## Accepted Manuscript

Enabling off-design linearised aerodynamics analysis using Krylov subspace recycling technique

## 



To appear in: Computers and Fluids
Received date: 14 March 2016
Revised date: 5 July 2016
Accepted date: 19 October 2016

Please cite this article as: Shenren Xu, Sebastian Timme, Kenneth J. Badcock, Enabling off-design linearised aerodynamics analysis using Krylov subspace recycling technique, Computers and Fluids (2016), doi: 10.1016/j.compfluid.2016.10.018

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

## Highlights

- A novel algorithm for solving large sparse linear systems of equations is proposed
- It combines an ILU preconditioner based on blended Jacóbian and GCRO-DR solver
- The solver implemented in DLR-TAU is applied to linearised aerodynamics analysis
- Superior performance regarding memory and runtime is shown on challenging cases
- Our work enabled rapid aeroelastic analysis using CFD at off-design conditions


# Enabling off-design linearised aerodynamics analysis using Krylov subspace recycling technique 

Shenren Xu*, Sebastian Timme, Kenneth J. Badcock<br>School of Engineering, University of Liverpool, Liverpool L69 3GH, United Kingdom


#### Abstract

The major computational challenge, when using frequency domain linearised computational fluid dynamics in the analysis of aeroelastic problems such as aircraft flutter, gust response or shock buffet, are the excessive memory and CPU time requirements to solve the large sparse linear systems of equations. To address these issues found with the generalised minimal residual linear equation solver, the generatised conjugate residual solver with deflated restarting is adopted here in which an invariant Krylov subspace is recycled both between restarts when solying a single linear frequency domain problem and for a sequence of equations when varying the system matrix and forcing terms. The proposed method is implemented in an industrial code and applied to three test cases including the forced excitation and buffet onset of a pitch-plunge aerofoil, a realistic passenger aircraft in inviscid transonic flow and a generic half wing-body model at a pre-buffet condition. The memory requirements for the problems investigated are reduced by up to an order of magnitude, while the CPU times are reduced by up to a factor of three.


[^0]Keywords: Linearized Computational Fluid Dynamics, Shock Buffet, Flutter, GCRO-DR, Krylov Subspace Recycling

## 1. Introduction

${ }_{20}$ projection basis [3, 4].
More specifically, to predict the flutter onset using CFD aerodynamics pre-buffet conditions exhibits resonance when excited near the frequency of the flow instability. This information can then serve as a good initial guess. Another computational challenge associated with tracking the destabilising

69 lar, thus causing tremendous convergence difficulty [10]. In this work, we form the ILU preconditioner from a blended Jacobian matrix (even though

$$
\dot{\mathbf{w}}_{f}=\mathbf{R}_{f}\left(\mathbf{w}_{f}, \mathbf{x}, \dot{\mathbf{x}}, \boldsymbol{\theta}\right)
$$

where $\mathbf{w}_{f}$ and $\mathbf{R}_{f}$ denote the fluid unknowns and corresponding residual vector, respectively, and $\boldsymbol{\theta}$ are the system parameters. The vectors $\mathbf{x}$ and $\dot{\mathbf{x}}$
are location and velocity of the fluid mesh points which are functions of the structural mode shapes and the modal amplitudes denoted $\boldsymbol{\eta}$. Linearising about an equilibrium point and assuming small amplitude harmonic motion, the latter equation can be re-formulated as

$$
\begin{equation*}
\left.\left(\frac{\partial \mathbf{R}_{f}}{\partial \mathbf{w}_{f}}-i \omega^{(k)} I\right) \boldsymbol{\phi}_{f}^{(j, k)}=-\left(\frac{\partial \mathbf{R}_{f}}{\partial \boldsymbol{\eta}}+i \omega^{(k)} \frac{\partial \mathbf{R}_{f}}{\partial \dot{\boldsymbol{\eta}}}\right) \boldsymbol{\phi}_{\eta}^{(j)}\right) \tag{1}
\end{equation*}
$$

where $\phi_{f}$ and $\phi_{\eta}$ are complex-valued amplitudes of fluid and structure, respectively. The equation gives the aerodynamic response $\phi_{f}$ following a disturbance of the structure $\phi_{\eta}$. To find aerodynamic data for further analysis, this equation usually has to be pre-computed for each structural mode shape (denoted by superscript $j$ ) and for a range of forced sinusoidal excitations in the modal amplitudes at different frequencies $\omega$ (denoted by superscript $k$ ). As mentioned earlier, several hundreds of this equation need to be solved for industrial problems with the number of fluid unknowns easily exceeding several tens of millions.

Adding equations to describe the unsteady motion of the structure in terms of the modal amplitudes $\boldsymbol{\eta}$ gives a coupled problem to be solved for investigation in flutter stability and also gust response behaviour. The basis of a reduced order model for such aeroelastic analyses can be calculated from the eigenvectors of the coupled system. The fluid part of the direct (i.e. right) eigenvalue problem is equivalent to eq. (1), except that the structural part $\phi_{y}$ of the eigenvector is now part of the solution rather than a pre-defined user input and that the eigenvalue (i.e. frequency) corresponds to a particular eigenvector (with superscript $k=j$ ). The corresponding adjoint (i.e. left)
eigenvalue problem for the fluid part $\boldsymbol{\psi}_{f}$ of the eigenvector can be derived as

$$
\begin{equation*}
\left(\left(\frac{\partial \mathbf{R}_{f}}{\partial \mathbf{w}_{f}}\right)^{T}+i \omega^{(j)} I\right) \boldsymbol{\psi}_{f}^{(j)}=-\left(\frac{\partial \mathbf{R}_{\dot{\eta}}}{\partial \mathbf{w}_{f}}\right)^{T} \boldsymbol{\psi}_{\dot{\eta}}^{(j)} \tag{2}
\end{equation*}
$$

where $\mathbf{R}_{\dot{\eta}}$ is the residual vector corresponding to the structural anknowns. Details of the mathematical formulation of linearised frequency domain aerodynamics and model reduction in the context of aeroelastic analysis can be found in [4].

Another type of problem arises when the fluid exhibits an instability without structural motion. One typical example is the shock-buffet problem where the shock wave interacts with the boundary layer and destabilises the steady flow beyond a critical parameter. To find the buffet onset, shifted inverse methods are an obvious choice to calculate few eigenvalues close to an initial guess. To choose such initial shift (i.e. a characteristic frequency of the instability), either engineering judgement is required or the resonant behaviour of the flow when excited at frequencies in the vicinity of the instability can be exploited $[8,9]$, which would be equivalent to solving eq. (1). The closer the shift to the target eigenvalue, the faster the algorithm converges. This howeyer leads to the second, even bigger challenge using shifted inverse methods. The linear system to be solved is nearly singular. Using a direct sparse linear equation solver quickly becomes infeasible for everything beyond two-dimensional problems. Thus, a preconditioned sparse iterative linear equation solver is a possible alternative.

We use the shifted inverse method from [16] referred to as inverse correc-
tion. The equation to be solved is

$$
\begin{equation*}
\left(\frac{\partial \mathbf{R}_{f}}{\partial \mathbf{w}_{f}}-\sigma I\right) \delta \boldsymbol{\phi}_{f}=-\left(\frac{\partial \mathbf{R}_{f}}{\partial \mathbf{w}_{f}}-\lambda I\right) \boldsymbol{\phi}_{f} \tag{3}
\end{equation*}
$$

where $\sigma$ is the constant complex-valued shift close to the target eigenvalue $\lambda$ and the eigenvector is updated as

$$
\phi_{f} \leftarrow \phi_{f}+\delta \phi_{f}
$$

until the norm of the right-hand side converges below a given tolerance. The right-hand side in eq. (3) represents the residual vector of the eigenvalue problem based on the current approximation to the eigenvector $\phi_{f}$ of unit length and eigenvalue based on the Rayleigh quotient

The eigenvector update $\delta \phi_{f}$ is always initialised to zero, which is convenient since for converging outer iterations the update will go to zero. The inner convergence is defined relative to the convergence of the outer iteration giving a nearly constant number of inner iterations. In this work, four orders of magnitude is chosen as stopping criterion of the inner linear system.
As can be seen from this brief introduction of linearised aerodynamics, the efficient solution of large sparse linear systems of equations is at the heart of it. For convenience in the following discussion, the coefficient matrix (i.e. fluid Jacobian matrix plus a complex-valued shift) is denoted $A$, while the various right-hand side terms are called $\mathbf{b}$.

## 3. Solving large sparse linear systems of equations

The main challenge in using linearised CFD aerodynamics is solving large sparse linear systems of equations. Restarted generalised minimal residual (GMRES) solver [17] was implemented in the DLR-TAU code to solve the linearised equations. As mentioned in the introduction, for stiff problems, restarted GMRES often suffers from stagnation unless a large number of Krylov vectors is kept. It is not uncommon to keep several hundred Krylov vectors in order to converge. This large memory requirement could then become the bottleneck when solving large cases. To ease these difficulties, we implemented generalised conjugate residual solver with deflated restarting (GCRO-DR), which converges almost like full GMRES (GMRES that does not restart and keeps expanding the Krylov subspace until the convergence criterion is met) but has small memory requirement. In addition, recycling a certain Krylov subspace between different equations is possible for GCRODR and is thus favourable for solving a sequence of linear equations with similar coefficient matrices.

In this section, both GMRES and GCRO-DR will be explained. Furthermore, GCRO-DR combined with inter-equation recycling, dubbed GCRO$D R-R$, is introduced.

### 3.1. GMRES

The baseline linear solver used is GMRES. The theory and implementation details are well documented in [17] and only a brief introduction is given here. When solving the linear system

$$
A \mathbf{x}=\mathbf{b}
$$

one first forms the Krylov subspace

$$
\mathcal{K}_{m}(A, \mathbf{b})=\operatorname{span}\left(\mathbf{b}, A \mathbf{b}, A^{2} \mathbf{b}, \ldots, A^{m-1} \mathbf{b}\right)
$$

using the Arnoldi iteration. After $m$ Arnoldi steps, a unit vector basis $V_{m}=$ $\left[\mathbf{v}_{1}, \mathbf{v}_{2}, \ldots, \mathbf{v}_{m}\right]$ that spans the Krylov subspace is constructed satisfying the Arnoldi relation

$$
A V_{m}=V_{m+1} \bar{H}_{m}
$$

where $\bar{H}_{m}$ is an upper Hessenberg matrix. The solution $\mathbf{X}$ is approximated as

$$
\mathbf{x}=\mathbf{x}_{0}+V_{m} \mathbf{d}_{m}
$$

where the coefficient vector $\mathbf{d}_{m}$ is solved through the least square problem minimising the resulting residual $\mathbf{r}=\mathbf{b}-A \mathbf{x}$. It can be shown that

$$
\begin{aligned}
& \|\mathbf{r}\|=\left\|\mathbf{r}_{0}-A V_{m} \mathbf{d}_{m}\right\|=\left\|\beta \mathbf{v}_{1}-V_{m+1} \bar{H}_{m} \mathbf{d}_{m}\right\| \\
& =\left\|V_{m+1}\right\|\left\|\beta \mathbf{e}_{1}-\bar{H}_{m} \mathbf{d}_{m}\right\|=\left\|\beta \mathbf{e}_{1}-\bar{H}_{m} \mathbf{d}_{m}\right\|
\end{aligned}
$$

where $\mathbf{e}_{1}$ is the first standard basis vector of $\mathbb{R}^{m+1}$ and $\mathbf{r}_{0}=\mathbf{b}-A \mathbf{x}_{0}$ is the initial residual. The least square problem is then reduced to a very low dimension $m$. The restarted version simply forms the Krylov subspace and solves the least square problem again from the updated initial solution and residual vectors.

In the current work, GMRES is preconditioned using ILU factorisation. Once the matrices $L$ and $U$ are computed, the only modification to GMRES without preconditioning is whenever a vector is multiplied by the coefficient
matrix, the resulting vector is further left-multiplied first by $L^{-1}$ and then by $U^{-1}$. The triangular matrices $L$ and $U$ are inverted using forward and backward substitutions. Alternatively, right or split preconditioning could also easily be achieved by plugging in these two additional matrix inversions in a slightly different manner [17]. There seems to be no obvious advantage for any type of preconditioning over the others, therefore we only used left preconditioning throughout this work as the least modification is required. In addition, due to memory considerations, a complex-valued version of ILU with low fill-in is used.

## 3.2. $G C R$

Generalised conjugate residual (GCR) [18] is a Krylov subspace method algorithmically equivalent to GMRES with different procedures. For the theory and the detailed implementation, refer to [17, 19]. Only a brief introduction is given here.

Standard GCR constructs two vector bases

$$
U_{m}=\left[\mathbf{u}_{1}, \mathbf{u}_{2}, \ldots, \mathbf{u}_{m}\right] \text { and } C_{m}=\left[\mathbf{c}_{1}, \mathbf{c}_{2}, \ldots, \mathbf{c}_{m}\right]
$$

satisfying

$$
\begin{equation*}
C_{m}=A U_{m} \text { and } C_{m}^{H} C_{m}=I . \tag{4}
\end{equation*}
$$

The solution is approximated on the subspace spanned by the column vectors of $U_{m}$

$$
\mathbf{x}=\mathbf{x}_{0}+U_{m} \mathbf{d}_{m}
$$

subject to the constraint that the resulting residual is perpendicular to the
subspace spanned by the column vectors of $C_{m}$

$$
C_{m}^{H} \mathbf{r}=C_{m}^{H}\left(\mathbf{r}_{0}-A U_{m} \mathbf{d}_{m}\right)=\mathbf{0}
$$

solvers is GCRO [20], which uses GCR for the outer loop and any Krylov subspace solver, such as GMRES, for the inner loop. The motivation behind
can best be explained by revisiting eq. (5). Instead of assigning the latest residual vector to $\mathbf{u}_{i}$, ideally we could set

$$
\mathbf{u}_{i} \leftarrow A^{-1} \mathbf{r}_{i-1}
$$

(7)

Then the solution is found immediately following the update step in eq. (6) since the resulting residual vector is zero. Although it is generally not possible to invert the matrix $A$, it does imply that we could assign to $\mathbf{u}_{i}$ an approximate solution to the equation $A \mathbf{u}_{i}=\mathbf{r}_{i-1}$ to accelerate the convergence. To get the approximate solution, GMRES is used to approximately solve for $\mathbf{u}_{i}$ in

$$
\left(I-C_{i-1} C_{i-1}^{H}\right) A \mathbf{u}_{i}=\mathbf{y}_{i-1}
$$

for $k$ iterations with initial solution of zero. The term in the bracket preceding $A$ ensures that the Krylov subspace formed in the inner loop is normal to the Krylov subspace in the outer GCR loop so that monotonic residual reduction is guaranteed.

### 3.4. GCRO-DR and GCRO-DR-R

For nested Krylov subspace solvers such as GCRO, the inner loop generates a Krylor subspace to approximately solve the equation, and the subspace is discarded when exiting the inner loop. It would be advantageous to recycle some information from the discarded subspace to aid the outer convergence. One approach is to truncate based on the principle angle between the Krylov vectors constructed during the inner loop and select the important ones to augment the outer Krylov subspace. The criterion to select them is to check how much worse the inner loop convergence would have been if the inner
loop has stopped before those vectors are formed. One solver based on this idea is generalised conjugate residual with optimal truncation [21]. Another approach is to select the interior eigenvectors (eigenvectors corresponding to the smallest-in-magnitude eigenvalues) that can be computed during the Arnoldi iteration from the inner loop and use them to augment the outer Krylov subspace. The two most important solvers following this idea are generalised minimal residual with deflated restarting [22] and GCRO-DR [19]. Both are nested Krylov subspace solvers with the main difference being that the former uses GMRES for the outer loop while the latter uses GCR. Although the latter requires more memory, it has the advantage of recycling eigenvectors both between restarts and between equations with different coefficient matrices and/or forcing terms. Due to this flexibility, GCRO-DR is used in this work.

The algorithm of GCRO-DR is now explained. It begins with a start-up GMRES cycle with $m$ Arnoldi iterations which produces the upper Hessenberg matrix $\bar{H}_{m}$ and the Krylov vectors $V_{m}$. The solution $\mathbf{x}$ and residual vectors $\mathbf{r}$ are first updated accordingly. Compared with restarted GMRES, an additional step to extract the approximate interior eigenvectors $\left\{\mathbf{y}_{1}, \mathbf{y}_{2}, \ldots, \mathbf{y}_{k}\right\}$ of the matrix $A$ is taken by first solving the eigenvalue problem

$$
\begin{equation*}
\left(H_{m}+h_{m+1, m}^{2} H_{m}^{-H} \mathbf{e}_{m} \mathbf{e}_{m}^{H}\right) \mathbf{p}_{i}=\theta_{i} \mathbf{p}_{i}, i=1, \ldots, m \tag{8}
\end{equation*}
$$

where the square matrix $H_{m}$ is $\bar{H}_{m}$ without the last row and $h_{m+1, m}$ is the non-zero entry of $\bar{H}_{m}$ on its last row, and then setting

$$
\left[\mathbf{y}_{1}, \mathbf{y}_{2}, \ldots, \mathbf{y}_{k}\right]=: Y_{k} \leftarrow V_{m} P_{k}
$$

where $P_{k}=\left[\mathbf{p}_{1}, \mathbf{p}_{2}, \ldots, \mathbf{p}_{k}\right]$ correspond to the $k$ smallest $\theta_{i}$. The matrices $C_{k}$ and $U_{k}$ are constructed from $Y_{k}$ by setting

$$
\begin{equation*}
C_{k} \leftarrow V_{m+1} Q \text { and } U_{k} \leftarrow Y_{k} R^{-1} \tag{9}
\end{equation*}
$$

where $[Q, R]$ is the $Q R$-factorisation of $\bar{H}_{m} P_{k}$. It can be verified that the resulting $C_{k}$ and $U_{k}$. satisfy the condition in eq. (4).

The start-up cycle is followed by a deflated GMRES cycle in which we perform $(m-k)$ Arnoldi iterations starting with $\mathbf{v}_{1}=\mathbf{r} /\|\mathbf{r}\|$ using the linear operator $\left(I-C_{k} C_{k}^{H}\right) A$ such that the Krylov vectors to be formed are orthogonal to $C_{k}$ as in GCR. The key step is to combine $U_{k}$ from the outer GCR and $V_{m-k}$ from the inner Arnoldi iterations to form a subspace to approximate the solution. Define

$$
\hat{V}_{m}=\left[U_{k} D_{k}, V_{m-k}\right], \hat{W}_{m+1}=\left[C_{k}, V_{m-k+1}\right], \bar{G}_{m}=\left[\begin{array}{cc}
D_{k} & B_{m-k} \\
0 & \bar{H}_{m-k}
\end{array}\right]
$$

which satisfy the generalised Arnoldi relation

$$
A \hat{V}_{m}=\hat{W}_{m+1} \bar{G}_{m}
$$

where $D_{k}=\operatorname{diag}\left(\left\|\mathbf{u}_{1}\right\|^{-1},\left\|\mathbf{u}_{2}\right\|^{-1}, \ldots,\left\|\mathbf{u}_{k}\right\|^{-1}\right)$ and $B_{m-k}=C_{k}^{H} A V_{m-k}$. The solution update $\delta \mathbf{x}$ is approximated over the subspace spanned by the columns of $\hat{V}_{m}$ and we solve for the coefficient vector $\mathbf{d}_{m}$ that minimises the norm of the resulting residual $\left\|\mathbf{r}-A \hat{V}_{m} \mathbf{d}_{m}\right\|$, which, due to the Arnoldi relation, is equivalent to $\left\|\hat{W}_{m+1}^{H} \mathbf{r}-\bar{G}_{m} \mathbf{d}_{m}\right\|$. The minimiser can be found by solving a least square problem. According to [20], a more efficient alternative is to first
solve for the last ( $m-k$ ) components of $\mathbf{d}_{m}$ using $\bar{H}_{m-k}$ only and then solve for the first $k$ components of $\mathbf{d}_{m}$ that correspond to the basis vectors $U_{k}$. This alternative approach could be more accurate for some cases, although no notable difference is observed in our work. The solution and residual vectors are then updated with $\mathbf{d}_{m}$. In addition, we compute $\theta_{i}$ and $\mathbf{p}_{i}$ of the generalised eigenvalue problem

$$
\bar{G}_{m}^{H} \bar{G}_{m} \mathbf{p}_{i}=\theta_{i} \bar{G}_{m}^{H} \hat{W}_{m+1}^{H} \hat{V}_{m} \mathbf{p}_{i}
$$

similar to eq. (8). The approximate interior eigenvectors of the coefficient matrix are $Y_{k}=\hat{V}_{m} P_{k}$ with $P_{k}$ containing the $k$ interior eigenvectors to the reduced system as its columns. To form $C_{k}$ and $U_{k}$, first perform QRfactorisation of $\bar{G}_{m} P_{k}$ and then set

$$
C_{k} \leftarrow \hat{W}_{m+1} Q \text { and } U_{k} \leftarrow Y_{k} R^{-1}
$$

similar to eq. (9).
Different from GCRO, after $C_{k}$ and $U_{k}$ are formed, the solution and residual vectors are not immediately updated in GCRO-DR. This is because the least square problem after the inner Arnoldi iteration has included the basis $U_{k}$ in the search subspace. The resulting algorithm is denoted as GCRO$\operatorname{DR}(m, k)$ where $m$ is the dimension of the Krylov subspace retained for approximating the solution while $k$ is the number of eigenvectors recycled. The total number of vectors that needs to be stored is thus $(m+k)$. The deflated GMRES cycle is repeated using the most recent solution and residual vectors along with the recycled eigenvectors until the stopping criterion is met.

In GCRO-DR, the eigenvectors $Y_{k}$, computed from the Arnoldi iterations, are 'recycled' to improve/deflate the next subsequent cycle. Due to the flexibility of GCRO, any set of vectors could be recycled and regularised to form $U_{k}$ and $C_{k}$, not only for the next restarted cycle, but also for solving a different equation. Instead of performing a start-up cycle of $m$ GMRES iterations to get $k$ approximate interior eigenvectors, those from a previous solve can be recycled using the QR-factorisation to generate $U_{k}$ and $C_{k}$ that satisfy the condition in eq. (4). We call this variant version that allows interequation recycling GCRO-DR-R. The algorithm for GCBO-DR-R is shown in Appendix A. It should be pointed out that the original GCRO-DR algorithm introduced in [19] is also capable of recycling from one equation to another. Here we denote this kind of inter-equation recycling technique with an additional '-R' just for clarity.

The effectiveness of recycling the eigenvectors between equations strongly depends on how good an approximation the eigenvectors from one equation are for another. For farying right-hand sides, there are mainly three scenarios: (i) identical coefficient matrices, (ii) diagonally shifted coefficient matrices and (iii) similar but different coefficient matrices. For case (i), GCRO-DR-R should be very effective as the spectral information of the coefficient matrix does not change. For case (ii), GCRO-DR-R is also expected to be quite effective. Although the eigenvectors are preserved despite the shifted spectrum, the smallest eigenvectors for $\left(A-\sigma_{1} I\right)$ are not necessarily the smallest eigenvectors for $\left(A-\sigma_{2} I\right)$. Thus the deflation may not be as effective as in the first case. Case (iii) is the most general one. In [23], GCRO is extended to solve a system of equations that have varying sym-
metric real-valued coefficient matrices as well as multiple complex shifts. In [23], some theory was developed regarding the convergence behaviour in the presence of small perturbation of the coefficient matrix, which indicates that the recycling in case (iii) may still be effective as long as the change in the coefficient matrix is small.

The other factor determining the effectiveness of recycling is the righthand side. However, a theory regarding the convergence due to the right-hand side does not seem to exist (for a survey of the existing theories regarding the convergence properties of various Krylov subspace methods, refer to [24]). While some attempts have recently been made to formulate an asymptotic convergence bound taking into account the right-hand sides [25], the more useful transient convergence behaviour is still unclear.

## 4. Results

The linear equation solvers outlined in the previous section are now applied to three test cases to demonstrate their effectiveness in reducing the memory requirements and in accelerating convergence. The governing equations of the flow are solved using the DLR-TAU code and all linear systems are obtained from this solver's discretisation scheme. For the twodimensional aerofoil test case and the three-dimensional half wing-body case, the Reynolds-averaged Navier-Stokes equations are used together with the one equation turbulence model of Spalart-Allmaras. The inviscid fluxes of the mean flow equations are discretised using the second-order central scheme with scalar artificial dissipation, while the first-order Roe scheme is used for the turbulence equation. Viscous fluxes follow the full gradient approach
with the gradients reconstructed using the Green-Gauss theorem. The Euler equations are solved for the full aircraft case using the same central scheme. The matrix, which the ILU factorisation is based on, is formed by linearly blending the Jacobian matrices of first- and second-order spatial discretisation [26]. Furthermore, ILU with zero fill-in is used throughout this work. ILU with one level of fill-in requires appreciably more memory but results in limited speedup.

### 4.1. Frequency response and buffet onset for an aerofoil

The first test case is a two-dimensional NACA 0012 aerofoil undergoing harmonic excitation in pitch and plunge modes at various frequencies in transonic flow near buffet onset. The freestreany Mach number is 0.76 with a Reynolds number of 10 million. Fully turbulent flow is assumed. The angle of attack is 3.5 deg. The aerofoil case has 30,000 grid points, corresponding to 150,000 complex-valued unknowns. The sparse Jacobian matrix has 16.7 million non-zero entries.

To investigate basic properties of the different linear solvers, we first solve the linear equation for a reduced frequency of 0.35 excited by the pitch mode. As will be explained below, the resulting linear system at this frequency is most difficult to solve. The convergence history for ten orders of magnitude residual drop using different solvers is plotted in Fig. 1. The residual here and in the following paragraphs is the normalised $\mathrm{L}_{2}$ norm of the preconditioned residual, defined as Res $=\left\|U^{-1} L^{-1}(\mathbf{b}-A \mathbf{x})\right\| /\left\|U^{-1} L^{-1} \mathbf{b}\right\|$. As reference, full GMRES is first used and it converges with 464 iterations. Restarted GMRES is then run using 50, 100 and 200 Krylov vectors, respectively. At least 100 vectors are needed to avoid convergence stagnation and GMRES


Figure 1: Case 1. Left: Convergence history using restarted GMRES and GCRO-DR; right: CPU time of GMRES and GCRO-DR using different number of vectors with ten orders of magnitude convergence.
restarted with 200 vectors is found to 末e optimal. Using more than 200 vectors does not further accelerate the convergence due to the increased cost in orthogonalisation. On the other hand, $\operatorname{GCRO}-\operatorname{DR}(20,10)$, requiring only 30 vectors to be stored, conyerges significantly faster, reducing the CPU time of the best performing GMRES solver by over a factor of three.

To examine the memory requirements and CPU time for both solvers for the same linear system, different numbers of Krylov vectors are tested. All solvers are required to converge by ten orders of magnitude. Since there are two parameters $m$ and $k$ that can vary independently in GCRO-DR, we simplify the parameter study by setting $k=m / 2$. Convergence stalls if less than 27 vectors are stored for GCRO-DR, above which the CPU time almost linearly increases with the fastest convergence achieved when storing 30 vectors. Therefore, $\operatorname{GCRO}-\mathrm{DR}(20,10)$ is used for the remaining computations of
this case. For GMRES, the restart number is decreased from $m=464$ until the convergence severely slows down and eventually stalls below $m=200$. The two dashed lines in Fig. 1 show the respective smallest number of yectors needed by the two solvers. Note that the reduced memory requirement is equivalent to storing six flow Jacobian matrices for this two-dimensional case.

Next, inter-equation recycling is investigated. The frequency response due to harmonic excitation in pitch and plunge modes is computed with three solvers: $\operatorname{GMRES}(200)$, $\operatorname{GCRO}-\operatorname{DR}(20,10)$ and GCRO-DR-R $(20,10)$. The reduced frequency varies from 0.1 to 0.6 with ah increment of 0.05 . For each frequency, the aerofoil is first excited using the pitch and then the plunge mode. The order of solving the sequence of equations is such that the coefficient matrices and their complex-valued $\operatorname{HU}(0)$ preconditioner are formed as few times as possible and the coefficient matrices vary monotonically to allow effective recycling between equations. A total of 22 linear solves are performed. The convergence criterion is again a residual drop of ten orders of magnitude.

As can be seen in Fig. 2, compared with GCRO-DR, GCRO-DR-R completely avoids the initial phase of the slow convergence from the second linear equation solve due to recycling. The effect however slows down and the asymptotid convergence of the nested solvers is comparable, both outperforming GMRES. The CPU time breakdown of each linear solve for all three solvers for different modes excited at different frequencies is shown in Fig. 3. Compared to GMRES(200), GCRO-DR(20,10) reduces the overall CPU time by $64 \%$ and using recycling between equations speeds up another $15 \%$.


Figure 2: Case 1. Convergence history of $\operatorname{GMRES}(200), \operatorname{GCRO}-\operatorname{DR}(20,10)$ and GCRO-DR-R $(20,10)$ for the first three equation solves.



Figure 3: Case 1: CPU time for computing the frequency response using different solvers.
a peak at the reduced frequency of 0.35 for both pitch and plunge modes, which is believed related to the near resonance fluid motion. This resonance behaviour is more evident from the plot on the left of Fig. 4 showing the


Figure 4: Case 1. Left: the complex lift derivative for 3.5 deg angle of attack excited by pitch and plunge modes over the range of reduced frequencies; right: convergence history of different linear solvers for the first outer iteration of the inverse correction solver.
magnitude of the unsteady lift derivative of the aerofoil excited by both modes at different reduced frequencies. This motivates the computation of the responsible fluid eigenvalue and eigenvector. To compute the eigen pair, inverse correction method [16] as in eq. (3), is used with shift $\sigma=0.35 i$ and a random initial guess for the eigenvector.

The first outer iteration is solved using GMRES with different restarts and GCRO-DR, for which the convergence history is shown on the right of Fig. 4. Restarting after every 300 vectors seems to be optimal for GMRES. For GCRO-DR $(20,10)$, the memory required is only one tenth of that of GMRES, while the convergence is accelerated by over a factor of four. The convergence criterion used here is ten orders of magnitude. In practice, since only the outer iteration convergence is related to the actual convergence of the eigenvalue problem, the inner iteration convergence could be relaxed to


Figure 5: Case 1. Left: CPU time comparison of different solvers for all nine outer iterations; right: convergence of the eigenvalue problem.

### 4.2. Basis of reduced order model for an aircraft model

The three-dimensional model is a wide-body civil aircraft research configuration with a semi-span of about 30 m and an overall length of about 65 m . Flow conditions are a freestream Mach number of 0.85 at 1.0 deg angle of attack. The computational mesh for the Euler CFD calculation has 0.74 million grid points, equivalent to 3.7 million complex-valued unknowns for the linear system. In terms of storage, the flow Jacobian matrix for the second-order spatial discretisation has around 451 million non-zero real-valued entries requiring around 3.3 GB of memory to be stored with double precision. The complex-valued ILU(0) preconditioning matrix stored in single precision requires another 3.3 GB of memory.

Using ten mode shapes of the structure(the first mode, dominant in wing bending, as mapped to the CFD surfaces is illustrated on the left of Fig. 6), ten eigenvalues are found using the Schur complement method. The Schur complement method can efficiently compute eigenvalues and the associated structural part of the right and left eigenvectors. The remaining task is to find the fluid part of the eigenvectors using eqs. (1) and (2). Once both the right and left eigenvectors of the coupled problems are found, a reduced order model can be constructed [4].

From previous experience, it is known that the left eigenvector problem is more challenging. Thus, GMRES and GCRO-DR using different parameters are first used to solve for the first left eigenvector with the convergence history shown on the right in Fig. 6. The convergence criterion has been set to six orders of magnitude, which is sufficient to converge the unsteady lift derivative to within $1 \%$ accuracy. GMRES with 100 vectors is found


Figure 6: Case 2. Left: aerodynamic surfaces of a civil aircraft model for projected first bending mode; right: convergence for computing the first left eigenvector. to be optimal, i.e., using 50 vectors leads to convergence stagnation while using more slows down the convergence due to the increased cost in orthogonalisation. For GCRO-DR, there is only marginal variation in CPU time for the three combinations of parameters used, and thus GCRO-DR $(30,10)$ is chosen for all the remaining eigenvector solves. Replacing GMRES(100) with GCRO-DR $(30,10)$ reduces the total memory requirement for the linear solve (flow Jacobian matrix, ILU preconditioning matrix and Krylov vectors combined) from 18.2 GB to 11.4 GB, with the difference being the 60 fewer vectors needed to be stored, while the CPU time is reduced by a factor of three. Note that the results for this case are computed using one CPU only. The CPU time breakdown of the 20 eigenvector solves using GMRES(100) and GCRO-DR $(30,10)$ is shown in Fig. 7. Using deflated restarting speeds up


Figure 7: Case 2. CPU time for solving all eigenvectors using the three different solvers.
455
the whole calculation by a factor of two. However, the recycling technique does not seems to be effective for this case. Using eigenvector recycling between solves even slows down the convergence for left eigenvector problems, while the speedup for the right eigenvector solves are marginal at around $6 \%$. This is presumably due to the fact that the approximate eigenvectors recycled from solving one linear system is no longer a good approximation for the eigenvectors of the next.

It was reported in [19] that GCRO-DR with recycling may not necessarily accelerate the convergence even when recycling is applied to the same equation with both identical coefficient matrices and right-hand sides. This same behaviour was found in this case when repeatedly solving the first left eigenvector problem, i.e., solving the equation again with recycled eigenvectors from the previous solve of the same equation does not speed up convergence.

Another feature of the convergence is that the left eigenvectors are in general more difficult to solve compared to the right eigenvectors, even though their coefficient matrices have the same spectrum, thus the same asymptotic convergence property. This implies that the right-hand side plays an important role in the transient convergence in these cases [25].

### 4.3. Frequency response for a pre-buffet half wing-body configuration

The frequency response for a generic half wing-body model is now computed. The freestream Mach number is 0.8 while the Reynolds number based on the aerodynamic mean chord is 3.5 million. Fully turbulent flow is assumed. Two angles of attack at 2.8 deg and 3.0 deg are considered, with the higher angle close to buffet onset. Similar to the aerofoil-near-buffet case, the higher angle of attack case exhibits significant stiffness in the linear system. The steady state solutions are converged by ten orders of magnitude, which is deemed sufficient by monitoring the lift, drag and moment coefficients. The surface pressure distribution for the steady state solution at 2.8 deg angle of attack is shown on the left in Fig. 8. To perform linearised aerodynamic analysis, four synthetic mode shapes (the first of which is shown on the right in Fig. 8) are used to harmonically excite the system at a range of reduced frequencies.

The test case consists of around 2.7 million grid points, resulting in a complex-valued linear system of equations with around 16.3 million degrees of freedom. The second-order Jacobian matrix has 3.14 billion real-valued entries corresponding to 24 GB of memory. Storing the Jacobian matrix is equivalent to 100 Krylov vectors. The $\operatorname{ILU}(0)$ preconditioning matrix, based on the blended first- and second-order Jacobian matrices, stores around 3.14


Figure 8: Case 3. Left: pressure coefficient distribution at 2.8 deg angle of attack; right: baseline wing shape and the deformed one with the first synthetic mode.
billion complex-valued entries in single precision. For comparison, ILU(1) would require 48 GB of memory. All simulations are run on 144 cores with 2 GB per core. The ILU(0) preconditioning matrix is computed locally without parallel communication. It will have deteriorated performance with increased number of cores as reported in [27]. The convergence tolerance is set to ten orders of magnitude throughout.

To perform a parameter study for both GMRES and GCRO-DR, the effect of two key parameters is examined: blending coefficient to combine first- and second-order Jacobian matrices and number of Krylov vectors. For each angle of attack, the blending coefficient is varied between 0 and 1. Then both $m$ for GMRES and $(m+k)$ for GCRO-DR vary from 50 to 500 for each value of the blending coefficient. For each combination of $m+k, m$ and $k$ are already optimised. More details on the best combination is discussed in the following paragraph. The runtime of both GMRES and


Figure 9: Case 3. Runtime of linear frequency domain solvers at 2.8 and 3.0 deg angle of attack, all converged ten orders of magnitude.
best-performing GCRO-DR with respect to the number of vectors stored and blending coefficient are shown in Fig. 9.

At 2.8 deg angle of attack, GCRO-DR roughly speeds up the solution by 499 a factor of two. In addition, the lower threshold of the number of Krylov

Table 1: Optimal number of recycled eigenvectors at 2.8 deg (left) and 3.0 deg (right) angle of attack. ( $\mathrm{NA}=$ Not Available)

| $k$ |  | $m+k$ |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 60 | 80 | 100 | 200 | 300 | 400 | 500 |
|  | 0.0 | 2 | 3 | 10 | 20 | 30 | 40 | 15 |
|  | 0.2 | 2 | 3 | 5 | 20 | 15 | 20 | 15 |
|  | 0.4 | 2 | 5 | 10 | 30 | 15 | 20 | 15 |
|  | 0.6 | 4 | 2 | 5 | 10 | 15 | 20 | 15 |
|  | 0.8 | NA | NA | 5 | 10 | 15 | 20 | 15 |
|  | 0.9 | NA | NA | 10 | 10 | 15 | 20 | 15 |


| $k$ |  | $m+k$ |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 80 | 100 | 150 | 200 | 300 | 400 | 0 |
|  | 0.0 | 10 | 15 | 30 | 30 | 30 | 20 | 20 |
|  | 0.2 | 10 | 15 | 25 | 30 | 30 | 15 | 20 |
|  | 0.4 | 10 | 15 | 25 | 30 | 30 | 15 | 20 |
|  | 0.6 | NA | 15 | 25 | 30 | 30 | 15 | 20 |
|  | 0.8 | NA | 15 | 20 | 20 | 30 | 20 | 20 |
|  | 0.9 | NA | NA | 25 | 25 | 35 | 15 | 20 |

${ }^{515}$, fastest convergence. As can be seen, there does not seem to be a general rule 516
vectors for reasonable convergence is reduced substantially, from 300 to below 100. At 3.0 deg angle of attack, GCRO-DR accelerates the convergence much more significantly, by a peak-to-peak ratio of 4.5 . For GMRES to converge in reasonable time, at least 300 vectors are required, while 150 Krylov vectors seem sufficient otherwise. Overall, the convergence is not very sensitive to the blending coefficient as long as it does not exceed 0.9 , beyond which the stiffness of the linear system increases drastically. For a blending coefficient of 1 , none of the linear solvers is able to converge with 500 Krylov vectors, neither with $\operatorname{ILU}(0)$ nor $\operatorname{ILU}(1)$. The extreme stiffness has been reported previously but a plausible elucidation is still missing.

Note that in Fig. 9 the runtime for GCRO-DR is for the optimal combination of $m$ and $k$, while keeping the sum constant, from a series of tests. To understand how the parameters should be chosen, Table 1 presents the values of $k$ for the plotted GCRO-DR convergence data. This optimal value is found by varying $k$ from 0 to $m / 2$ and identifying the one that results in the for choosing $k$, except some guiding principles such as $k$ should be around

## 5. Conclusions

In this paper, generalised conjugate residual solver with deflated restarting is applied to a few typical problems of linearised aerodynamic analy-


Figure 10: Case 3. Runtime of all solves at 3.0 deg angle of attack using three different solvers with tolerance of ten orders of magnitude.
sis and the significant improvement over the baseline generalised minimal residual method, regarding both the memory requirement and CPU time, is demonstrated. The cases investigated are the frequency response computation of an aerofoil undergoing pitch or plunge motions in transonic flow at
near buffet onset condition, computing the left and right eigenvectors for a full aircraft model in inviscid transonic flow, and the frequency response of a half wing-body model in viscous transonic flows at below- and near-buffet, both of which are challenging off-design conditions. All the test cases involve solving large sparse linear systems of equations arising from linear frequency domain Euler or Navier-Stokes equations.

The deflation technique significantly reduces both the CPU time and the number of Krylov vectors that need to be stored for all cases. Although recycling eigenvectors between equations does not necessarily improve the asymptotic convergence rate, it does significantly improve the transient convergence by overcoming the initial stagnation, making it ideal for solving a sequence of linear systems of equations with relaxed convergence criteria.

## 6. Acknowledgements

The research leading to these results was co-funded by Innovate UK, the UK's innovation agency, as part of the Enhanced Fidelity Transonic Wing project.

## References

[1] R. Thormann, M. Widhalm, Linear-frequency-domain predictions of dynamic-response data for viscous transonic flows, AIAA Journal 51 (11) (2013) 2540-2557.
[2] S. Timme, S. Marques, K. Badcock, Transonic aeroelastic stability analysis using a Kriging-based Schur complement formulation, AIAA Journal 49 (6) (2011) 1202-1213.
[3] A. Da Ronch, K. J. Badcock, Y. Wang, A. Wynn, R. N. Palacios, Nonlinear model reduction for flexible aircraft control design, AIAA Paper 2012-4404.
[4] S. Timme, K. Badcock, A. Da Ronch, Linear reduced order modelling for gust response analysis using the DLR-TAU code, Proc. IFASD, 2013.
[5] J. Crouch, A. Garbaruk, D. Magidov, Predicting the onset of flow unsteadiness based on global instability, Journal of Computational Physics 224 (2) (2007) 924-940.
[6] M. Iorio, L. González, E. Ferrer, Direct and adjoint global stability analysis of turbulent transonic flows over a NACA0012 profile, International Journal for Numerical Methods in Fluids 76 (3) (2014) 147-168.
[7] F. Sartor, C. Mettot, D. Sipp, Stability, receptivity, and sensitivity analyses of buffeting transonic flow over a profile, AIAA Journal 50 (4) (2014) 880-890.
[8] J. Nitzsche, A numerical study on aerodynamic resonance in transonic separated How, Proc. IFASD, 2009.
[9] M. Iovnovich, D. E. Raveh, Reynolds-averaged Navier-Stokes study of the shock-buffet instability mechanism, AIAA Journal 50 (4) (2012) 880-890.
[10] R. Thormann, J. Nitzsche, M. Widhalm, Time-linearized simulation of unsteady transonic flows with shock-induced separation, in: 6th European Congress on Computational Methods in Applied Sciences and Engineering, 2012.
[11] X.-C. Cai, W. D. Gropp, D. E. Keyes, M. D. Tidriri, Newton-KrylovSchwarz methods in CFD, in: Numerical methods for the Navier-Stokes equations, Springer, 1994, pp. 17-30.
[12] W. Gropp, D. Keyes, L. C. Mcinnes, M. Tidriri, Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD, International Journal of High Performance Computing Applications 14 (2) (2000) 102-136.
[13] M. Osusky, D. W. Zingg, Parallel Newton-Krylov-Schur flow solver for the Navier-Stokes equations, AIAA journal 51 (12) (2013) 2833-2851.
[14] S. Xu, D. Radford, M. Meyer, J.-D. Müller, Stabilisation of discrete steady adjoint solvers, Journal of Computational Physics 299 (2015) 175-195.
[15] R. Dwight, Efficiency improvements of RANS-based analysis and optimization using implicit and adjoint methods on unstructured grids, Ph.D. thesis, University of Manchester (2006).
[16] U. Rüde, W. Schmid, Inverse multigrid correction for generalized eigenvalue computations, Technical Report, Universität Augsburg, 1995.
[17] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856-869.
[18] S. Eisenstat, H. Elman, M. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM Journal on Numerical Analysis 20 (2) (1983) 345-357.
[19] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, S. Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM Journal on Scientific Computing 28 (5) (2006) 1651-1674.
[20] E. de Sturler, Nested Krylov methods based on GCR, Journal of Computational and Applied Mathematics 67 (1) (1996) 15-41.
[21] E. de Sturler, Truncation strategies for optimal Krylov subspace methods, SIAM Journal on Numerical Analysis 36 (3)(1999) 864-889.
[22] R. B. Morgan, GMRES with deflated restarting, SIAM Journal on Scientific Computing 24 (1) (2002) 20-37.
[23] M. E. Kilmer, E. de Sturler, Recycling subspace information for diffuse optical tomography, SIAM Journal on Scientific Computing 27 (6) (2006) 2140-2166.
[24] J. Liesen, P. Tichỳ, Convergence analysis of Krylov subspace methods, GAMM-Mitteilungen 27(2) (2004) 153-173.
[25] D. Titley-Peloquin, J. Pestana, A. J. Wathen, GMRES convergence bounds that depend on the right-hand-side vector, IMA Journal of Numerical Analysis 34 (2) (2014) 462-479.
[26] A. J. McCracken, S. Timme, K. J. Badcock, Accelerating convergence of the CFD linear frequency domain method by a preconditioned linear solver, in: 6th European Congress on Computational Methods in Applied Sciences and Engineering, 2012.
[27] R. Thormann, M. Widhalm, Forced motion simulations using a linear frequency domain solver for a generic transport aircraft, Proc. IFASD, 2013.

```
Algorithm 1 GCRO-DR-R \(\left(A, \mathbf{x}_{0}, \mathbf{b}, m, k\right.\), iter_max, tol \(, U, L\), Recycle \()\)
    \(\mathbf{r} \leftarrow U^{-1}\left(L^{-1}\left(\mathbf{b}-A \mathbf{x}_{0}\right)\right) ;\) iter \(\leftarrow 0\)
    if (Recycle=True) then \(\% Q R\) factorisation of recycled \(Y_{k}\)
        \([Q, R] \leftarrow \mathrm{QR}\left(Y_{k}\right) ; C_{k} \leftarrow Q ; U_{k} \leftarrow Y_{k} R^{-1}\)
        \(\mathbf{x} \leftarrow \mathbf{x}_{0}+U_{k} C_{k}^{H} \mathbf{r} ; \mathbf{r} \leftarrow \mathbf{r}-C_{k} C_{k}^{H} \mathbf{r}\)
    else \(\% m\) Arnoldi iterations to calculate \(Y_{k}\)
        \(\beta \leftarrow\|\mathbf{r}\| ; \mathbf{v}_{1} \leftarrow \mathbf{r} / \beta\)
        for \(i=1,2, \ldots, m\) do
            \(\mathbf{v}_{i+1} \leftarrow U^{-1}\left(L^{-1}\left(A \mathbf{v}_{i}\right)\right)\); iter \(\leftarrow\) iter +1
            for \(j=1,2, \ldots, i\) do \(\%\) Orthogonalisation
                \(h_{j, i} \leftarrow \mathbf{v}_{j}^{H} \mathbf{v}_{i+1} ; \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1}-h_{j, i} \mathbf{v}_{j}\)
                \(h_{i+1, i} \leftarrow\left\|\mathbf{v}_{i+1}\right\| ; \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1} / h_{i+1, i}\)
            Solve for \(\mathbf{d}_{m}\) that minimises \(J\left(\mathbf{d}_{m}\right)=\left\|\beta \mathbf{e}_{1}-\bar{H}_{m} \mathbf{d}_{m}\right\|\)
            \(\mathbf{x} \leftarrow \mathbf{x}_{0}+V_{m} \mathbf{d}_{m} ; \mathbf{r} \leftarrow V_{m+1}\left(\beta \mathbf{e}_{1}-\bar{H}_{m} \mathbf{d}_{m}\right)\)
            Compute \(k\) eigenvectors \(\mathbf{p}_{i}\) in \(\left(H_{m}+h_{m+1, m}^{2} H_{m}^{-H} \mathbf{e}_{m} \mathbf{e}_{m}^{H}\right) \mathbf{p}_{i}=\theta_{i} \mathbf{p}_{i}\)
            \([Q, R] \leftarrow \operatorname{QR}\left(\bar{H}_{m} P_{k}\right) ; C_{k} \leftarrow V_{m+1} Q ; Y_{k} \leftarrow V_{m} P_{k} ; U_{k} \leftarrow Y_{k} R^{-1}\)
    \(\mathbf{v}_{1} \leftarrow \mathbf{r} /\|\mathbf{r}\|\)
    for \(i=1,2, \ldots, m-k\) do
            \(\mathbf{v}_{i+1} \leftarrow\left(I-C_{k} C_{k}^{H}\right)\left(U^{-1}\left(L^{-1}\left(A \mathbf{v}_{i}\right)\right)\right) ;\) iter \(\leftarrow\) iter +1
            for \(j=1,2, \ldots i\) do \(\%\) Orthogonalisation
                \(h_{j, i} \leftarrow \mathbf{v}_{j}^{H} \mathbf{v}_{i+1} ; \mathbf{y}_{i+1} \leftarrow \mathbf{v}_{i+1}-h_{j, i} \mathbf{v}_{j}\)
        \(h_{i+1, i} \longleftarrow\left\|\mathbf{v}_{i+1}\right\| ; \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1} / h_{i+1, i}\)
    \(D_{k} \leftarrow \operatorname{Diag}\left(\left\|\mathbf{u}_{1}\right\|^{-1},\left\|\mathbf{u}_{2}\right\|^{-1}, \ldots,\left\|\mathbf{u}_{k}\right\|^{-1}\right)\)
    \(B_{k} \leftarrow C_{k}^{H}\left(U^{-1}\left(L^{-1}\left(A V_{m-k}\right)\right)\right) ; \bar{G}_{m} \leftarrow\left[D_{k}, B_{k} ; 0, \bar{H}_{m-k}\right]\)
    \(\hat{V} \leftarrow\left[U_{k} D_{k}, V_{m-k}\right] ; \hat{W}_{m+1} \leftarrow\left[C_{k}, V_{m-k+1}\right]\)
    Solve for \(\mathbf{d}_{m}\) that minimises \(J\left(\mathbf{d}_{m}\right):=\left\|\hat{W}_{m+1}^{H} \mathbf{r}-\bar{G}_{m} \mathbf{d}_{m}\right\|\)
    \(\mathbf{x} \leftarrow \mathbf{x}+\hat{V}_{m} \mathbf{d}_{m} ; \mathbf{r} \leftarrow \mathbf{r}-\hat{W}_{m+1} \bar{G}_{m} \mathbf{d}_{m}\)
27: Compute \(k\) eigenvectors \(\mathbf{p}_{i}\) in \(\bar{G}_{m}^{H} \bar{G}_{m} \mathbf{p}_{i}=\theta_{i} \bar{G}_{m}^{H} \hat{W}_{m+1}^{H} \hat{V}_{m} \mathbf{p}_{i}\)
    \([Q, R] \leftarrow \operatorname{QR}\left(\bar{G}_{m} P_{k}\right) ; C_{k} \leftarrow \hat{W}_{m+1} Q ; Y_{k} \leftarrow \hat{V}_{m} P_{k} ; U_{k} \leftarrow Y_{k} R^{-1}\)
29: if \(\left(J\left(\mathbf{d}_{m}\right)<\right.\) tol or iter \(>\) iter_max \()\) then
30: \(\quad Y_{k}=U_{k}\); Terminate programme.
```

31: Goto line 16


[^0]:    *Corresponding author. Tel.: +44 (0)151 7948035
    Email address: shenren.xu@liverpool.ac.uk (Shenren Xu)

