ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

SOLVING LINEAR SYSTEMS WITH MULTIPLE RIGHT-HAND SIDES WITH GMRES: AN APPLICATION TO AIRCRAFT DESIGN

Aloïs Bissuel[1], Grégoire Allaire[2], Laurent Daumas[1], Frédéric Chalot[1] and Michel Mallet[1]

[1] Dassault Aviation
78, quai Marcel Dassault 92552 Saint-Cloud CEDEX 300 France
e-mail: {alois.bissuel, laurent.daumas, frederic.chalot, michel.mallet}@dassault-aviation.com

[2] CMAP
UMR 7641 École Polytechnique CNRS
Route de Saclay, 91128 Palaiseau CEDEX France
e-mail: allaire@cmap.polytechnique.fr

Keywords: Linearized Navier-Stokes, GMRes, Block GMRes, Finite Elements, HPC

Abstract. *To create efficient new aerodynamic designs or predict the onset of flutter, the linearised Navier-Stokes equations might be used. In some cases, many right-hand sides must be solved keeping the same matrix. In this paper, techniques which enable to solve several right-hand sides at the same time, such as Block* GMR*es, or reuse pieces of information computed in the previous solves, such as Krylov space recycling, are investigated. They will be tested on both simple and industrial test cases.*

1 INTRODUCTION

To create efficient new designs, aircraft manufacturers may resort to automatic optimisations tools relying on linearised Navier-Stokes solver [3]. Flutter analysis is another application where one can use time-domain linearisation of the Navier-Stokes equations [2]. In both cases, one may have to solve the same linear system with different right-hand sides, be it one for each geometric variable (direct method) or for each cost function (adjoint method) in the case of optimization, or one for each structural eigenmode in the case of flutter analysis. Due to their size and sparsity, these linear systems are solved using iterative methods such as GMRes [14].

Extensions of the GMRes method to multiple right-hand sides were consequently investigated. They can be grouped in two main classes. The first one is based on reusing information from the first solve to speed-up convergence of the subsequent systems. Using a projector, spectral information from the first solve can spectrally deflate at lower cost a GMRes method [10]. It is expected to attain near asymptotic convergence right from the beginning, removing transient plateaus. Another method could be to project all the right hand-sides on the Krylov space generated during the GMRes solve of the first vector to make these converge at least partially, lowering the number of iterations necessary for their complete solving.

The second class of extension of GMRes to several right-hand sides is to solve them all at once, using block Krylov spaces [5]. To improve the convergence when using small Krylov spaces with restarts, spectral deflation was implemented [9]. Decreased computation time is expected to come from both an accelerated convergence and a reduced parallel communication time due to block communications.

These methods will be tested on 2D and 3D examples, ranging from one hundred thousand to thirty million unknowns, on parallel architectures. Both convergence and timing results will be presented.

2 AETHER

2.1 Compressible Navier-Stokes code

AETHER is a compressible Navier-Stokes code developed in-house by Dassault Aviation. It is a finite elements solver on unstructured mesh, stabilized by the SUPG method [7]. To enable better properties of the implicit matrices, the compressible Navier-Stokes equations are written in entropic variables, so that they become symmetric. Several turbulence models are available (Spalart-Allmaras, k-ϵ, k-ω, K-KL, *etc*).The non-linear code is fully implicit, using GMRes to solve the Newton iterations. The code is made parallel using MPI as message passing library.

2.2 Linearised Navier-Stokes equations

For aeroelasticity problems, a simple way to understand the linearised Navier-Stokes equations is to consider the non-linear ones in a reduced form:

$$E(\mathbf{V}, \mathbf{x}) = 0 \tag{1}$$

Here, $\mathbf{V}$ is the physical state of the systems (*i.e.* the flow variables), and $\mathbf{x}$ corresponds to coordinates of the aeroplane. If the surface of the aircraft is slightly modified by a small displacement $\delta\mathbf{x}$, for instance a twist, then the flow variables will change accordingly by an increment $\delta\mathbf{V}$:

$$E\left(\mathbf{V} + \delta\mathbf{V}, \mathbf{x} + \delta\mathbf{x}\right) = 0 \tag{2}$$

A linear expansion of this equation, remembering that the base flow state verifies (1), yields

$$\frac{\partial E}{\partial \mathbf{V}}\delta\mathbf{V} = -\frac{\partial E}{\partial \mathbf{x}}\delta\mathbf{x} \tag{3}$$

Thus, if a small known displacement $\delta\mathbf{x}$ is applied to the aircraft, solving this linear system enables to know the variation of the flow variables $\delta\mathbf{V}$. If a harmonic movement needs to be applied to the structure, complex numbers are used. For details, see [2]. When doing aerodynamic optimisation, either a direct approach (comparable to the derivation in real number made before) or a adjoint one can be used, as explained in [3].

3 BLOCK GMRES

When solving for several right-hand sides, it might be assumed that searching iteratively the solution in several directions at the same time could be advantageous. Herein lies the main idea of the block GMRes method.

3.1 Standard block GMRes

Block GMRes is simply a generalisation of the standard iterative GMRes method [14] for vector with several columns. We want to solve the following linear problem :

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \text{with} \quad \mathbf{A} \in \mathbb{R}^{N\times N}, \quad \mathbf{b} \in \mathbb{R}^{N\times s}, \quad \mathbf{x} \in \mathbb{R}^{N\times s} \tag{4}$$

Here, N is the dimension of the linear system, and s is the number of right-hand side we want to solve simultaneously. Krylov spaces, which are necessary to GMRes, can be extended to a block definition.

$$\mathcal{B}_n(\mathbf{A}, \mathbf{y}) = \text{block span}\left(\mathbf{y}, \mathbf{A}\mathbf{y}, \ldots, \mathbf{A}^{n-1}\mathbf{y}\right) \tag{5}$$

Block span means that the linear combinations of the individual columns of the block vector are taken into account, i.e.

$$\forall \mathbf{u} \in \mathcal{B}_n(\mathbf{A}, \mathbf{y}), \exists\, (\boldsymbol{\gamma}_k) \in \mathbb{R}^{s\times s} \quad \mathbf{u} = \sum_{k=0}^{n-1} \mathbf{A}^k \mathbf{y} \boldsymbol{\gamma}_k$$

Starting from an initial guess $\mathbf{x}_0$, a block Krylov method for solving the linear system (4) finds an approximate solution $\mathbf{x}_n$ such that

$$\mathbf{x}_n - \mathbf{x}_0 \in \mathcal{B}_n(\mathbf{A}, \mathbf{r}_0) \tag{6}$$

with $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ being the initial residual.

The algorithm of the block GMRes is presented in [5]. Its implementation is summarised in algorithm 1. Compared to standard GMRes, there is little conceptual difference. Only the normalisations are transformed into QR-factorisations and the Hessenberg matrix is defined by blocks.

The Hessenberg matrix $\overline{\mathbf{H}}_n = \left(\widetilde{\boldsymbol{\eta}}_{i,j}\right) \in \mathbb{R}^{n+1\times n}$ is upper triangular with one extra lower diagonal. To solve the least-square problem with only a simple triangular solve as in line 17 of algorithm 1, the matrix $\overline{\mathbf{H}}_n$ needs to be reduced to a purely upper triangular matrix. A complete QR decomposition would be both too costly and impractical to update, as $\overline{\mathbf{H}}_n$ is created column-by-column. As introduced in [6], Householder transformations can be put to use efficiently for this task. These transformations are applied to blocks of dimension $2s \times s$ to reduce them to

Algorithm 1 Non-restarted Block GMRes with m Krylov vectors

$\mathbf{v}_0\boldsymbol{\rho}_0 = QR(\mathbf{r}_0)$ ▷ QR factorisation of the initial residual
$\mathbf{q} \leftarrow \mathbf{e_1}\boldsymbol{\rho}_0$ ▷ Least-square problem right-hand side
for $n = 1{,}m$ **do**
  $\tilde{\mathbf{v}} \leftarrow \mathbf{A}\mathbf{v}_{n-1}$
  for $k = 0{,}n-1$ **do** ▷ Arnoldi orthonormalisation of the next vector
    $\widetilde{\boldsymbol{\eta}}_{k,n-1} \leftarrow \langle \mathbf{v}_k, \tilde{\mathbf{v}} \rangle$
    $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} - \mathbf{v}_k \widetilde{\boldsymbol{\eta}}_{k,n-1}$
  end for
  $\mathbf{v}_n \widetilde{\boldsymbol{\eta}}_{n,n-1} = QR(\tilde{\mathbf{v}})$
  Apply previous Householder transformations to $\left(\widetilde{\boldsymbol{\eta}}_{k,n}\right), k = 0, \ldots, n-2$
  Create and apply Householder transformation $\mathbf{Q}_n$ on $\begin{pmatrix} \widetilde{\boldsymbol{\eta}}_{n-1,n-1} \\ \widetilde{\boldsymbol{\eta}}_{n,n-1} \end{pmatrix}$
  Apply Householder transformation $\mathbf{Q}_n$ to $\mathbf{q}$
  Compute residual : norm of the last block of $\mathbf{q}$
  If residual lower than ε **exit loop**
end for
Apply Householder transformations backwards to $\mathbf{q}$
$\mathbf{k}_n \leftarrow \mathbf{H}_n^{-1}\mathbf{q}$ ▷ Triangular solve
$\mathbf{x}_n \leftarrow \mathbf{x}_0 + \mathbf{V}_{n+1}\mathbf{k}_n$ ▷ New approximate solution
$\mathbf{r}_n \leftarrow \mathbf{V}_{n+1}\mathbf{k}_n$ ▷ New residual

upper triangular form. When a column is added to $\overline{\mathbf{H}}_n$, all previous Householder transformations need to be applied so as to get the modified $\boldsymbol{\eta}_{i,n}$. This transformed column is added as the last column of $\mathbf{H}_n$ which is upper triangular.

Restarting the block GMRes algorithm is simply a matter of restarting it with the new residual.

3.2 Spectral deflation of block GMRes

When solving a linear problem with the GMRes method, restarts are a necessity for memory reasons and computational time, as each new Krylov vector must be orthogonalised against all previous vectors. When restarted, Krylov methods have a known tendency to stall convergence. To alleviate these convergence problems, Morgan introduced in [8] a deflation of the smallest eigenvector of the matrix $\mathbf{A}$, which are hard to find during the iterations. Some numerical tricks are introduced by Röllin in [13]. This theory can be adapted to the block case.

The deflated restart is done in several steps as explained in algorithm 2. First of all, the harmonic Ritz vectors are computed, solving a standard eigenvalue problem and not a generalized one (see [13] for details). The k lowest ones are stored and orthogonalized. To force the new iterations to explore directions yet unsearched, the residual of the previous iteration is added to this basis. The new Krylov basis and Hessenberg matrix are also created. Finally, the residual in the new Krylov basis has to be computed. The idea for this critical step came from [12] : $\mathbf{q}$ is simply filled with the orthonomalising coefficients created during the orthogonalisation of the residual against the harmonic Ritz vectors.

Deflation of the right-hand sides, which is a way to eliminate linear dependence between the right-hand sides [5], was not used for several reasons. First of all, it is hard to correctly implement it jointly with spectral deflation. At restarts with spectral deflation, the new starting Krylov space and Hessenberg matrix must follow the Arnoldi relation. Moreover, the residual is a

Algorithm 2 Deflated restart of the block GMRes algorithm with k harmonic Ritz vectors

1: Compute the k first harmonic Ritz vectors $\mathbf{p}_i$ of $\mathbf{H}_n$
2: Orthonormalise $\mathbf{P}_k = (\mathbf{p}_i)_{i=1,k}$
3: Append $\mathbf{q}_n$ to $\mathbf{P}_k$ which becomes $\mathbf{P}_{k+1}$
4: Orthonormalise the last column of $\mathbf{P}_{k+1}$ against the first columns
5: Store the orthonormalisation coefficients in $\mathbf{q}$
6: $\mathbf{V}_{k+1}^{new} \leftarrow \mathbf{V}_{n+1}\mathbf{P}_{k+1}$
7: $\overline{\mathbf{H}}_k^{new} \leftarrow \mathbf{P}_{k+1}^T\overline{\mathbf{H}}_n\mathbf{P}_k$

linear combination of the harmonic Ritz vectors [13]. Deflation of the right-hand sides can delete and add (previously deleted) right-hand sides, and to enforce the aforementioned Arnoldi relation is difficult. To circumvent this problem, one could use the GCRO-DR algorithm [11], which will be introduced shortly below. The second reason is that a deflation of the right-hand sides was implemented (but not along spectral deflation), and no linear (near-)dependency could be found. As in aeroelasticity applications, the right-hand sides are quite similar, and the solutions have some sort of commonality between them, deflation was hoped to be quite powerful, but ultimately it was not found to be of any effect in our problems.

4 KRYLOV SPACE RECYCLING

Instead of solving all the right-hand sides at the same time, the information created during the solving of one right-hand side can be reused to solve all the other ones. Several methods can be devised, depending on which piece of information is to be recycled. When a single right-hand side is solved using the GMRes method, a Krylov space is generated, which in itself might be interesting to recycle. Moreover, if deflation is used, the smallest harmonic Ritz vectors are also computed. They are hard to find, and their approximation might take several restarts to be correct, so they might be of higher value than a single Krylov space.

4.1 Harmonic Ritz vector reuse

A first way to effectively reuse the harmonic Ritz vectors created during a previous solve with a deflated GMRes method is GCRO-DR [11]. It consists in a GMRes loop nested within a GCR (Gradient Conjuguate Residual). The necessary orthogonality constraint for GCR is enforced through a carefully chosen projector, akin to deflation projectors. See [4] for an interesting comparison between deflation projectors and deflation as treated in section 3.2. GCRO-DR was put to use in [15] for linearised Navier-Stokes computational studies with some success. It was chosen not to implement this algorithm due to its complexity and its likeness to deflated GMRes (see [11]).

Instead, a simpler method was tried. As explained in [10], a projection onto the small harmonic Ritz vectors at each restart of a standard GMRes (*i.e.* non deflated) could be sufficient to attain near-asymptotic convergence right from the start, and also exhibit good deflation properties at restart (no significant change of the convergence rate). For the projection and the GMRes method to interweave properly, the projection has to be carefully crafted. The projection, dubbed MinRes in [10], is explained in algorithm 3.

The MinRes projection should be used in the following way. A first right-hand side is solved using deflated GMRes. At the last restart, the Krylov space and the Hessenberg matrix created by the deflation and corresponding to the smallest harmonic Ritz vectors are saved. The other right-hand sides are then solved : before each restart of a standard GMRes cycle, the residual and

Algorithm 3 MinRes projection

Require: $\mathbf{x}_0$ approximate solution, $\mathbf{r}_0$ residual, $\mathbf{V}_{k+1}$ and $\overline{\mathbf{H}}_k$ verifying Arnoldi relation
1: $\mathbf{c} \leftarrow \mathbf{V}_{k+1}^T \mathbf{r}_0$
2: Solve $\min \|\mathbf{c} - \overline{\mathbf{H}}_k \mathbf{d}\|$ ▷ Simple triangular solve after upper-triangularising $\overline{\mathbf{H}}_k$
3: New solution is $\mathbf{x}_k \leftarrow \mathbf{x}_0 + \mathbf{V}_k \mathbf{d}$
4: New residual is $\mathbf{r}_k \leftarrow \mathbf{r}_0 - \mathbf{V}_{k+1} \overline{\mathbf{H}}_k \mathbf{d}$

the approximate solution are updated with the MinRes projection, using the saved Krylov space and Hessenberg matrix.

Another variant which was tested is to solve all the right-hand sides at the same time, one of them being solved with a deflated GMRes, and all the other being projected on the same Krylov space using the Minres projection. The idea is to leverage the similarities between the right-hand sides, and solving one may lower to a large extent the other residuals. Unfortunately, this technique proved unsuccessful in reducing with more than one order of magnitude the residuals of the other right-hand sides.

5 TEST CASES

These two techniques for solving linear systems with multiple right-hand sides will be tested on a two dimensional test-case, and a three-dimensional one, the latter being more representative of industrial problems.

5.1 2D test case: RAE2822

The first test case is a classic 2D wing profile. An illustration of the mesh used is shown in figure 1. This unstructured mesh has 35,000 nodes. A simulation on this test case has 140,000 unknowns.

This test case was used at a Mach number of 0.734 and an angle of attack of 3°. The base flow around which the linearised calculations are made does not exhibit any shocks or flow separation. Four movements of the profile were used for the purpose of creating four right-hand sides. The deformations corresponds in that order to a thickening, a pitching, a vertical displacement of the profile and an aileron deflection.

5.1.1 Block GMRes

To test the efficiency of the block GMRes method, several combinations of the number of right-hand sides and of Krylov space were tried. The figure 2 indicates the results.

The test runs are sorted according to the order of the block Krylov space, which is equivalently the number of iterates with the matrix $\mathbf{A}$. The parameter “p” indicates the number of right-hand sides solved at the same time. The dotting pattern is identical when solving the same number of right-hand sides. The smaller the dots, the lower the number of right-hand sides. The abscissa was chosen to be the number of matrix block-vector product (instead of matrix vector product). This enables to compare easily a block GMRes run with a standard one. For instance, if a solve with four right-hand sides takes less iterations than a solve with a single right-hand side (and that the four right-hand sides take roughly the same number of standard GMRes iterations to be solved), then solve the four of them at the same time is advantageous in terms of matrix-vector products.

The first comment to be made is that as a general rule, when a higher number of right-hand

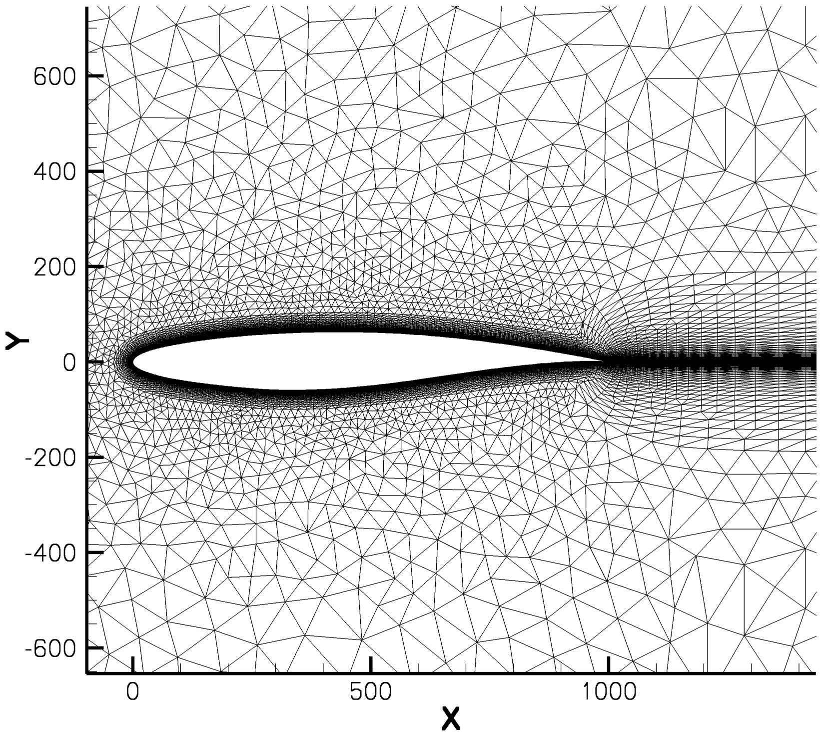

Figure 1: Mesh used for the 2D test case RAE2822

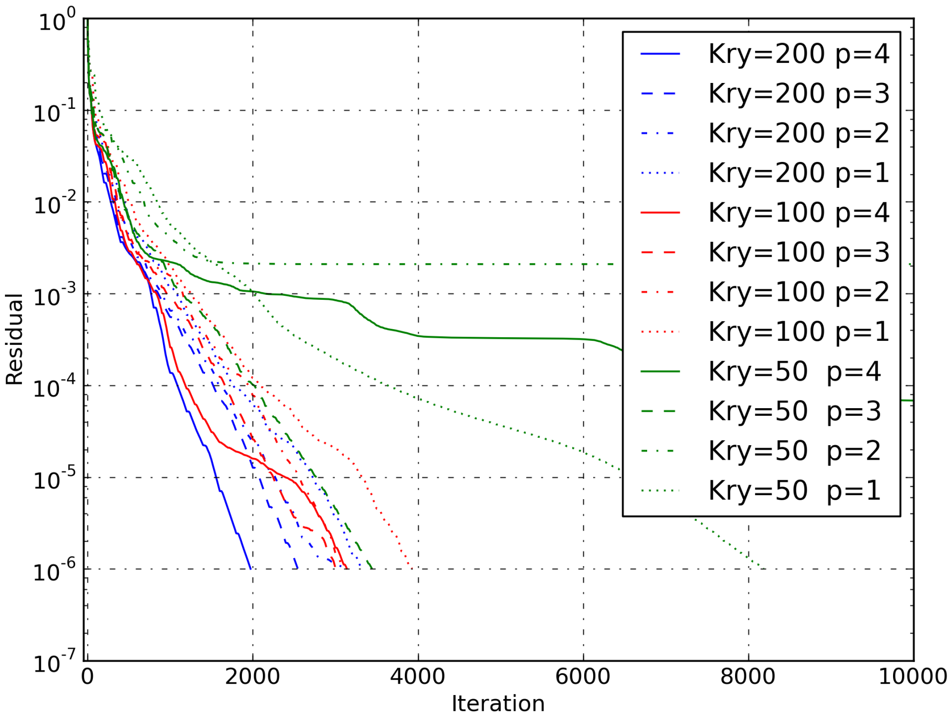

Figure 2: Convergence of the right-hand sides with block GMRes. In abscissa is the number of matrix block-vector products. “kry” is the order of the block Krylov space

Krylov	200	200	200	200	100	100	100	100
p	4	3	2	1	4	3	2	1
Time (s)	356	244	135	35	272	149	61	25
Time/p (s)	89	81	68	35	68	50	30	25
Iterations	1974	2546	3174	3315	3140	3021	3092	3931
Iterations/s	5.5	10.4	23.4	95.2	11.6	20.2	51.1	155.2
Iterations*p/s = matvec/s	22.2	31.3	46.9	95.2	46.2	60.7	102.2	155.2

Krylov	50	50	50	50
p	4	3	2	1
Time (s)	—	80	—	33
Time/p (s)	—	27	—	33
Iterations	—	3444	—	8185
Iterations/s	—	43.0	—	245.3
Iterations*p/s = matvec/s	—	128.9	—	245.3

Table 1: 2D test case: timing of block GMRes solve. Hyphen indicates a lack of convergence

sides are computed at the same time, the convergence is achieved in less iterations. This is verified for the solves with a block Krylov space of 200. It would have also been the case for the block Krylov space of 100, if it were not for the solve with four right-hand sides, whose convergence exhibits a strange slowing around 2000 iterations. Also, when the order of the block Krylov space is reduced, the robustness of the solve is decreased.For the Krylov space of order 50, with four and two right-hand sides, the convergence stalled for unexplained reasons.

Some timing details are provided in table 1. From these results, it can be inferred that block GMRes is slower than standard GMRes. The small convergence acceleration is not sufficient to overcome the the extra work of mainly orthonormalisation due to the larger number of vectors in memory.

To make a fairer comparison between block GMRes and its single right-hand side counterpart, table 2 shows a comparison this time grouped by equal memory footprint of the Krylov space. Here the block GMRes method is faster than the standard GMRes. First of all, the block GMRes needs less iterations to converge. Also, due to the blocking of the parallel communications and of the scalar products in the orthonormalisations, the matrix vector product is faster in block GMRes. Both of these effects combine to give a significant timing edge to the blocked Krylov method. This comparison is not realistic of industrial applications because of the large number of vectors used. Deflated GMRes has enabled Dassault Aviation to use quite small Krylov spaces without loss of robustness. Its blocked version is not yet robust enough.

	800 vectors			400 vectors		
Kry	200	400	800	100	200	400
p	4	2	1	4	2	1
Time (s)	356	208	129	272	135	177
Time/p (s)	89	104	129	68	68	177
Iterations	1974	2336	2819	3140	3174	3023
Iterations/s	5.5	11.2	21.8	11.6	23.4	17.1
Iterations*p/s = matvec/s	22.2	22.5	21.8	46.2	46.9	17.1

Table 2: 2D test case: timing with equal memory footprint

5.1.2 Krylov space recycling

Krylov space recycling was tested. The first right-hand side was solved using a standard deflated GMRes, so as to get the lowest harmonic Ritz vectors of the matrix. The convergence curve is shown on figure 3.

In this test case, the Krylov space size was of 200 vectors. The number of harmonic Ritz vectors which were kept was 60. The target residual was a normalized 10^{-5}, meaning that the residual had to be 5 orders of magnitude lower for the computation to be stopped. From the second right-hand side onward, the system was solved using non-deflated GMRes with Minres projection. For the first three orders of magnitude, it is competitive with deflated GMRes, but then the convergence stalls (except for the second right-hand side for which it happens one order of magnitude lower). For clarity, the graph was trimmed at 12 000 iterations.

5.2 3D test case

The 3D test case is a composed of a dummy half-fuselage with a generic swept wing. The profile has a 10% thick symmetric airfoil section (NACA 64A010 profile). The wing has a 1m span. It has been tested previously in CFD and in wind tunnel [2]. A view of the half-model is shown on figure 4. The flow conditions are a Mach number of 0.88 and an angle of attack of 0°. The computational mesh totals more than 6 million nodes, which makes more than 30 million unknowns to be solved. This test case is deemed to be representative of industrial applications.

The movements corresponding to the right-hand sides are test polynomials on the wing of the coordinates x, y and z of increasing degree. Seven of them are shown on figure 5. They are a blend of a pitching, heaving, bending, twisting movement of the wing. They seem quite similar. It is thus expected that solving one may considerably help solving the others.

5.2.1 Block GMRes

The block GMRes solver was tested on the 3D case. Ten right-hand sides were solved at the same time. Due to memory restrictions, we were forced to use a block Krylov space of order 50. The convergence curve is shown on figure 6. The convergence for the overall residual is presented with a bold line, and the residual of the single vectors are plotted with thin lines. After the 2000th iteration, the convergence is slowing down, and is completely stalled at the end of the computation. The residual failed to be lowered by 5 orders of magnitude, which is by experience

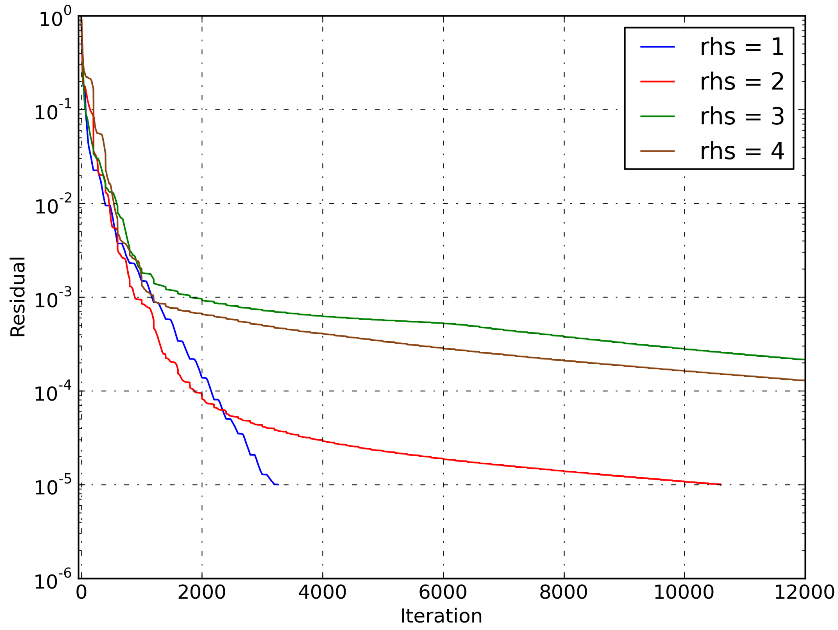

Figure 3: 2D test case: convergence of the Krylov space recycling method

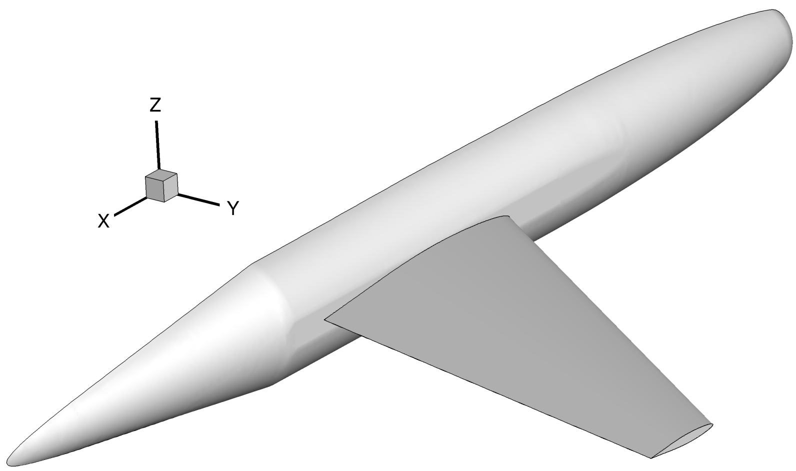

Figure 4: General view of the 3D test case

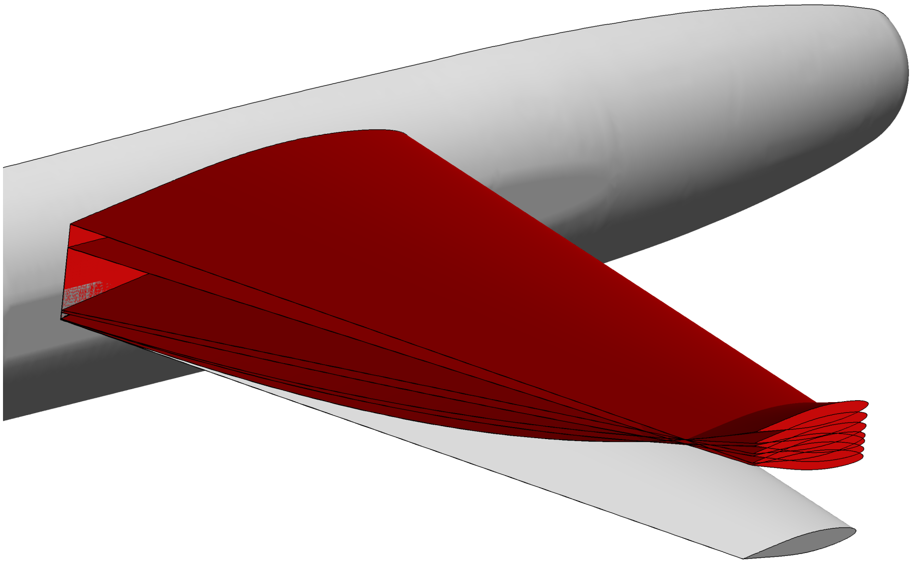

Figure 5: Seven movements corresponding to test polynomials on the wing

a requirement to get a good solution. By 4000 iterations, a standard GMRes computation would have already converged to that precision. As the convergence curve is not conclusive, no timings are shown.

5.2.2 Krylov space recycling

Recycling the harmonic Ritz vectors proved more interesting on the 3D case than on the 2D one. Figure 7 shows the convergence of the GMRes solver coupled with the Minres projection. The Krylov space was composed of 200 vectors, and 40 harmonic Ritz vectors were kept and used for the Minres projection. It is interesting to see that all the right-hand sides have the same asymptotic convergence rate. Nevertheless, this rate is smaller than the one achieved with the deflated GMRes solve (first right-hand side). Finally, the asymptotic convergence was not reached rapidly. The fact that no plateaus are found in the GMRes solve shows that this test case might not be the one where this method could shine [10]. To conclude with, the convergence penalty is far larger than the overhead time imposed by the deflation in standard GMRes.

6 CONCLUSION

In this paper, extensions of the GMRes algorithm to simultaneously solve several right-hand sides were implemented and tested on simple and industrial test cases. Block GMRes, the natural extension of GMRes to several right-hand sides, can accelerate matrix vector products on parallel computing architectures. Its lack of robustness on small Krylov space does put it at a disadvantage compared to standard GMRes. To compare favourably, it needs too large a Krylov space to be fast enough for industrial use. Krylov space recycling, which is a simpler

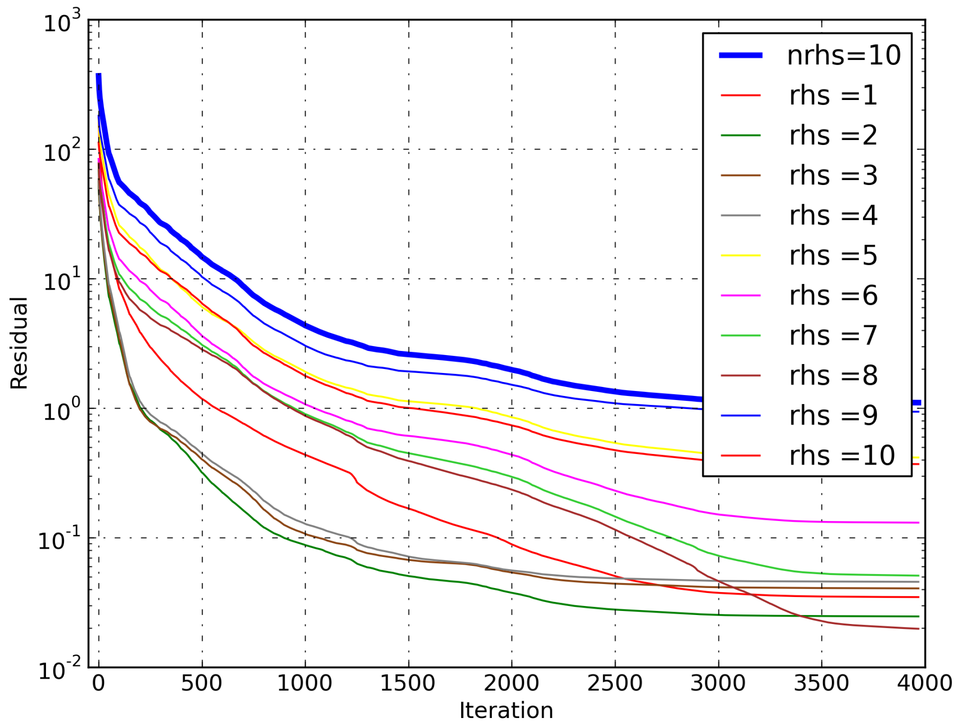

Figure 6: 3D test case: convergence of the block GMRes solver. Bold line is the total residual.

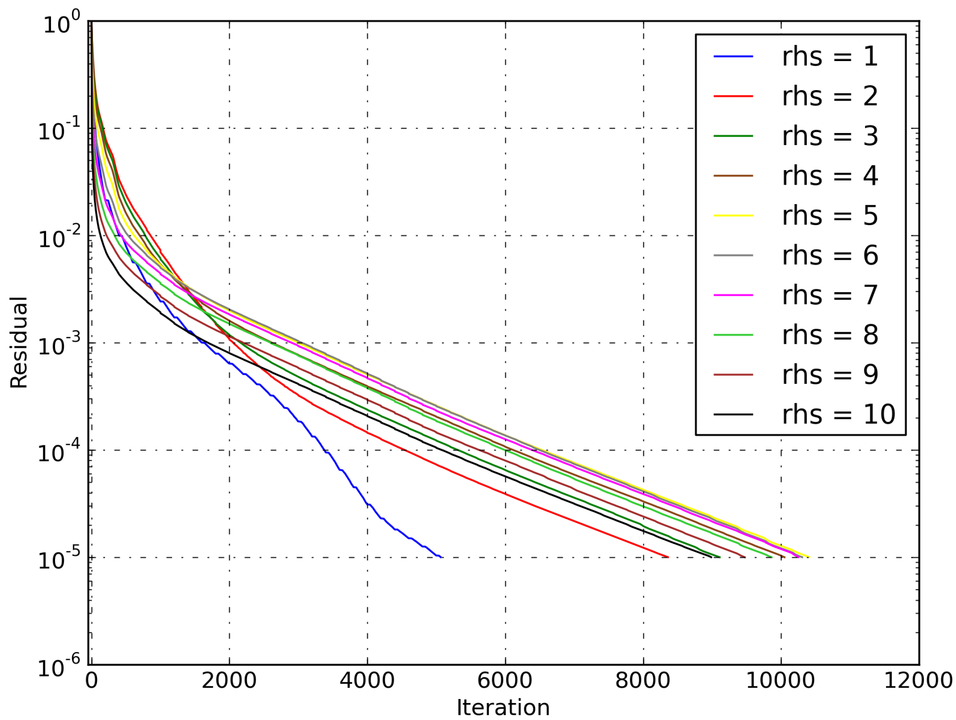

Figure 7: 3D test case: Krylov recycling. First right-hand side is solved with deflated GMRes

way of solving several right-hand sides at the same time, was disappointing on the simpler test case, where it stalled the convergence. On the industrial test case, though all right hand sides converged, the asymptotic convergence rate was not good enough and not reached soon enough for this method to be competitive.

This study reversely highlighted the performance of the deflated GMRes method for solving large linear systems, as various extensions of this method proved to be not as fast. Thus, future works will be devoted to accelerating the solves without changing the GMRes method. A study is currently underway to look at two-level preconditionners, which have shown to be promising to reduce the impact of domain decomposition on convergence in CFD computations[1].

ACKNOWLEDGEMENT

This work is supported by a CIFRE fellowship funded in part by ANRT (National Association for Research and Technology) on behalf of the French Ministry of Higher Education and Research, and in part by Dassault Aviation.

REFERENCES

[1] H. Alcin, B. Koobus *et al.*, *Efficiency and scalability of a two-level Schwarz algorithm for incompressible and compressible flows*, Int. J. Number. Meth. Fluids, 2012

[2] L. Daumas, F. Chalot, N. Forestier, Z. Johan, *Industrial use of linearized CFD tools for aeroelastic problems*, IFASD, 2009

[3] L. Daumas, Q. Dinh, S. Kleinveld, G. Rogé, *Automatic shape optimization using parametric CAD applied to sonic boom reduction*,3rd AIAA Multidisciplinary Design Optimization Specialist Conference, pp. 23-26. 2007.

[4] A. Gaul, M. Gutknecht, J. Liesen, R. Nabben, *A Framework for Deflated and Augmented Krylov Subspace Methods*, arXiv:1206.1506v2, Feb. 2013

[5] M. H. Gutknecht, *Block Krylov Spaces Methods for Linear Systems With Multiple Right-Hand Sides: An Introduction*, Feb. 2006

[6] M. Gutknecht, T. Schmelzer, *Updating the QR decomposition of block tridiagonal and block Hessenber matrices generated by block Krylov space methods*, Research Report No. 2005-10, 2005

[7] M. Mallet, *A Finite Element Method for Computational Fluid Dynamics*, PhD Thesis, Stanford University, 1985

[8] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[9] R. B. Morgan, *Restarted Block GMRes With Deflation Of Eigenvalues*, Applied Numerical Mathematics, 54.2 (2005): 222-236

[10] R. B. Morgan, W. Wilcox, *Deflated Iterative Methods for Linear Equations with Multiple Right- Hand Sides*, arXiv 0707.0505 2007

[11] M. L. Parks, E. de Sturler *et al.*, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., Vol. 28, No. 5, pp. 1651–1674

[12] X. Pinel, M. Montagnac, *Block Krylov Methods to Solve Adjoint Problems in Aerodynamic Design Optimization*, AIAA Journal, Vol. 51, No. 9, Sept 2013

[13] S. Röllin, W. Fichtner, *Improving the accuracy of GMRes with deflated restarting*, SIAM J. Sci. Comput. Vol. 30, No. 1, pp. 232–245

[14] Y. Saad, M. Schultz, *GMRes: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., Vol. 7, No. 3, July 1986

[15] S. Xu, S. Timme, K. J. Badcock, *Krylov Subspace Recycling for Linearised Aerodynamics Analysis using DLR-TAU*, IFASD 2015