Improving the Numerical Simulation of an Airflow Problem with the BlockCGSI Algorithm

C. Balsa[1], M. Braza[2], M. Daydé[3], J. Palma[1], and D. Ruiz[3]

[1] FEUP, Porto, Portugal
{cbalsa, jpalma}@fe.up.pt
[2] IMFT–CNRS, Toulouse, France
braza@imft.fr
[3] ENSEEIHT–IRIT, Toulouse, France
{dayde, ruiz}@enseeiht.fr

Abstract. Partial spectral information associated with the smallest eigenvalues can be used to improve the solution of successive linear systems of equations, namely in the simulation of time-dependent partial differential equations, where at each time step there are several systems with the same spectral properties to be solved. We propose to perform a partial spectral decomposition with the BlockCGSI algorithm in the first time step, and exploit this information to improve the convergence of the Conjugate Gradient algorithm in the solution of the following linear systems. We describe in summary the BlockCGSI algorithm, that is a combination of the block Conjugate Gradient (blockCG) with the Inverse Subspace Iteration. Then, we validate the accelerating strategy in the simulation of the flow around an airplane wing, where the Conjugate Gradient is accelerated through the deflation of the starting residual.

1 Introduction

Partial spectral information associated with the smallest eigenvalues can be used to improve the solution of successive linear systems of equations, namely in the simulation of time-dependent partial differential equations, where at each global iteration there are several systems with the same spectral properties to be solved. We propose a two-phase acceleration technique, where in the first phase we perform a partial spectral decomposition of the system solved in the first global iteration, with the BlockCGSI algorithm. In a second phase we exploit this information to improve the convergence of the Conjugate Gradient algorithm in the solution of the following linear systems.

The two-phase acceleration strategy has been initially proposed in the experimental work in Ref. [1]. In the first phase, the BlockCGSI algorithm computes a *near*-invariant subspace associated with the smallest eigenvalues, and in the second phase this spectral information is used to deflate the eigencomponents associated with the smallest eigenvalues with an appropriate starting guess. We concluded [2] that this strategy has a good potential to reduce the computing

time of a fluid flow simulation algorithm, and the success of this approach depends on the appropriate monitoring of the BlockCGSI algorithm that combines the blockCG iterative solver with the Subspace Iteration. In [3], we analyzed the convergence of the BlockCGSI algorithm from an inner-outer iteration point of view. We establish how the eigenvalue error of the Subspace Iteration varies along the inner iteration, when the system is solved with the blockCG algorithm. In agreement with these results, we proposed an appropriate stopping criterion for each level of iteration, that enables to reduce the computational costs.

In the present work we validate the two phase accelerating strategy in the simulation of a flow around an airplane wing (see [4]). Firstly, in section 2, we describe in short, the BlockCGSI algorithm, that is a combination of the block Conjugate Gradient (blockCG) [5, 1] with the Subspace Iteration [6]. We analyze the computational costs involved in the two phases, as a function of the dimension of the computed *near*-invariant subspace, and compute the *a posteriori* optimal dimension (section 3.1). Based on these results, we propose a strategy for choosing dynamically the dimension of the basis, that does not need *a priori* informations about the spectrum of the coefficient matrix. We conclude (section 3.2), showing the benefits resulting from the application of the two-phase approach to present airflow problem.

2 The BlockCGSI algorithm

The BlockCGSI algorithm is used to compute an M-orthonormal basis W of a *near*-invariant subspace associated with the smallest eigenvalues in the preconditioned matrix $M^{-1}A$. If this basis incorporates, for instance, all the eigenvalues of $M^{-1}A$ in the range $[0, \mu]$, we can expect, when using it later as a second level of preconditioning, that the condition number of the coefficient matrix will be reduced to about $\kappa = \lambda_{\max}/\mu$, where $\lambda_{\max}$ is the largest eigenvalue in $M^{-1}A$. In Algorithm 1, $\lambda_{\max}$ and μ are considered as input parameters (a rough upper bound on $\lambda_{\max}$ is usually enough). Another input concerns the choice of the block size s that defines the dimension of the working subspace at each inverse iteration; it also gives the number of right-hand sides and solutions vectors of the multiple linear systems solved by the blockCG algorithm, and consequently the amount of memory required as working space.

As a starting point, the algorithm requires the generation of an M-orthonormal basis of size s. The closer are these vectors to the targeted *near*-invariant subspace, the faster will be the convergence of the inverse iteration. The scope of steps 1 to 4 in Algorithm 1, is to generate an initial M-orthonormal set $V^{(0)}$ of s vectors with eigencomponents corresponding to eigenvalues in the range $[\mu_f, \lambda_{\max}]$ below some predetermined value $\xi \ll 1$ (denoted as the "filtering level"). This filtering technique is based on Chebyshev polynomials (step 3) and details about it can be found in [3]. The idea behind the use of these Chebyshev filters at the starting point is to put the inverse subspace iteration in the situation of working in the orthogonal complement of a large number of eigenvectors, e.g. all those associated with the eigenvalues in the range $[\mu_f, \lambda_{\max}]$.

Algorithm 1: BlockCGSI algorithm

Inputs: $A, M = R^T R \in \mathbb{R}^{n\times n}, \mu, \lambda_{max} \in \mathbb{R}, s \in \mathbb{N}$
Output: a *near*-invariant subspace W associated with all eigenvalues in the range $]0, \mu]$

Begin

Generate the initial subspace (with filtering)

1. $Z^{(0)}$ =RANDOM(n, s)
2. $Y^{(0)} = R^{-1}Z^{(0)}\Psi$ such that ${Y^{(0)}}^T M Y^{(0)} = I_{s\times s}$
3. $Q^{(0)}$=Chebyshev-Filter$(Y^{(0)}, \xi, [\mu_f, \lambda_{\max}], A, R)$
4. $V^{(0)} = Q^{(0)}\Gamma$ such that ${V^{(0)}}^T M V^{(0)} = I_{s\times s}$
5. $W^{(0)}$ = empty
6. **For** $k = 1, ...,$ until convergence **Do:**

Orthogonal iteration

i. Solve $M^{-1}AZ^{(k)} = V^{(k-1)}$ with blockCG
ii. $P^{(k)} = Z^{(k)} - W^{(k-1)}{W^{(k-1)}}^T M Z^{(k)}$
iii. $Q^{(k)}\Gamma_k = P^{(k)}$ such that ${Q^{(k)}}^T M Q^{(k)} = I_{s\times s}$
iv. $Q^{(k)} = [W^{(k-1)}\ Q^{(k)}]$

Ritz acceleration

v. $\beta_k = {Q^{(k)}}^T A Q^{(k)}$
vi. Diagonalize $\beta_k = U_k \Delta_k U_k^T$
where $U_k^T = U_k^{-1}$
and Δ_k =Diag$(\delta_1, ..., \delta_{p+s})$ (Ritz Values)
vii. $V^{(k)} = Q^{(k)}U_k$ (Ritz Vectors)

Update the computational window

viii. $W^{(k)}$ = converged columns of $V^{(k)}$
ix. $V^{(k)}$ = non-converged columns of $V^{(k)}$
x. (n, p) = size$(W^{(k)})$
xi. Incorporate new vectors in $(V^{(k)})$

7. **EndDo**

End

We can also expect that the resulting filtered right-hand sides will present more favorable spectral properties that can improve the convergence behavior of the blockCG. Obviously, there is some compromise to achieve, in the sense that very small values of μ_f and ξ will minimize the number of inverse and blockCG iterations, but will also increase the computational efforts in the Chebyshev initial filtering step.

The essence of the inverse subspace iteration is the orthogonal iteration. It consists in multiplying a set of vectors by $A^{-1}M$ and M-ortonormalizing it in

turn. In step i, the multiplication by $A^{-1}M$ is performed implicitly through the iterative solution of the system $M^{-1}AZ^{(k)} = V^{(k-1)}$ via the blockCG. In order to reduce the computational costs, this system is solved with an accuracy determined by an appropriate residual threshold ε (for details see [3]). In step ii, the approximate solution vectors $Z^{(k)}$ are then projected onto the orthogonal complement of the converged vectors $W^{(k-1)}$, in order to remove the influence of eigencomponents associated with the converged eigenvalues. The set of projected vectors $P^{(k)}$ is then M-orthonormalized (step iii), and gathered together with $W^{(k-1)}$.

To improve the rate of convergence of the subspace iteration, the orthogonal iteration is followed by the Ritz acceleration (steps v to vii), as suggested by [6]. The spectral information contained in $Q^{(k)}$ is thus redistributed in the column vectors of $V^{(k)}$, that will contain each better approximations of individual eigenvectors. Steps v, vi, and vii, yield the Ritz values, $diag(\Delta) = \delta_1, ..., \delta_{p+s}$, ranged in increasing order, and the associated Ritz vectors, $[v_1, v_2, ..., v_p, ..., v_{p+s}]$, where p is the dimension of $W^{(k-1)}$ and s is the current block size.

The end of the BlockCGSI algorithm consists in testing the convergence and updating the computational window. In step viii, all the Ritz vectors that are considered as *near*-invariant (with respect to the given accuracy) are assigned to $W^{(k)}$ (more details are given in [3]). Step xi consists in incorporating new vectors in the current set of vectors $V^{(k)}$. The operation that consists in introducing a set of ℓ new vectors, after some of the Ritz vectors have converged, is detailed in Algorithm 2. We denote this algorithmic issue in the BlockCGSI algorithm as *"sliding window"*. Its purpose is to enable the approximation of a number of eigenvectors greater than the block size s. Basically, we generate randomly a linear combination of the filtered vectors $V^{(0)}$, generated in the starting steps (1 and 2) of the BlockCGSI algorithm. Then, these vectors are projected in the M-orthogonal complement of the converged ones, in order to remove the corresponding eigencomponents. Note that we can also opt to reduce or enlarge the block size s at this stage, when setting the value of ℓ (i.e. the number of newly incorporated vectors).

ALGORITHM 2: INCORPORATE NEW VECTORS

Inputs: $\ell \in \mathbb{N}, M = R^T R \in \mathbb{R}^{n\times n}, V^{(0)} \in \mathbb{R}^{n\times s}, W^{(k)} \in \mathbb{R}^{n\times p}, V^{(k)} \in \mathbb{R}^{n\times(s-\ell)}$

Begin

a) Y =RANDOM(s, ℓ)

b) $P = V^{(0)}Y$

c) $P = Q\Gamma$ such that $Q^T MQ = I_{\ell\times\ell}$

d) $P = Q - W^{(k)}{W^{(k)}}^T MQ$

e) $V^{(k)} = [V^{(k)}P]$

End

3 Some numerical experiments in an airflow problem

We present some numerical results concerning the exploitation of a *near*-invariant subspace W, with dimension q, associated with the eigenvalues of $M^{-1}A$ in the range $]0, \mu[$. The spectral information, computed with BlockCGSI algorithm, is used to improve an airflow simulation code of the flow around a wing (see details in [7, 4, 8]). The Navier-Stokes equations are solved by finite elements in a 2D field, through a prediction-correction algorithm and a semi-implicit time discretization scheme. To obtain all the physical structures of the flow, long periods of simulation ($T = 10$ or $T = 20$) are required. In each time step (typically 0.01 s) iteration, we need to solve a system of linear equation (Poisson type), with the same coefficient matrix and changing right-hand sides, of size $n = 27283$ and $nz = 187487$ non-zero elements.

3.1 Optimal dimension of the basis

After preconditioning, by means of the classical Incomplete Cholesky ($M = R^T R = IC(0)$), the spectrum is distributed from $\lambda_{\min} = 6.5e-05$ to $\lambda_{\max} = 1.7e+00$, which corresponds to a spectral condition number κ of order $2.6e+04$. After the preconditioning, there are still few eigenvalues on the left of the spectrum that are responsible for the non-linear convergence of the Conjugate Gradient. To remove these problems we propose to deflate this part of the spectrum through an initial projection on the CG algorithm. The spectral projector is built with the basis W of the *near*-invariant subspace computed with the BlockCGSI algorithm. We denote the technique that combines the Conjugate Gradient with the initial deflation as the Init-CG algorithm (see details in [3]). One open question is how many eigenvalues we must compute to improve the convergence of the Init-CG algorithm. For instance, if we want to reduce the condition number to 100, we need to cancel the effect of the 48 smallest eigenvalues ($\mu \approx 1.7e-02$). The desirable choice is that μ falls between two clusters.

The optimal dimension of the basis W will be the one that minimizes the total cost when solving all the systems during the simulation, with our two-phase approach. This cost is given by

$$\texttt{Total cost} = \mathcal{C}_{BCGSI} + \mathcal{C}_{InitCG} \times \texttt{NGits}, \tag{1}$$

where `NGits` is the number of global iteration (or time steps), i.e. the total number of systems to be solved. The cost of pre-computing the spectral information with the BlockCGSI algorithm is given by $\mathcal{C}_{BCGSI}$ and the cost of solving one system with Init-CG is given by $\mathcal{C}_{InitCG}$.

3.1.1 Pre-computational cost. The cost of pre-computing the spectral information depends on the dimension q of the basis W and on working parameters in the BlockCGSI algorithm, as for instance the block size s, the filtering level ξ, and the cut-off value μ_f for the filtering. In our experiments with the current test problem, the value of μ_f was automatically set as $\mu_f = \mu$, and ξ was fixed to

$1e-10$. As we have seen in [3], the filtering level is important but does not need to be very small to reduce substancialy the costs in the BlockCGSI algorithm. In an efficient implementation of the BlockCGSI algorithm, Level 3 BLAS kernels can be incorporated in order to maximize the Megaflops rate, and the value of s can also be determined only on the bais of such computer aspects, keeping in mind that the *sliding window* technique adjusts the dimension of the basis W automaticaly.

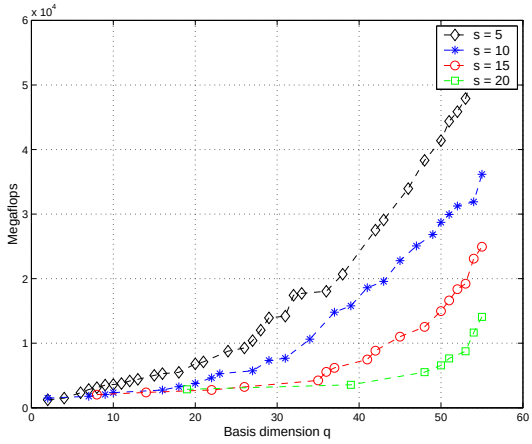


Fig. 1. Costs of pre-computing the *near*-invariant subspace ($\mathcal{C}_{BCGSI}$) for different block sizes s

Figure 1 displays the values of $\mathcal{C}_{BCGSI}$ for a block size s equal to 5, 10, 15 and 20. We can see that lower pre-computational costs are obtained with larger block sizes, specially for high dimensions of q. The principal reasons for that are the *guard vector* effect [3] and the costs of incorporating new vectors by Algorithm 2. As indicated in step b of Algorithm 2, we inject a random linear combination of the filtered starting vectors $V^{(0)}$ generated in step 4 of Algorithm 1. With this practical simplification we call the Chebyshev filtering routine only once and do not need to filter the newly incorporated vectors. The idea behind that, is that the starting vectors include already some information concerning all the eigenvalues in the range $]0, \mu[$, and to recover it we just need to redistribute this information over each of the newly incorporated vectors. In some cases, specially if we want an accurate spectral information, a breakdown can occur due to the *near*-collinearity of these new vectors relatively to the converged ones in $W^{(k)}$, which can be avoided if we force the blockCG to do a minimum number of iterations (for instance $i_{min} = 4$).

3.1.2 Solution cost. If we analyze now the behavior of the solution costs $\mathcal{C}_{InitCG}$, as shown in figure 2. At the beginning a large decrease of the costs until q reaches approximately the dimension 20, above which the improvements are minimal ($q_{sol} \approx 20$). This occurs because until q is lower than 20 we are interpolating the extremal eigenvalues of the cluster and after that we are in the middle of the cluster. As the basis is enlarged from $q = 20$ to $q = 30$ the value of μ is shifted from $6.63e-3$ to $1.02e-2$, which corresponds to a small reduction of the condition number from $\kappa = 2.54e+02$ to $\kappa = 1.68e+02$. Additionally, the costs of the initial and restarted oblique projection in the INIT-CG algorithm also contribute to a constant level of solution costs $\mathcal{C}_{InitCG}$ when q is larger than approximately 20. This does not mean that we could not use the proposed two-phase approach with dimension larger than $q = 20$. As we will show, the large number of times that the system, with the same coefficient matrix, is solved enables the computation of larger dimension basis W. In the right of figure 2, we plot the costs of pre-computing the spectral information in terms of number of right-hand side computed through the formula

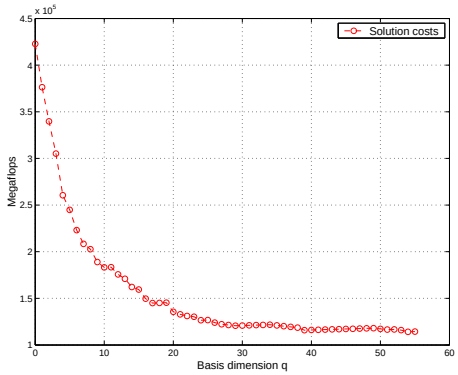
$$\texttt{Amor. rhs} = \Big\lfloor \frac{\mathcal{C}_{BCGSI}}{\mathcal{C}_{InitCG} - \mathcal{C}_{CG}} \Big\rfloor + 1,$$

where $\mathcal{C}_{CG}$ is the cost of classical CG algorithm, and $\mathcal{C}_{BCGSI}$ is the cost of pre-computing the spectral information with the BlockCGSI algorithm using a block size $s = 10$ (see figure 1). In the case of 100 systems to be solved ($\texttt{NGits} = 100$) the pre-computational costs are amortized until the basis reaches a critical value of $q = 50$. Under these conditions, if a larger number of times-steps is needed (for instance $\texttt{NGits} = 1000$) larger will be the critical value of q.

3.1.3 Minimizing the total cost function. Much more important than the critical values of q is the optimal value of q that minimizes the total cost function given by (1). As we have seen, the parameter `Total cost` is the addition of two other cost functions that are inversely proportional, namely $\mathcal{C}_{BCGSI}$ and $\mathcal{C}_{InitCG} \times \texttt{NGits}$. We have seen that the cost of pre-computing the spectral information $\mathcal{C}_{BCGSI}$ increases with q, while the solution costs $\mathcal{C}_{InitCG} \times \texttt{NGits}$ decrease as q increases. In figure 3, we plot all these two costs as well as the sum of two (the `Total cost`). The cost $\mathcal{C}_{BCGSI}$ was computed with block size $s = 10$, and the plot on the left corresponds to a simulation time of $T = 1$ where $\texttt{NGits} = T/\Delta t = 100$, and the plot on the right to $T = 10$ with $\texttt{NGits} = T/\Delta t = 1000$. The minimal value of `Total cost` occurs before $\mathcal{C}_{BCGSI} \approx \mathcal{C}_{InitCG} \times \texttt{NGits}$ because the solution cost $\mathcal{C}_{InitCG} \times \texttt{NGits}$ decrease very slightly when q is greater than 20 and the cost $\mathcal{C}_{BCGSI}$ grows in a larger scale. The optimal value of q (q_{opt}) is near 20 when $\texttt{NGits} = 100$, and near 30 when $\texttt{NGits} = 1000$.

The optimal value of q confirms that we must stop the BlockCGSI algorithm when the Ritz values are very close to each other. There is no benefit in approximating all the 48 eigenvalues corresponding to the targeted condition number $\kappa = 100$. The cost of solving all the systems given by $\mathcal{C}_{BCGSI}$ increases if we continue the subspace iteration when q is larger than 20. Even if the total number

(a) Solution costs



(b) Amortization Right-hand sides

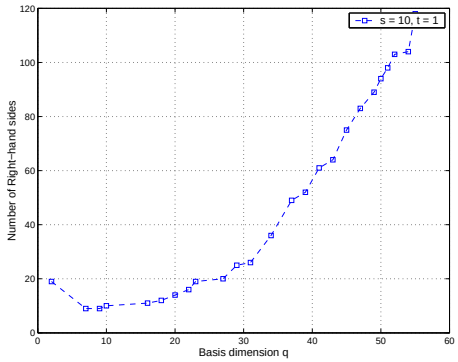


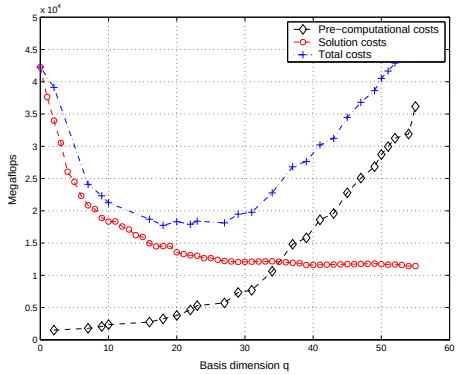
Fig. 2. Solution costs and amortization right-hand sides with INIT-CG (b).

of systems to be solved `NGits` is large, as for instance 1000, there is no effective reduction of the total costs when we compute a basis of higher dimension. As shown in figure 3b, the `Total cost` corresponding to $q = 20$ is nearly $1.5e+05$ `Mflops` and the value corresponding to $q = 30$ is nearly $1.3e+05$ `Mflops`. We confirm the idea that if the spectrum is very clustered (as in the present case) the two-phase accelerating strategy is more effective if we compute only the *near*-invariant subspace associated with the extremal eigenvalues.

3.2 Costs-benefits of the two-phase approach

Table 1 shows the cost-benefits of accelerating strategy. We consider that the INIT-CG algorithm has converged when the backward error is below 10^{-8}. In this case the classical Conjugate Gradient (INIT-CG with $q = 0$) performs 423 `Mflops`. As before, we indicate the number of floating-point operations in

(a) Total costs, `NGits` = 100



(b) Total costs, `NGits` = 1000

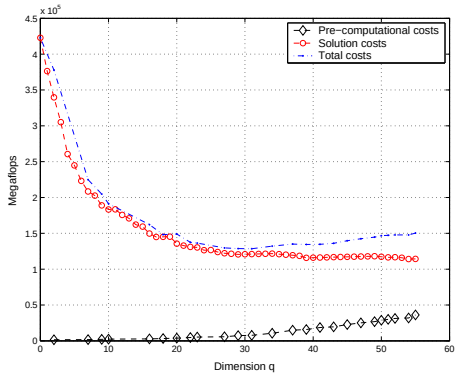


Fig. 3. Total costs of the two-phase approach.

Megaflops (`Mflops`) and the number of amortization right-hand sides by `Amor.rhs`. The pre-computational costs $\mathcal{C}_{BCGSI}$ are obtained with a block size of $s = 15$.

Firstly we can observe that when `NGits` = 100 the minimum value of `Total cost`, obtained with $q = 25$, is 15925 `Mflops`. The optimal value of q is greater than in the previous section ($q = 20$) because we run the BlockCGSI with $s = 15$ instead of $s = 10$. The same occurs when `NGits` = 1000 where the optimal value of q is equal to 40 instead of 30. This indicates that, if there were no computational restrictions, the BlockCGSI algorithm would run with a block size s as large as possible.

If we apply our accelerating technique to the current problem, and compute a basis W with optimal dimension ($q = 25$), 3225 `Mflops` are needed for the spectral pre-computation, out of which the Init-CG achieves convergence in

Table 1. Cost-benefits of the two-phase accelerating technique

.

Spectral fact.		INIT-CG	`Amor. rhs`	`Total cost`	
q	$\mathcal{C}_{BCGSI}$	`Mflops`		`NGIts` = 100	`NGIts` = 1000
0	–	423	–	42300	423000
5	2041	261	13	28141	263041
10	2377	189	11	21277	191377
15	2767	162	11	18967	164767
20	2767	145	10	17267	147767
25	3225	127	11	15925	130225
30	4242	121	15	16342	125242
35	4242	122	15	16442	126242
40	7473	116	25	19073	123473
45	11015	117	36	22715	128015
50	14972	118	50	26772	132972
55	24950	114	81	36350	138950

127 `Mflops`, i.e. a reduction of 70% compared with the work needed to solve one system with the classical Conjugate Gradient (423 `Mflops`). Therefore, the 3225 extra `Mflops` are paid back after 11 consecutive global iterations of the simulation code. And, in the case of `NGits` = 100, the value of `Total cost` is reduced from 42300 to 15925 `Mflops`, which corresponds to a reduction of 62% of the total amount of work required to solve 100 consecutive linear systems. In the case of `NGits` = 1000, if a basis of size $q = 30$ is used, the value of `Total cost` is reduced from 423000 to 125242 `Mflops`, which is a reduction of order 70% over all the computational work needed to solve the 1000 systems.

4 Conclusions

A two-phase approach was suggested to improve the numerical simulation of an airflow problem. In the first phase we have computed with the BlockCGSI algorithm a *near*-invariant subspace linked to the smallest eigenvalues. In the second phase, the basis of this subspace is used in each run of the Conjugate Gradient to deflate the starting residual (INIT-CG) and improve the consecutive solutions of the linear systems with the same coefficient matrix and changing right-hand sides.

The key question of this strategy is the dimension q of the *near*-invariant subspace to be computed. The optimal dimension depends on a compromise between the pre-computational (first phase) costs and the solution (second phase) costs. The cost of pre-computing the spectral information, which increases with the dimension of the basis, depends also on the block size s used on the BlockCGSI algorithm. The results show that larger block sizes reduce the pre-computational costs. On the other hand, the solution costs decrease with the increasing dimension of the *near*-invariant subspace q, until a q value (q_{sol}) is reached, above

which the solution costs stagnates. The optimal dimension (q_{opt}) is thus, the one that minimizes the sum of the two costs (pre-computing and solution) over all the systems to be solved. The results showed that q_{opt} and q_{sol} are close to each other.

The stagnation of the solution costs occur because the convergence rate of CG is not sufficiently improved, since the effect of all the extremal eigenvalues, separated from the main cluster, was removed. As the remaining eigenvalues are very close to each other, their deflation yields only a low reduction on the condition number that governs the convergence rate of the CG.

As a consequence of the previous results, we suggest a dynamical strategy to set up the dimension q of the *near*-invariant subspace associated with the smallest eigenvalues without *a priori* knowledge of the spectrum. At the beginning, after setting a reduced condition number $\kappa = \lambda_{\max}/\mu$ in agreement with the convergence rate of the CG, we request to approximate all the eigenvectors corresponding to the eigenvalues smaller that μ. At each iteration of the BlockCGSI algorithm, if the request is not satisfied, we compute the gaps between the approximated eigenvalues (Ritz values). As soon as the larger gap is below a given preset tolerance (which means that we are in the middle of a cluster), we stop the BlockCGSI algorithm and switch to the CG improved with precomputed basis (Init-CG), to compute the remaining system solutions of the simulation problem.

References

1. Arioli, M., Ruiz, D.: Block conjugate gradient with subspace iteration for solving linear systems. In: Iterative Methods in Linear Algebra, Second IMACS Symposium on Iterative Metohds in Linear Algebra, Blagoevgrad, Bulgaria, S. Margenov and P. Vassilevski (eds.) (June, 1995) pp. 64–79
2. Balsa, C., Palma, J., Ruiz, D.: Partial spectral information from linear systems to speed-up numerical simulations in computational fluid dynamics. In Daydé, M., Dongarra, J., Hernandez, V., Palma, J., eds.: High Performance Computing for Computational Science, 6th Int. Meeting, VECPAR'04. LNCS 3402, Berlin, Springer-Verlag (2005) pp. 699–715
3. Balsa, C., Daydé, M., Palma, J., Ruiz, D.: Inexact subspace iteration to exploit partial spectral information. Technical Report TR/TLSE/05/09, Institut National Polytechnique de Toulouse, LIMA-IRIT (2005)
4. Bergmann, M.: Analyse physique de la Transition Laminaire-Turbulent 2D dans des écoulements Cisaillés a l'Aide d'un Code de Navier-Stokes en éléments Finis. Rapport de stage de DEA, Toulouse (2001)
5. O'Leary, D.P.: The block conjugate gradient algorithm and related methods. Linear Algebra and its Applications (1980) 293–322
6. Parlett, B.N.: The Symmetric Eigenvalue Problem. SIAM, Philadelphia (1998)
7. Braza, M.: Analyse Physique du Comportement Dynamique d'un Écoulement Externe, Décollé, Instationnaire en Transition Laminaire-Turbulent. Application: Cylindre Circulaire. Thse d'état, INPT, Toulouse (1986)
8. Martinat, G.: Analyse physique de la Transition Laminaire-Turbulent sous l'Effect de la Rotation par un Code en Élement Finis. Rapport de stage de DEA, Toulouse (2003)