Efficient stabilization and acceleration of numerical simulation of fluid flows by residual recombination

V. Citro[a], P. Luchini[a], F. Giannetti[a], F. Auteri[b]

[a] *DIIN, Universitá degli Studi di Salerno, Via Giovanni Paolo II, 132, 84084, Italy*
[b] *Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, via La Masa 34, 20156, Italy*

Abstract

The study of the stability of a dynamical system described by a set of partial differential equations (PDEs) requires the computation of unstable states as the control parameter exceeds its critical threshold. Unfortunately, the discretization of the governing equations, especially for fluid dynamic applications, often leads to very large discrete systems. As a consequence, matrix based methods, like for example the Newton-Raphson algorithm coupled with a direct inversion of the Jacobian matrix, lead to computational costs too large in terms of both memory and execution time.

We present a novel iterative algorithm, inspired by Krylov-subspace methods, which is able to compute unstable steady states and/or accelerate the convergence to stable configurations. Our new algorithm is based on the minimization of the residual norm at each iteration step with a projection basis updated at each iteration rather than at periodic restarts like in the classical GMRES method. The algorithm is able to stabilize any dynamical system without increasing the computational time of the original numerical procedure used to solve the governing equations. Moreover, it can be easily inserted into a pre-existing relaxation (integration) procedure with a call to a single *black-box* subroutine.

The procedure is discussed for problems of different sizes, ranging from a small two-dimensional system to a large three-dimensional problem involving the Navier-Stokes equations. We also show that the proposed algorithm is able to improve the convergence of existing iterative schemes. In particular, the procedure is applied to the subcritical flow inside a lid-driven cavity. We also discuss the application of `Boostconv` to compute the unstable steady flow past a fixed circular cylinder (2D) and boundary-layer flow over a hemi-

spherical roughness element (3D) for supercritical values of the Reynolds number. We also show that `Boostconv` can be used effectively with any spatial discretization, be it a finite-difference, finite-volume, finite-element or spectral method.

Keywords: steady solution, iterative procedure, stabilization algorithm

1. Introduction

The knowledge of fixed points or periodic solutions of a dynamical system is important both for stability analysis and the development of flow control strategies. The first step, within the framework of a classical stability analysis, is the computation of a reference state around which the governing equations are linearized [1]: this can be either a steady or a periodic solution of the nonlinear governing equations. The stability of such states usually depends on the value of a given parameter. In general, when a critical value is reached, a bifurcation occurs and the original solution becomes linearly unstable, with the system tending towards a new state. A classical example of such behaviour in fluid dynamics is the instability occurring in the wake of a circular cylinder. At low Reynolds number (precisely for $Re < 46.7$) the flow is steady and symmetric, but for larger values of Re a global instability arises in the flow field [2] leading to the well-known von Kármán vortex street. In order to perform stability computations beyond the critical threshold, we need a numerical method which is able to pass through the bifurcation point, selecting the unstable branch of the solution. Unfortunately, this cannot be achieved by using a standard time integration of the governing equations.

For low-dimensional systems, *e.g.*, models of chemical reactions, coupled oscillators or lumped element models for fluid flows [3], several continuation and bifurcation packages like AUTO [4] or CONTENT [5] are freely available on the net. These packages are based on Newton's method coupled to a direct linear solver applied to an augmented algebraic system of equations. They have been designed also to deal with higher-codimension bifurcations and continuation of periodic orbits.

Unfortunately, the numerical treatment of PDEs often involves the solution of very large systems of algebraic equations which do not allow the use of such packages. In particular, Newton's algorithm, involving matrix inversion, cannot be used to solve large problems because of large memory requirements and computational costs (CPU time). In these cases an alternative approach

is to employ a Newton-Krylov algorithm, where a Krylov subspace method (like a GMRES or a BCGSTAB) is used to solve linear systems at each Newton substep.

For large-scale systems, computations of unstable states have been performed in a limited number of studies [6], [7]. Van Noorden et al.[9] discussed the application of a continuation method with subspace iterations to compute periodic orbits of high-dimensional systems. Newton-Krylov techniques are used by Sanchez et al. [10] to obtain the fixed points of a Poincaré map. Shroff & Keller [8] proposed the Recursive Projection Method (RPM) which stabilizes an unstable iterative procedure by splitting the solution space into the direct sum of a large subspace spanned by the stable modes and a small subspace containing the unstable ones: the algorithm applies Newton's method only on the small subspace while it retains the original iterative procedure on its complement. Mittelmann & Weber[12] proposed a continuation strategy coupled with a multigrid algorithm to compute solutions of nonlinear eigenvalue problems near turning points.

Akervik et al. [13], instead, proposed to apply selective frequency damping (SFD) to recover the steady states of the Navier-Stokes equations. The key idea is to apply a temporal low-pass filter to damp the oscillations of the unsteady part of the solution. It was successfully adopted by Bagheri et al.[14] and Ilak et al.[15] to investigate the stability of a jet in cross-flow and by Nichols & Schmid[16] and Qadri et al.[17] to study lifted flames. However, the SFD algorithm needs an estimate of the global mode frequency and it cannot be applied to compute unstable states in presence of stationary bifurcations. Moreover, all the cited techniques involve significant coding and/or changes in the original numerical algorithm.

The aim of the present work is to propose a new algorithm, inspired by Krylov-subspace methods, able to efficiently compute unstable steady states of a high-dimensional dynamical system. This method is based on the minimization of the residual norm at each integration step and can be applied as a black-box procedure in any iterative or time marching algorithm without negatively impacting the computational time of the original code.

2. Iterative solution of a linear system: Krylov methods

The iterative solution of the linear system

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

using Krylov subspace methods has been widely studied in the last decades [18]. In the previous equation $\mathbf{A}$ is a $\mathbb{R}^{N\times N}$ matrix and $\mathbf{x}$ and $\mathbf{b}$ are the $\mathbb{R}^N$ vectors representing the solution and the known term of the linear system respectively. The approximation of the solution $\mathbf{x}$ is sought such that $\mathbf{x}_j$ belongs to the shifted Krylov spaces $\mathscr{S}_j = \mathbf{x}_0 + \mathscr{K}_j(\mathbf{A}, \mathbf{r}_0)$ with

$$\mathscr{K}_j(\mathbf{A}, \mathbf{r}_0) = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, ..., \mathbf{A}^{j-1}\mathbf{r}_0\}, \quad \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0, \tag{2}$$

where $\mathscr{K}_j$ is the $j-th$ Krylov subspace and $\mathbf{r}_0 \in \mathbb{R}^N$ is the residual vector with respect to the initial guess $\mathbf{x}_0$. The residual vector $\mathbf{r}_j$ lies in the Krylov residual subspace $\mathscr{R}_j$ defined as

$$\mathscr{R}_j = \mathbf{r}_0 + \mathbf{A}\mathscr{K}_j(\mathbf{A}, \mathbf{r}_0). \tag{3}$$

The main idea of this iterative procedure is that the $j-th$ approximation of the solution $\mathbf{x}_j \in \mathscr{S}_j$ is found by requiring the minimization of a functional. Thus, different Krylov methods result from different choices of this functional, from the characteristics of the matrix and from implementation details [18]. A possible choice is to select the approximation $\mathbf{x}_j$ such to minimize the 2-norm $||\cdot||_2$ of the residual,

$$\mathbf{x}_j = \min_{\mathbf{x}_j \in \mathscr{S}_j} ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2. \tag{4}$$

Such method is usually referred as the *minimal residual approach* (MR) and it is largely adopted: a typical example is offered by the popular GMRES method by Saad and Schultz [19].

The implementation of GMRES is based on the solution of the *least squares* problem (4) obtained through an orthonormal basis of the Krylov subspace produced by the Arnoldi procedure. It is worthwhile noting that during the execution of GMRES the basis grows and, as a consequence, the storage requirements grow accordingly. In case of a large system, the number of iterations needed to achieve a sufficiently accurate solution can be excessive and the resulting Arnoldi matrix becomes unacceptably large to be stored: usually, a restarted procedure is adopted. It consists in restarting the algorithm when the subspace dimension reaches a maximum value p. In particular, after p iterations the current new approximation $\mathbf{x}_j$ and inherent residual $\mathbf{r}_j$ are computed and GMRES is stopped. These arrays become the starting point for a new call to the algorithm. Unfortunately, the restarted algorithm usually shows a slow convergence rate [18].

Preconditioning techniques can be adopted to improve the performances and reliability of Krylov subspace methods: it is recognized, in fact, that preconditioning is the most critical ingredient in the development of efficient solvers for challenging problems in scientific computation [18] and represents the real key ingredient to achieve an acceptable convergence rate.

3. `BoostConv` algorithm

The main idea which inspired the proposed algorithm is similar to the one at the basis of GMRES, but in a reverse logic sequence. We start from an *existing* iterative algorithm that is modified to `Boost` the `Conv`ergence of the overall procedure.

A generic linear iteration for the solution of the linear system (1) can be expressed as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{B}\mathbf{r}_n \tag{5}$$

where $\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n$ is the residual and $\mathbf{B}$ is a matrix representing the particular iterative scheme used to solve the problem (see e.g., Ch.4 [20]). For example, experssion (5) can result from a classical Jacobi or Gauss-Seidel method or from a pseudo-temporal discretization of a dynamical system. In general, the matrix B can also contain a preconditioner.

The convergence of procedure (5) is governed by the eigenvalues of the iteration matrix $(\mathbf{I} - \mathbf{B}\mathbf{A})$: the algorithm converges if and only if the spectral radius of the iteration matrix is smaller than 1. In that case, the asymptotic convergence rate is dictated by the slowly decaying modes. Usually, only a small part of the spectrum strongly influences the convergence rate. Even in case of a nonlinear system, the behavior will become linear when the approximate solution $\mathbf{x}_n$ is close enough to the solution $\mathbf{x}$ of the nonlinear governing equation. On the other hand, the algorithm could diverge because of a small set of unstable modes. Thus, the purpose of `BoostConv` is to modify only the part of the spectrum characterized by these slowly decaying or amplified modes, while letting the original algorithm damp the remaining (decaying) modes.

In order to obtain this stabilization, like in the GMRES algorithm, we use the information provided by the state $\mathbf{x}_n$ and by the residual $\mathbf{r}_n$. The residual satisfies the homogeneous equation obtained by applying the operator $-\mathbf{A}$ to (5) and then adding $\mathbf{b}$ to both sides. In this way we easily obtain the

following evolution equation for $\mathbf{r}_n$:

$$\mathbf{b} - \mathbf{A}\mathbf{x}_{n+1} = \mathbf{b} - \mathbf{A}\left[\mathbf{x}_n + \mathbf{B}(\mathbf{b} - \mathbf{A}\mathbf{x}_n)\right], \text{i.e.}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A}\mathbf{B}\mathbf{r}_n. \tag{6}$$

Our key idea is to improve the existing procedure (5) by replacing the residual vector $\mathbf{r}_n$ with a modified residual $\boldsymbol{\xi}_n$ such that the improved algorithm reads as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{B}\boldsymbol{\xi}_n(\mathbf{r}_n). \tag{7}$$

In the previous equation $\boldsymbol{\xi}_n$ is a suitable function of $\mathbf{r}_n$ and can be interpreted as the feedback term of a closed loop control algorithm or a structural perturbation of the original iteration matrix. In order to guarantee the consistency of the modified algorithm with the original iterative procedure and recover the solution of the linear system (1), it is sufficient that $\boldsymbol{\xi}_n \to 0$ as $\mathbf{r}_n \to 0$. The introduction of the vector $\boldsymbol{\xi}_n$ modifies equation (6), leading to the new residual equation

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A}\mathbf{B}\boldsymbol{\xi}_n, \tag{8}$$

or equivalently

$$\mathbf{r}_n - \mathbf{r}_{n+1} = \mathbf{A}\mathbf{B}\boldsymbol{\xi}_n. \tag{9}$$

We now minimize $\mathbf{r}_{n+1}$ by choosing a suitable function $\boldsymbol{\xi}_n = \boldsymbol{\xi}_n(\boldsymbol{r}_n)$. If we knew $(\mathbf{AB})^{-1}$ we could exactly annihilate $\mathbf{r}_{n+1}$ by computing $\boldsymbol{\xi}_n(\mathbf{r}_n)$ as

$$\boldsymbol{\xi}_n = (\mathbf{AB})^{-1}\mathbf{r}_n. \tag{10}$$

However, for large systems, the exact inversion of $\mathbf{AB}$ is out of reach or too expensive to be performed. We therefore approximate the solution of (10) by using a classical least-squares method.

The action of the operator $\mathbf{AB}$ can be represented by storing a set of N vector pairs $(\mathbf{u}_i, \mathbf{v}_i)$, where the second member is produced by the action of $\mathbf{AB}$ on the first. The least-squares method is then adopted to approximate the solution of the algebraic linear system $\mathbf{AB}\boldsymbol{\xi}_n = \mathbf{r}_n$, through its projection on the subspace spanned by $\mathbf{u}_i$, as

$$\boldsymbol{\xi}_n \simeq \sum_{i=1}^{N} c_i \mathbf{u}_i. \tag{11}$$

In our case the vectors $\mathbf{u}_i$ and $\mathbf{v_i}$ are related to each other by

$$\mathbf{v_i} = \mathbf{AB}\mathbf{u_i} \quad \text{for } i = 1, .., N. \tag{12}$$

while the coefficients c_i are chosen to minimize $|\mathbf{r}_n - \mathbf{AB}\boldsymbol{\xi}_n|^2$. The standard least-squares procedure leads to a system of equations for the coefficients c_i of the form

$$D_{kl}\, c_l = t_k \tag{13}$$

where $t_k = \mathbf{v}_k \cdot \mathbf{r}_n$ and $D_{kl} = \mathbf{v}_k \cdot \mathbf{v}_l$ is a small $N \times N$ matrix. Matrix $\mathbf{D}$ is often ill-conditioned and an orthogonalization procedure (QR decomposition) is usually needed to find the solution. However, when N is small, as for the cases we are considering, the solution can be found with a classical LU decomposition, which is computationally faster. If the algorithm only consisted of looping eq. (11) it would never converge because the subspace would not change and $\boldsymbol{\xi}_n$ could converge to zero even when the residual $\mathbf{r}_n$ is not identically zero, but simply orthogonal to the leading N (basis-)vectors $\mathbf{u}_i$. We can improve the approximation if we remark that the least-squares solution produces a new residual $\boldsymbol{\rho} = \mathbf{r_n} - \mathbf{AB}\xi_\mathbf{n}$ which can be expressed in terms of $\mathbf{v_i}$ as

$$\boldsymbol{\rho} = \mathbf{r_n} - \mathbf{AB}\xi_\mathbf{n} = \mathbf{r_n} - \mathbf{AB}\left(\sum_{i=1}^{N} c_i \mathbf{u_i}\right) = \mathbf{r_n} - \sum_{i=1}^{N} c_i \mathbf{AB}\mathbf{u_i} = \mathbf{r_n} - \sum_{i=1}^{N} c_i \mathbf{v_i}. \tag{14}$$

Remembering that the original iterative algorithm (5) was designed to somewhat work, if slowly, with $\boldsymbol{\xi}_n$ equal to the old residual $\mathbf{r}_n$, we restore a convergent procedure by adding the new residual $\boldsymbol{\rho} = \mathbf{r}_n - \sum_i c_i \mathbf{v}_i$ to eq. (11), so that the complete algorithm now reads

$$\boldsymbol{\xi}_n = \sum_i c_i \mathbf{u}_i + \mathbf{r}_n - \sum_i c_i \mathbf{v}_i. \tag{15}$$

The rationale behind this procedure is to invert exactly the part of the problem represented by the dominant and slower decaying modes while letting the original iterative algorithm to handle the remaining modes. We now go back to the issue of selecting a convenient set of vectors $\mathbf{u_i}$: in `BoostConv` algorithm, both $\mathbf{u}_i$ and $\mathbf{v}_i$ are conveniently calculated by observing that, according to (8), for each n we have

$$\mathbf{r}_n - \mathbf{r}_{n+1} = \mathbf{AB}\boldsymbol{\xi}_n. \tag{16}$$

For a given N, in a cyclic fashion, we add, at the beginning of a new iteration, a new vector pair by selecting $\mathbf{u}_N = \boldsymbol{\xi}_{n-1}$ and $\mathbf{v}_N = \mathbf{r}_n - \mathbf{r}_{n-1}$. In order to keep the size of the basis constant, another pair must be discarded which typically will be the oldest. Such choice is dictated by the fact that by applying the algorithm to a nonlinear system it is beneficial to use the freshest information on the system dynamics in order to account for the change of the system Jacobian (in our case represented by the linear operator $\mathbf{A}$). We also note that in matrix D_{kl} of (13) only the row and the column involving a new pair need to be updated. Such selection procedure works when we already have N vector pairs. At the beginning of the algorithm (for $n < N$) we can still use the same procedure but we continuously increase the basis dimension from 1 to the chosen value of N. In this first stage, no vector pairs are discharged. Note that the basis dimension N must satisfy $N > M$, where M is the number of unstable modes of the system.

From a programming viewpoint, `BoostConv` algorithm can be encapsulated in a black-box procedure where the only input is $\mathbf{r}_n$ and the only output is $\boldsymbol{\xi}_n$. If $\boldsymbol{\xi}_n$ is returned in the same vector where $\mathbf{r}_n$ was provided, the only modification necessary to boost the convergence of the pre-existing iterative algorithm (5) is a single line of code containing the call to `BoostConv`. We can do this in a very compact way because the original algorithm can be seen as a simple feedback loop as illustrated in figure 1: as a consequence, we can simply insert `BoostConv` in the subroutine evaluating the residual as depicted in figure 2.

4. Implementation details

As mentioned in the previous sections, a key feature of the outlined method is the possibility to code it into a black-box computer algorithm. Here, we provide some useful programming guidelines concerning the proposed procedure summarized in Algorithm 1. The only required input at each step is the residual $\mathbf{r}_n$ at the current iteration n: using such information the procedure calculates the modified residual vector $\boldsymbol{\xi}_n$ (of the same dimension). Here, in order to compact the resulting algorithm, we define an auxiliary vector $\mathbf{w}_h = \mathbf{u}_h - \mathbf{v}_h$.

Once provided the dimension N of the vector basis, the procedure can be divided in four parts: i) from line 2 to 6 of Algorithm 1, we discard the

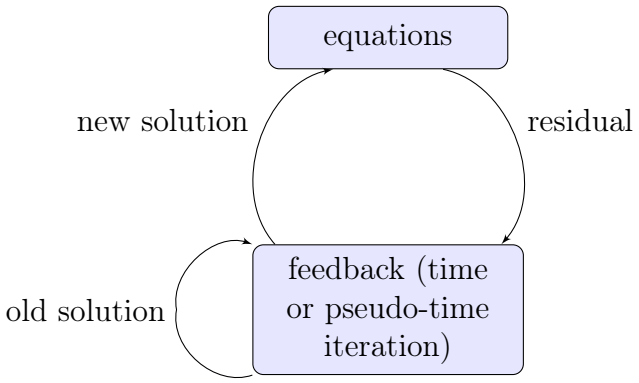

Figure 1: Original iterative procedure. Unmodified integration of governing equations.

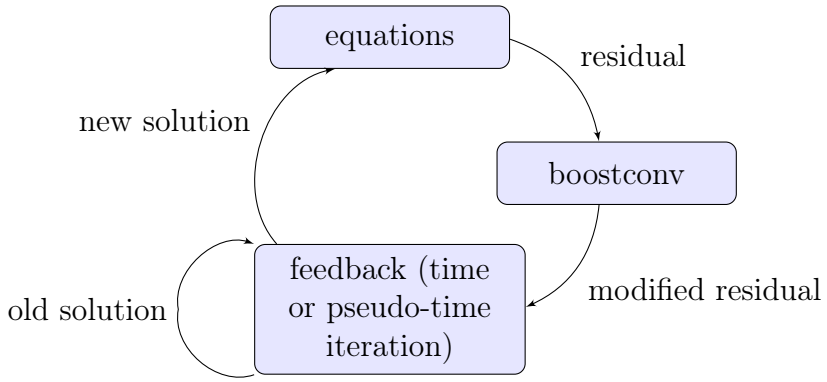

Figure 2: Modification of an existing iterative scheme: stabilized iterative procedure.

oldest pair of vectors $\mathbf{u}$ and $\mathbf{v}$ to store the new pair computed by using the data provided at the current step n; ii) from line 7 to 13, we build the least-squares matrix $\mathbf{D}$ of eq. 13 and the known term $\mathbf{t}$; iii) in line 14 we solve the linear problem arising from the least-squares method by using a simple LU decomposition on the small $N \times N$ system; iv) in line 15 we compute the new modified residual $\boldsymbol{\xi}_n$ according to eq. 15.

Algorithm 1 `BoostConv`

1: **Input:** $\mathbf{r}_n$(current residual); **Output:** $\boldsymbol{\xi}_n$(modified residual);
2: **for** $\{h = 1$ to $N - 1\}$ **do**
3: $\quad \mathbf{v}_h = \mathbf{v}_{h+1}$; $\quad \mathbf{w}_h = \mathbf{w}_{h+1}$; ▷ Discard the oldest vectors
4: ▷ (remember that $\mathbf{w}_h = \mathbf{u}_h - \mathbf{v}_h$)
5: **end for**
6: $\mathbf{v}_N = \mathbf{r}_{n-1} - \mathbf{r}_n$; $\quad \mathbf{w}_N = \boldsymbol{\xi}_{n-1} - \mathbf{v}_n$; ▷ Update the vector basis
7: **for** $\{m = 1$ to $N\}$ **do**
8: $\quad D_{N,m} = \mathbf{v}_N \cdot \mathbf{v}_m$; ▷ Update the least-square matrix (eq.13)
9: **end for**
10: $D_{m,N} = D_{N,m}$; ▷ Least-square matrix is symmetric
11: **for** $\{k = 1$ to $N\}$ **do**
12: $\quad t_k = \mathbf{v}_k \cdot \mathbf{r}_n$; ▷ Build known term of eq.13
13: **end for**
14: $\mathbf{c} = \mathbf{D}^{-1} \cdot \mathbf{t}$; ▷ Solve the least-square problem (eq.13)
15: $\boldsymbol{\xi}_n = \mathbf{r}_n + \sum_i c_i \mathbf{w}_i$; ▷ Compute the modified residual (eq.15)

5. Application to nonlinear systems

Let us consider the nonlinear problem

$$\mathbf{F}(\mathbf{x}) = 0, \tag{17}$$

where $\mathbf{F}$ represents a generic nonlinear operator (for example the Navier-Stokes equations). Linearization around the current solution $\mathbf{x}_n$ of eq. (17) provides the linear system

$$\partial \mathbf{F}/\partial \mathbf{x} \; (\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{F}(\mathbf{x}_n). \tag{18}$$

Equation (18) can be framed as an instance of equation (1) by defining $\mathbf{A} = \partial \mathbf{F}/\partial \mathbf{x}$. If the matrix $\mathbf{A}$ is inverted exactly by a direct or iterative algorithm, eq. (18) represents the classical Newton method, while if a

single iteration of an iterative solver is applied, we recover a quasi-Newton method. Every linear iterative algorithm can thus be converted into a quasi-Newton method. However, there is a fundamental difference between linear and nonlinear problems: matrix $\mathbf{A}$, i.e., the Jacobian matrix of the system, changes during the iterative procedure in the case of nonlinear problems. Classical Krylov methods are not able to take such changes into account. As the Krylov basis grows in size at each iteration, the oldest vectors acquired become inadequate to represent the continuously changing $\mathbf{A}$ matrix. In addition to round off error, this is the main reason that makes a restarting procedure necessary. However, restarting involves the ad hoc choice of new numerical parameters such as how often and for what fraction of the basis to restart. `BoostConv`, on the other hand, continuously adapts to a changing $\mathbf{A}$ by simply discarding the oldest vector at each iteration.

6. Numerical Results

We now apply the procedure to the Ginzburg-Landau equation and a set of fluid problems in order to show its performance.

6.1. Ginzburg-Landau: a low-dimensional system

In this section, we will discuss the application of the proposed procedure to a simple low-dimensional system described by the Ginzburg-Landau equation. Such equation has been widely studied and used to model vortex shedding phenomena in the wake of bluff bodies (see e.g. Cossu & Chomaz[21], Chomaz[22]). Following Chomaz et al.[23], Chomaz[22], Bagheri et al.[24], we write the Ginzburg-Landau equation as

$$\frac{\partial \mathcal{A}}{\partial t} + \nu \frac{\partial \mathcal{A}}{\partial x} - \gamma \frac{\partial^2 \mathcal{A}}{\partial x^2} - \mu(x)\mathcal{A} + |\mathcal{A}|^2 \mathcal{A} = 0. \tag{19}$$

This equation is of convection-diffusion type and is characterized by a complex convection coefficient $\nu = U + 2ic_u$ and a diffusion coefficient $\gamma = 1 + ic_d$. In order to model non-parallel flows, Hunt & Crighton[25] and Bagheri et al.[24] chose μ as a quadratic function: $\mu(x) = (\mu_0 - c_u^2) + \mu_2 x^2/2$. Further details about this equation can be found in Bagheri et al.[24], where the meaning of each term is carefully explained.

Here, we focus our attention on the system dynamics for $\gamma = 1 - i$, $\mu_0 = 0.52$, $\mu_2 = -0.01$ and $\nu = 2 + 0.2i$; the system behavior, in this region of the parameter space is periodic. A fourth order Runge-Kutta scheme is

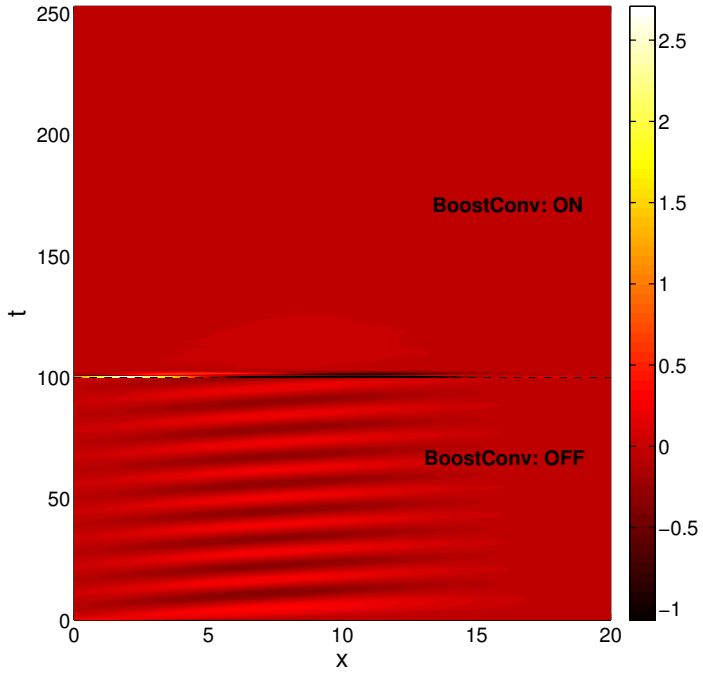

Figure 3: Stabilization of the Ginzburg-Landau system by using `BoostConv`. In the lower part of the figure (up to time $t = 100$), the system evolves to a limit cycle. At $t = 100$, we applied the `BoostConv` algorithm that is able to rapidly stabilize the system recovering its fixed point.

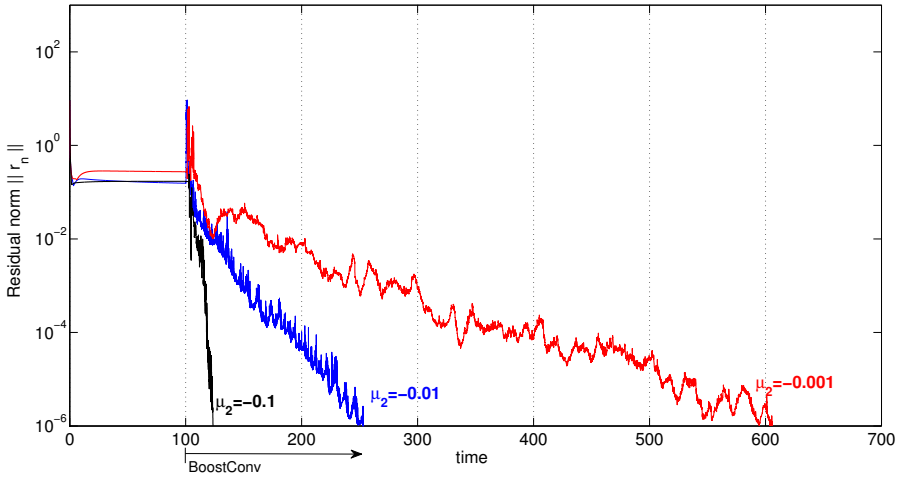

Figure 4: Stabilization of Ginzburg-Landau equation by using `BoostConv`. We show the evolution of the norm of the residual as a function of time. In particular, we focus our attention on the effect of non-normality of the system by changing μ_2[24]. The solution for $\mu_2 = -0.01$ is depicted in figure 3.

used here to march the equation in time. Figure 3 shows the application of `BoostConv` algorithm to recover the fixed point. In the first part of the simulation ($t < 100$) the system naturally evolves towards a limit cycle. Once a saturated periodic solution is reached, we apply `BoostConv` ($t > 100$) to our time-integration scheme: results shows that the new stabilization procedure is able to rapidly recover the fixed point of the Ginzburg-Landau equation. The results presented in this section are obtained by using a constant time step $\Delta t = 0.001$ and a Krylov basis of dimension $N = 15$.

Figure 4 shows the evolution of the norm of the residual for different cases. The parameter μ_2 is changed to investigate the effect of the system non-normality [24] on the stabilization procedure. This parameter has been varied from -10^{-1} to -10^{-3}, passing from a moderately to a highly non-normal system. We found that `BoostConv` is always able to stabilize the system but, as expected, the non-normality of the operator influences the convergence rate of the algorithm.

6.2. Acceleration of a stable procedure: the lid-driven cavity flow

The flow inside a lid-driven cavity has been extensively studied in the last decades and it is usually taken as a benchmark solution for CFD problems

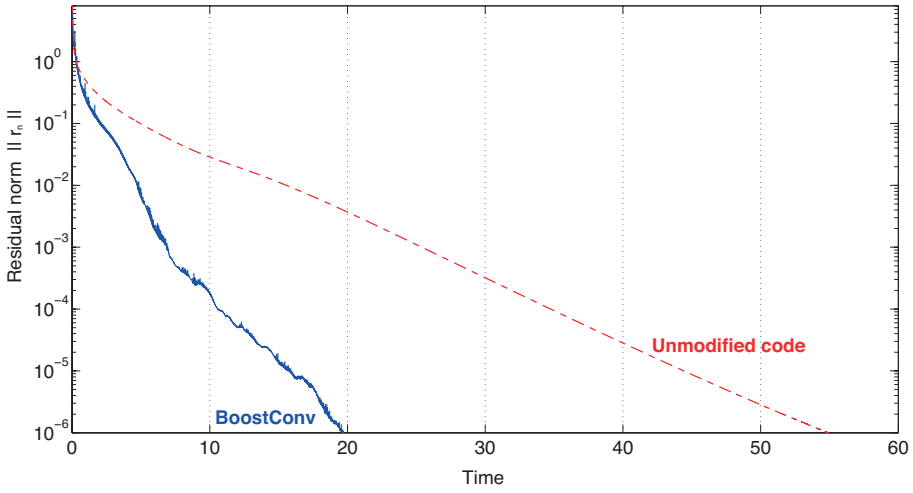

Figure 5: Two-dimensional flow inside a lid-driven cavity at $Re = 500$. In particular, we show the evolution of the residual norm as a function of the time.

(see e.g. [26],[27]). This configuration presents a singularity at the corners where the lid moves. Recently, Auteri *et al.*[28] obtained an accurate solution of this flow by using a second-order spectral projection method. The critical Reynolds number (based on the lid velocity U_{lid} and the height of the square cavity H) for the first Hopf bifurcation of this flow was calculated to be $Re_{cr} \approx 8018$.

The `BoostConv` algorithm is applied to compute the steady state at $Re = 500$. At this Reynolds number the flow is stable and a classical time integration converges towards a steady state solution. Our aim is to show the effect of the present algorithm on the convergence rate of the existing iterative procedure. We use the Spectral element code *Nek5000* to accurately solve the governing equations. The results presented in this section are obtained by selecting in *Nek5000* a first order temporal scheme and a constant time step $\Delta t = 0.001$, while the dimension of the `BoostConv` basis is set to $N = 15$.

Figure 5 shows the evolution of the residual as a function of time ($t = k\Delta t$, where k is the number of time steps). We observe that the residual norm obtained from the unmodified time-integration procedure, depicted using a dashed line, reaches the target value of 10^{-6} at $t \approx 54$. The solid line, on the other hand, represents the evolution of the residual when `BoostConv` is

used. We note that the application of the proposed algorithm accelerates the convergence of the procedure by a factor of ≈ 2.85. Finally, we recall that the introduction of `BoostConv` implies only the solution of a linear system associated with the least-squares method: consequently, the computational time of each time step is increased by only 5%.

6.3. Two-dimensional flow past a circular cylinder

The problem of viscous incompressible flow past a circular cylinder has received great attention both from a theoretical and a numerical viewpoint [2]. At low Reynolds number, the steady flow is symmetric and is characterized by a small recirculation bubble behind the cylinder. When the Reynolds number based on the cylinder diameter D exceeds the critical value of $Re^I_{cr} = 46.7$ the flow becomes unstable and a periodic Von-Kármán vortex street appears [29]. A linear stability analysis performed on the unstable steady base flow shows the existence of a second unstable wake mode for $Re > Re^{II}_{cr} = 110.8$.

In the present section, we consider the flow at the (supercritical) Reynolds number of $Re = 120 > Re^{II}_{cr}$. The natural evolution of the governing equations produces a saturated limit cycle depicted in figure 6a). We used `BoostConv` to compute the base flow in three-different codes: 1) a time-marching finite-difference immersed boundary code [29] (time integration is achieved using a semi-implicit CN-RK3 method (explicit third order Runge-Kutta for the convection term and implicit Crank-Nicolson scheme for the diffusion term); 2) the spectral element (SEM) code *Nek5000* [30] (both first order and second order time integration scheme have been tested with success); 3) the finite-element code Freefem++ [31],[32] (the characteristic-Galerkin method has been used for the time integration). As in the previous section, for each code, the modification of the existing time integration procedure consists in a single line of code that contains the call to `BoostConv`. All the results presented in this section are obtained by using a time step equal to $\Delta t = 0.001$ that corresponds for the adopted spatial discretizations to a Courant-Friedrichs-Lewy (CFL) number of ≈ 0.4 for all the simulations performed using the three different codes. We chose to adopt a `BoostConv` basis of 15 vectors. In general, the choice of the basis dimension and the time step Δt depends on particular problem considered. In fact, as mentioned before, the convergence rate of the code is dictated by the structure of the spectrum.

Figure 7 shows the evolution of the norm of the residual ($||\mathbf{r}_n||$) computed using Nek5000 as function of time. The blue line represents the case in

Table 1: Application of *BoostConv* to stabilize the flow past a circular cylinder at $Re = 120$. We report the effect of the basis dimension N and the time step Δt on convergence of the code. We report the computational time obtained running $Nek5000$ on an $Intel\ i7-6700HQ$ (4 cores, 3.50 GHz).

Basis Dim.	Δt	Computational time
$N = 15$	0.0001	3702.3 s
$N = 15$	0.0005	703.9 s
$N = 15$	0.001	419.7 s
$N = 15$	0.00125	376.6 s
$N = 50$	0.001	489.1 s
$N = 10$	0.001	407.0 s
$N = 5$	0.001	404.3 s

which the standard time integration procedure is not modified (leading to the saturated limit cycle). For this case, we show that the algorithm is able to compute the base flow starting from two different initial conditions: 1) a uniform flow; 2) the saturated limit cycle. We note that the residual target value is obtained for both the initial conditions at $t \approx 170$. We computed also the unstable base flow by using the SFD algorithm. Figure 7 clearly shows that `BoostConv` is able to obtain the stabilized flow faster than the SFD algorithm: the resulting base flow is depicted in Figure 6b).

The velocity and pressure distributions of the computed steady flow along the vertical line $x = D$ are shown in figure 8. An equivalent mesh resolution is adopted for all codes. We note that each code converges (when `BoostConv` is called during the time integration) to the base flow computed by using a Newton method which serves as a reference solution.

Table 1 shows the effect of the basis dimension N and time step Δt on the convergence. We note that a large time step is fundamental to quickly get the unstable base flow. On the other hand, the basis dimension N weakly affects the convergence rate of the code. It must be stressed, however, that the effect of these parameters on the convergence rate strongly depends on the problem.

6.4. High-dimensional problem: three-dimensional DNS of a boundary layer flow over a hemispherical roughness element

As discussed in the previous sections, `BoostConv` is conceived to stabilize the dynamical system without a negative impact on the computational

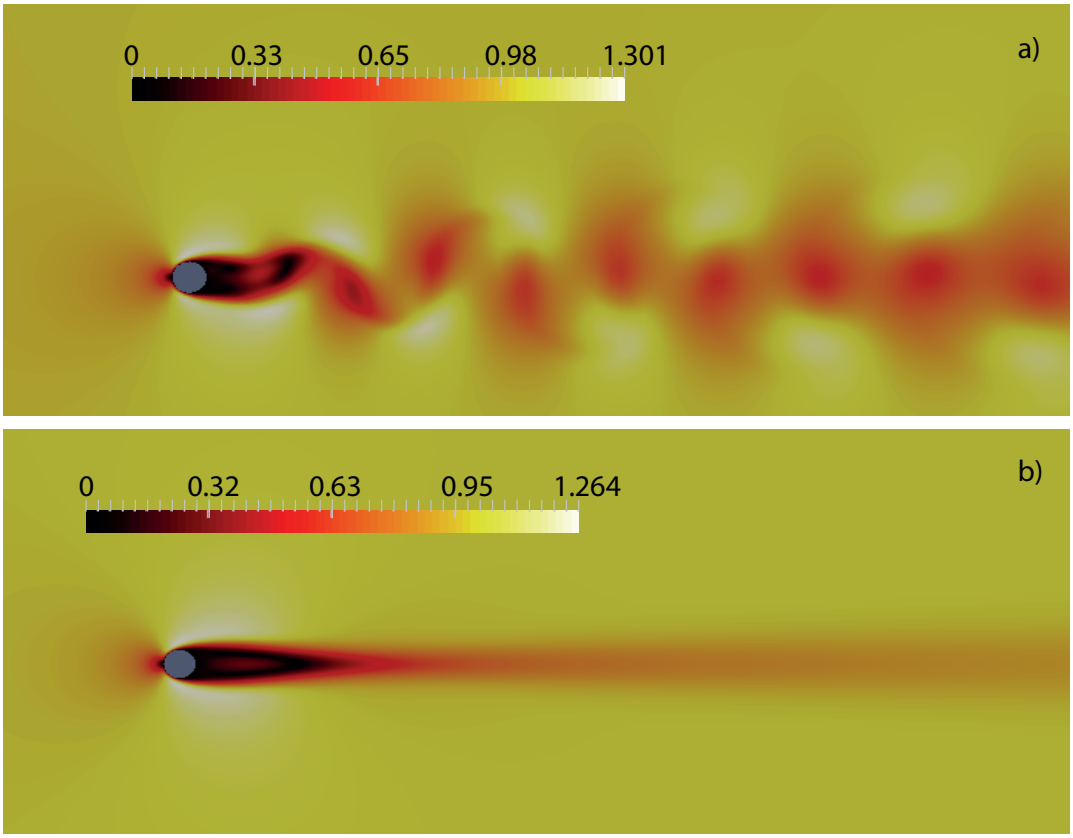

Figure 6: Two-dimensional flow past a circular cylinder at $Re = 120 > Re_{cr}^{II}$. a) Snapshot of the saturated limit cycle (unmodified integration of Navier-Stokes equations). b) Base flow: stabilized simulation using `BoostConv`. The evolution of the residual norm associated with these simulations is depicted in Figure 7.

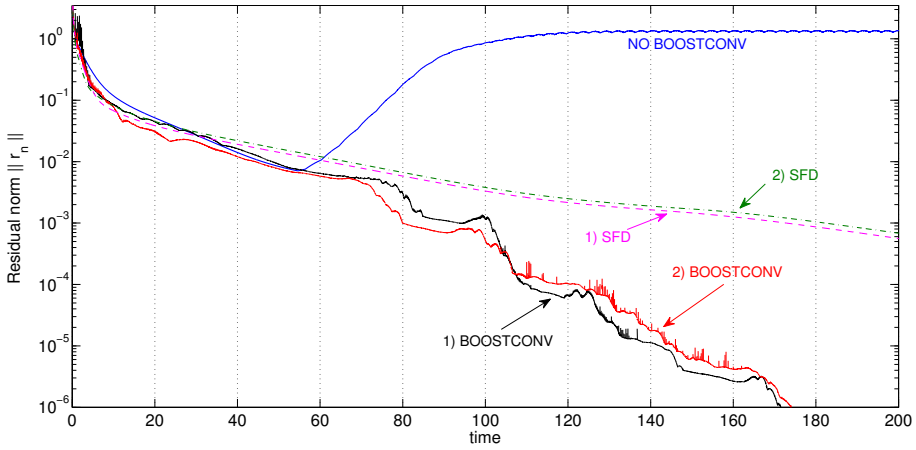

Figure 7: Stabilization of the cylinder flow by using `BoostConv` at $Re = 120$. We started from two different initial conditions: 1) saturated limit cycle and 2) uniform flow in the streamwise direction. We carried out the stabilization of this flow also by the SFD algorithm [13]: 1) $\chi = \omega_c = 0.3$; 2) $\chi = \omega_c = 0.5$. The flow simulations are carried out by using the code *Nek5000*.

burden of the simulation. In this section we show the application of this algorithm to a high-dimensional test case that would be infeasible with a matrix-based method.

In particular, we consider the three-dimensional flow past a hemispherical roughness element immersed in a laminar Blasius boundary layer. The computational domain is discretized by approximately 24 million points. This geometrical configuration is the same chosen by Klebanoff et al. [33] in their experimental investigations. As considered by Tani et al.[34] and Citro et al.[35], the Navier-Stokes equations are made dimensionless using the total height k of the roughness element as the characteristic length scale and the velocity U_k of the incoming uniform stream that would exist in the boundary layer at the height k if the roughness element is absent. The resulting Reynolds number can be written as $Re_k = U_k k/\nu$, with ν being the kinematic viscosity of the fluid. The other important parameter that influences the flow dynamics is the ratio between the displacement thickness δ_k^* of the incoming boundary layer and the height k of the hemisphere.

For this test case, we chose $k/\delta_k^* = 2.62$ and $Re_k = 450$ that are the same

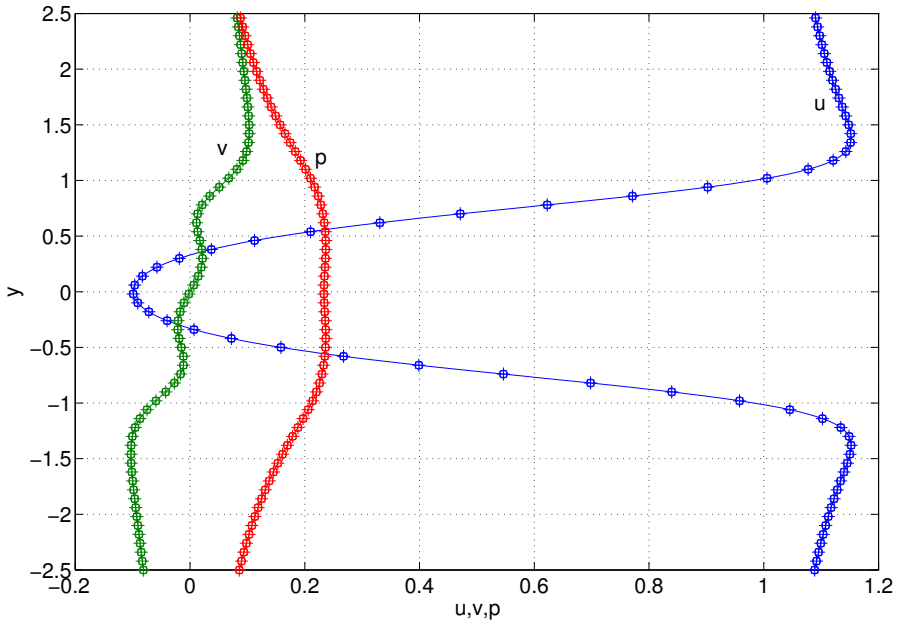

Figure 8: Velocity and pressure distributions in the wake of a circular cylinder at $x = D$ ($Re = 50$). Here, we compare the solutions obtained using (o) finite-difference immersed-boundary code, ($+$) spectral element code (*Nek5000*) and ($\square$) finite element (time-marching) code. The profiles depicted using solid lines ($-$) are obtained using the Newton method.

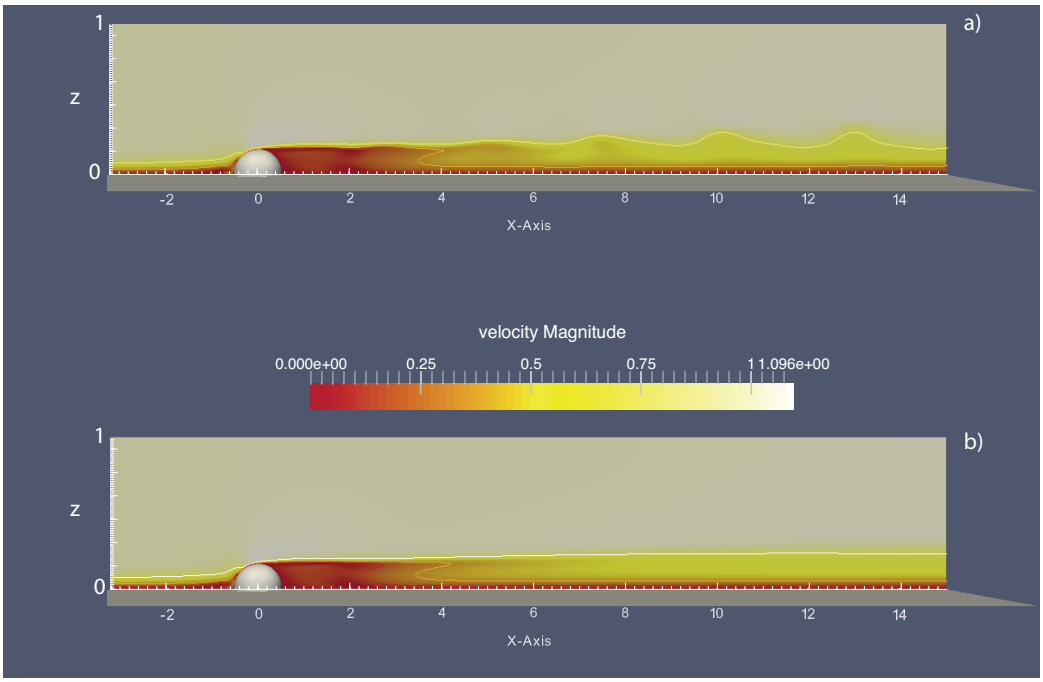

Figure 9: Three-dimensional flow past a hemispherical roughness element: contour plot of the velocity magnitude. Perspective view of a) unsteady supercritical flow (without the application of `BoostConv`) and b) stabilized flow obtained by using the `BoostConv` algorithm. Parameter settings: $Re_k = 450$ and $k/\delta_k^* = 2.62$.

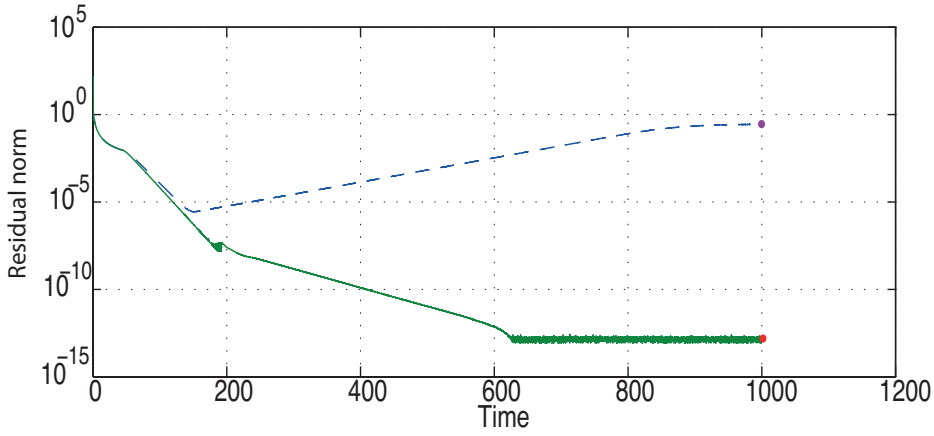

Figure 10: Flow past a hemispherical roughness element. Solid line (−): convergence history of `BoostConv`. Dashed line (−−): time integration of the governing equations.

conditions in which Klebanoff et al. [33] documented their experimental results. As also shown by Citro et al.[35], the inherent three-dimensional flow pattern beyond the Hopf bifurcation is characterized by coherent vortical structures called *hairpin vortices* that periodically detach from the hemisphere. A contour plot of a snapshot of the resulting periodic flow is depicted in Figure 9a) by taking a mid-plane slice. In this side view of the unsteady supercritical flow, we plotted the iso-contours of the velocity magnitude. As discussed before for the cylinder and the lid-driven cavity cases, we plot the evolution of the residual as a function of time for both the original time-stepper (dashed line) and for the modified procedure involving `BoostConv`. The convergence history is reported in figure 10. For this case, we performed a DNS using the SEM code *Nek5000*: the dimension of the basis used is $N = 15$ (see section 3) and the time step is $\Delta t = 0.001$. The initial field is a uniform flow as for the case of the circular cylinder. The run was performed on an IBM BG/Q Supercomputer at CINECA and we found that the stabilization procedure increased the simulation time by about 9%. The resulting base flow is shown in figure 9b).

7. Stabilization of periodic orbits

Finally, we present an example in which `BoostConv` is used to stabilize periodic unstable orbits. We consider the two-dimensional flow past two side-by-side circular cylinders. This flow can be considered a prototype to investigate wake interference phenomena past bluff bodies. The bifurcation scenario is strongly influenced by the gap between the cylinders [36]. In this example, we focus on the gap value $g = 0.7D$. The flow pattern, at $Re = 62$, is characterized by two asymmetric unsteady wakes deflecting alternatively toward one of the cylinders. The analysis performed by Carini et al.[36] demonstrated that the flip-flopping state, developing at this Reynolds number, can be considered as an instability of the in-phase synchronized vortex shedding. As a consequence, the in-phase base flow cannot be simply recovered by using a standard time integration of governing equations. Figure 11 shows the results of the application of `BoostConv` to this case to compute the unstable periodic orbit. In particular, Figure 11a) shows the stabilization of the value of Strouhal number as a function of the time-integration periods. The abatement of the infinity norm of the residual is depicted in Figure 11b). Snapshots of the stabilized periodic field are reported in Figure 12.

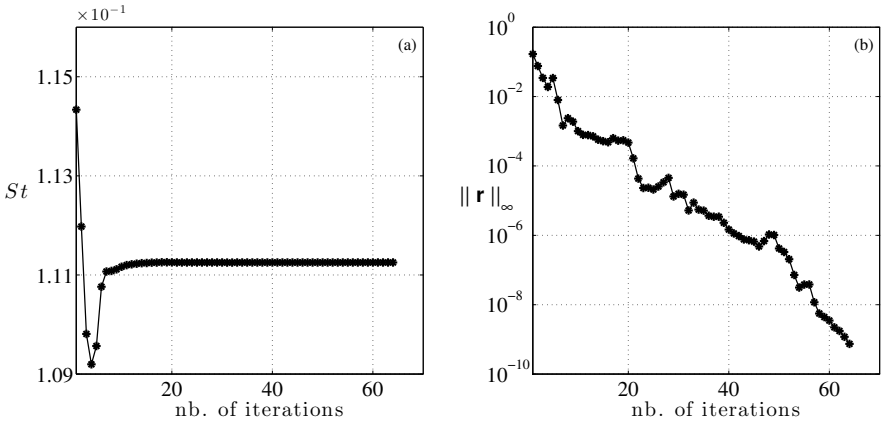

Figure 11: Stabilization of the flow past two side-by-side circular cylinders. The gap between the cylinders is $g = 0.7D$ and the Reynolds number is $Re = 62 > Re_{cr}$. We depict (a) the resulting Strouhal number $St = 1/T$ and (b) the infinity norm of the residual on the periodicity condition $||\mathbf{r}||_\infty$ as a function of the number of iterations (periods involved in the computation).

8. Conclusions

In this paper we presented a novel, efficient and easy-to-implement algorithm inspired by the Krylov-subspace projection methods, which is able to compute unstable steady states of any dynamical system. `BoostConv` is based on the least-squares minimization of the residual norm at each step of an existing iterative procedure. The key idea of the proposed method is to invert only the small part of the problem represented by the dominant and slower decaying/growing modes exactly, while letting the original iterative algorithm handle the remaining modes. As a consequence, the proposed algorithm is very efficient because the linear system that has to be inverted at each time step is small compared with the dimension of the problem. From a programming viewpoint, `BoostConv` can be encapsulated in a black-box routine where the only required input is the residual $\mathbf{r}_n$ and the only output is the modified residual $\boldsymbol{\xi}_n$. We underline that the only modification necessary to stabilize or boost the convergence of a pre-existing iterative algorithm is a single line of code containing the call to `BoostConv`.

We also report numerical results obtained with the new procedure. First

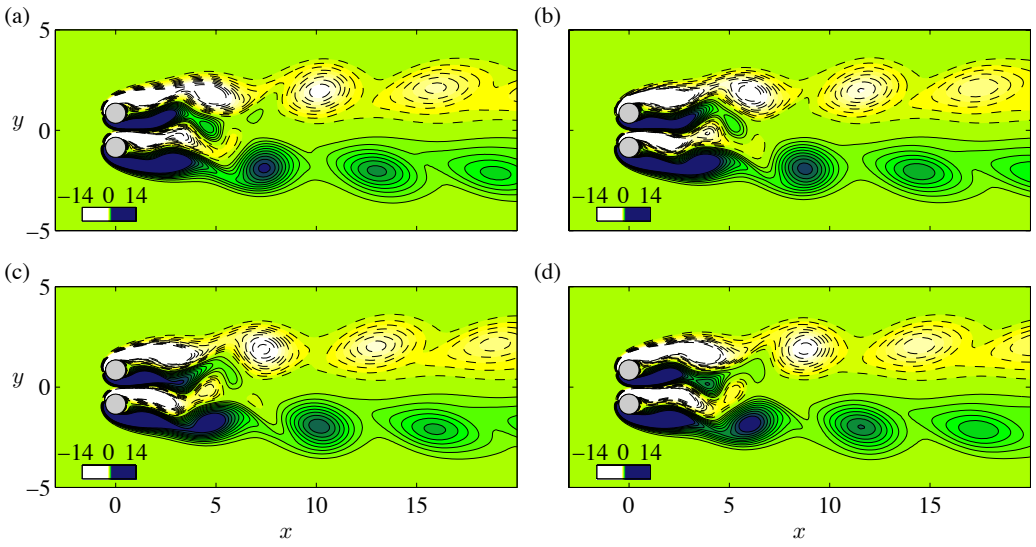

Figure 12: Vorticity snapshots of the periodic stabilized base flow using `BoostConv`. We equally divided the shedding period and reported here the resulting phases: (a) $t = T/4$; (b)$t = T/2$; (c)$t = 3T/4$; (d)$t = T$. Parameter settings as in Fig. 11: $g = 0.7D$ and $Re = 62$.

of all, we applied `BoostConv` to the low dimensional system consisting of the Ginzburg-Landau equation. We showed that the `BoostConv` is always able to stabilize the system but the non-normality can play an important role in the convergence rate. The classical case of the two-dimensional lid-driven cavity flow has also been considered. At a subcritical Reynolds number, we showed that `BoostConv` is able to accelerate the convergence of the existing time integration procedure. Moreover, we dealt with the two-dimensional flow past an infinitely long circular cylinder showing, with several different codes, that `BoostConv` is able to drive the iterative procedure to the base flow computed using Newton's method. We note that the initial condition, did not affect the stabilization performance. A three-dimensional case has also been considered to demonstrate the application of `BoostConv` to a high-dimensional problem. In the case of the flow past a hemispherical roughness element immersed in a laminar Blasius boundary layer, we found that the resulting modification of the time integration does not significantly affect the computational burden of the simulation. Furthermore, we showed that the proposed algorithm can be used also to stabilize unstable periodic orbits. In

fact, recently, thanks to the application of this procedure, Carini et al.[36] were able to explain the flip-flop mechanism in the flow past two side-by-side circular cylinders. Finally, we note that `BoostConv` has been recently applied to compute unstable states in presence of a stationary bifurcation [37].

Unlike Krylov methods such as GMRES, `BoostConv` takes into account for the change of the system Jacobian during the time evolution of the dynamical system. We underline also that the proposed algorithm can be combined with any spatial discretization method and can be used to stabilize a dynamical system even in presence of bifurcations with higher codimension.

[1] Drazin, P., Reid, W. *Hydrodynamic Stability*, Cambridge University Press, 1981

[2] Williamson, C.H.K. *Vortex Dynamics in the Cylinder Wake*, Annu. Rev. Fluid Mech., Vol. 28, pp. 477-539, 1996

[3] Govaerts, W.J.F. *Numerical Methods for Bifurcations of Dynamical Equilibria*, Soc. for Industr. & Appl. Math., 2000

[4] Doedel, E. *AUTO: Software for continuation and bifurcation problems in ordinary differential equations*, Report Applied Mathematics, California Institute of Technology, Pasadena, USA, 1986

[5] Kuznetsov, Y.A., Levitin, V.V. *CONTENT, a multiplatform continuation environment*, Technical report, Amsterdam, The Netherlands, 1996

[6] Lust, K., Roose, D., Spence, A., Champneys, A.R. *An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions*, SIAM J. Sci. Comput., Vol. 19, pp. 1188-1209, 1998

[7] van Noorden, T.L., Verduyn, S.M., Bliek, A. *The efficient computation of periodic states of cyclically operated chemical processes*, IMA J. Appl. Math., Vol. 68, pp. 149-166, 2003

[8] Shroff, G.M., Keller, H.B. *Stabilization of unstable procedures: the recursive projection method*, SIAM J. Numer. anal., Vol. 30, pp. 1099-1120, 1993

[9] van Noorden, T.L., Verduyn, L., Bliek, A. *A Broyden Rank $p+1$ Update Continuation Method with Subspace Iteration*, SIAM J. Sci. Comput., Vol. 25, pp. 1921-1940, 2004

[10] Sanchez, J., Net, M., Garcia-Archilla, B., Simo, C. *Newton-Krylov continuation of periodic orbits for Navier–Stokes flows*, J. Comput. Phys., Vol. 201, pp. 13-33, 2004

[11] Sanchez, J., Marques, F., Lopez, J.M. *A continuation and Bifurcation Technique for Navier-Stokes Flows*, J. Comput. Phys., Vol. 180, pp. 78-98, 2002

[12] Mittelmann, H.D., Weber, H. *Multi-grid solution of bifurcation problems*, SIAM J. Sci. Stat. Comput., Vol. 6, 1985

[13] Akervik, E., Brandt, L., Henningson, D.S., Hoepffner, J., Marxen, O., Schlatter, P. *Steady solutions of the Navier–Stokes equations by selective frequency damping*, Phys. of Fluids, Vol. 18, pp. 068102, 2006

[14] Bagheri, S., Schlatter, P., Schmid, P.J., Henningson, D.S. *Global stability of a jet in crossflow*, J. Fluid Mech., Vol. 624, pp. 33–44, 2009

[15] Ilak, M., Schlatter, P., Bagheri, S., Henningson, D.S. *Bifurcation and stability analysis of a jet in cross-flow: onset of global instability at a low velocity ratio*, J. Fluid Mech., Vol. 696, pp. 94–121, 2012

[16] Nichols, J.W., Schmid, P.J. *The effect of a lifted flame on the stability of round fuel jets*, J. Fluid Mech., Vol. 609, pp. 275–284, 2008

[17] Qadri, U.A., Chandler, G.J., Juniper, M.P. *Self-sustained hydrodynamic oscillations in lifted jet diffusion flames: origin and control*, J. Fluid Mech., Vol. 775, pp. 201–222, 2015

[18] Simoncini, V., Szyld, D.B. *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., Vol. 14, pp. 1–59, 2007

[19] Saad, Y., Schultz, M.H. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., Vol. 7, pp. 856–869, 1986.

[20] Saad, Y., *Iterative Methods for Sparse Linear Systems (2nd edn)*, SIAM, 2003.

[21] Cossu, C., Chomaz, J. M., *Global measures of local convective instabilities*, Phys. Rev. Lett., 78(23), 4387, 1997.

[22] Chomaz, J. M., *Global Instabilities in Spatially Developing Flows: Non-Normality and Nonlinearity*, Annu. Rev. Fluid Mech., Vol. 37, pp. 357–392, 2005.

[23] Chomaz, J. M., Huerre, P., and Redekopp, L. G., *The Effect of Nonlinearity and Forcing on Global Modes*, New Trends in Nonlinear Dynamics and Pattern-Forming Phenomena NATO Advanced Series Institute, Series B: Physics, P. Coullet and P. Huerre, eds., Springer, New York, pp. 259–274.

[24] Bagheri, S., Henningson, D. S., Hoepffner, J., Schmid, P. J., *Input-output analysis and control design applied to a linear model of spatially developing flows*, Appl. Mech. Rev., Vol. 62, 020803, 2009.

[25] Hunt, R. E., Crighton, D. G., *Instability of Flows in Spatially Developing Media*, Proc. R. Soc. London, Ser. A, Vol. 435, pp. 109–128, 1991.

[26] Ghia, U., Ghia K.N., Shin, C.T. *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys., Vol. 48, pp. 387–411, 1982

[27] Schreiber R., Keller H.B., *Driven cavity flows by efficient numerical techniques*, J. Comput. Phys., Vol. 49, pp. 310–333, 1983

[28] Auteri F., Parolini N., Quartapelle L., *Numerical investigation on the stability of singular driven cavity flow*, J. Comput. Phys., Vol. 183, pp. 1–25, 2002

[29] Giannetti, F., Luchini, P. *Structural sensitivity of the first instability of the cylinder wake*, J. Fluid Mech., Vol. 581, pp. 167–197, 2007

[30] Fischer, P.F. *An overlapping schwarz method for spectral element solution of the incompressible Navier-Stokes equations*, J. Comput. Phys., Vol. 133, pp. 84–101, 1997

[31] Hecht, F. *New development in FreeFem++*, J. Num. Math., Vol. 20, pp. 251–265, 2012

[32] Citro, V., Giannetti, F., Brandt, L., Luchini, P. *Linear three-dimensional global and asymptotic stability analysis of incompressible open cavity flow*, J. Fluid Mech., Vol. 768, pp. 113–140, 2015

[33] Klebanoff, P.S., Cleveland, W.G., Tidstrom, K.D. *On the evolution of a turbulent boundary layer induced by a three-dimensional roughness element*, J. Fluid Mech., Vol. 237, pp. 101–187, 1992

[34] Tani, I., Komoda, H., Komatsu, Y., Iuchi, M. *Boundary-Layer Transition by Isolated Roughness*, Aeronautical research institute, Internal report, Vol. 375, pp. 129–143, 1962

[35] Citro, V., Giannetti, F., Luchini, P., Auteri, F. *Global stability and sensitivity analysis of boundary-layer flows past a hemispherical roughness element*, Phys. Fluids, Vol. 27, pp. 084110, 2015

[36] Carini, M., Giannetti, F., Auteri, F. *On the origin of the flip-flop instability of two side-by-side cylinder wakes*, J. Fluid Mech., Vol. 742, pp. 552–576, 2014

[37] Citro, V., Tchoufag, J., Fabre, D., Giannetti, F., Luchini P. *Linear stability and weakly nonlinear analysis of the flow past rotating spheres*, J. Fluid Mech., Vol. 807, pp. 62–86, 2016