{n \times k}$. From the NACA0012 case, we saw that employing too few of fixed vectors may not be sufficient to ensure convergence; while in the case of the CRM wing, we observe that preselecting too large of a number can have a detrimental impact on the computational cost.

We finally perform another study on the importance of continuing to identify the number of Ritz values. In Figure 9, we disable deflated restarting after the 15th cycle, and the two convergence rates and the numbers of identified Ritz values are plotted. As expected, for the disabled case, none appear after the 15th cycle. Most importantly, both curves have identical convergence rates up to approximately the 1100th iteration, where the developed *dyn*DR identifies two negative Ritz values and the convergence rate continues to improve, while the

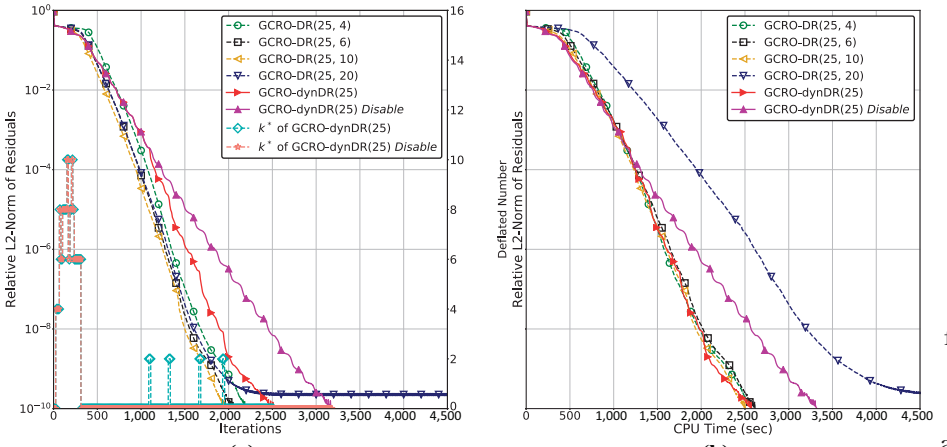

Figure 9. Convergence curves of solving an adjoint system of equations for the 3 D CRM wing with the developed GCRO-DR(25, k) where k = 4,6,10,20, and GCRO-*dyn*DR(25) solvers in terms of (a) iteration cycles, and (b) CPU time

Note: In addition, GCRO-*dyn*DR(25) *Disable* is the GCRO-*dyn*DR(25) solver with deflation disabled after the 15th cycle

disabled version experiences a reduction. The impact on the computational cost is shown in Figure 9(b).

Furthermore, to efficiently harness the concurrent computational resources, scalability of linear system solvers has become an essential performance benchmark. As observed in Figure 10, due to weaker global preconditioning with an increase in the number of cores, restarted GMRES experiences a long plateau in its convergence. However, with the same preconditioner, GCRO- *dyn*DR does not reveal any slowdown during its convergence, and keeps a relatively higher scalability as presented in Figure 10.

To better study the memory footprint, Table I summarizes the convergence performance amongst the solvers shown in Figures 8 and 10 with the required total memory per core. The information is also shown in Figure 11 for an easier comparison where the total memory requirement is reported. The memory includes the storage of the first- and second-order

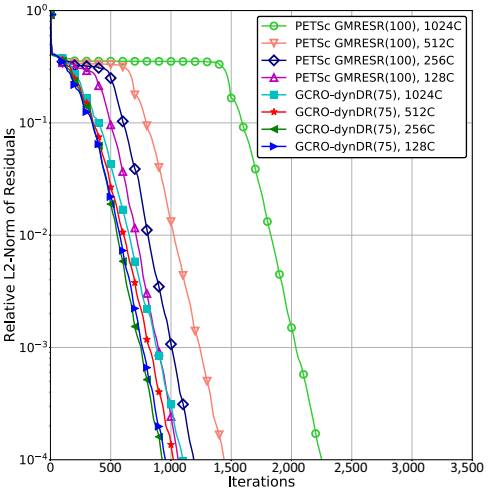

Figure 10.
Convergence curves of solving an adjoint system of equations for the 3 D CRM wing with a general restarted GMRES and the developed GCRO-*dyn*DR solver in terms of iteration cycles with varied number of cores

		Solving adjoint equation		
		Iterations	Time (sec)	Memory (Gb/Core)
PETSc Full GMRES	ILU0	1530	6151	5.72
	ILU1	1340	5919	6.11
PETSc GMRESR(75)	ILU0	*	*	1.58
	ILU1	2109	5192	1.94
PETSc GMRESR(100)	ILU0	2658	3518	1.69
	ILU1	1925	3630	1.97
GMRES- *dyn*DR(75)	ILU0	2320	2107	1.65
GMRES- *dyn*DR(25)	ILU0	2527	2612	1.58
GCRO- *dyn*DR(75)	ILU0	2322	2445	1.67
GCRO- *dyn*DR(25)	ILU0	2516	2687	1.60

Table I.
Summary of the performance of solving the adjoint equation for 3 D CRM wing

Notes: The symbol * denotes stalled convergence. All the reported results based on convergence level reaching 1.0×10^{-10}. The monitored memory usage includes the first- and second-order Jacobians, the preconditioners and the adjoint solver

Jacobians, the global preconditioner with domain decomposition, the local ILU preconditioners, as well as the linear solver with deflated and Krylov vectors. At equivalent Krylov vectors, ILU with a larger fill-in always converges with a smaller number of iterations but not necessarily in terms of CPU time. As expected the figure illustrates a rightward shift of the results with ILU(1) and clearly demonstrates the advantage for both the memory consumption and computational cost with both the GCRO-*dyn*DR and GMRES-*dyn*DR solver.

6. Conclusions

The paper proposes an extension of dynamic deflated restarting into the traditional GCRO method to improve convergence performance with a significant reduction in the memory usage. The novel deflation strategy involves selecting the number of deflated vectors per restart cycle based on the number of negative harmonic Ritz eigenpairs and defaulting to standard restarted GMRES within the inner loop if none, and restricts the deflated vectors to the smallest eigenvalues present in the modified Hessenberg matrix. The results demonstrate the effectiveness of GCRO-*dyn*DR(*m*) at converging at a lower storage cost when compared to Full PETSc GMRES. Both, GCRO-DR and GMRES-DR converge at identical rates in terms of the number of iterations; however, the GMRES variant is slightly more computationally attractive. Furthermore the test cases have demonstrated that deflating the critical number of vectors proves beneficial; where too few of a number may be insufficient to restore convergence and too large of a fixed number of vectors led to an increase in the computational cost. In addition, the solver has shown to be scalable and less sensitive to a weaker global preconditioner.

We note that several areas related to this work remain to be explored. First, alternate approaches to dynamically identify degenerate Ritz values are warranted; such as Ritz values away from clusters and the presence of a larger number of values with a small magnitude. Second, the accuracy of the extracted Ritz values for the GCRO approach might need to be investigated. Lastly, we plan to further apply the developed solver for more challenging cases, in terms of the size of the computational grid as well as the complexity of the flow field.

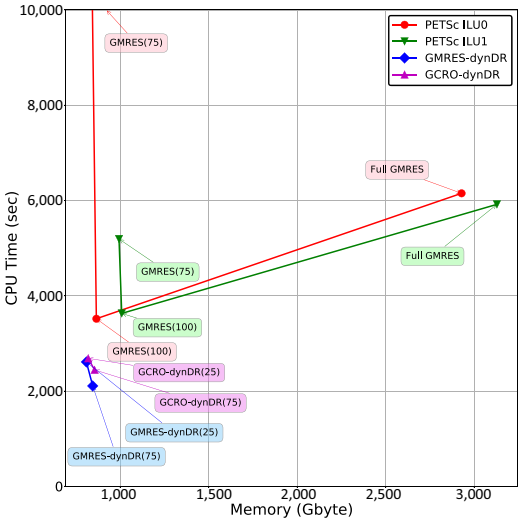

Figure 11.
CPU time versus memory used in solving the adjoint equation for the 3 D CRM case with PETSc GMRES solver, GMRES-*dyn*DR and GCRO-*dyn*DR solvers from Table I

References

Amritkar, A., de Sturler, E., Świrydowicz, K., Tafti, D. and Ahuja, K. (2015), "Recycling Krylov subspaces for CFD applications and a new hybrid recycling solver", *Journal of Computational Physics*, Vol. 303, pp. 222-237, doi: 10.1016/j.jcp.2015.09.040.

Brezillon, J., Dwight, R.P. and Widhalm, M. (2009), "Aerodynamic optimization for cruise and high-lift configurations", *Megadesign and Megaopt – German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, Vol. 107, pp. 249-262, doi: 10.1007/978-3-642-04093-1_18.

Brezillon, J. and Gauger, N. (2004), "2D and 3D aerodynamic shape optimisation using the adjoint approach", *Aerospace Science and Technology*, Vol. 8 No. 8, pp. 715-727, doi: 10.1016/j.ast.2004.07.006.

Cambier, L. and Kroll, N. (2008), "MIRACLE – a joint DLR/ONERA effort on harmonization and development of industrial and research aerodynamic computational environment", *Aerospace Science and Technology*, Vol. 12 No. 7, pp. 555-566, doi: 10.1016/j.ast.2008.01.007.

Chen, C.-H., Nadarajah, S. and Castonguay, P. (2019), "A dynamically deflated GMRES adjoint solver for aerodynamic shape optimization", *Computers and Fluids*, Vol. 179, pp. 490-507, doi: 10.1016/j.compfluid.2018.11.016.

de Sturler, E. (1996), "Nested Krylov methods based on GCR", *Journal of Computational and Applied Mathematics*, Vol. 67 No. 1, pp. 15-41, doi: 10.1016/0377-0427(94)00123-5.

de Sturler, E. (1999), "Truncation strategies for optimal Krylov subspace methods", *Siam Journal on Numerical Analysis*, Vol. 36 No. 3, pp. 864-889, doi: 10.1137/S0036142997315950.

Dwight, R. (2006), "Efficiency improvements of RANS-based analysis and optimization using implicit and adjoint methods on unstructured grids", Ph.D. thesis, University of Manchester

Erhel, J., Burrage, K. and Pohl, B. (1996), "Restarted GMRES preconditioned by deflation", *Journal of Computational and Applied Mathematics*, Vol. 69 No. 2, pp. 303-318, doi: 10.1016/0377-0427(95)00047-X.

Giraud, L., Gratton, S., Pinel, X. and Vasseur, X. (2010), "Flexible GMRES with deflated restarting", *SIAM Journal on Scientific Computing*, Vol. 32 No. 4, pp. 1858-1878, doi: 10.1137/080741847.

He, P., Mader, C.A., Martins, J.R. and Maki, K.J. (2018), "An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM", *Computers and Fluids*, Vol. 168, pp. 285-303, doi: 10.1016/j.compfluid.2018.04.012.

Heuveline, V. and Strauß, F. (2009), "Shape optimization towards stability in constrained hydrodynamic systems", *Journal of Computational Physics*, Vol. 228 No. 4, pp. 938-951, doi: 10.1016/j.jcp.2008.06.030.

Hicken, J.E. and Zingg, D.W. (2010), "A simplified and flexible variant of GCROT for solving nonsymmetric linear systems", *SIAM Journal on Scientific Computing*, Vol. 32 No. 3, pp. 1672-1694, doi: 10.1137/090754674.

Hicken, J.E. and Zingg, D.W. (2010), "Induced-drag minimization of nonplanar geometries based on the euler equations", *AIAA Journal*, Vol. 48 No. 11, pp. 2564-2575, doi: 10.2514/1.j050379.

Jameson, A. (1994), R. I. for Advanced Computer Science (US), Optimum aerodynamic design via boundary control.

Jameson, A. and Reuther, J. (1994), R. I. for Advanced Computer Science (US), Control theory based airfoil design using the Euler equations.

Kim, S., Alonso, J.J. and Jameson, A. (2004), "Multi-element high-lift configuration design optimization using viscous continuous adjoint method", *Journal of Aircraft*, Vol. 41 No. 5, pp. 1082-1097, doi: 10.2514/1.17.

Kroll, N., Gauger, N.R., Brezillon, J., Dwight, R., Fazzolari, A., Vollmer, D., Becker, K., Barnewitz, H., Schulz, V. and Hazra, S.U. (2007), "Flow simulation and shape optimization for aircraft design", *Journal of Computational and Applied Mathematics*, Vol. 203 No. 2, pp. 397-411, doi: 10.1016/j.cam.2006.04.012.

Lions, J.L. (1971), *Optimal Control of Systems Governed by Partial Differential Equations, Die Grundl Ehren Der Mathematischen Wissenschaften in Einzeldarstellungen*, Springer-Verlag, Berlin, New York, NY,

Lyu, Z., Kenway, G.K. and Martins, J.R. (2014), "Aerodynamic shape optimization investigations of the common research model wing benchmark", *AIAA Journal*, Vol. 53 No. 4, pp. 968-985, doi: 10.2514/1.J053318.

Morgan, R.B. (1995), "A restarted GMRES method augmented with eigenvectors", *Siam Journal on Matrix Analysis and Applications*, Vol. 16 No. 4, pp. 1154-1171, doi: 10.1137/S0895479893253975.

Morgan, R.B. (2000), "Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations", *SIAM Journal on Matrix Analysis and Applications*, Vol. 21 No. 4, pp. 1112-1135, doi: 10.1137/S0895479897321362.

Morgan, R.B. (2002), "GMRES with deflated restarting", *Siam Journal on Scientific Computing*, Vol. 24 No. 1, pp. 20-37, doi: 10.1137/S1064827599364659.

Morgan, R.B. (2005), "Restarted block-GMRES with deflation of eigenvalues", *Applied Numerical Mathematics*, Vol. 54 No. 2, pp. 222-236, doi: 10.1016/j.apnum.2004.09.028.

Morgan, R.B. and Zeng, M. (2006), "A harmonic restarted Arnoldi algorithm for calculating eigenvalues and determining multiplicity", *Linear Algebra and Its Applications*, Vol. 415 No. 1, pp. 96-113, doi: 10.1016/j.laa.2005.07.024. Special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems.

Nadarajah, S. and Jameson, A. (2007), "Optimum shape design for unsteady three-dimensional viscous flows using a nonlinear frequency-domain method", *Journal of Aircraft*, Vol. 44 No. 5, pp. 1513-1527, doi: 10.2514/1.27601.

Nadarajah, S.K., McMullen, M.S. and Jameson, A. (2006), "Aerodynamic shape optimization for unsteady three-dimensional flows", *International Journal of Computational Fluid Dynamics*, Vol. 20 No. 8, pp. 533-548, doi: 10.1080/10618560601088343.

Paige, C.C., Parlett, B.N. and Van der Vorst, H.A. (1995), "Approximate solutions and eigenvalue bounds from Krylov subspaces", *Numerical Linear Algebra with Applications*, Vol. 2 No. 2, pp. 115-133.

Papadimitriou, D.I. and Giannakoglou, K.C. (2007), "A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows", *Computers and Fluids*, Vol. 36 No. 2, pp. 325-341, doi: 10.1016/j.compfluid.2005.11.006.

Parks, M.L., de Sturler, E., Mackey, G., Johnson, D.D. and Maiti, S. (2006), "Recycling Krylov subspaces for sequences of linear systems", *SIAM Journal on Scientific Computing*, Vol. 28 No. 5, pp. 1651-1674, doi: 10.1137/040607277.

Pashos, G., Kavousanakis, M.E., Spyropoulos, A.N., Palyvos, J.A. and Boudouvis, A.G. (2009), "Simultaneous solution of large-scale linear systems and eigenvalue problems with a parallel GMRES method", *Journal of Computational and Applied Mathematics*, Vol. 227 No. 1, pp. 196-205, doi: 10.1016/j.cam.2008.07.012..

Pinel, X. and Montagnac, M. (2013), "Block krylov methods to solve adjoint problems in aerodynamic design optimization", *AIAA Journal*, Vol. 51 No. 9, pp. 2183-2191, doi: 10.2514/1.j052113.

Saad, Y. (2003), *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia.

Spalart, P. and Allmaras, S. (1992), "A one-equation turbulence model for aerodynamic flows", in *30th aerospace sciences meeting and exhibit*, p. 439.

Tatossian, C.A., Nadarajah, S.K. and Castonguay, P. (2011), "Aerodynamic shape optimization of hovering rotor blades using a non-linear frequency domain approach", *Computers and Fluids*, Vol. 51 No. 1, pp. 1-15, doi: 10.1016/j.compfluid.2011.06.014.

Van der Vorst, H.A. and Vuik, C. (1993), "The superlinear convergence behavior of GMRES", *Journal of Computational and Applied Mathematics*, Vol. 48 No. 3, pp. 327-341, doi: 10.1016/0377-0427(93)90028-A.

Walther, B. and Nadarajah, S. (2012), "Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor", *Journal of Turbomachinery*, Vol. 135 No. 2, p. 021017, doi: 10.1115/1.4007502.

Xu, S. and Timme, S. (2017), "Robust and efficient adjoint solver for complex flow conditions", *Computers and Fluids*, Vol. 148, pp. 26-38, doi: 10.1016/j.compfluid.2017.02.012. iD: 271457.

Yamazaki, I., Tomov, S. and Dongarra, J. (2014), "Deflation strategies to improve the convergence of communication-avoiding GMRES", in *Proceedings of the 5th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems, ScalA '14, IEEE Press, Piscataway, NJ, USA*, pp. 39-46, doi:10.1109/ScalA.2014.6.

Yu, Y., Lyu, Z., Xu, Z. and Martins, J.R. (2018), "On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization", *Aerospace Science and Technology*, Vol. 75, pp. 183-199, doi: 10.1016/j.ast.2018.01.016.

Appendix 1 GMRES with dynamic deflated restarting

Algorithm 1: GMRES-*dyn*DR

Initialization: $\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{v}_1 \leftarrow \mathbf{r}_0 / \|\mathbf{r}_0\|$

Perform m steps of the Arnoldi process: $\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$

$\mathbf{c} \leftarrow \mathbf{V}_{m+1}^T \mathbf{r}_0$

$\mathbf{y}^* = argmin \|\mathbf{c} - \overline{\mathbf{H}}_m \mathbf{y}\|$

$\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{M}^{-1}\mathbf{V}_m \mathbf{y}^*$

$\mathbf{r}_m \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_m$

while $\|\mathbf{r}_m\| / \|\mathbf{b}\| > tol$ **do**

  Evaluate the eigenvectors of $\left[\mathbf{H}_m + h_{m+1,m}^2 \mathbf{H}_m^{-T} \mathbf{e}_m \mathbf{e}_m^T\right] \mathbf{g}_i = \lambda_i \mathbf{g}_i$

  $k^* \leftarrow$ the number of the negative eigenvalues

  if $k^* \geq 1$ **then**

    Select $\mathbf{g}_i$, where $0 \leq i \leq k^*$, associated with k^* smallest eigenvalues by magnitude, and $\mathbf{G}_{k^*} = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k^*-1}]$

    QR factorization of $\left[\begin{bmatrix} \mathbf{G}_{k^*} \\ \mathbf{0}_{1\times k^*} \end{bmatrix}, \mathbf{c} - \overline{\mathbf{H}}_m \mathbf{y}^*\right] = \mathbf{P}_{k^*+1} \mathbf{\Gamma}_{k^*+1}$

    New Krylov subspace basis vectors: $\mathbf{V}_{k^*}^{new} \leftarrow \mathbf{V}_{m+1}\mathbf{P}_{k^*}$

    $\overline{\mathbf{H}}_{k^*}^{new} \leftarrow \mathbf{P}_{k^*+1}^T \overline{\mathbf{H}}_m \mathbf{P}_{k^*}$

    Perform $m - k^*$ steps of the Arnoldi process with $\mathbf{V}_{k^*}^{new}$ and $\overline{\mathbf{H}}_{k^*}^{new}$, $\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$

  else

    Perform m steps of the Arnoldi process: $\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$

  $\mathbf{c} \leftarrow \mathbf{V}_{m+1}^T \mathbf{r}_m$

  $\mathbf{y}^* = argmin \|\mathbf{c} - \overline{\mathbf{H}}_m \mathbf{y}\|$

  $\mathbf{x}_m \leftarrow \mathbf{x}_m + \mathbf{M}^{-1}\mathbf{V}_m \mathbf{y}^*$

  $\mathbf{r}_m \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_m$

Appendix 2. GCRO with dynamic deflated restarting

Algorithm 2: GCRO-*dyn*DR

$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{v}_1 \leftarrow \mathbf{r}_0 / \|\mathbf{r}_0\|$, $\mathbf{c} = \|\mathbf{r}_0\|\mathbf{e}_1$

Perform m steps of the Arnoldi process: $\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$

$\mathbf{x}_m \leftarrow \mathbf{x}_0 + \mathbf{M}^{-1}\mathbf{V}_m\mathbf{y}^*$ where $\mathbf{y}^* = argmin\|\mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}\|$

$\mathbf{r}_m \leftarrow \mathbf{V}_{m+1}\left(\mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}\right)$, $\mathbf{r}_i \leftarrow \mathbf{r}_m$

Evaluate the eigenvectors of $\left[\mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-T}\mathbf{e}_m\mathbf{e}_m^T\right]\mathbf{g}_j = \lambda_j\mathbf{g}_j$, where $0 \le j \le m$

$k^* \leftarrow$ the number of the negative eigenvalues

QR factorization of $\mathbf{G}_{k^*} = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k^*-1}] = \mathbf{P}_{k^*}\mathbf{\Gamma}_{k^*}$ to obtain $\mathbf{P}_{k^*}$

$\mathbf{Y}_{k^*} = \mathbf{V}_m\mathbf{P}_{k^*}$, $\overline{\mathbf{H}}_m\mathbf{P}_{k^*} = \mathbf{Q}\mathbf{R}$

$\mathbf{C}_{k^*} \leftarrow \mathbf{V}_{m+1}\mathbf{Q}$, $\mathbf{U}_{k^*} \leftarrow \mathbf{Y}_{k^*}\mathbf{R}^{-1}$, $\mathbf{C}_{k^*} = \mathbf{A}\mathbf{U}_{k^*}$

while $\|\mathbf{r}_i\| > tol$ **do**

- $i \leftarrow i + 1$
- **if** $k^* \ge 1$ **then**
  - Algorithm 3: Inner GMRES
- **else**
  - Perform m steps of the Arnoldi process: $\mathbf{A}\mathbf{M}^{-1}\mathbf{V}_m = \mathbf{V}_{m+1}\overline{\mathbf{H}}_m$
  - $\mathbf{c} \leftarrow \mathbf{V}_{m+1}^T\mathbf{r}_{i-1}$
  - $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{M}^{-1}\mathbf{V}_m\mathbf{y}^*$ where $\mathbf{y}^* = argmin\|\mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}\|$
  - $\mathbf{r}_i \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_i$
  - Evaluate the eigenvectors of $\left[\mathbf{H}_m + h_{m+1,m}^2\mathbf{H}_m^{-H}\mathbf{e}_m\mathbf{e}_m^T\right]\mathbf{g}_j = \lambda_j\mathbf{g}_j$, where $0 \le j \le m$
  - $k^* \leftarrow$ the number of the negative eigenvalues, and $\mathbf{G}_{k^*} = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k^*-1}]$
  - QR factorizatioon of $\left[\begin{bmatrix}\mathbf{G}_{k^*}\\ \mathbf{0}_{1\times k^*}\end{bmatrix}, \mathbf{c} - \overline{\mathbf{H}}_m\mathbf{y}^*\right]$ to obtain $\mathbf{P}_{k^*}$
  - $\mathbf{Y}_{k^*} = \mathbf{V}_m\mathbf{P}_{k^*}$, $\overline{\mathbf{H}}_m\mathbf{P}_{k^*} = \mathbf{Q}\mathbf{R}$
  - $\mathbf{C}_{k^*} \leftarrow \mathbf{V}_{m+1}\mathbf{Q}$, $\mathbf{U}_{k^*} \leftarrow \mathbf{Y}_{k^*}\mathbf{R}_{39}^{-1}$, $\mathbf{C}_{k^*} = \mathbf{A}\mathbf{U}_{k^*}$

Algorithm 3: Inner GMRES for GCRO-*dyn*DR

Perform $m - k^*$ steps of the Arnoldi process based on $\mathbf{C}_{k^*}$ and $\mathbf{U}_{k^*}$ from the previous resatrt cycle:

$$\mathbf{A}\mathbf{V}_{m-k^*} = \mathbf{C}_{k^*}\mathbf{C}_{k^*}^T\mathbf{A}\mathbf{V}_{m-k^*} + \mathbf{V}_{m-k^*+1}\overline{\mathbf{H}}_{m-k^*}$$

$\tilde{\mathbf{U}}_{k^*} = \mathbf{U}_{k^*}\mathbf{D}_{k^*}$

$\widehat{\mathbf{V}}_m = \left[\tilde{\mathbf{U}}_{k^*} \quad \mathbf{V}_{m-k^*}\right]$

$\widehat{\mathbf{W}}_{m+1} = \left[\mathbf{C}_{k^*} \quad \mathbf{V}_{m-k^*+1}\right]$

$\mathbf{A}\widehat{\mathbf{V}}_m = \widehat{\mathbf{W}}_{m+1}\overline{\mathbf{G}}_m$, where $\overline{\mathbf{G}}_m = \begin{bmatrix} \mathbf{D}_{k^*} & \mathbf{C}_{k^*}^T\mathbf{A}\mathbf{V}_{m-k^*} \\ 0 & \overline{\mathbf{H}}_{m-k^*} \end{bmatrix}$

$\mathbf{x}_i = \mathbf{x}_{i-1} + \widehat{\mathbf{V}}_m\mathbf{y}^*$ where $\mathbf{y}^* = argmin\|\widehat{\mathbf{W}}_{m+1}^T\mathbf{r}_{i-1} - \overline{\mathbf{G}}_m\mathbf{y}\|$

$\mathbf{r}_i = \mathbf{r}_{i-1} - \widehat{\mathbf{W}}_{m+1}\overline{\mathbf{G}}_m\mathbf{y}$

Evaluate the eigenvectors of $\overline{\mathbf{G}}_m^T\overline{\mathbf{G}}_m\mathbf{g}_j - \lambda_j\left(\overline{\mathbf{G}}_m^T\widehat{\mathbf{W}}_{m+1}^T\widehat{\mathbf{V}}_m\right)\mathbf{g}_j = 0$, where $0 \le j \le m$

$k^* \leftarrow$ the number of the negative eigenvalues

QR factorization of $\mathbf{G}_{k^*} = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k^*-1}] = \mathbf{P}_{k^*}\mathbf{\Gamma}_{k^*}$ to obtain $\mathbf{P}_{k^*}$

$\mathbf{Y}_{k^*} = \widehat{\mathbf{V}}_m\mathbf{P}_{k^*}$, $\overline{\mathbf{G}}_m\mathbf{P}_{k^*} = \mathbf{Q}\mathbf{R}$

$\mathbf{C}_{k^*} \leftarrow \widehat{\mathbf{W}}_{m+1}\mathbf{Q}$, $\mathbf{U}_{k^*} = \mathbf{Y}_{k^*}\mathbf{R}^{-1}$

Corresponding author
Chih-Hao Chen can be contacted at: chih-hao.chen2@mail.mcgill.ca

For instructions on how to order reprints of this article, please visit our website:
www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: **permissions@emeraldinsight.com**