The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

A THOROUGH ANALYSIS OF DEFLATION TECHNIQUES APPLIED TO CFD: FROM RPM TO BOOSTCONV AND BEYOND

JEREMY KALFOUN[1], GUILLAUME PIERROT[2] AND JOHN CAGNOL[3]

[1] Université Paris-Saclay, CentraleSupélec, MICS and FdM CNRS FR-3487
ICON Technology & Process Consulting
e-mail: jeremy.kalfoun@centralesupelec.fr

[2] ICON Technology & Process Consulting
email: g.pierrot@iconcfd.com, www.iconcfd.com

[3] Université Paris-Saclay, CentraleSupélec, MICS and FdM CNRS FR-3487
e-mail: john.cagnol@centralesupelec.fr

Key words: CFD, Convergence acceleration, Deflation methods, Fixed-Point iterations, Stabilization of solution algorithms

Summary. Iterative methods such as Fixed-Point Iterations arising from the discretization of PDEs by standard methods (FE, FV, etc.) offer their share of challenges when applied to large systems due to their poor conditioning. Slow convergence, limit cycle oscillations and, in the worst case, divergence are regularly observed. Deflation-based techniques such as the Recursive Projection Method (RPM), are particularly appealing for tackling these kinds of issues without sacrificing the physical fidelity as they allow for the discrimination in between slow and fast modes, the first one being solved implicitly while the baseline algorithm is still relied upon to solve the latter. More recently, an alternative approach, called BoostConv, has emerged around the same idea, although the connection between the two has received little attention. In this paper, we offer to comprehensively revisit this family of methods for fixed-point iterations, giving rise to a new unified framework that encompasses both approaches and more and brings to light a number of as yet unexplored degrees of freedom, auguring further potential improvements in terms of performance and robustness. The present analysis focuses on three primary elements: choice of a projector, trouble modes recruitment, and rescue mode. A comparison of existing approaches is provided along with the benefits of proposed new variants on a selection of random matrices and CFD cases with different levels of complexity for the sake of completeness.

1 Introduction

Iterative methods for solving large linear systems $Ax = b$ (or nonlinear $A_x x = b$) have been a topic of intense research since the very begining of numerical analysis. The goal of these methods is to find a numerical solution to the system of equations with as few iterations as possible, while ensuring stability and accuracy of the solution. In particular, these linear systems are found when discretizing a set of Partial Differential Equations (PDE) with Finite Element or Finite Volume methods which is, for example, the case in Computational Fluid Dynamics [1].

A number of different approaches have been developed for solving linear systems. Among them, two families of methods have been widely studied: fixed-point iterations and Krylov subspace methods which lies inside the projection methods framework [2, 3].

Since it's simpler and more flexible to implement, in a Matrix-Free environment, fixed-point iterations are often preferred. This is the case for SIMPLE-like algorithms, PISO inner-loops, etc. They consist of iteratively calculating x^{k+1} as a function of x^k that can be written on the form $x^{k+1} = Gx^k + b$ where G is a given iteration matrix. If the spectral radius of G is strictly less than one then the method is guaranteed to converge [2], and the convergence rate directly depends on the largest eigenvalue's modulus. On the other hand, if the spectral radius is greater than one, the method will diverge. Other issues such as slow convergence due to eigenvalues of the iteration matrix very close to the unit circle or limit cycle oscillations due to non-linear effects are often observable.

To recover or speed up the convergence, deflation methods have been introduced. They are based on a divide and conquer idea. The solution space is divided into a (hopefully small) troublespace and an easygoing-space, and the solution algorithm applies different strategies to each of them. The troublespace can be approached in different ways. It can be done a priori by using coarse grids as low frequency modes in space are often the troublemaking ones, it is linked to subdomain deflation and multigrid methods [6] or by recycling previous solutions to guesstimate the starting point and deflate the search directions which is linked to model reduction. It can also be done on the fly by massaging the Krylov vectors to approximate dominant eigenvectors. From now on we will refer to this approach as "Krylov Deflation" [7, 4]. All these types of deflation can be applied in the context of the Krylov method, but no general framework exists in the context of fixed-point iterations.

In this work, a general framework of Deflated Fixed-Point Iterations (DFPI) is presented, which will prove to cover a number of methods that do not appear to be deflation at first glance (e.g. RPM). It enables obtaining general results on convergence, exploring degrees of freedom, discussing the choice of projectors, trouble-space, vector recruitment and having a simplified general implementation. Finally, the Deflated Fixed-Point Iterations framework will be applied to random matrices to expose the different behaviors and CFD cases.

2 DFPI Framework

The aim of Deflated Fixed-Point Iterations is to find an approximate solution of the matrix system:

$$Ax^{\infty} = b \tag{1}$$

where $A \in \mathcal{M}_N(\mathbb{R})$ non-singular and $b \in \mathbb{R}^N$ are known, while $x^{\infty} \in \mathbb{R}^N$ is the solution of the system. The non-linear aspects are to be covered in a future work. From now on we'll focus on the linear framework in order to settle the main concepts and results

2.1 Formulation of the DFPI

The usual Preconditioned Fixed-Point iterations (Richardson methods) to solve the linear system 1 can be read:

$$\begin{aligned} x^{n+1} &= x^n + P^{-1}(b - Ax^n) \\ &= x^n + P^{-1}A(x^{\infty} - x^n) \end{aligned} \tag{2}$$

with $P \in \mathcal{G}l_N(\mathbb{R})$. The convergence of these iterations depends on the largest eigenvalue in modulus of $Id - P^{-1}A$ [2].

Let's define a projection space approaching the troublespace $\boldsymbol{Z} = \text{span}\,\{Z_i\}_{i=1,\dots,M}$, $Z_i \in \mathbb{R}^N$ with $M << N$, and a projection Q_R on $\boldsymbol{Z}$.

The projection Q_R is fully defined by the projection space $\text{Im}\,(Q_R) = \boldsymbol{Z}$ and $\ker\,(Q_R)$. As we only have access to Ax^∞ and not x^∞ on its own, $\ker\,(Q_R)$ is defined throughout the subspace $\boldsymbol{Y} = A^{-T}(\ker\,(Q_R))^\perp$. By definition,

$$\forall x \in \mathbb{R}^N, \forall y \in \boldsymbol{Y}, \left\langle (Id - Q_R)\, x, A^T y \right\rangle = 0$$

And we recognize the Petrov-Galerkin projection [2]

$$\forall x \in \mathbb{R}^N, \forall y \in \boldsymbol{Y}, \langle AQ_R x, y\rangle = \langle Ax, y\rangle$$

$$\Rightarrow Q_R = Z\left(Y^T A Z\right)^{-1} Y^T A$$

where $N \times M$ matrices Z and Y, are constructed such that column vectors form a basis of subspaces $\boldsymbol{Z}$ and $\boldsymbol{Y}$. Some particular cases worth mentioning:

- The Galerkin projection when $\boldsymbol{Y} = \boldsymbol{Z}$,
- The Least Square projection when $\boldsymbol{Y} = A\boldsymbol{Z}$,
- The Orthogonal projection when $\boldsymbol{Y} = A^{-T}\boldsymbol{Z}$

As a general deflation method, the idea of the DFPI is to solve directly onto the trouble-space and iterate on the easygoing one. This is achieved by re-projecting the current error $x^n - x^\infty$ on $ker\,(Q_R)$ at the beginning of each iteration:

$$x^{n+\frac{1}{2}} = (Id - Q_R)\, x^n + Q_R x^\infty$$

Then, the iteration from the baseline algorithm is performed:

$$x^{n+1} = x^{n+\frac{1}{2}} + P^{-1}A\left(x^\infty - x^{n+\frac{1}{2}}\right)$$

This leads to the first formulation of the Deflated Fixed-Point Iterations (DFPI):

$$\begin{cases} x^{n+\frac{1}{2}} = x^n + Q_R\,(x^\infty - x^n) \\ x^{n+1} = x^{n+\frac{1}{2}} + P^{-1}A\,(Id - Q_R)\,(x^\infty - x^n) \end{cases} \tag{3}$$

Equivalently, the projection of the error onto $ker\,(Q_R)$ can be performed after iterating. This leads to a post-projection formulation. The deflated fixed point iterations can be implemented as a blackbox which performs a projection before or after iterating with the usual fixed-point algorithm.

A preconditioner formulation can also be derived. Let's define $Q_L^A \in \mathcal{M}_N(\mathbb{R})$ such as $Q_L^A A = AQ_R$

Proposition 1. *If matrix* $\left(A^{-1}Q_L^A + P^{-1}\left(Id - Q_L^A\right)\right)$ *is non-singular, DPFI (Eq. 3) can be written as Fixed-Point iterations with a modified preconditioner* $\underline{P} = \left(A^{-1}Q_L^A + P^{-1}\left(Id - Q_L^A\right)\right)^{-1}$

Proof. The main formulation of the DFPI can be written:

$$\begin{aligned}\underline{x}^{n+1} &= \underline{x}^n + Q_R\left(x^\infty - \underline{x}^n\right) + P^{-1}A\left(Id - Q_R\right)\left(x^\infty - \underline{x}^n\right)\\ &= \underline{x}^n + \left(A^{-1}Q_L^A + P^{-1}\left(Id - Q_L^A\right)\right)A\left(x^\infty - \underline{x}^n\right)\end{aligned}$$

□

This formulation is equivalent to the previous ones. The modified preconditioner $\underline{P}$ behaves like A on the trouble-space and like P on the easygoing-space.

2.2 Convergence of the DFPI

Proposition 2. *If iterations Eq. 3 converges and $\left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)$ is non-singular, then they converge to the solution of Eq. 1 for any projection.*

Proof. In Section 2.1, the main formulation of the DFPI has been written as fixed-point iterations with a modified preconditioner $\underline{P}$. If these iterations converge, the limit will be the solution if $\underline{P}$ is well defined and non-singular which is the case if $Q_R + P^{-1}A\left(Id - Q_R\right)$ is non-singular. □

Proposition 3. *$\boldsymbol{Z}$ stable by $P^{-1}A$ is a sufficient condition for $\left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)$ to be non-singular*

Proof. $x \in \mathbb{R}^N$ such that $\left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)x = 0$

$$\begin{aligned}&\Rightarrow \left(Id - Q_R\right)P^{-1}A\left(Id - Q_R\right)x = 0\\ &\Rightarrow P^{-1}A\left(Id - Q_R\right)x = z \in \boldsymbol{Z}\end{aligned}$$

If $\boldsymbol{Z}$ is stable by $P^{-1}A$, then $\left(Id - Q_R\right)x = A^{-1}Pz \in \boldsymbol{Z}$ and so $\left(Id - Q_R\right)x = 0$. And finally $Q_R x = 0$, leading to $x = 0$ and $\underline{P}$ defined and non-singular.

□

Just as iterations with preconditioner P converge if and only if $\forall\lambda \in \mathrm{Sp}\left\{Id - P^{-1}A\right\}, |\lambda| < 1$, iterations with $\underline{P}$ will converge if and only if:

$$\forall\lambda \in \mathrm{Sp}\left\{Id - \left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)\right\}, |\lambda| < 1$$

Let's consider $\boldsymbol{Z}$ such that it contains all the generalized eigenvectors associated to trouble making eigenvalues of $\mathrm{Sp}\left\{Id - P^{-1}A\right\}$. Let's write $\lambda_i, 1 \le i \le M$ these eigenvalues, and $\mathrm{Sp}\left\{Id - P^{-1}A\right\} = \left\{\lambda_i\right\}_{1\le i\le N}, M < N$. With $\boldsymbol{Z}$ stable, for any projection:

Proposition 4. *$Sp\left\{Id - \left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)\right\} = \left(0, 0, ..., 0, \lambda_{M+1}, ..., \lambda_N\right)$ and so iterations with preconditioner $\underline{P}$ converge at a rate given by the largest non-deflated eigenvalue in modulus.*

Proof. One can show that eigenvalues of the matrices $M = P^{-1}A$ and $N = Q_R + P^{-1}A\left(Id - Q_R\right)$ are the union of eigenvalues of:

- $\left(\left(Id - Q_R\right)P^{-1}A\left(Id - Q_R\right)\right)|_{\ker(Q_R)}$
- $\left(Q_R P^{-1}AQ_R\right)|_{\mathrm{Im}(Q_R)}$ for M and $Q_R|_{\mathrm{Im}(Q_R)}$ for N

Hence $\mathrm{Sp}\left\{Id - \left(Q_R + P^{-1}A\left(Id - Q_R\right)\right)\right\} = (0, 0, ..., 0, \lambda_{M+1}, ..., \lambda_N)$ where $(\lambda_{M+1}, ..., \lambda_N)$ are eigenvalues of $\left(\left(Id - Q_R\right)P^{-1}A\left(Id - Q_R\right)\right)|_{\ker(Q_R)}$ and so iterations with preconditioner $\underline{P}$ converge.

□

An example where convergence is recovered for a 900×900 random sparse matrix is shown in Figure 1. A diagonal preconditioner is used and $\boldsymbol{Z}$ is enhanced a posteriori with dominant eigenvectors. The largest eigenvalue modulus is $|\lambda_{max}| = 16.78$ (not shown on the figure for the sake of clarity, as for smallest eigenvalues). The red eigenvalues correspond to those of the base iteration matrix that do not appear in the modified iteration matrix at the end. The black ones are in both iteration matrices. Each drop in the residuals and in their slope correspond to $\boldsymbol{Z}$ enhancement with a dominant eigenvector (or pair of conjugate ones). Three different projections are used, and remarkably enough, the choice of projection does not seem to have much impact, the three of them leading to very similar results.

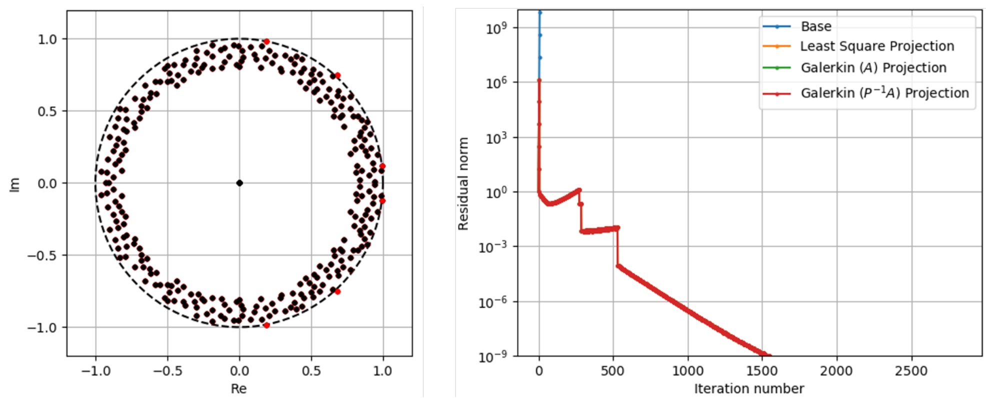

Figure 1: Test case on a random sparse matrix. Left : Eigenvalues of the iteration matrices. Right : Residuals plot for several projections

3 RPM and BoostConv

In this section, we show how two well-known methods for stabilizing fixed-point iterations, RPM and BoostConv, fit within the Deflated Fixed Point Iterations framework. Although these methods are typically applied to general non-linear cases, we will focus on their application within the linear framework for the purposes of our study. For the sake of simplicity, we assume the spaces are fixed. The issue of recruiting eigenvectors will be addressed in a subsequent paragraph.

3.1 RPM in the DFPI framework

The Recursive Projection Method (RPM) was first introduced by Schroff and Keller [8] to stabilize fixed-point iterative methods by using Newton iterations on the unstable space and fixed point iterations on the algebraic complement. A description and evaluation of the method is conducted in [9]. The method is based on the fixed-point iterations which, in the case of Preconditioned Richardson Iterations, is written:

$$x^{n+1} = F(x^n) = x^n + P^{-1}A\,(x^\infty - x^n) \tag{4}$$

By introducing the subspace $\boldsymbol{Z}$ formed by the eigenvectors associated to dominant eigenvalues, and the orthogonal projection $\mathcal{O}$ on $\boldsymbol{Z}$, the iterative scheme is modified as follows in the usual case of additive RPM:

$$\begin{cases} x^n = \hat{x}^n + \tilde{x}^n = \mathcal{O}x^n + (Id - \mathcal{O})\,x^n \\ \hat{x}^{n+1} = \mathcal{O}F(\hat{x}^{n+1} + \tilde{x}^n) \\ \tilde{x}^{n+1} = (Id - \mathcal{O})\,F(\hat{x}^n + \tilde{x}^n) \end{cases} \tag{5}$$

In our work, we present multiplicative RPM, where $\hat{x}^{n+1}$ is used to calculate $\tilde{x}^{n+1}$, as it has been calculated in the previous step:

$$\tilde{x}^{n+1} = (Id - \mathcal{O})\,F(\hat{x}^{n+1} + \tilde{x}^n)$$

Correspondence between DFPI and multiplicative RPM is more straightforward as it does not require the introduction of a new projection Q_L and the hypothesis that $\boldsymbol{Z}$ is stable by $P^{-1}A$ (inherent to RPM in any case). In the following, we will focus on multiplicative RPM, but a similar reasoning exists for additive RPM.

In classical non-linear RPM, $\hat{x}^{n+1}$ is calculated implicitly using a Newton iteration. In the linear case, only one Newton step is necessary.

Developing the fixed-point iteration for $\hat{x}^{n+1}$:

$$\hat{x}^{n+1} = \mathcal{O}\left(\hat{x}^{n+1} + \tilde{x}^n + P^{-1}A\left(x^\infty - \tilde{x}^n - \hat{x}^{n+1}\right)\right)$$

Using the fact that $\mathcal{O}\hat{x}^{n+1} = \hat{x}^{n+1}$, $\mathcal{O}\tilde{x}^n = 0$ and $\tilde{x}^n = x^n - \hat{x}^n$:

$$\mathcal{O}\left[P^{-1}A\,(x^\infty - x^n) - P^{-1}A\left(\hat{x}^{n+1} - \hat{x}^n\right)\right] = 0$$

It means that $\forall j \in [1, ..., m]$, $\left(P^{-1}A\left[(x^\infty - x^n) - \left(\hat{x}^{n+1} - \hat{x}^n\right)\right], Z_j\right) = 0$ with $\hat{x}^{n+1} - \hat{x}^n \in \boldsymbol{Z}$ which means that $\hat{x}^{n+1} - \hat{x}^n$ is the Galerkin projection of $x^\infty - x^n$ associated to $P^{-1}A$ on $\boldsymbol{Z}$:

$$\hat{x}^{n+1} = \hat{x}^n + Q_\xi\,(x^\infty - x^n) \tag{6}$$

In the case of multiplicative RPM, $\tilde{x}^{n+1} = (Id - \mathcal{O})\,F(\hat{x}^{n+1} + \tilde{x}^n)$:

$$\begin{aligned} \Rightarrow \tilde{x}^{n+1} &= \tilde{x}^n + (Id - \mathcal{O})\,P^{-1}A\left(x^\infty - \tilde{x}^n - \hat{x}^{n+1}\right) \\ &= \tilde{x}^n + (Id - \mathcal{O})\,P^{-1}A\,(x^\infty - \tilde{x}^n - \hat{x}^n - Q_\xi\,(x^\infty - x^n)) \\ &= \tilde{x}^n + (Id - \mathcal{O})\,P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n) \end{aligned}$$

But $\forall x \in \mathbb{R}^N$, $Z^T A\,(Id - Q_\xi)\,x = 0$, and so:

$$\mathcal{O}P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n) = Z\left(Z^T Z\right)^{-1}\left[Z^T P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n)\right] = 0$$

$$\Rightarrow \tilde{x}^{n+1} = \tilde{x}^n + P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n)$$

Taking

$$\begin{cases} x^{n+\frac{1}{2}} = \tilde{x}^n + \hat{x}^{n+1} \\ x^{n+1} = \tilde{x}^{n+1} + \hat{x}^{n+1} \end{cases}$$

$$\begin{cases} x^{n+\frac{1}{2}} = \tilde{x}^n + \hat{x}^n + Q_\xi\,(x^\infty - x^n) = x^n + Q_\xi\,(x^\infty - x^n) \\ x^{n+1} = \tilde{x}^n + \hat{x}^{n+1} + P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n) = x^{n+\frac{1}{2}} + P^{-1}A\,(Id - Q_\xi)\,(x^\infty - x^n) \end{cases} \tag{7}$$

which correspond to the main formulation of the DFPI.

3.2 BosstConv in the DFPI framework

The BoostConv method is presented by its author Citro [10] as based on the minimization of the residual norm at each integration step and described as inspired by the Krilov methods in order to stabilize the computation of unstable steady states. It is also implemented in [11] where satisfactory results are presented.

The method is widely described in [10]. It starts from an iterative algorithm to solve Eq. 1:

$$x^{n+1} = x^n + P^{-1} r^n$$

with $r^n = b - Ax^n$. This corresponds to the Preconditioned Fixed-Point Iterations. These iterations, and more precisely the residual, are modified by the method as follows:

$$x^{n+1} = x^n + P^{-1}\xi_1^n \tag{8a}$$
$$\xi_1^n = \xi_0^n + \rho^n \tag{8b}$$
$$\rho^n = r^n - AP^{-1}\xi_0^n \tag{8c}$$
$$\xi_0^n = \text{argmin}_{\xi\in\boldsymbol{Z}_0} ||r^n - AP^{-1}\xi||^2 \tag{8d}$$

where $\boldsymbol{Z}_0$ is spanned by a certain set of vectors ($\boldsymbol{u}_i$ in [10]).

The minimization problem defined in Eq. 8d can be written differently, with a change of variable $\zeta = P^{-1}\xi$ and writing $r^n = A(x^\infty - x^n)$:

$$P^{-1}\xi_0^n = \text{argmin}_{\zeta\in\boldsymbol{Z}} ||A\left((x^\infty - x^n) - \zeta\right)||^2 \tag{9}$$

with $\boldsymbol{Z}$ being the subspace given by $P^{-1}\boldsymbol{Z}_0$. The solution of this minimization problem is none other than the Least Square Projection of $(x^\infty - x^n)$ on $\boldsymbol{Z}$. Hence, the solution is written:

$$P^{-1}\xi_0^n = Q_R^{LSQ}(x^\infty - x^n) \tag{10}$$

with the Least square projection operator on the subspace $\boldsymbol{Z}$:

$$Q_R^{LSQ} = Z\left(Z^T A^T A Z\right)^{-1} Z^T A^T A$$

Then the correction $\rho^n = r^n - AP^{-1}\xi_0^n$ is applied to ξ_0^n such that

$$\begin{aligned}\xi_1^n &= \xi_0^n + r^n - AP^{-1}\xi_0^n \\ &= PQ_R(x^\infty - x^n) + A(x^\infty - x^n) - AQ_R(x^\infty - x^n)\end{aligned}$$

(with $Q_R = Q_R^{LSQ}$)

Hence, the modified iteration in Eq. 8a can be read as:

$$\begin{aligned}x^{n+1} &= x^n + P^{-1}\xi_1^n \\ &= x^n + Q_R(x^\infty - x^n) + P^{-1}A\left(Id - Q_R\right)(x^\infty - x^n)\end{aligned}$$

Finally, the BoostConv iterations can be written as follows:

$$\begin{cases} x^{n+\frac{1}{2}} = x^n + Q_R\left(x^\infty - x^n\right) \\ x^{n+1} = x^{n+\frac{1}{2}} + P^{-1}A\left(Id - Q_R\right)(x^\infty - x^n) \end{cases} \tag{11}$$

which corresponds exactly to the iterations presented in Eq. 3.

Both RPM and BoostConv are the same methods with a different projection.

4 Eigenvectors Selection

The first observable difference between RPM and BoostConv is the projection they use. But as previously mentioned, it does not consistently affect the results. The selection strategy of the eigenvectors plays a crucial role. While both methods store increments to approach the dominant eigenvectors using a generalized power iteration method, BoostConv adds a vector at each iteration, whereas RPM waits approximately 100 iterations to select a few well-approximated eigenvectors.

Both strategies have their own advantages and disadvantages. RPM is more selective and parsimonious in its recruitment, enriching the projection base only with sufficiently well-approximated generalized eigenvectors, while BoostConv, like GMRES, indiscriminately recruits all Krylov vectors until the storage is saturated.

In this work, a *Just-in-time* strategy is introduced and consists of two steps:

- Establishing a working subspace until a certain level of stability is achieved
- Using the Rayleigh-Ritz method to approximate the eigenvectors within this subspace and determining whether to enrich the subspace based on a stability criterion

The second step consists of calculating eigenvectors of the small projected iteration matrix $M_p = Z_t^T \left(Id - \underline{P}^{-1}A\right) Z_t$. Writing μ_i an eigenvalue of M_p associated to v_i, the Ritz pair $(\mu_i, Z_t v_i)$ is an approximate solution of the eigenvalue problem on the iteration matrix.

Then, $|| \left(Id - \underline{P}^{-1}A\right) Z_t v_i - \mu_i Z_t v_i||_2$ gives an indication on how good the approximate eigenvector $Z_t v_i$ is. It can be added to $\boldsymbol{Z}$ if smaller than a given tolerance and if the associated eigenvalue modulus is greater than a certain limit.

This method enables the storeage of dominant eigenvectors as soon as possible while ensuring accuracy and stability of $\boldsymbol{Z}$.

Three strategies are applied on a random sparse matrix case as before Fig. 2. The number of stored vectors is limited to 36. We utilize three tolerance levels: a loose tolerance, where a vector is added every two iterations; a tight tolerance, involving a significant number of iterations before adding eigenvectors; and an intermediate tolerance, which corresponds to our 'Just-in-time' strategy.

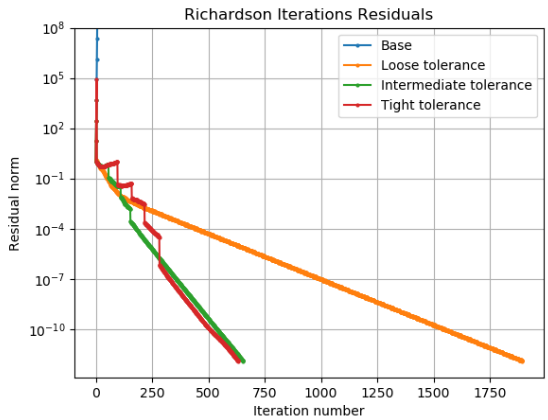

Figure 2: Residuals for several eigenvectors selection strategy on a random sparse matrix.

While the loose tolerance proves more efficient in the initial iterations, the tight tolerance demonstrates superiority towards the end. The intermediate tolerance combines the best of both approaches by adding eigenvectors as soon as possible, ensuring subspace stability without the need to wait for a specific number of iterations.

5 Rescue Mode

The Krylov method GMRES without restart is guarantied to converge. Here, in some cases, when dominant eigenvectors are associated to very large eigenvalues, the algorithm can blow up almost immediately before we even had the time to recruit a first stable vector. To avoid this, a rescue mode has been designed. The approach involves iterative updates without updating the solution directly, instead normalizing the residual used in each iteration while aligning with the dominant eigenvector.

This is achieved by modifying the residual in the DFPI iterations:

$$x^{n+1} = x^n + \underline{P}^{-1}\tilde{r}^n \tag{12}$$

where $\tilde{r}^n = \alpha^n \left(r^n + \tilde{r}^{n-1} - r^{n-1}\right)$ and $\tilde{r}^0 = r^0$. α^n can be chosen as $\frac{1}{||x^n - x^{n-1}||}$.

Doing so, Arnoldi iterations are performed to keep aligning with the dominant eigenvector without diverging. Once a decent approximation of the dominant eigenmode has been achived, the basis is enriched and standard DFPI iterations are applied again if no further divergence is detected. An example of the rescue mode is demonstrated in a similar test case as before, but with larger eigenvalues.

Figure 3: Residuals for several eigenvectors selection strategy on a random sparse matrix.

6 CFD Case

Finally, DFPI are tested on a well known CFD validation case, known as the transonic bump. A dual time-stepping methodology is used and DFPI are performed during the linear inner loop. For a given tolerance on the residuals, a 40% decrease in the number of iterations is observed. When tolerances are tightened, which can be necessary in industrial cases, the improvement is even more pronounced.

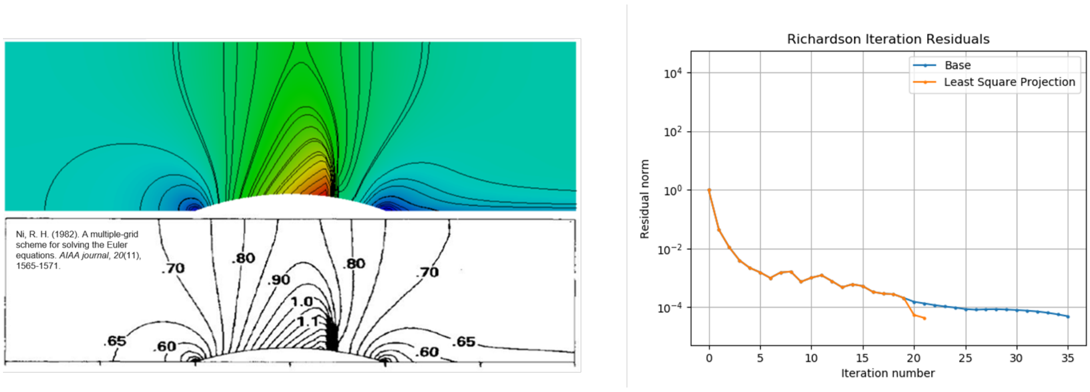

Figure 4: DFPI applied to the transonic test case.

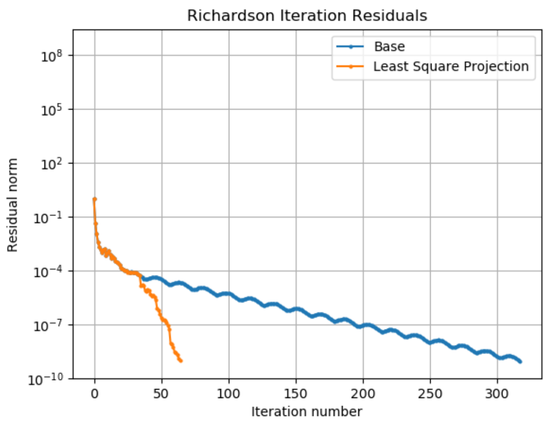

Figure 5: DFPI applied to the transonic test case with tighter tolerance.

7 Conclusion

This paper introduces a general framework for deflation methods in the linear setting, accompanied by a convergence proof for any projection. This framework encompasses several existing methods, such as BoostConv and RPM, for trouble-space computed on the fly from Krylov vectors. These methods differ primarily in their projection and vector selection strategies. We present an optimized method for eigenvector selection and a rescue mode that guarantees prevention of divergence. This general framework has demonstrated favorable results on random matrices and a transonic bump CFD case. Future work is underway to cover the non-linear case, context adaptative tolerance, and implementation in an industrial CFD code.

Acknowledgments

We would like to thank Jacques Papper and other colleagues from ICON Technology & Process Consulting and from the MICS for the fruitful, constructive and enlightened discussions that enabled us to make progress on this project.

REFERENCES

[1] Anderson, J. D. and Wendt, J. 1995. "Computational fluid dynamics." In Springer 206.

[2] Saad, Y. 2003. "Iterative methods for sparse linear systems." SIAM.

[3] Saad, Y. and Schultz, M. H. 1986. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." In SIAM Journal on Scientific and Statistical Computing 7(3), 856-869.

[4] Frank, J. and Vuik, C. 2001. "On the construction of deflation-based preconditioners." In SIAM Journal on Scientific Computing 23, 442-462.

[5] Campobasso, M. S. and Giles, M. B. 2003. "Effects of flow instabilities on the linear analysis of turbomachinery aeroelasticity." In Journal of propulsion and power 19(2), 250-259.

[6] Wesseling, P. 1995. "Introduction to multigrid methods." Wesseling, P. (1995). In NASA-CR-195045.

[7] Nicolaides, R. A. 1987. "Deflation of conjugate gradients with applications to boundary value problems." In SIAM Journal on Numerical Analysis 24(2), 355-365.

[8] Shroff, G. M. and Keller, H. B. 1993. "Stabilization of unstable procedures: the recursive projection method." In SIAM Journal on numerical analysis, 30(4), 1099-1120.

[9] Görtz, S. and Möller, J. 2004. "Evaluation of the recursive projection method for efficient unsteady turbulent cfd simulations." In 24th International Congress of the Aeronautical Sciences, 1-13.

[10] Citro, V. , Luchini, P. , Giannetti, F. and Auteri, F. 2017. "Efficient stabilization and acceleration of numerical simulation of fluid flows by residual recombination." In Journal of computational physics 344, 234-246.

[11] Dicholkar, A. , Zahle, F. , and Sørensen, N. N. 2022. "Convergence enhancement of SIMPLE-like steady-state RANS solvers applied to airfoil and cylinder flows." In Journal of Wind Engineering and Industrial Aerodynamics 220.