KRYLOV RECYCLING TECHNIQUES FOR UNSTEADY SIMULATION OF TURBULENT AERODYNAMIC FLOWS

Kaveh Mohamed*, Siva Nadarajah*, Marius Paraschivoiu**
*McGill University, Montreal, Canada, **Concordia University, Montreal, Canada

Keywords: *Iterative Solvers, Krylov Subspace Recycling, CFD, Convergence Acceleration*

Abstract

The performance of the Krylov subspace recycling methods to accelerate the convergence of large scale unsteady aerodynamic simulations are investigated in this paper. The generalized conjugate residual technique with deflated restarting (GCRO-DR) is adopted for recycling Krylov vectors; its performance is then compared with that of the generalized minimal residual (GMRES) solver for several unsteady aerodynamic simulations. The main conclusion of this study is that no significant gain in the CPU time is obtained as a result of the Krylov vector recycling. In particular, for large scale problems, the recycling leads to more computational time spent in the linear solver.

1 Introduction

Convergence accelerating techniques are required to make the large-scale unsteady aerodynamic simulations computationally efficient. The implicit discretization of the Navier-Stokes equations on a real engineering mesh geometry and the subsequent linearization of the non-linear fluxes lead to a large and sparse linear system of equations. This system is usually solved using a Krylov subspace iterative techniques such as GMRES [13]. For unsteady problems, a new linear system is formed and should be solved with the required accuracy at each time step. This makes unsteady problems computationally expensive. Krylov subspace recycling methods are among the techniques that aim to decrease the number of iterations in the linear solver and hence to reduce the computational cost.

The main idea of the recycling techniques is to keep some of the search vectors calculated from the previous cycle or time step for the calculation of the solution at the next cycle or time step. The GMRES method, which is popular for solving large nonsymmetric linear systems of equations, is usually used with restarting in order to reduce the storage and computational costs. This tends to slow down the convergence. With a Krylov recycling technique, we try to save some important vectors from the previous cycle at the time of the restart. Furthermore, in an unsteady problem, some important Krylov vectors can be retained from the previous time step and used for the calculation in the next time step. Different recycling techniques are distinguished by the specific approaches they adopt to choose the recycled subspace and by the scheme they use to incorporate the recycled subspace in the next cycle or time step.

The available work in the literature on this topic mainly focuses on the solution of model partial differential equations for simple geometries [12, 2, 9]. This work aims to implement recycling techniques to accelerate the numerical solution of the unsteady compressible Navier-Stokes equations for several aerodynamic problems. We consider the problems where the convergence of the original restarted solver (GMRES) is already impaired due to some numerical and/or physical features such as the mesh skewness, separated flow regions and the existence of various time-scales in the flow field. We compare the convergence behavior of the GCRO-DR solver with that of the original GMRES method.

In the next section, the numerical discretization schemes for the flow equations as well as the algorithms for the linear iterative solvers with and without recycling are described. In section 3, the convergence acceleration efficiency of the recycling technique is investigated in four test problems. Conclusions are given in section 3.6.

2 Methodology

2.1 Flow Solver

The viscous compressible fluid flow is modeled with the Navier-Stokes (NS) equations. The spatial discretization of the flow equations is achieved via a mixed finite-volume-finite-element method [5] on unstructured tetrahedral elements. Convective fluxes are calculated over median dual control volumes with a second order MUSCL Roe scheme whereas viscous fluxes are discretized using linear finite-element basis functions. A second-order implicit time integration scheme is adopted. Newton's method is implemented to linearize implicit convective and viscous fluxes. As the calculation of the second order convective flux Jacobians is difficult, only their first order approximations are computed. To achieve the required spatial accuracy for unsteady problems, several Newton iterations are performed per time step [7]. The Generalized Minimal residual (GMRES) method [13] is originally used to solve the resulting linearized system of equation.

The effect of turbulence is modeled via an eddy viscosity approach where the one-equation Spalart-Allmaras (SA) turbulence model [14] is used to calculate the turbulence viscosity. Molecular viscosity, μ, and thermal conductivity, k, are replaced by $\mu+\mu_t$ and $k+k_t$, respectively, in NS equations. Turbulent thermal conductivity is calculated from turbulent viscosity using the assumption of a constant turbulent Prandtl number of $\mathrm{Pr}_t = \frac{c_p\mu_t}{k_t} = 0.9$. Besides Reynolds Averaged Navier-Stokes (RANS) methodology, SA based Detached-eddy simulation (DES) technique [16, 15] is also implemented to be used for the problems involving the simulation of massively separated flows. The SA equation is discretized with the same scheme as the one used for NS equations. The flow and turbulence model equations are loosely coupled during the flow calculation, i.e. first NS equations are solved then the turbulence viscosity is updated by solving the SA equation with recently updated flow variables. The linear system of discretized SA equations are solved via GMRES.

The treatment of moving and deformable meshes are accomplished by the arbitrary Lagrangian Eulerian (ALE) kinematical description of the fluid domain [3]. The time integration of fluxes and Jacobians are performed over an intermediate mesh position satisfying geometric conservation laws developed by Lesoinne et al. [8], Farhat et al. [4] and Nkonga et al. [11].

NS and SA discretized systems of equations are preconditioned using a local Jacobi method to accelerate the convergence of the iterative solver. All numerical algorithms are parallelized with the message passing standard (MPI).

2.2 Krylov Subspace Methods and Recycling

2.2.1 GMRES Method

GMRES is one of the more popular iterative algorithms for solving linear systems of equations, which is based on projection methods [13]. A projection technique looks for an approximate solution to the linear system of equations given by

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1}$$

in a *search subspace* denoted by $\mathcal{K}$. Here, $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. If the dimension of the search subspace is $m \leq n$, then m conditions are required in order to uniquely determine $\mathbf{x} \in \mathcal{K}$. These conditions are usually given as m orthogonality constraints, such that the residual vector $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$ is normal to another m-dimensional subspace called *subspace of constraints* and denoted by $\mathcal{L}_m$. In GMRES, the search subspace is the Krylov subspace,

$$\mathcal{K}_m(\mathbf{A},\mathbf{r}_0) = \operatorname{span}\left\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \cdots, \mathbf{A}^{m-1}\mathbf{r}_0\right\}, \tag{2}$$

where $\mathbf{r}_0$ is the initial residual vector calculated based on the initial guess to the solution, $\mathbf{x}_0$, i.e. $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; the subspace of constraint is given by $\mathcal{L}_m = \mathbf{A}\mathcal{K}_m$.

The heart of GMRES iterations is the Arnoldi orthogonalization method in which an orthogonal basis is built for the Krylov subspace $\mathcal{K}_m$. The basis vectors have the following property,

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\bar{\mathbf{H}}_m. \tag{3}$$

The Arnoldi vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ are the normalized orthogonal basis of $\mathcal{K}_m$ while the last Arnoldi vector $\mathbf{v}_{m+1}$ is orthogonal to $\mathcal{K}_m$. They constitute the columns of the $n \times (m+1)$ matrix $\mathbf{V}_{m+1}$. The $(m+1) \times m$ matrix $\bar{\mathbf{H}}_m$ has an upper Hessenberg structure. The matrix obtained by deleting the last row of $\bar{\mathbf{H}}_m$ is denoted by $\mathbf{H}_m$. No orthogonal basis vector for the subspace of constraints, $\mathcal{L}_m$, is explicitly built or stored in the GMRES algorithm.

In practical engineering problems, the GMRES method is usually used with a restarting technique. The number of degrees of freedom in the problem of interest represents the size of the Arnoldi vectors, n. Many engineering CFD problems involve millions of degrees of freedom. This limits the maximum number of Arnoldi vectors that can be saved and used in the orthogonalization process. Hence the dimension of the Krylov subspace can not usually exceed 100 due to the storage limitations, i.e. $m_{\max} \leq 100$. If the residual remains greater than the required tolerance after $m_{\max}$ iterations, GMRES will be restarted, whereby all previously calculated Arnoldi vectors will be erased and a new Krylov subspace will be built based on the current residual. The period between two subsequent restarts is called a cycle. The consequent loss of orthogonality between Arnoldi vectors in the new and previous cycles degrades GMRES convergence.

2.2.2 GCRO-DR Recycling Method

The convergence of the restarted GMRES can be improved by recycling selected Krylov vectors between two cycles at the time of the restart. Recycling can also be performed while solving a sequence of linear systems of equations, where the Krylov vectors will be recycled between two subsequent systems. For these purposes, we adopted the generalized conjugate residual solver with deflated restarting (GCRO-DR) developed by Parks et al. [12]. This solver is based on Morgan's GMRES with deflated restarting (GMRES-DR) [9] and de Sturler's generalized conjugate residual with optimal truncation (GCROT) [2] recycling iterative schemes. It is designed to solve a series of linear systems of equations such as system (1), where the coefficient matrix $\mathbf{A}$ and the right hand side (RHS) vector b can be changed arbitrarily from one system to another. The only restriction is that the change of the matrix $\mathbf{A}$ between two consecutive systems should be gradual.

GCRO-DR recycles matrices $\mathbf{U}_k$ and $\mathbf{C}_k$ between two systems or cycles. Each matrix contains k n-dimensional recycled vectors as its columns. The following relations hold between $\mathbf{U}_k$ and $\mathbf{C}_k$,

$$\begin{aligned} \mathbf{A}\mathbf{U}_k &= \mathbf{C}_k, && (4)\\ \mathbf{C}_k^H\mathbf{C}_k &= \mathbf{I}_k, && (5) \end{aligned}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix, and superscript H represents the transpose conjugate of the matrix. The calculation starts by finding the solution of the system (1) via a projection step onto the subspace $\mathbf{U}_k$ and orthogonal to $\mathbf{C}_k$, which leads to,

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_0 + \mathbf{U}_k\mathbf{C}_k^H\mathbf{r}_0, && (6)\\ \mathbf{r} &= \mathbf{r}_0 - \mathbf{C}_k\mathbf{C}_k^H\mathbf{r}_0. && (7) \end{aligned}$$

In the next step, $m-k$ Arnoldi iterations are performed in order to construct $m-k$ orthogonal bases of the Krylov subspace starting with the new residual $\mathbf{r}$ in (7). Arnoldi vectors are built such that they maintain orthogonality to the subspace spanned by columns of $\mathbf{C}_k$. The following relation is satisfied,

$$(\mathbf{I}_n - \mathbf{C}_k\mathbf{C}_k^H)\mathbf{A}\mathbf{V}_{m-k} = \mathbf{V}_{m-k+1}\bar{\mathbf{H}}_{m-k}. \tag{8}$$

Equation (8) can be recast into the original form (3) using the definition, $\mathbf{B}_k = \mathbf{C}_k^H \mathbf{A}\mathbf{V}_{m-k}$,

$$\mathbf{A}\left[\begin{array}{cc} \mathbf{U}_k & \mathbf{V}_{m-k} \end{array}\right] = \left[\begin{array}{cc} \mathbf{C}_k & \mathbf{V}_{m-k+1} \end{array}\right] \left[\begin{array}{cc} \mathbf{I}_k & \mathbf{B}_k \\ \mathbf{0} & \bar{\mathbf{H}}_{m-k} \end{array}\right]. \tag{9}$$

New matrices are defined to further simplify equation (9),

$$\begin{aligned} \tilde{\mathbf{U}}_k &= \mathbf{U}_k \mathbf{D}_k \\ \hat{\mathbf{V}}_m &= \left[\begin{array}{cc} \tilde{\mathbf{U}}_k & \mathbf{V}_{m-k} \end{array}\right], \\ \hat{\mathbf{W}}_{m+1} &= \left[\begin{array}{cc} \mathbf{C}_k & \mathbf{V}_{m-k+1} \end{array}\right], \\ \bar{\mathbf{G}}_m &= \left[\begin{array}{cc} \mathbf{D}_k & \mathbf{B}_k \\ \mathbf{0} & \bar{\mathbf{H}}_{m-k} \end{array}\right], \end{aligned}$$

where the diagonal matrix $\mathbf{D}_k$ is introduced to normalize the column vectors of $\mathbf{U}_k$. The final form of the Arnoldi relation that incorporates the recycled vectors is given by,

$$\mathbf{A}\hat{\mathbf{V}}_m = \hat{\mathbf{W}}_{m+1}\mathbf{G}_m. \tag{10}$$

The solution to the linear system of equations (1) is then obtained with a projection step onto the range of $\hat{\mathbf{V}}_m$ and orthogonal to the range of $\mathbf{A}\hat{\mathbf{V}}_m$.

In GCRO-DR, the recycled vectors $\mathbf{U}_k$ are selected from the span of the approximate eigenvectors associated with the smallest eigenvalues of $\mathbf{A}$ [12]. In many problems, eigenvalues of smallest magnitude are responsible for the slow convergence. If an eigenvector exists in the subspace spanned by the Krylov vectors, then its corresponding eigenvalue will be deflated or removed during the projection iterations. The idea is to recycle the smallest existing eigenvectors to the next cycle for deflation purposes.

Harmonic Ritz vectors associated with the smallest harmonic Ritz values of $\mathbf{A}$ are accurate approximations to the small eigenvectors of $\mathbf{A}$. Morgan et al. [10] developed the algorithm for the computation of the corresponding Ritz vectors. At the end of the first cycle, k smallest eigenvectors of the following problem are calculated and stored in columns of matrix $\mathbf{T}_k$,

$$(\mathbf{H}_m + h_{m+1,m}^2 \mathbf{H}_m^{-H} \mathbf{e}_m \mathbf{e}_m^H)\mathbf{r} = \alpha \mathbf{r}, \tag{11}$$

where α and $\mathbf{r}$ are an eigenvalue and its corresponding eigenvector, respectively. The subspace of k smallest eigenvectors of $\mathbf{A}$ is then approximated by the range of $\mathbf{V}_m\mathbf{T}_k$. If Arnoldi iterations are performed with previously recycled vectors, as represented by equation (10), then the problem (11) will be replaced by,

$$\bar{\mathbf{G}}_m^H \bar{\mathbf{G}}_m \mathbf{r} = \alpha \bar{\mathbf{G}}_m{}^H \hat{\mathbf{W}}_{m+1}^H \hat{\mathbf{V}}_m \mathbf{r}. \tag{12}$$

Similarly, the subspace of k smallest eigenvectors of $\mathbf{A}$ is approximated by the range of $\hat{\mathbf{V}}_m\mathbf{T}_k$. The calculation details and the pseudo-code for the GCRO-DR solver can be found in reference [12].

3 Results

3.1 Overview

The convergence-acceleration efficiency of the GCRO-DR was studied for five different problems. All test cases except the first one were built based on the flow solver described in section 2.1. The case 1 was one of the test problems originally investigated by Parks et al. [12]. The validation of our implementation was the main purpose to repeat the same problem herein. Furthermore, it served as a comparison basis where the GCRO-DR solver exhibited the best performance.

In the following subsections, the GMRES solver is specified with the maximum size of its Krylov subspace, e.g. GMRES(m) indicates that the maximum number of Arnoldi vectors created and stored in the orthogonalization process is equal to m. In other words, if the norm of the residual is greater than the given tolerance at the end of the mth iteration, GMRES will restart, whereby all previously computed Arnoldi vectors will be erased and a new orthogonalization step will begin. The GCRO-DR solver requires two specification parameters: the maximum size of the Krylov subspace, m, and the size of the recycled subspace, k. At the end of each cycle, before the restart, the Krylov subspace of GCRO-DR(m,k) contains k recycled vectors coming from the previous cycle and $m-k$ Arnoldi vectors built in the current cycle.

3.2 Convection-Diffusion

The linear two-dimensional convection-diffusion partial differential equation,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + c\frac{\partial u}{\partial x} = 0, \qquad (13)$$

was solved over $(0,1) \times (0,1)$ unit square domain with the following boundary conditions,

$$\begin{aligned} u &= 0 \quad \text{on x = 0 \& y = 0,} \\ u &= 1 \quad \text{on x = 1 \& y = 1.} \end{aligned}$$

A central finite difference scheme was used to discretize equation (13) on a uniform mesh with an edge length of $1/41$. The resulting system of equations had 1600 degrees of freedom, i.e. $n = 1600$. The convective velocity was fixed at $c = 40$.

The solution to this problem was computed by reducing the second norm of the relative residual up to eight orders of magnitude using GMRES(25) and GCRO-DR(25,10) solvers. The convergence plots are depicted in figure 1. The GCRO-DR solver was run twice for the same problem. The projection step in the second run began with 10 recycled vectors calculated at the end of the first run. GMRES converged after 267 iterations with 10 restarts while the first and the second runs of GCRO-DR required 112 and 82 iterations, respectively, to reach to the specified residual reduction. This was equal to 7 and 6 restarts for the first and the second runs. The resulting speed-ups in terms of the iteration count as well as the CPU time are compared in table 1, where nit and CPU stand for the number of iterations and the CPU time respectively; the first and the second runs are denoted by GCRO-DR and GCRO-DR+. Only the CPU time spent to solve the linear system of equations was reported. The gain in the CPU time was less than the corresponding iteration reduction due to the additional calculations that was performed in the GCRO-DR solver for eigenvalue problems.

The distribution of 25 smallest harmonic Ritz values calculated at each restart (cycle) of the first run of GCRO-DR(25,10) are shown in figure 2. Since the coefficient matrix was real, the complex Ritz values appeared as pairs of complex conjugates. The Ritz value distribution changed significantly from one cycle to another, specially in the area close to the origin (0,0). This indicated the deflation of the recycled vectors in the course of the projection steps.

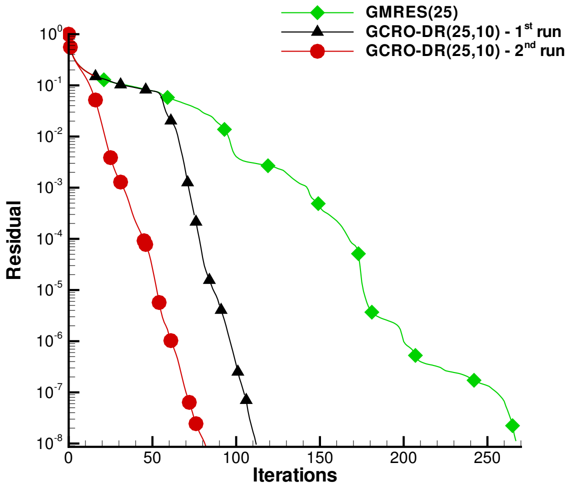


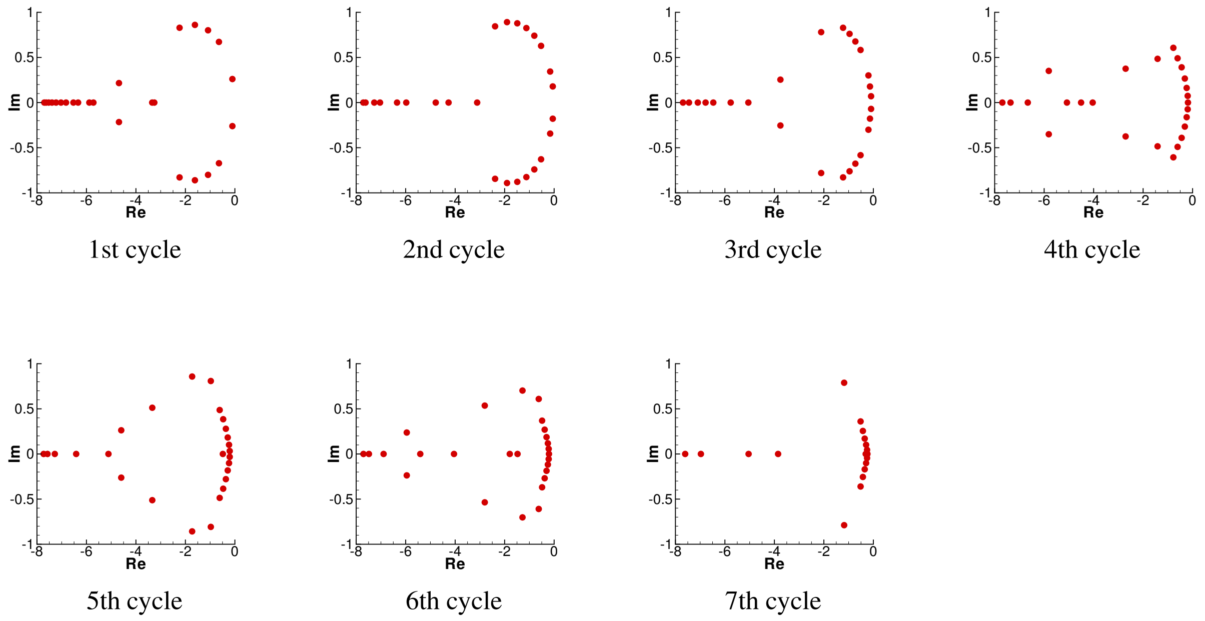
Fig. 1 Residual norm versus iteration number for linear convection-diffusion problem with $c = 40$ and $h = 1/41$.

3.3 Flat Plate Boundary Layer

The unsteady turbulent boundary layer over a flat plate of length l was simulated using the RANS methodology. The mean freestream Mach and Reynolds numbers were 0.4 and 4.82×10^6, respectively. The flow unsteadiness was gen-

Table 1 Comparison between the performance of GMRES(25) and GCRO-DR(25,10) solvers for convection-diffusion problem with $c = 40$ and $h = 1/41$.

solver	nit	CPU	speed-up%	
			nit	CPU
GMRES	267	0.48 sec.	0	0
GCRO-DR	112	0.34 sec.	58.0	29.2
GCRO-DR+	82	0.28 sec.	69.3	41.7



1st cycle

2nd cycle

3rd cycle

4th cycle

5th cycle

6th cycle

7th cycle

Fig. 2 Distribution of the real and imaginary parts of the 25 smallest Ritz values for the linear convection-diffusion problem at various restarts in the first run of GCRO-DR(25,10).

erated by superimposing an unsteady component to the mean freestream velocity, U_∞. The unsteady component consisted of harmonic oscillations with an amplitude of $0.05U_\infty$ and a period of $\frac{l}{2U_\infty}$. The implicit simulation time-step was equal to $\frac{0.015l}{U_\infty}$, which corresponded to a CFL (Courant-Friedrichs-Lewy) number of 7000. The 3D unstructured tetrahedral mesh contained $47,831$ nodes and $256,458$ elements resulting in a linear system of equations with over $239,000$ degrees of freedom. The boundary layer was resolved through highly skew tetrahedral elements with aspect ratios up to $100,000$ satisfying $y^+ \approx 1$ condition in the law-of-wall units. The presence of highly skew elements, in general, degrades the convergence of iterative solvers, and makes this problem a worthy candidate to examine the performance of the recycling scheme.

The resulting linear system of equations was solved with three different schemes where the norm of the residual vector was decreased up to four orders of magnitude at each time-step. The first method involved the GMRES(25) solver while in the second and third schemes the GCRO-DR(25,10) solver was used. In the second method, the Krylov vectors were only recycled between consecutive cycles within each time-step, whereas the recycling was performed, in the third scheme, not only within each time-step but also between successive time-steps. The third scheme will be denoted by GCRO-DR(m,k)+ hereafter.

The residual reductions within a time-step at $t = \frac{l}{2U_\infty}$ are depicted in figure 3 and the corresponding speed-ups are listed in table 2. The specified residual reduction was achieved after 16, 21, and 22 restarts for GMRES, GCRO-DR and GCRO-DR+ solvers, respectively. Recycling the smallest Ritz vectors between time-steps in GCRO-DR+ led to 6% more iterations and a 36% increase in the CPU time spent to solve the linear system. Although GCRO-DR reduced the number of iterations by 22.7%, the gain in the CPU time was very modest; only a 1.6% decrease was observed. This was due to the extra computations that should be performed at each restart in order to solve eigenvalue problems. Hence, the GCRO-DR recycling technique will be efficient if it can lead to a significant iteration reduction in a prob-

Table 2 Comparison between the performance of GMRES(25), GCRO-DR(25,10), and GCRO-DR(25,10)+ schemes for unsteady flat plate problem at $t = \frac{l}{2U_\infty}$.

solver	nit	CPU	speed-up%	
			nit	CPU
GMRES	410	427 sec.	0	0
GCRO-DR	317	421 sec.	22.7	1.4
GCRO-DR+	435	581 sec.	-6.1	-36.1

lem.

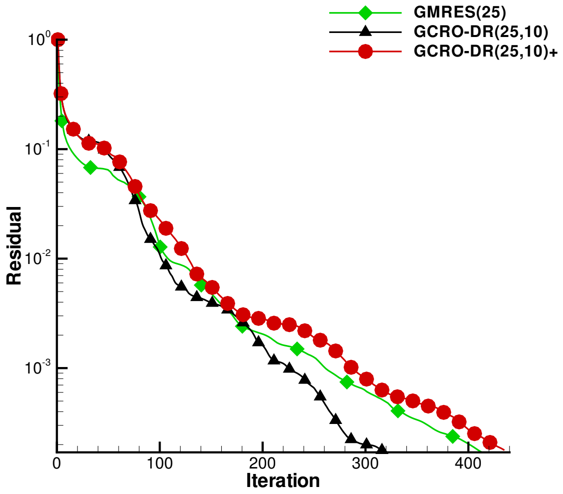


Fig. 3 Residual norm versus iteration number for unsteady flat plate problem at $t = \frac{l}{2U_\infty}$.

3.4 Flow Around Circular Cylinder

The recycling methods were applied to the problem of von Kármán vortex shedding off a circular cylinder. The flow was laminar with Mach and Reynolds numbers of 0.2 and 100, respectively. The unstructured tetrahedral mesh contained 572,599 nodes and 3,258,631 elements of high quality; no skew tetrahedron was present. The mesh was decomposed into 10 subdomains using METIS [6] for parallel computation on a distributed memory machine. The simulation time-step was equal to $0.15\frac{D}{U_\infty}$ representing a CFL number of 70. Our objective was to investigate the performance of the recycling techniques for unsteady problems involving separated flow regions.

The linear system of equations followed from the discretization scheme had 2,862,995 degrees of freedom. The solution was found via GMRES(25), GCRO-DR(25,10), and GCRO-DR(25,10)+ solvers by reducing the residual up to seven orders of magnitude at each time step. The convergence plot at $t = 200\frac{D}{U_\infty}$ is shown in figure 4 and the resulting speed-ups are compared in table 3. All three schemes exhibited more or less the same convergence behavior, with GCRO-DR(25,10) requiring one percent more iterations to produce the desired residual reduction. However, recycling techniques took almost twice as much CPU time as the amount spent in GMRES iterations. Thus, for this problem recycling failed to provide any convergence acceleration.

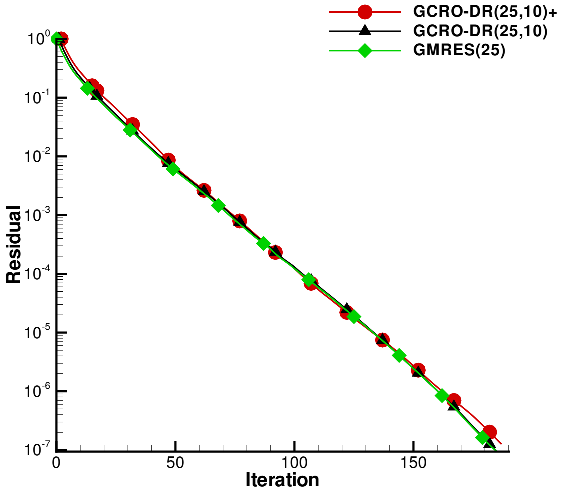
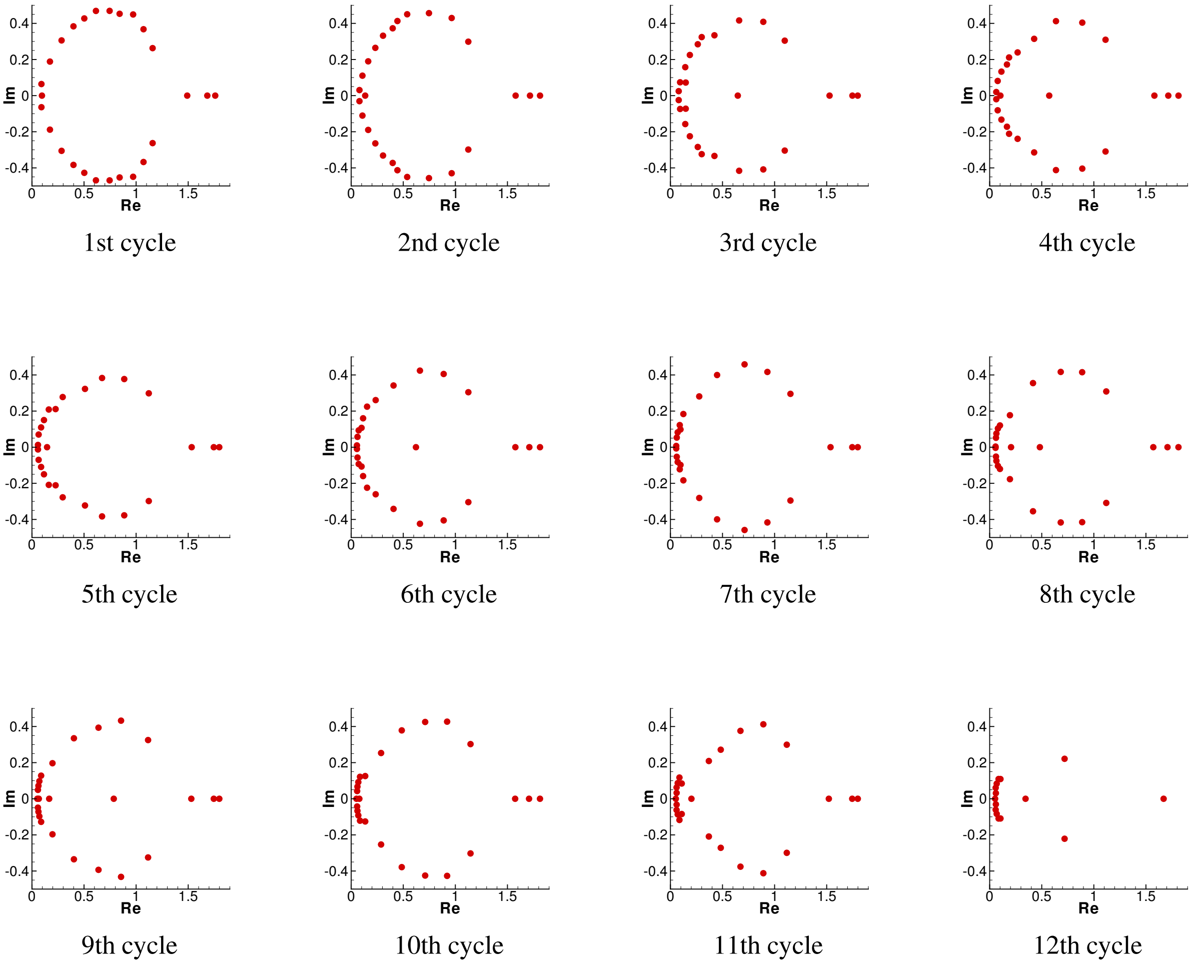


Fig. 4 Residual norm versus iteration number for laminar unsteady flow around a circular cylinder $t = 200\frac{D}{U_\infty}$.

The distribution of the 25 smallest Ritz values in 12 cycles of GCRO-DR(25,10) iterations at $t = 200\frac{l}{U_\infty}$ is shown in figure 5. The closest 10 Ritz values to the origin (0,0) correspond to the recycled vectors. As depicted in this figure, from the 6^{th} to the 10^{th} cycle, the same



1st cycle

2nd cycle

3rd cycle

4th cycle

5th cycle

6th cycle

7th cycle

8th cycle

9th cycle

10th cycle

11th cycle

12th cycle

Fig. 5 Distribution of the real and imaginary parts of the 25 smallest Ritz values for the problem of laminar unsteady flow around a cylinder at various restarts of GCRO-DR(25,10) iterations.

Table 3 Comparison between the performance of GMRES(25), GCRO-DR(25,10), and GCRO-DR(25,10)+ schemes for laminar unsteady flow around a circular cylinder at $t = 200\frac{l}{U_\infty}$.

solver	nit	CPU	speed-up%	
			nit	CPU
GMRES	185	70 sec.	0	0
GCRO-DR	185	144 sec.	0	-105.7
GCRO-DR+	187	145 sec.	-1.0	-107.1

Ritz values were recycled between the restarts. In other words, the recycled vectors were not deflated throughout these cycles. This was contrary to the behavior observed for the convection-diffusion problem in figure 5.

3.5 Static Stall of NACA0015 Wing

A detached-eddy simulation (DES) was performed over a square NACA0015 wing at an angle of attack (AOA) of 18°. The freestream Reynolds and Mach numbers were 1.86×10^5 and 0.04. The simulation AOA well exceeded the reported static stall angle for this configuration, $\alpha_{ss} \approx 14.5°$ [1]. An unstructured mesh with $2,247,174$ nodes and $13,300,570$ tetrahedral elements was used for the simulation. Although the wing geometry and the freestream condition were fixed, due to the transient and chaotic nature of captured turbulent eddies, a time-accurate unsteady simulation was performed. The implicit time-step was $0.0078\frac{c}{U_\infty}$ where c represents the chord of the wing's cross section. This was equivalent to a CFL number of 5000. Figure 6 shows the computational mesh, the structure of turbulent eddies, and the strong wing-tip vortex extended downstream of the tip section.

The convergence rate of the DES is generally much slower than a similar RANS simulation owing to the resolution of the small and unsteady structures in the flow field. A comparison between convergence rates for a DES and a RANS simulation is given in figure 7-a. Both simulations were performed for the configuration described above, on the same computational mesh, using GMRES(50). The DES was approximately four times slower. This motivated us to study the convergence acceleration efficiency of Krylov recycling techniques for the same problem.

The resulting system of equations had $11,235,870$ degrees of freedom. The norm of the residual was decreased up to eight orders of magnitude at each time-step via GMRES(25), GCRO-DR(25,10), and GCRO-DR(25,10)+ iterations. The convergence plot is depicted in figure 7-b and the resulting speed-ups are reported in table 4. Recycling solvers showed an adverse efficiency for both the convergence rate and the CPU time. The increase in the iteration count was approximately 20%, which almost doubled the CPU time spent for the solution of the linear system as compared to the GMRES(25)'s performance.

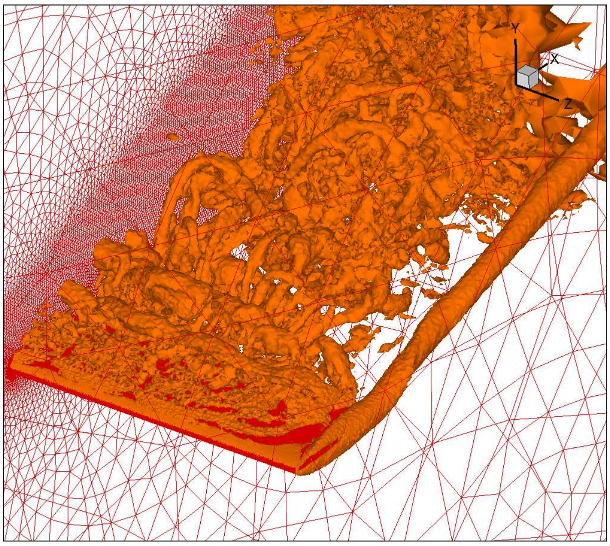
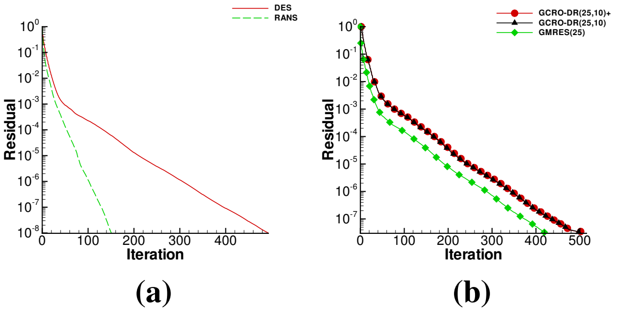


Fig. 6 Swirl contours around NACA0015 wing at static stall condition.



(a) **(b)**

Fig. 7 Residual norm versus iteration number for the flow around NACA0015 wing at static stall condition. **(a)** comparison between RANS and DES using GMRES(50) solver. **(b)** comparison between different solvers in DES.

3.6 Conclusions

The performance of a Krylov subspace recycling algorithm, GCRO-DR, was investigated for various CFD problems. Two recycling strategies were studied: in the first one, Krylov vectors were only recycled between restarts of the solver in a single system of equations, whereas in the second scheme, the recycling was extended over subsequent systems of equations. Each CFD problem was characterized by a special feature such as the mesh skewness, the flow separation, and the presence of turbulent eddies of various scales, which were responsible for the slow convergence of the linear solver. The following observations have been made after implementing the recycling technique:

- Recycling have had the best performance for the problem of the unsteady turbulent boundary layer over a flat plate which in-

Table 4 Comparison between the performance of GMRES(25), GCRO-DR(25,10), and GCRO-DR(25,10)+ schemes for DES of the flow around a NACA0015 wing at static stall condition.

solver	nit	CPU	speed-up%	
			nit	CPU
GMRES	421	97 sec.	0	0
GCRO-DR	506	199 sec.	-20.2	-105.1
GCRO-DR+	510	201 sec.	-21.1	-107.2

volved very skew elements, however the gain in the CPU time is very modest;

- As the size of the problem increases the performance of the recycling method deteriorates;
- The convergence of the DES is impaired by recycling;
- A better performance is obtained when the recycling was limited within a single system of equations;
- For the recycling method to be efficient in terms of CPU time reduction, the gain in the iteration count should be significant, e.g., more than 25%.

4 Acknowledgment

This research was supported by the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ).

References

[1] D. Birch and T. Lee. Tip vortex behind a wing undergoing deep-stall oscillation. *AIAA Journal*, 43(10):2081 – 2092, 2005.

[2] Eric de Sturler. Truncation strategies for optimal krylov subspace methods. *SIAM Journal on Numerical Analysis*, 36(3):864–889, 1999.

[3] J. Donea, S. Giuliani, and J. P. Halleux. Arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689 – 723, 1981.

[4] C. Farhat, M. Lesoinne, and P. Stern. High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: methodology and preliminary results. *Advances in Engineering Software*, 28(1):43 – 61, 1997.

[5] L. Hallo, C. Le Ribault, and M. Buffat. Implicit mixed finite-volume-finite-element method for solving 3D turbulent compressible flows. *International Journal for Numerical Methods in Fluids*, 25(11):1241 – 1261, 1997.

[6] G. Karypis and V. Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing ordering of sparse matrices. Technical report, University of Minnesota, Department of Computer Science/Army HPC Research Center, 1998.

[7] B. Koobus and C. Farhat. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170(1-2):103 – 29, 1999/02/26.

[8] Michel Lesoinne and Charbel Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134(1-2):71 – 90, 1996.

[9] Ronald B. Morgan. Gmres with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, 2002.

[10] Ronald B. Morgan and Min Zeng. Harmonic projection methods for large non-symmetric eigenvalue problems. *Numerical linear algebra with applications*, 5(1):33–55, January/ February 1998.

[11] R. Nkonga and H. Guillard. Godunov type method on non-structural meshes for three-dimensional moving boundary problems. *Computer Methods in Applied Mechanics and Engineering*, 113(1-2):183 – 204, 1994.

[12] Michael L. Parks, Eric de Sturler, Greg Mackey, Duane D. Johnson, and Spandan Maiti. Recy-

cling krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, 2006.

[13] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics SIAM, 3600 University City Science Center, Philadelphia, PA 19104-2688, second edition, 2003.

[14] P.R. Spalart and S.R. Allmaras. One-equation turbulence model for aerodynamic flows. *Recherche Aerospatiale*, 1:5 – 21, 1994.

[15] P.R. Spalart, S. Deck, M.L. Shur, K.D. Squires, M.Kh. Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, 20(3):181 – 195, 2006.

[16] P.R. Spalart, W.-H. Jou, M. Strelets, and S.R. Allmaras. Comments on the feasibility of les for wings, and on a hybrid rans/les approach. In *Proceedings - First AFOSR International Conference on DNS/LES*, pages 137 – 147, Ruston, Louisiana, United States, 1997.

5 Copyright Statement

The authors confirm that they, and/or their company or institution, hold copyright on all of the original material included in their paper. They also confirm they have obtained permission, from the copyright holder of any third party material included in their paper, to publish it as part of their paper. The authors grant full permission for the publication and distribution of their paper as part of the ICAS2008 proceedings or as individual off-prints from the proceedings.