# Accepted Manuscript

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

Shenren Xu, Sebastian Timme, Kenneth J. Badcock

 PII:
 S0045-7930(16)30320-6

 DOI:
 10.1016/j.compfluid.2016.10.018

 Reference:
 CAF 3306

To appear in: Computers and Fluids

Received date:14 March 2016Revised date:5 July 2016Accepted 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 Jacobian 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<sup>\*</sup>, Sebastian Timme, Kenneth J. Badcock 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 generalised conjugate residual solver with deflated restarting is adopted here in which an invariant Krylov subspace is recycled both between restarts when solving 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.

Preprint submitted to Journal of Computers and Fluids

October 19, 2016

<sup>\*</sup>Corresponding author. Tel.: +44 (0)151 794 8035

*Email address:* shenren.xu@liverpool.ac.uk (Shenren Xu)

*Keywords:* Linearized Computational Fluid Dynamics, Shock Buffet, Flutter, GCRO-DR, Krylov Subspace Recycling

#### 1. Introduction

This paper presents the details of an investigation to improve the gen-1 eralised minimal residual (GMRES) linear equation solver both in terms 2 f memory requirements and convergence rates when using the generalised 3 conjugate residual solver with deflated restarting (GCRO-DR). Linearised 4 computational fluid dynamics (CFD) is chosen to model the transonic aero-5 dynamics in the aeroelastic test cases presented. The range of potential ap-6 plications include flutter and gust response analysis as well as the prediction 7 of transonic shock buffet onset using eigenvalue calculation methods. 8

For modern aircraft design, the accurate and efficient calculation of both the flutter onset and gust response at an early design stage is of critical im-10 portance as it, to large extent, determines the flight envelope and structural 11 sizing. Despite significant advances in computational algorithms for CFD and 12 structural dynamics, using time-domain simulation for such routine analyses 13 is still not a viable option. Linear frequency domain solutions can be used to 14 efficiently compute the aerodynamic response due to structural excitation [1], 15 the information of which is then needed in a flutter analysis [2] similar to 16 industrial approaches using the classical doublet lattice method (DLM). In 17 addition, linear and nonlinear reduced order models can be constructed for 18 gust response analysis when using eigenmodes of the aeroelastic system as a projection basis [3, 4]. 20

21

More specifically, to predict the flutter onset using CFD aerodynamics

in a DLM-like approach, the aerodynamic response due to excitation in the 22 structural modes at various frequencies is first pre-computed, and then vari-23 ous interpolation techniques can be used to reconstruct a response surface to 24 quickly provide aerodynamic data when solving the small flutter eigenvalue 25 problem. Typically, for a complete passenger aircraft, the part of the analy-26 sis, requiring access to the CFD solver functionality with millions of degrees 27 of freedom, involves up to 100 structural modes and seven to ten excitation 28 frequencies resulting in several hundred solves per configuration. 29

An even more computationally challenging situation arises when the fluid 30 part exhibits an instability, which further complicates the fluid-structure in-31 teraction. An example of such flow instability with aeronautical relevance 32 transonic shock buffet. For small-scale problems, the onset of the flow 33 instability can be found by solving for a few eigenvalues of the system, us-34 ing a direct sparse equation solver, at each flow condition and tracking the 35 eigenvalue which first crosses the imaginary axis [5, 6, 7]. For realistic wing 36 cases, a direct method is not a viable option and an iterative sparse equation 37 solver combined with a shifted inverse eigenvalue method could be used in-38 stead to calculate the relevant eigenvalue. Different from a flutter analysis, 39 where the frequency range of interest is dictated by the frequencies of the 40 structural dynamics in vacuum, self-induced flow instability is independent 41 of the structural motion. A good initial guess for the shift is thus not easy 42 to find. In [8, 9] it was discussed however that the aerodynamic response at 43 pre-buffet conditions exhibits resonance when excited near the frequency of 45 the flow instability. This information can then serve as a good initial guess. Another computational challenge associated with tracking the destabilising 46

eigenvalue is that the shifted fluid Jacobian matrix is nearly singular. For
instance, in [10] the preconditioned restarted GMRES solver was used to
compute the frequency response of an aerofoil at pre-buffet conditions and
the linear solver failed to converge in many cases, presumably due to the
worsened stiffness at those conditions.

Krylov subspace solvers, such as GMRES, have been applied in the past 52 to converge the nonlinear Reynolds-averaged Navier-Stokes (RANS) equa-53 tions to a steady state [11, 12, 13]. In such implicit nonlinear RANS solvers, 54 however, the numerical stiffness of the inner linear system is usually reduced 55 due to an approximate or first-order Jacobian formulation [14]. The overall 56 solver efficiency is usually achieved by using an approximate Newton formu-57 lation for the inner system and solving it only inexactly. It is shown in [15] 58 that even for a two-dimensional RANS case, the linear system resulting from 50 the exact second-order Jacobian formulation is extremely stiff and an incom-60 plete lower-upper (ILU) factorisation preconditioner with fill-in level of four 61 is needed to precondition the GMRES solver in order to allow convergence. 62 For three-dimensional cases with industrial relevance, the exact Newton-63 Krylov approach is not practical for the nonlinear RANS flow solver. Note 64 that by 'first/second-order Jacobian', we mean the Jacobian matrix based on 65 the first/second-order spatial discretisation, not a Hessian matrix. 66

The numerical stiffness is much more severe for the linearised RANS equations (as discussed in the current paper) because the coefficient matrix is now not only of second-order, but also shifted often close to become singular, thus causing tremendous convergence difficulty [10]. In this work, we form the ILU preconditioner from a blended Jacobian matrix (even though

the coefficient matrix itself is still the second-order one) and show that a 72 fill-in level of zero rather than four is necessary for effective precondition-73 ing, thus significantly reducing the memory requirement. In addition, the 74 most commonly used Krylov solver, GMRES, is replaced with a deflated 75 solver, GCRO-DR, to substantially reduce the minimal Krylov subspace size 76 required to avoid stagnation. Our proposed algorithm is much more effi-77 cient than the state-of-the-art ILU-Newton-Krylov algorithm currently used 78 in industry for linearised aerodynamics calculations. The application of our 79 proposed algorithm to edge-of-envelope test cases shows that both the mem-80 ory requirement and the CPU time can be significantly reduced, enabling the 81 linearised aerodynamics analysis of a much wider range of problems. 82

The theoretical formulation of the linearised CFD aerodynamics is introduced in Section 2, while the numerical methods used for solving the resulting large sparse linear systems of equations are discussed in Section 3. Results demonstrating the memory and runtime efficiency of the GCRO-DR solver compared to the baseline GMRES solver are shown in Section 4 for three test cases.

## <sup>89</sup> 2. Linearised frequency domain aerodynamics

The transient nonlinear equation describing the unsteady aerodynamics is written in semi-discrete form as

$$\dot{\mathbf{w}}_f = \mathbf{R}_f(\mathbf{w}_f, \mathbf{x}, \dot{\mathbf{x}}, \boldsymbol{\theta})$$

<sup>90</sup> where  $\mathbf{w}_f$  and  $\mathbf{R}_f$  denote the fluid unknowns and corresponding residual <sup>91</sup> vector, respectively, and  $\boldsymbol{\theta}$  are the system parameters. The vectors  $\mathbf{x}$  and  $\dot{\mathbf{x}}$  <sup>92</sup> are location and velocity of the fluid mesh points which are functions of the <sup>93</sup> structural mode shapes and the modal amplitudes denoted  $\eta$ . Linearising <sup>94</sup> about an equilibrium point and assuming small amplitude harmonic motion, <sup>95</sup> the latter equation can be re-formulated as

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

where  $\phi_f$  and  $\phi_\eta$  are complex-valued amplitudes of fluid and structure, re-96 spectively. The equation gives the aerodynamic response  $\phi_f$  following a dis-97 turbance of the structure  $\phi_n$ . To find aerodynamic data for further analysis, 98 this equation usually has to be pre-computed for each structural mode shape 99 (denoted by superscript i) and for a range of forced sinusoidal excitations 100 in the modal amplitudes at different frequencies  $\omega$  (denoted by superscript 101 k). As mentioned earlier, several hundreds of this equation need to be solved 102 for industrial problems with the number of fluid unknowns easily exceeding 103 several tens of millions. 104

Adding equations to describe the unsteady motion of the structure in 105 terms of the modal amplitudes  $\eta$  gives a coupled problem to be solved for 106 investigation in flutter stability and also gust response behaviour. The basis 107 of a reduced order model for such aeroelastic analyses can be calculated from 108 the eigenvectors of the coupled system. The fluid part of the direct (i.e. right) 109 eigenvalue problem is equivalent to eq. (1), except that the structural part 110 111  $\phi_n$  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 112 eigenvector (with superscript k = j). The corresponding adjoint (i.e. left) 113

eigenvalue problem for the fluid part  $\psi_f$  of the eigenvector can be derived as

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

where  $\mathbf{R}_{\dot{\eta}}$  is the residual vector corresponding to the structural unknowns. 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 119 without structural motion. One typical example is the shock-buffet problem 120 where the shock wave interacts with the boundary layer and destabilises the 121 steady flow beyond a critical parameter. To find the buffet onset, shifted 122 inverse methods are an obvious choice to calculate few eigenvalues close to 123 an initial guess. To choose such initial shift (i.e. a characteristic frequency 124 of the instability), either engineering judgement is required or the resonant 125 behaviour of the flow when excited at frequencies in the vicinity of the insta-126 bility can be exploited [8, 9], which would be equivalent to solving eq. (1). 127 The closer the shift to the target eigenvalue, the faster the algorithm con-128 verges. This however leads to the second, even bigger challenge using shifted 129 inverse methods. The linear system to be solved is nearly singular. Using a 130 direct sparse linear equation solver quickly becomes infeasible for everything 131 beyond two-dimensional problems. Thus, a preconditioned sparse iterative 132 linear equation solver is a possible alternative. 133

134

We use the shifted inverse method from [16] referred to as inverse correc-

<sup>135</sup> tion. The equation to be solved is

$$\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{w}_f} - \sigma I\right) \delta \phi_f = -\left(\frac{\partial \mathbf{R}_f}{\partial \mathbf{w}_f} - \lambda I\right) \phi_f \tag{3}$$

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

$$\boldsymbol{\phi}_{f} \leftarrow \boldsymbol{\phi}_{f} + \delta \boldsymbol{\phi}_{f}$$

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

$$\lambda = oldsymbol{\phi}_f^H rac{\partial \mathbf{R}_f}{\partial \mathbf{w}_f} oldsymbol{\phi}_f$$

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 **b**.

#### <sup>146</sup> 3. Solving large sparse linear systems of equations

The main challenge in using linearised CFD aerodynamics is solving large 147 sparse linear systems of equations. Restarted generalised minimal residual 148 (GMRES) solver [17] was implemented in the DLR-TAU code to solve the 149 linearised equations. As mentioned in the introduction, for stiff problems, 150 restarted GMRES often suffers from stagnation unless a large number of 151 Krylov vectors is kept. It is not uncommon to keep several hundred Krylov 152 vectors in order to converge. This large memory requirement could then 153 become the bottleneck when solving large cases. To ease these difficulties, 154 we implemented generalised conjugate residual solver with deflated restarting 155 (GCRO-DR), which converges almost like full GMRES (GMRES that does 156 not restart and keeps expanding the Krylov subspace until the convergence 157 criterion is met) but has small memory requirement. In addition, recycling 158 a certain Krylov subspace between different equations is possible for GCRO-159 DR and is thus favourable for solving a sequence of linear equations with 160 similar coefficient matrices. 161

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

165 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

$$\mathcal{K}_m(A, \mathbf{b}) = \operatorname{span}(\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \dots, A^{m-1}\mathbf{b})$$

using the Arnoldi iteration. After *m* Arnoldi steps, a unit vector basis  $V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$  that spans the Krylov subspace is constructed satisfying the Arnoldi relation

$$AV_m = V_{m+1}\bar{H}_m$$

where  $\bar{H}_m$  is an upper Hessenberg matrix. The solution **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

$$\|\mathbf{r}\| = \|\mathbf{r}_0 - AV_m \mathbf{d}_m\| = \|\beta \mathbf{v}_1 - V_{m+1} \bar{H}_m \mathbf{d}_m\|$$
$$= \|V_{m+1}\| \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{d}_m\| = \|\beta \mathbf{e}_1 - \bar{H}_m \mathbf{d}_m\|$$

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 176 by  $U^{-1}$ . The triangular matrices L and U are inverted using forward and 177 backward substitutions. Alternatively, right or split preconditioning could 178 also easily be achieved by plugging in these two additional matrix inversions 179 in a slightly different manner [17]. There seems to be no obvious advantage 180 for any type of preconditioning over the others, therefore we only used left 181 preconditioning throughout this work as the least modification is required. 182 In addition, due to memory considerations, a complex-valued version of ILU 183 with low fill-in is used. 184

185 *3.2. GCR* 

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

$$U_m = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$$
 and  $C_m = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m]$ 

190 satisfying

$$C_m = AU_m \text{ and } C_m^H C_m = I.$$
(4)

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  $C_m$ 

$$C_m^H \mathbf{r} = C_m^H (\mathbf{r}_0 - A U_m \mathbf{d}_m) = \mathbf{0}$$

which is equivalent to  $\mathbf{d}_m$  being the minimiser of  $\|\mathbf{r}\|$ . Similar to the Arnoldi iteration generating the Krylov vectors in GMRES, GCR uses a recursive procedure to generate the column vectors of  $C_m$  and  $U_m$ . At the *i*-th iteration,  $i \ge 1$ , set

$$\mathbf{u}_i \leftarrow \mathbf{r}_{i-1} \text{ and } \mathbf{c}_i \leftarrow A \mathbf{u}_i$$
 (5)

which are first orthogonalised against  $C_{i-1}$  if  $i \neq 1$ ,

$$\mathbf{u}_i \leftarrow \mathbf{u}_i - (C_{i-1}C_{i-1}^H)\mathbf{u}_i$$
 and  $\mathbf{c}_i \leftarrow \mathbf{c}_i - (C_{i-1}C_{i-1}^H)\mathbf{c}_i$ 

and then normalised by  $\|\mathbf{c}_i\|$ . The solution and residual vectors are then updated as

$$\mathbf{x}_i \leftarrow \mathbf{x}_{i-1} + \mathbf{u}_i(\mathbf{c}_i^H \mathbf{r}_{i-1}) \text{ and } \mathbf{r}_i \leftarrow \mathbf{r}_{i-1} - \mathbf{c}_i(\mathbf{c}_i^H \mathbf{r}_{i-1}).$$
 (6)

A preconditioned version of GCR is easily obtained by adding the matrix vector multiplication step for the preconditioning matrices as in GMRES.

199 3.3. GCRO

Nested Krylov subspace solvers wrap one Krylov solver outside another and use the inner solver to precondition the outer one. One of those nested 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

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

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

$$(I - C_{i-1}C_{i-1}^H)A\mathbf{u}_i = \mathbf{r}_{i-1}$$

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.

### 210 3.4. GCRO-DR and GCRO-DR-R

For nested Krylov subspace solvers such as GCRO, the inner loop gener-211 ates a Krylov subspace to approximately solve the equation, and the subspace 212 is discarded when exiting the inner loop. It would be advantageous to recycle 213 some information from the discarded subspace to aid the outer convergence. 214 One approach is to truncate based on the principle angle between the Krylov 215 vectors constructed during the inner loop and select the important ones to 216 augment the outer Krylov subspace. The criterion to select them is to check 217 how much worse the inner loop convergence would have been if the inner 218

loop has stopped before those vectors are formed. One solver based on this 219 idea is generalised conjugate residual with optimal truncation [21]. Another 220 approach is to select the interior eigenvectors (eigenvectors corresponding 221 to the smallest-in-magnitude eigenvalues) that can be computed during the 222 Arnoldi iteration from the inner loop and use them to augment the outer 223 Krylov subspace. The two most important solvers following this idea are 224 generalised minimal residual with deflated restarting [22] and GCRO-DR 225 [19]. Both are nested Krylov subspace solvers with the main difference being 226 that the former uses GMRES for the outer loop while the latter uses GCR. 227 Although the latter requires more memory, it has the advantage of recycling 228 eigenvectors both between restarts and between equations with different co-229 efficient matrices and/or forcing terms. Due to this flexibility, GCRO-DR is 230 used in this work. 231

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 GM-RES, an additional step to extract the approximate interior eigenvectors  $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$  of the matrix A is taken by first solving the eigenvalue problem

$$(H_m + h_{m+1,m}^2 H_m^{-H} \mathbf{e}_m \mathbf{e}_m^H) \mathbf{p}_i = \theta_i \mathbf{p}_i, \ i = 1, \dots, m$$
(8)

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

$$[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k] =: Y_k \leftarrow V_m P_k$$

where  $P_k = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k]$  correspond to the k smallest  $\theta_i$ . The matrices  $C_k$ and  $U_k$  are constructed from  $Y_k$  by setting

$$C_k \leftarrow V_{m+1}Q$$
 and  $U_k \leftarrow Y_k R^{-1}$ .

where [Q, R] is the QR-factorisation of  $\overline{H}_m P_k$ . 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  $(I - C_{k}C_{k}^{H})A$  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

$$\hat{V}_m = [U_k D_k, V_{m-k}], \ \hat{W}_{m+1} = [C_k, V_{m-k+1}], \ \bar{G}_m = \begin{bmatrix} D_k & B_{m-k} \\ 0 & \bar{H}_{m-k} \end{bmatrix}$$

which satisfy the generalised Arnoldi relation

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

where  $D_k = \text{diag}(\|\mathbf{u}_1\|^{-1}, \|\mathbf{u}_2\|^{-1}, \dots, \|\mathbf{u}_k\|^{-1})$  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  $\|\mathbf{r} - A\hat{V}_m\mathbf{d}_m\|$ , which, due to the Arnoldi relation, is equivalent to  $\|\hat{W}_{m+1}^H\mathbf{r} - \bar{G}_m\mathbf{d}_m\|$ . 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

$$C_k \leftarrow \hat{W}_{m+1}Q$$
 and  $U_k \leftarrow Y_k R^{-1}$ .

 $_{255}$  similar to eq. (9).

Different from GCRO, after  $C_k$  and  $U_k$  are formed, the solution and resid-256 ual vectors are not immediately updated in GCRO-DR. This is because the 257 least square problem after the inner Arnoldi iteration has included the basis 258  $U_k$  in the search subspace. The resulting algorithm is denoted as GCRO-259 DR(m,k) where m is the dimension of the Krylov subspace retained for 260 approximating the solution while k is the number of eigenvectors recycled. 261 The total number of vectors that needs to be stored is thus (m + k). The 262 deflated GMRES cycle is repeated using the most recent solution and resid-263 ual vectors along with the recycled eigenvectors until the stopping criterion 264 is met. 265

In GCRO-DR, the eigenvectors  $Y_k$ , computed from the Arnoldi itera-266 tions, are 'recycled' to improve/deflate the next subsequent cycle. Due to 267 the flexibility of GCRO, any set of vectors could be recycled and regularised 268 to form  $U_k$  and  $C_k$ , not only for the next restarted cycle, but also for solving 269 a different equation. Instead of performing a start-up cycle of m GMRES 270 iterations to get k approximate interior eigenvectors, those from a previous 271 solve can be recycled using the QR-factorisation to generate  $U_k$  and  $C_k$  that 272 satisfy the condition in eq. (4). We call this variant version that allows inter-273 equation recycling GCRO-DR-R. The algorithm for GCRO-DR-R is shown 274 in Appendix A. It should be pointed out that the original GCRO-DR algo-275 rithm introduced in [19] is also capable of recycling from one equation to 276 another. Here we denote this kind of inter-equation recycling technique with 277 an additional '-R' just for clarity. 278

The effectiveness of recycling the eigenvectors between equations strongly 270 depends on how good an approximation the eigenvectors from one equation 280 are for another. For varying right-hand sides, there are mainly three sce-281 narios: (i) identical coefficient matrices, (ii) diagonally shifted coefficient 282 matrices and (iii) similar but different coefficient matrices. For case (i), 283 GCRO-DR-R should be very effective as the spectral information of the co-284 efficient matrix does not change. For case (ii), GCRO-DR-R is also expected 285 to be quite effective. Although the eigenvectors are preserved despite the 286 shifted spectrum, the smallest eigenvectors for  $(A - \sigma_1 I)$  are not necessar-287 ily the smallest eigenvectors for  $(A - \sigma_2 I)$ . Thus the deflation may not be 288 as effective as in the first case. Case (iii) is the most general one. In [23], 289 GCRO is extended to solve a system of equations that have varying sym-290

#### ACCEPTED MANUSCRIPT

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.

#### 303 4. Results

The linear equation solvers outlined in the previous section are now ap-304 plied to three test cases to demonstrate their effectiveness in reducing the 305 memory requirements and in accelerating convergence. The governing equa-306 tions of the flow are solved using the DLR-TAU code and all linear sys-307 tems are obtained from this solver's discretisation scheme. For the two-308 dimensional aerofoil test case and the three-dimensional half wing-body case, 309 the Reynolds-averaged Navier–Stokes equations are used together with the 310 one equation turbulence model of Spalart–Allmaras. The inviscid fluxes of 311 the mean flow equations are discretised using the second-order central scheme 312 with scalar artificial dissipation, while the first-order Roe scheme is used for 313 the turbulence equation. Viscous fluxes follow the full gradient approach 314

#### ACCEPTED MANUSCRIPT

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.

# 322 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 freestream 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 330 the linear equation for a reduced frequency of 0.35 excited by the pitch mode. 331 As will be explained below, the resulting linear system at this frequency is 332 most difficult to solve. The convergence history for ten orders of magnitude 333 residual drop using different solvers is plotted in Fig. 1. The residual here and 334 in the following paragraphs is the normalised  $L_2$  norm of the preconditioned 335 residual, defined as  $Res = \|U^{-1}L^{-1}(\mathbf{b} - A\mathbf{x})\|/\|U^{-1}L^{-1}\mathbf{b}\|$ . As reference, 336 full GMRES is first used and it converges with 464 iterations. Restarted 337 GMRES is then run using 50, 100 and 200 Krylov vectors, respectively. At 338 least 100 vectors are needed to avoid convergence stagnation and GMRES 339



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 be optimal. Using more than 200 vectors does not further accelerate the convergence due to the increased cost in orthogonalisation. On the other hand, GCRO-DR(20,10), requiring only 30 vectors to be stored, converges 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, GCRO-DR(20,10) is used for the remaining computations of

#### ACCEPTED MANUSCRIPT

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 vectors 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 350 to harmonic excitation in pitch and plunge modes is computed with three 360 solvers: GMRES(200), GCRO-DR(20,10) and GCRO-DR-R(20,10). The re-361 duced frequency varies from 0.1 to 0.6 with an increment of 0.05. For each 362 frequency, the aerofoil is first excited using the pitch and then the plunge 363 mode. The order of solving the sequence of equations is such that the coef-364 ficient matrices and their complex-valued ILU(0) preconditioner are formed 365 as few times as possible and the coefficient matrices vary monotonically to 366 allow effective recycling between equations. A total of 22 linear solves are 367 performed. The convergence criterion is again a residual drop of ten orders 368 of magnitude. 369

As can be seen in Fig. 2, compared with GCRO-DR, GCRO-DR-R com-370 pletely avoids the initial phase of the slow convergence from the second lin-371 ear equation solve due to recycling. The effect however slows down and the 372 asymptotic convergence of the nested solvers is comparable, both outper-373 forming GMRES. The CPU time breakdown of each linear solve for all three 374 solvers for different modes excited at different frequencies is shown in Fig. 3. 375 Compared to GMRES(200), GCRO-DR(20,10) reduces the overall CPU time 376 by 64% and using recycling between equations speeds up another 15%. 377



Figure 2: Case 1. Convergence history of GMRES(200), GCRO-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.

The CPU time of GMRES for the frequency sweep in this case shows 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  $\sigma = 0.35i$  and a random initial guess for the eigenvector.

The first outer iteration is solved using GMRES with different restarts 387 and GCRO-DR, for which the convergence history is shown on the right of 388 Fig. 4. Restarting after every 300 vectors seems to be optimal for GMRES. 389 For GCRO-DR(20,10), the memory required is only one tenth of that of 390 GMRES, while the convergence is accelerated by over a factor of four. The 391 convergence criterion used here is ten orders of magnitude. In practice, since 392 only the outer iteration convergence is related to the actual convergence of 393 the eigenvalue problem, the inner iteration convergence could be relaxed to 394



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

achieve better overall performance. It is found that converging four orders of
magnitude during inner iterations is sufficiently efficient, thus is chosen for
the entire calculation.

For the outer eigenvalue problem, GMRES(300), GCRO-DR(20,10) and 398 GCRO-DR-R(20,10) are used. The CPU time of all three solvers for all nine 399 outer iterations is plotted on the left in Fig. 5 together with the convergence 400 of the outer eigenvalue problem shown on the right. Inter-equation recycling 401 is very effective for computing the eigenvalue with a speedup by a factor of 402 two, because the coefficient matrix does not change over the outer iterations. 403 The final converged eigenvalue corresponding to the critical mode is  $\lambda =$ 404 -0.014 + 0.354i. Note that all the results for this test case are computed 405 using one CPU only. 406

#### 407 4.2. Basis of reduced order model for an aircraft model

The three-dimensional model is a wide-body civil aircraft research con-408 figuration with a semi-span of about 30 m and an overall length of about 409 65 m. Flow conditions are a freestream Mach number of 0.85 at 1.0 deg 410 angle of attack. The computational mesh for the Euler CFD calculation 411 has 0.74 million grid points, equivalent to 3.7 million complex-valued un-412 knowns for the linear system. In terms of storage, the flow Jacobian matrix 413 for the second-order spatial discretisation has around 451 million non-zero 414 real-valued entries requiring around 3.3 GB of memory to be stored with 415 double precision. The complex-valued ILU(0) preconditioning matrix stored 416 in single precision requires another 3.3 GB of memory. 417

Using ten mode shapes of the structure (the first mode, dominant in wing 418 bending, as mapped to the CFD surfaces is illustrated on the left of Fig. 6), 419 ten eigenvalues are found using the Schur complement method. The Schur 420 complement method can efficiently compute eigenvalues and the associated 421 structural part of the right and left eigenvectors. The remaining task is to 422 find the fluid part of the eigenvectors using eqs. (1) and (2). Once both the 423 right and left eigenvectors of the coupled problems are found, a reduced order 424 model can be constructed [4]. 425

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 432 using more slows down the convergence due to the increased cost in orthog-433 onalisation. For GCRO-DR, there is only marginal variation in CPU time 434 for the three combinations of parameters used, and thus GCRO-DR(30,10)435 is chosen for all the remaining eigenvector solves. Replacing GMRES(100)436 with GCRO-DR(30,10) reduces the total memory requirement for the linear 437 solve (flow Jacobian matrix, ILU preconditioning matrix and Krylov vectors 438 combined) from 18.2 GB to 11.4 GB, with the difference being the 60 fewer 439 vectors needed to be stored, while the CPU time is reduced by a factor of 440 three. Note that the results for this case are computed using one CPU only. 441 The CPU time breakdown of the 20 eigenvector solves using GMRES(100) 442 and GCRO-DR(30,10) is shown in Fig. 7. Using deflated restarting speeds up 443



Figure 7: Case 2. CPU time for solving all eigenvectors using the three different solvers.

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].

#### 462 4.3. Frequency response for a pre-buffet half wing-body configuration

The frequency response for a generic half wing-body model is now com-463 puted. The freestream Mach number is 0.8 while the Reynolds number based 464 on the aerodynamic mean chord is 3.5 million. Fully turbulent flow is as-465 sumed. Two angles of attack at 2.8 deg and 3.0 deg are considered, with the 466 higher angle close to buffet onset. Similar to the aerofoil-near-buffet case, the 467 higher angle of attack case exhibits significant stiffness in the linear system. 468 The steady state solutions are converged by ten orders of magnitude, which is 469 deemed sufficient by monitoring the lift, drag and moment coefficients. The 470 surface pressure distribution for the steady state solution at 2.8 deg angle 471 of attack is shown on the left in Fig. 8. To perform linearised aerodynamic 472 analysis, four synthetic mode shapes (the first of which is shown on the right 473 in Fig. 8) are used to harmonically excite the system at a range of reduced 474 frequencies 475

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 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.

<sup>482</sup> billion complex-valued entries in single precision. For comparison, ILU(1)
<sup>483</sup> would require 48 GB of memory. All simulations are run on 144 cores with 2
<sup>484</sup> GB per core. The ILU(0) preconditioning matrix is computed locally without
<sup>485</sup> parallel communication. It will have deteriorated performance with increased
<sup>486</sup> number of cores as reported in [27]. The convergence tolerance is set to ten
<sup>487</sup> 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

<sup>499</sup> a factor of two. In addition, the lower threshold of the number of Krylov

k		m+k									
		60	80	100	200	300	400	500			
nding coefficient	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			
ble	0.9	NA	NA	10	10	15	20	15			

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

k		m+k							
		100	150	200	300	400	500		
0.0	10	15	30	30	30	20	20		
0.2	10	15	25	30	30	15	20	1	
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		
	k 0.0 0.2 0.4 0.6 0.8 0.9	80           0.0         10           0.2         10           0.4         10           0.6         NA           0.8         NA           0.9         NA	80         100           0.0         10         15           0.2         10         15           0.4         10         15           0.6         NA         15           0.8         NA         15           0.9         NA         NA	80         100         150           0.0         10         15         30           0.2         10         15         25           0.4         10         15         25           0.6         NA         15         25           0.8         NA         15         20           0.9         NA         NA         25	m + k           80         100         150         200           0.0         10         15         30         30           0.2         10         15         25         30           0.4         10         15         25         30           0.6         NA         15         25         30           0.8         NA         15         20         20           0.9         NA         NA         25         25	m + k           80         100         150         200         300           0.0         10         15         30         30         30           0.2         10         15         25         30         30           0.4         10         15         25         30         30           0.6         NA         15         25         30         30           0.8         NA         15         20         20         30           0.9         NA         NA         25         25         35	m + k           80         100         150         200         300         400           0.0         10         15         30         30         30         20           0.2         10         15         25         30         30         15           0.4         10         15         25         30         30         15           0.6         NA         15         25         30         30         15           0.8         NA         15         20         20         30         20           0.9         NA         NA         25         25         35         15	m+k           80         100         150         200         300         400         500           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	

vectors for reasonable convergence is reduced substantially, from 300 to below 500 100. At 3.0 deg angle of attack, GCRO-DR accelerates the convergence much 501 more significantly, by a peak-to-peak ratio of 4.5. For GMRES to converge in 502 reasonable time, at least 300 vectors are required, while 150 Krylov vectors 503 seem sufficient otherwise. Overall, the convergence is not very sensitive to 504 the blending coefficient as long as it does not exceed 0.9, beyond which the 505 stiffness of the linear system increases drastically. For a blending coefficient 506 of 1, none of the linear solvers is able to converge with 500 Krylov vectors, 507 neither with ILU(0) nor ILU(1). The extreme stiffness has been reported 508 previously but a plausible elucidation is still missing. 500

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 fastest convergence. As can be seen, there does not seem to be a general rule for choosing k, except some guiding principles such as k should be around <sup>517</sup> 10% of m + k and below a maximum of 30 in general. Some attempts have <sup>518</sup> been made to adaptively determine the number of recycled vectors in [22], <sup>519</sup> however, as mentioned therein, even the simplest strategy does not seem to <sup>520</sup> apply to all the cases tested. One strategy, which seems to be quite robust <sup>521</sup> in our numerical experiments, is to keep the ratio k/(m + k) at around 10%, <sup>522</sup> and increase m and k proportionally whenever convergence stall is detected, <sup>523</sup> until the memory limit is reached.

Once the parameter study is done, GMRES(300) and GCRO-DR(125,25), 524 both with blending coefficient of 0.6, are used to perform a frequency response 525 study at 3.0 deg angle of attack. The reduced frequency ranges from 0 to 1. 526 The deflated linear solver is run with and without inter-equation recycling, 527 denoted by GCRO-DR-R and GCRO-DR respectively. The CPU time of 528 all solves using three different solvers are shown in Fig. 10. The goal being 529 the shortest overall turnover time, inter-equation recycling is applied to a 530 sequence of equations with different forcing terms, while recycling between 531 different frequencies is not considered. This sequencing strategy results in 532 eleven groups of equations being solved simultaneously, each group with one 533 frequency and varying forcing terms. Other sequencing strategies can be 534 considered as well. Although using 150 fewer Krylov vectors, GCRO-DR 535 speeds up the overall convergence by a factor of over three. At no extra cost, 536 GCRO-DR-R further accelerates convergence by over 20%. 537

## 538 5. Conclusions

<sup>539</sup> In this paper, generalised conjugate residual solver with deflated restart-<sup>540</sup> ing 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

#### ACCEPTED MANUSCRIPT

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.

#### 557 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

 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 flow, Proc. IFASD, 2009.
- 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-Krylov– Schwarz 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.

ACERTICAL

Algorithm 1 GCRO-DR-R $(A, \mathbf{x}_0, \mathbf{b}, m, k, \text{iter\_max}, \text{tol}, U, L, \text{Recycle})$ 1:  $\mathbf{r} \leftarrow U^{-1}(L^{-1}(\mathbf{b} - A\mathbf{x}_0))$ ; iter  $\leftarrow 0$ 2: if (Recycle=True) then % QR factorisation of recycled  $Y_k$  $[Q, R] \leftarrow QR(Y_k); C_k \leftarrow Q; U_k \leftarrow Y_k R^{-1}$ 3:  $\mathbf{x} \leftarrow \mathbf{x}_0 + U_k C_k^H \mathbf{r}; \ \mathbf{r} \leftarrow \mathbf{r} - C_k C_k^H \mathbf{r}$ 4: 5: else % m Arnoldi iterations to calculate  $Y_k$  $\beta \leftarrow \|\mathbf{r}\|; \mathbf{v}_1 \leftarrow \mathbf{r}/\beta$ 6: for i = 1, 2, ..., m do 7:  $\mathbf{v}_{i+1} \leftarrow U^{-1}(L^{-1}(A\mathbf{v}_i)); \text{ iter } \leftarrow \text{ iter } + 1$ 8: for  $j = 1, 2, \ldots, i$  do % Orthogonalisation 9:  $h_{i,i} \leftarrow \mathbf{v}_i^H \mathbf{v}_{i+1}; \ \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1} - h_{i,i} \mathbf{v}_i \checkmark$ 10:  $h_{i+1,i} \leftarrow \|\mathbf{v}_{i+1}\|; \ \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1}/h_{i+1,i}$ 11: Solve for  $\mathbf{d}_m$  that minimises  $J(\mathbf{d}_m) := \|\beta \mathbf{e}_1\|$  $\bar{H}_m \mathbf{d}_m \|$ 12: $\mathbf{x} \leftarrow \mathbf{x}_0 + V_m \mathbf{d}_m; \ \mathbf{r} \leftarrow V_{m+1}(\beta \mathbf{e}_1 - H_m \mathbf{d}_m)$ 13:Compute k eigenvectors  $\mathbf{p}_i$  in  $(H_m + h_{m+1,m}^2 H_m^{-H} \mathbf{e}_m \mathbf{e}_m^H) \mathbf{p}_i = \theta_i \mathbf{p}_i$ 14: $[Q, R] \leftarrow QR(\bar{H}_m P_k); \ C_k \leftarrow V_{m+1}Q; \ Y_k \leftarrow V_m P_k; \ U_k \leftarrow Y_k R^{-1}$ 15:16:  $\mathbf{v}_1 \leftarrow \mathbf{r} / \|\mathbf{r}\|$ 17: for  $i = 1, 2, \ldots, m - k$  do  $\mathbf{v}_{i+1} \leftarrow (I - C_k C_k^H) (U^{-1}(L^{-1}(A\mathbf{v}_i))); \text{ iter } \leftarrow \text{ iter } + 1$ 18:for j = 1, 2, ..., i do % Orthogonalisation  $h_{j,i} \leftarrow \mathbf{v}_j^H \mathbf{v}_{i+1}; \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1} - h_{j,i} \mathbf{v}_j$ 19:20: $h_{i+1,i} \leftarrow \|\mathbf{v}_{i+1}\|; \mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1}/h_{i+1,i}$ 21: 22:  $D_k \leftarrow \text{Diag}(\|\mathbf{u}_1\|^{-1}, \|\mathbf{u}_2\|^{-1}, \dots, \|\mathbf{u}_k\|^{-1})$ 23:  $B_k \leftarrow C_k^H(U^{-1}(L^{-1}(AV_{m-k}))); \ \bar{G}_m \leftarrow [D_k, B_k; 0, \bar{H}_{m-k}]$ 24:  $\hat{V}_m \leftarrow [U_k D_k, V_{m-k}]; \ \hat{W}_{m+1} \leftarrow [C_k, V_{m-k+1}]$ 25: Solve for  $\mathbf{d}_m$  that minimises  $J(\mathbf{d}_m) := \|\hat{W}_{m+1}^H \mathbf{r} - \bar{G}_m \mathbf{d}_m\|$ 26:  $\mathbf{x} \leftarrow \mathbf{x} + \hat{V}_m \mathbf{d}_m; \mathbf{r} \leftarrow \mathbf{r} - \hat{W}_{m+1} \bar{G}_m \mathbf{d}_m$ 27: Compute k eigenvectors  $\mathbf{p}_i$  in  $\bar{G}_m^H \bar{G}_m \mathbf{p}_i = \theta_i \bar{G}_m^H \hat{W}_{m+1}^H \hat{V}_m \mathbf{p}_i$ 28:  $[Q, R] \leftarrow QR(\bar{G}_m P_k); C_k \leftarrow \hat{W}_{m+1}Q; Y_k \leftarrow \hat{V}_m P_k; U_k \leftarrow Y_k R^{-1}$ 29: if  $(J(\mathbf{d}_m) < \text{tol or iter} > \text{iter\_max})$  then 30:  $Y_k = U_k$ ; Terminate programme. 31: Goto line 16