

RESEARCH ARTICLE

Open Access



An efficient quasi-optimal space-time PGD application to frictional contact mechanics

Anthony Giacoma^{1*}, David Dureisseix² and Anthony Gravouil^{2,3}

*Correspondence:

anthony.giacoma@ansys.com

¹ANSYS France, 11 avenue Albert
Einstein, 69100 Villeurbanne,
France

Full list of author information is
available at the end of the article

Abstract

The proper generalized decomposition (PGD) aims at finding the solution of a generic problems into a low rank approximation. On the contrary to the singular value decomposition (SVD), such a low rank approximation is generally not the optimal one leading to memory issues and loss of computational efficiency. Nonetheless, the computational cost of the SVD is generally prohibitive to be performed. In this paper, authors suggest an algorithm to address this issue. First, the algorithm is described and studied in details. It consists in a cheap iterative method compressing a low rank expansion. It will be shown that given a low rank approximation, the SVD of a provided low rank approximation can be reached at convergence. Behavior of the method is exhibited on a numerical application. Second, the algorithm is embedded into a general space-time PGD solver to compress the iterated separated form for the solution. An application to a quasi-static frictional contact problem is illustrated. Then, efficiency of such a compressing method will be demonstrated.

Keywords: Low rank approximation, Proper generalized decomposition, Singular value decomposition, Principal component analysis, Quasi-static contact

Background

Computational mechanics tackles nowadays large models involving huge amount of data to provide fine description of physics or accurate forecasts. For that purpose, several and various numerical methods have to be taken into account in order to perform efficiently these large scale simulations (both accurate and computationally cheap). To address this issue, both computational hardware and algorithms have to progress. During the last decades, a specific class of algorithms based on model reduction methods has been developed. They consist basically in focusing on dominant trends of the problem. Then, a large amount of computational time can be spared and accurate and well representative solution can be captured. These methods rely strongly on basis design for approximated solution which has to span the dominant trends and perhaps weaker ones up to a desired level of accuracy.

Given a collection of data (also called snapshots), the well-known canonical method to design the optimal basis is the Singular Value Decomposition (SVD). Such a decomposition may lead to prohibitive computational times in an industrial context due to its complexity. In addition reduced order modeling methods often require strategies to adapt online the reduced basis in order to include uncaptured trends of the problem. In other

words, if one has a SVD basis and wants to add some vectors, one has to recompute the SVD with new data. For such situations updating strategies are proposed in [1,2].

Usual SVD algorithms [3] compute SVD modes one-by-one “incrementally” until having a basis satisfying a certain level of accuracy. These algorithms iterate until finding a mode. Once the precision criterion is reached, the basis is ensured to be optimal because each found mode is the most representative one.

In this paper, we propose a different approach. Given a set of vectors, a basis is defined. The hereinafter suggested approach iterates over the whole basis in order to make all of its vectors closer to optimal ones until having SVD basis. Doing so, after each iteration a “quasi-optimal” basis is computed and few iterations are expected to provide a quite optimal basis. Such an approach ensures to have at each iteration a basis which spans the whole considered space to detriment of its optimality. Such an iterated basis could be sufficient to perform reliable computation or data analysis. One expects that the computational effort to get a quasi-optimal basis is low whereas classic SVD algorithms prescribing the optimality property are expensive.

In the following sections, the proposed strategy is first described on a rank-2 expansion. Convergence proof, analysis and results are exposed. Second, this strategy is generalized for rank- p expansion with a global convergence proof. Afterwards, this strategy is tested by computing the SVD of a matrix. Finally, an application case is performed. It deals with a combination of the suggested method and the proper generalized decomposition (PGD) method. On this basis, the efficiency of quasi-optimal approaches will be exemplified.

An iterative process to compute the SVD

In the following, we will denote with $\mathbf{A} \in \mathbb{R}^{n \times m}$ a real rectangular matrix. Without loss of generality, we will assume that $n \geq m$ (if not the case, we simply consider the transpose of \mathbf{A}). Given two column vectors of same size \mathbf{u} and \mathbf{v} , the associated inner product is denoted with $(\mathbf{u} | \mathbf{v})$; since in this article we consider the euclidean canonical inner product associated to the euclidean norm $\|\cdot\|$, $(\mathbf{u} | \mathbf{v}) = \mathbf{u}^T \mathbf{v}$.

Given a collection of m vectors $\mathbf{s}_i \in \mathbb{R}^n$ (e.g. experimental results called snapshots), they can be cast into a real rectangular matrix \mathbf{A} (the snapshot matrix) for which a low rank expansion (with $p \leq m$) is

$$\mathbf{A} = [\mathbf{s}_1 \cdots \mathbf{s}_m] = \sum_{i=1}^p \mathbf{u}_i \mathbf{v}_i^T = [\mathbf{u}_1 \cdots \mathbf{u}_p] [\mathbf{v}_1 \cdots \mathbf{v}_p]^T = \mathbf{U} \mathbf{V}^T \quad (1)$$

Matrices \mathbf{U} and \mathbf{V} are composed by respectively column vectors $\mathbf{u}_i \in \mathbb{R}^n$ (left vectors) and $\mathbf{v}_i \in \mathbb{R}^m$ (right vectors). Left vectors \mathbf{u}_i form an a priori non-unique basis for snapshots \mathbf{s}_k which coordinates are contained in \mathbf{v}_i . p corresponds to the size of the expansion (i.e. basis size). In this article, a method aiming at finding a suited basis for snapshots \mathbf{s}_k (i.e. the smallest basis) is proposed. The optimal basis is known to be given by left vectors of the SVD.

In order to obtain a decomposition (1), several methods could be used. They are expected to be able to prescribe specific properties such as orthonormality condition for the involved vectors. Three of them are listed below.

Decomposition according to the canonical basis. Given the matrix \mathbf{A} , each snapshot can be written in the canonical basis leading to $\mathbf{U} = \mathbf{I}_n$ (square $n \times n$ identity matrix, $p = n$) and $\mathbf{V} = \mathbf{A}^T$. Hence, vectors \mathbf{u}_i are orthonormal and columns of \mathbf{A} correspond to vectors \mathbf{v}_i .

Cholesky orthonormalization. Given the matrix \mathbf{A} , Cholesky factorization method can be used to find an orthonormal basis for snapshots in \mathbf{A} . Let us define the positive symmetric matrix $\mathbf{M} = \mathbf{A}^T \mathbf{A}$; it is assumed definite to fulfill the Cholesky factorization requirements (i.e. \mathbf{A} is assumed to be full rank).

$$\mathbf{M} = \mathbf{A}^T \mathbf{A} = \mathbf{L} \mathbf{L}^T \Rightarrow \mathbf{L}^{-1} \mathbf{A}^T \mathbf{A} \mathbf{L}^{-T} = \mathbf{I}_m \quad (2)$$

Matrix \mathbf{L} is a lower triangular matrix. Hence, $\mathbf{U} = \mathbf{A} \mathbf{L}^{-T}$ is an orthonormal basis for snapshots: $\mathbf{U}^T \mathbf{U} = \mathbf{I}_m$ and $\mathbf{V} = \mathbf{L}$ provide a low rank expansion. The overall complexity of this method is dominated by the Cholesky factorization for full matrices (about $\frac{1}{3}n^3$ operations).

Gram-Schmidt-based orthonormalization. If the matrix \mathbf{A} is already provided with a low rank expansion of size $p \leq m$, the Cholesky factorization can be replaced by a Gram-Schmidt procedure to provide an orthogonality condition for left vectors, i.e. $\forall(i, j), (\mathbf{u}_i | \mathbf{u}_j) = \delta_{ij}$ with δ_{ij} the Kronecker symbol. For that purpose, the Algorithm 1 can be used. The overall complexity of the Gram-Schmidt process is about $2np^2$ operations [3]. One can note that Algorithm 1 does not change the considered approximation, i.e. $\mathbf{A}_p = \tilde{\mathbf{A}}_p$. There is also some flexibility concerning the choice of the inner product.

Algorithm 1: Left-orthonormalization.

Input: $\mathbf{A}_p = \sum_{i=1}^p \mathbf{u}_i \mathbf{v}_i^T$
Output: $\tilde{\mathbf{A}}_p = \sum_{i=1}^p \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^T$ with $(\tilde{\mathbf{u}}_i | \tilde{\mathbf{u}}_j) = \delta_{ij}$

```

1 First left vector  $\tilde{\mathbf{u}}_1 \leftarrow \mathbf{u}_1 / \|\mathbf{u}_1\|$ 
2 First right vector  $\tilde{\mathbf{v}}_1 \leftarrow \|\mathbf{u}_1\| \times \mathbf{v}_1$ 
3 for  $k = 2$  to  $p$  do
4   for  $i = 1$  to  $k - 1$  do
5     Projection  $\alpha \leftarrow (\tilde{\mathbf{u}}_i | \mathbf{u}_k)$ 
6     Update right modes  $\tilde{\mathbf{v}}_i \leftarrow \tilde{\mathbf{v}}_i + \alpha \mathbf{v}_k$ 
7     Subtract projected component  $\mathbf{u}_k \leftarrow \mathbf{u}_k - \alpha \tilde{\mathbf{u}}_i$ 
8   end
9   New left vector  $\tilde{\mathbf{u}}_k \leftarrow \mathbf{u}_k / \|\mathbf{u}_k\|$ 
10  New right vector  $\tilde{\mathbf{v}}_k \leftarrow \|\mathbf{u}_k\| \times \mathbf{v}_k$ 
11 end

```

Other standard methods aim also at providing a first guess of the low rank expansion (QR factorization) and may have suitable advantages like numerical complexity or numerical stability. Nevertheless, one has to keep in mind that such pre-orthogonalization processes have a numerical cost.

Iterative singular value decomposition for a rank-2 matrix

Definition of the compression function F

In this section, we consider a rank-2 approximation denoted by \mathbf{A}_2 defined as follows:

$$\mathbf{A}_2 = \mathbf{u}_1 \mathbf{v}_1^T + \mathbf{u}_2 \mathbf{v}_2^T \quad \text{with} \quad \begin{cases} (\mathbf{u}_i | \mathbf{u}_j) = \delta_{ij} \\ \|\mathbf{v}_1\|^2 \geq \|\mathbf{v}_2\|^2 > 0 \end{cases} \quad (3)$$

Let F be the function providing a new iterate:

$$(\tilde{\mathbf{u}}_1, \tilde{\mathbf{v}}_1, \tilde{\mathbf{u}}_2, \tilde{\mathbf{v}}_2) = F(\mathbf{u}_1, \mathbf{v}_1, \mathbf{u}_2, \mathbf{v}_2) \quad (4)$$

and defined as:

1. Right vector \mathbf{v}_2 is written as $\mathbf{v}_2 = \alpha \mathbf{v}_1 + \tilde{\mathbf{v}}_2$ so that $\mathbf{v}_1^T \tilde{\mathbf{v}}_2 = 0$ and $\alpha = (\mathbf{v}_1 | \mathbf{v}_2) / (\mathbf{v}_1 | \mathbf{v}_1)$. $\mathbf{A}_2 = \tilde{\mathbf{u}}_1 \mathbf{v}_1^T + \mathbf{u}_2 \tilde{\mathbf{v}}_2^T$ with $\tilde{\mathbf{u}}_1 = \mathbf{u}_1 + \alpha \mathbf{u}_2$. One can remark that $\tilde{\mathbf{u}}_1^T \mathbf{u}_2 \neq 0$ a priori.
2. Left vectors are reorthogonalized using $\mathbf{u}_2 = \beta \tilde{\mathbf{u}}_1 + \tilde{\mathbf{u}}_2$ with $\tilde{\mathbf{u}}_1^T \tilde{\mathbf{u}}_2 = 0$ so that $\beta = (\tilde{\mathbf{u}}_1^T \mathbf{u}_2) / (\tilde{\mathbf{u}}_1^T \tilde{\mathbf{u}}_1) = \alpha / (1 + \alpha^2)$. $\mathbf{A}_2 = \tilde{\mathbf{u}}_1 \tilde{\mathbf{v}}_1^T + \tilde{\mathbf{u}}_2 \tilde{\mathbf{v}}_2^T$ with $\tilde{\mathbf{v}}_1 = \mathbf{v}_1 + \beta \tilde{\mathbf{v}}_2$.
3. Denoting $\gamma = \sqrt{1 + \alpha^2}$, since $\|\tilde{\mathbf{u}}_1\| = \gamma$ and $\|\tilde{\mathbf{u}}_2\| = 1/\gamma$, the left vectors are normalized with $\tilde{\mathbf{u}}_1 = \tilde{\mathbf{u}}_1/\gamma$, $\tilde{\mathbf{u}}_2 = \gamma \tilde{\mathbf{u}}_2$ and $\tilde{\mathbf{v}}_1 = \gamma \tilde{\mathbf{v}}_1$, $\tilde{\mathbf{v}}_2 = \tilde{\mathbf{v}}_2/\gamma$.

The new rank-2 approximation is $\tilde{\mathbf{A}}_2 = \tilde{\mathbf{u}}_1 \tilde{\mathbf{v}}_1^T + \tilde{\mathbf{u}}_2 \tilde{\mathbf{v}}_2^T = \mathbf{A}_2$. The following expressions are obtained:

$$\alpha = \frac{(\mathbf{v}_1 | \mathbf{v}_2)}{(\mathbf{v}_1 | \mathbf{v}_1)} \quad (5)$$

$$\tilde{\mathbf{u}}_1 = \mathbf{u}_1 + \alpha \mathbf{u}_2 \quad (6)$$

$$\tilde{\mathbf{v}}_1 = \mathbf{v}_1 + \frac{\alpha(\mathbf{v}_2 - \alpha \mathbf{v}_1)}{1 + \alpha^2} = \frac{\mathbf{v}_1}{1 + \alpha^2} + \frac{\alpha \mathbf{v}_2}{1 + \alpha^2} \quad (7)$$

$$\tilde{\mathbf{u}}_2 = \mathbf{u}_2 - \frac{\alpha(\mathbf{u}_1 + \alpha \mathbf{u}_2)}{1 + \alpha^2} = -\frac{\alpha \mathbf{u}_1}{1 + \alpha^2} + \frac{\mathbf{u}_2}{1 + \alpha^2} \quad (8)$$

$$\tilde{\mathbf{v}}_2 = \mathbf{v}_2 - \alpha \mathbf{v}_1 \quad (9)$$

$$\|\tilde{\mathbf{u}}_1\| = \sqrt{1 + \alpha^2} \quad (10)$$

$$\|\tilde{\mathbf{u}}_2\| = \frac{1}{\sqrt{1 + \alpha^2}} \quad (11)$$

Note that if $\alpha = 0$, the function is the identity and the algorithm is terminated. Otherwise, $\alpha \neq 0$ and

$$(\tilde{\mathbf{u}}_i | \tilde{\mathbf{u}}_j) = \delta_{ij} \quad \text{orthonormality} \quad (\clubsuit)$$

$$(\mathbf{v}_1 | \mathbf{v}_2) > (\tilde{\mathbf{v}}_1 | \tilde{\mathbf{v}}_2) \quad \text{compression} \quad (\spadesuit)$$

$$\|\tilde{\mathbf{v}}_1\|^2 > \|\tilde{\mathbf{v}}_2\|^2 \quad \text{order preservation} \quad (\diamond)$$

Proofs are given in the next Sections. The considered algorithm is the recursive application of function F .

Algorithm study

Using the Cauchy–Schwarz inequality and properties of right vectors in (3) yield to:

$$0 \leq (\mathbf{v}_1 | \mathbf{v}_2) \leq \|\mathbf{v}_1\| \times \|\mathbf{v}_2\| \leq \|\mathbf{v}_1\|^2 \quad (12)$$

squaring these inequalities, with $(\mathbf{v}_1 | \mathbf{v}_2) = \alpha \|\mathbf{v}_1\|^2$ and defining $\eta = \|\mathbf{v}_2\|^2 / \|\mathbf{v}_1\|^2$, one obtains:

$$0 \leq \alpha^2 \leq \eta \leq 1 \quad (13)$$

which is verified at each iteration ξ of the previous algorithm. Moreover, a recursion formula can be obtained, once given η_0 and α_0 :

$$\begin{cases} \eta_{\xi+1} = \frac{\eta_{\xi} - \alpha_{\xi}^2}{1 + \alpha_{\xi}^2(2 + \eta_{\xi})} \\ \alpha_{\xi+1} = \eta_{\xi+1} \alpha_{\xi} \end{cases} \quad (14)$$

α_ξ is an error measure at iteration ξ , and η_ξ is linked to the convergence rate. If at an iteration ξ , $\alpha_\xi \neq 0$, one gets with the previous inequalities

$$1 + \alpha_\xi^2(2 + \eta_\xi) > 1 + \alpha_\xi^2 > 1 - \alpha_\xi^2 \geq \eta_\xi - \alpha_\xi^2 > 0$$

so: $0 < \eta_{\xi+1} < 1$, $\eta_{\xi+1} \neq \alpha_{\xi+1}$, $\alpha_{\xi+1} \neq 0$, $\alpha_{\xi+1} < \alpha_\xi$. As a consequence, if $\alpha_0 = 0$ we get the solution without iterating, otherwise

$$\forall \xi \geq 1, \alpha_\xi \neq 0 \text{ and } 0 < \alpha_\xi^2 < \eta_\xi < 1$$

As a decreasing and lower-bounded α_ξ serie, it converges to a value α ; the fixed point of (14) is $\alpha = \eta\alpha$, so $\alpha = 0$ (the error decreases towards 0).

Since one also has $1 + \alpha_\xi^2(2 + \eta_\xi) > 1 + \alpha_\xi^2 > 1 - \alpha_\xi^2/\eta_\xi$, the following property holds: $0 < \eta_{\xi+1} < \eta_\xi$. As a decreasing and lower-bounded η_ξ serie, it converges to a value η (the convergence rate increases).

After algebraic manipulations, the recurrence formula (14) allows to prove that there is a preserved quantity along iterations:

$$\eta_\xi + 1/\eta_{\xi+1} = \eta_0 + 1/\eta_1 = \frac{1 + 2\alpha_0^2 + \eta_0^2}{\eta_0 - \alpha_0^2} := 2\delta$$

The fixed point of this quantity therefore allows to get the asymptotic value $\eta = \delta - \sqrt{\delta^2 - 1}$.

Algorithm properties

The orthogonality property (\clubsuit) results directly from the orthogonalization in the algorithm. Since $\eta_\xi < 1$, the order preservation (\diamond) is proved. Finally, one can obtain

$$\frac{(\tilde{\mathbf{v}}_1 | \tilde{\mathbf{v}}_2)}{(\mathbf{v}_1 | \mathbf{v}_2)} = \frac{(\tilde{\mathbf{v}}_1 | \tilde{\mathbf{v}}_2)}{(\mathbf{v}_1 | \mathbf{v}_2)} = \frac{\eta - \alpha^2}{1 + \alpha^2}$$

with previous inequalities, this ratio is strictly less than 1, proving the compression property (\spadesuit).

Numerical example

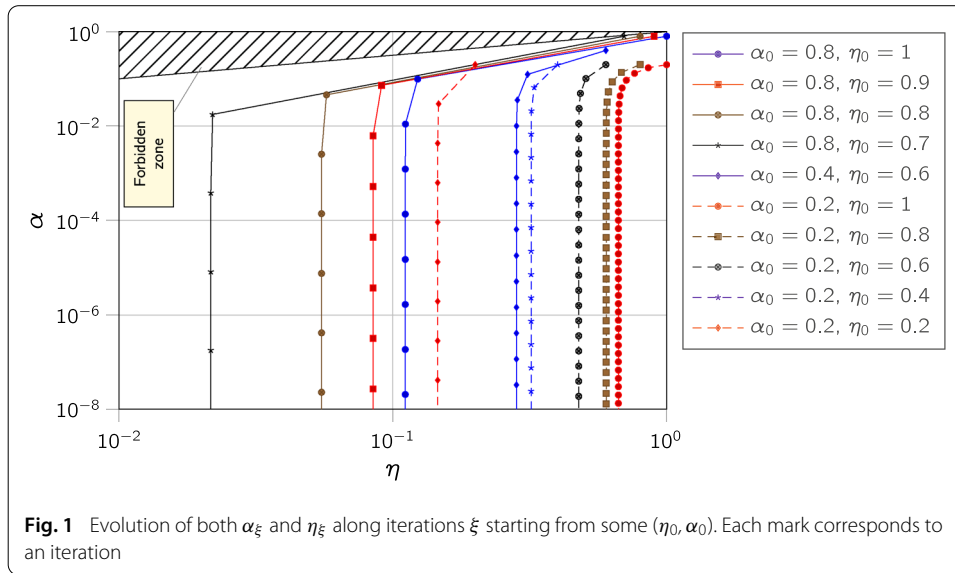
The first terms α_0 and η_0 depend on the initial given low rank approximation to compress. To illustrate briefly the behavior of the suggested algorithm, a plot (Fig. 1) is proposed. It depicts trajectories (η_ξ, α_ξ) starting from some different cases with different values of (η_0, α_0) . This confirms the previous analysis of the algorithm about convergence and convergence rate:

- The higher α_0 , the lower becomes η and the higher the convergence rate is.
- The lower η_0 , the faster the convergence is.
- The lower α_0 (right vectors are poorly correlated) and the higher η_0 (amplitudes of right vectors are similar), the lower the convergence rate is.

The last observation is perturbing because in this extend one has quasi-orthogonal right vectors and in other words, the job is quite completed. But the discrepancy between right vectors amplitudes is not enough to distinguish the most contributory rank one approximation.

Generalization to higher rank expansions

In this section, the compression stage is generalized to higher rank expansions (1), with $p > 2$ and $(\mathbf{u}_i | \mathbf{u}_j) = \delta_{ij}$ and $\forall i \leq j$, $0 < \|\mathbf{v}_i\| \leq \|\mathbf{v}_j\|$. With $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_p]$, we introduce the following symmetric matrix \mathbf{W} :



$$\mathbf{W} = \mathbf{V}^T \mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \mathbf{v}_1 & \mathbf{v}_1^T \mathbf{v}_2 & \cdots & \mathbf{v}_1^T \mathbf{v}_p \\ \mathbf{v}_2^T \mathbf{v}_1 & \mathbf{v}_2^T \mathbf{v}_2 & \cdots & \mathbf{v}_2^T \mathbf{v}_p \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_p^T \mathbf{v}_1 & \mathbf{v}_p^T \mathbf{v}_2 & \cdots & \mathbf{v}_p^T \mathbf{v}_p \end{bmatrix} \quad (15)$$

If \mathbf{W} is diagonal, then right vectors \mathbf{v}_i are orthogonal. In this case and since left vectors \mathbf{u}_i are orthonormal, \mathbf{A} is written under its (unique) SVD. The compression function defined by (4) can be applied for two pairs of vectors (to be chosen). Both pairs have to fulfill conditions (3). Resulting pairs and their properties have been studied in previous sections. As the expansion is of rank higher than 2, investigation about properties of the two resulting pairs of vectors regarding the others have to be carried out.

Orthogonality (\clubsuit) and compression (\spadesuit) properties have to be checked for all vectors. Ordering property (\diamond) can be always ensured by sorting dyads at the end of the application of F .

First of all, as resulting $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{u}}_j$ vectors are obtained by a linear combination of \mathbf{u}_i and \mathbf{u}_j , it turns out clearly that orthogonality property (\clubsuit) regarding other vectors pertains. Then, a non-declining compression property (\spadesuit) has to be checked. We denote by \mathbf{A} the initial expansion as defined in (1) and $\tilde{\mathbf{A}}$ the resulting expansion after the application of F for two pairs of vectors with:

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}} \tilde{\mathbf{V}}^T = [\mathbf{u}_1 \cdots \tilde{\mathbf{u}}_i \tilde{\mathbf{u}}_j \cdots \mathbf{u}_p] [\mathbf{v}_1 \cdots \tilde{\mathbf{v}}_i \tilde{\mathbf{v}}_j \cdots \mathbf{v}_p]^T \quad (16)$$

Thanks to (4) and orthonormality for vectors \mathbf{U} , the matrix \mathbf{W} can be expressed as follows:

$$\mathbf{A} = \tilde{\mathbf{A}} \Leftrightarrow \mathbf{A} \mathbf{A}^T = \tilde{\mathbf{A}} \tilde{\mathbf{A}}^T \Leftrightarrow \mathbf{U} \mathbf{V}^T \mathbf{V} \mathbf{U}^T = \tilde{\mathbf{U}} \tilde{\mathbf{V}}^T \tilde{\mathbf{V}} \tilde{\mathbf{U}}^T \Leftrightarrow \mathbf{W} = \mathbf{R} \tilde{\mathbf{W}} \mathbf{R}^T \quad (17)$$

With $\mathbf{R} = \mathbf{U}^T \tilde{\mathbf{U}}$ and $\tilde{\mathbf{W}} = \tilde{\mathbf{V}}^T \tilde{\mathbf{V}}$. Matrix \mathbf{R} is of the following form:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{P} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

With \mathbf{P} the transformation between the two chosen pairs. It can be exhibited from equations (5)–(11):

$$\mathbf{P} = \begin{bmatrix} \frac{1}{\sqrt{1+\alpha^2}} & \frac{\alpha}{\sqrt{1+\alpha^2}} & 0 \\ \frac{-\alpha}{\sqrt{1+\alpha^2}} & \frac{1}{\sqrt{1+\alpha^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

As a consequence, it turns out that matrices \mathbf{R} and \mathbf{P} are rotation matrices. In other words, the two chosen vectors \mathbf{v}_i and \mathbf{v}_j are rotated around a subspace of dimension $p - 2$. They are rotated in such a way that their associated extra-diagonal terms in \mathbf{W} diminish whereas other terms remain the same in norm. Then, compression property (♠) holds. Finally, the compression function can be applied for high rank expansion making it converge to its SVD expansion.

Various combinations of $p - 2$ dimensional subspace rotations can be chosen. It is a compromise between efficiency and computational sustainability (parallel computation).

Algorithm

We propose to apply iteratively the compression function F according to the Algorithm 2. In the suggested algorithm, all rotations are swept at in the while-loop (precision loop).

Algorithm 2: Recursive SVD

```

1  $\xi \leftarrow 1$ 
2 while  $\xi \leq \xi_{\max}$  do
3   for  $i = p$  down to 2 do
4     Sort vectors  $\mathbf{v}$  such that  $\|\mathbf{v}_1\|_2 \geq \dots \geq \|\mathbf{v}_p\|_2$ 
5     for  $j = 1$  up to  $i - 1$  do
6        $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{u}_j, \mathbf{v}_j) \leftarrow F(\mathbf{u}_i, \mathbf{v}_i, \mathbf{u}_j, \mathbf{v}_j)$ 
7     end
8   end
9    $\xi \leftarrow \xi + 1$ 
10 end

```

Algorithm 2 consists in applying compression function F to two rank-2 approximations composing the whole approximation of \mathbf{A} . This is achieved in a such way that conditions (♣), (♠) and (◇) are fulfilled. Doing so, previously given proofs can be reused. This algorithm may run until producing the SVD of \mathbf{A} .

With the proposed algorithm, loops dependencies do not enable their execution in parallel. Several for-loop strategies can be implemented, but are not studied herein.

Rank adaptation and downsizing

Let n be the size of \mathbf{u} and m the size of \mathbf{v} . The complexity of one instance of the compression function F is $c_F = 6n + 10m + 6$. One loop (indexed by ξ) involves $\frac{1}{2}p(p - 1)$ occurrences of F . All in all, complexity of Algorithm 3 can be estimated to $c = \xi_{\max}[3np(p - 1) + 5mp(p - 1) + 3p(p - 1)]$. This complexity is evaluated assuming that expansion size $q = p$ remains constant. Nonetheless, during iterative process, one is able to eliminate pairs of vectors of poor contribution by prescribing a threshold ϵ for the norms of the right vectors.

Algorithm 3: Downsizing algorithm for a low rank approximation.

```

1  $\xi \leftarrow 1$ 
2  $q \leftarrow p$ 
3 while  $\xi \leq \xi_{\max}$  do
4   for  $i = q$  down to 2 do
5     Sort vectors  $\mathbf{v}$  such that  $\|\mathbf{v}_1\|_2 \geq \dots \geq \|\mathbf{v}_p\|_2$ 
6     Cut at threshold  $\epsilon$ :  $\|\mathbf{v}_1\|_2 \geq \dots \geq \|\mathbf{v}_q\|_2 \geq \epsilon$ 
7     to produce  $q$  and set  $i \leftarrow \min(i, q)$ 
8     for  $j = 1$  up to  $i - 1$  do
9        $\alpha_{ij} \leftarrow (\mathbf{v}_j | \mathbf{v}_i) / (\mathbf{v}_j | \mathbf{v}_j)$ 
10       $\rho \leftarrow \sqrt{1 + \alpha_{ij}^2}$ 
11       $\mathbf{u}_j \leftarrow (1/\rho)\mathbf{u}_j + (\alpha_{ij}/\rho)\mathbf{u}_i$ 
12       $\mathbf{u}_i \leftarrow \rho\mathbf{u}_i - \alpha_{ij}\mathbf{u}_j$ 
13       $\mathbf{v}_j \leftarrow (1/\rho)\mathbf{v}_j + (\alpha_{ij}/\rho)\mathbf{v}_i$ 
14       $\mathbf{v}_i \leftarrow \rho\mathbf{v}_i - \alpha_{ij}\mathbf{v}_j$ 
15    end
16  end
17   $\xi \leftarrow \xi + 1$ 
18 end

```

Thus, a computational expense could be spared and the analysis focused on dimensions of interest.

Numerical application

To illustrate the previously described algorithm, the singular value decomposition of a given matrix is performed. This matrix is picked up from Matrix Market¹ and is called `rbs480a.mtx`. First, its SVD is computed using the standard Matlab solver. Singular values (σ_i^{ref}), reference left ($\mathbf{u}_i^{\text{ref}}$) and right ($\mathbf{v}_i^{\text{ref}}$) singular modes are therefore provided.

Previously, convergence properties have been enlightened according to mode amplitude properties. To exemplify those convergence behaviors, several configurations are built to affect amplitude ratio between modes (σ_i^{ref} is transformed into σ_i^{mod}); left and right singular modes remain the same and only mode contribution is affected. For each configuration, a whole modified matrix is rebuilt.

The square matrix `rbs480a.mtx` is full rank and has 480 singular values and singular modes. Three configurations are studied:

- No modification. Original singular value amplitudes of the matrix decrease slowly along the 400 first modes.
- Medium slope for singular value amplitudes. In a semilog diagram, a linear slope for mode amplitudes is prescribed.
- Strong initial slope for singular value amplitudes. A small amplitude ratio is prescribed for first successive modes.

These modifications are shown on Fig. 2.

¹<http://math.nist.gov/MatrixMarket/>.

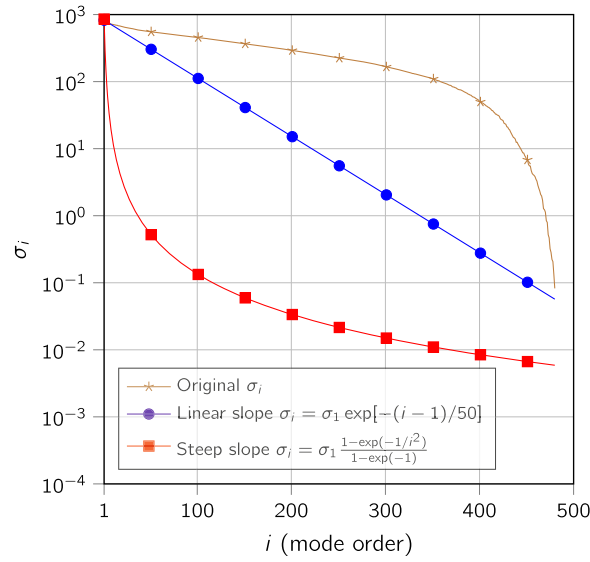


Fig. 2 Singular value amplitudes according to modifications applied on reference SVD of `rbs480a.mtx`.

For each of the three previously described matrices, the proposed algorithm is applied. Error \mathcal{E} and indicator \mathcal{I} are followed throughout iterations on Fig. 3. The error is $\mathcal{E} = \|\varepsilon\|_F$ with:

$$\varepsilon_i = \mathbf{u}_i^{\text{ref}} \sigma_i^{\text{mod}} [\mathbf{v}_i^{\text{ref}}]^T - \mathbf{u}_i \mathbf{v}_i^T \quad \text{for } 1 \leq i \leq p \quad (20)$$

The convergence indicator corresponds to the root mean square of all computed α_{ij} . On Fig. 3, one can note for the different configurations:

- No modification. As it may have been expected, the convergence is quite slow during the first iterations, because the first successive modes do have a high (close to 1) amplitude ratio. Nevertheless, the last 80 modes are more rapidly found.

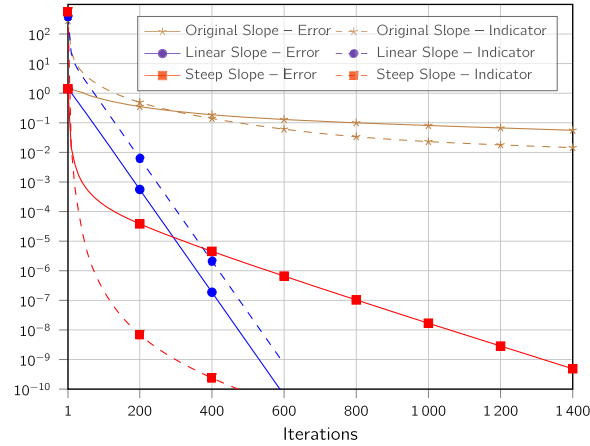


Fig. 3 Convergence of the proposed algorithm to compute the SVD of `rbs480a.mtx` test

- Medium slope. The amplitude ratios are all the same, and all iterations provide a similar convergence rate.
- Strong initial slope. The small amplitude ratio for the first successive modes lead to a large convergence rate during first iterations. The converse is obtained for the latest iterations.

As a summary, the first modes are converged during the first iterations, and the convergence rate is related to their amplitude ratio. Progressively, the following modes are detected by the algorithm and the convergence rate adapts to their new amplitude ratios. Note also that a constant amplitude ratio leads to a uniform convergence rate.

Application to SVD-free quasi-optimal space-time PGD

During the last decade, a novel generation of solvers based on model reduction techniques has been fostered for both linear and non-linear problem solving. These solvers aim at reducing drastically the computational time and the memory usage to store the solution. They consist in different strategies:

- a posteriori approaches (POD/SVD, surrogate modeling) using prior knowledge about the solution (sampling, snapshots, etc.) to compute desired new solutions.
- a priori approaches which do not require previous knowledge about the solution and aim at computing a desired solution in a convenient form (memory and cost efficient).

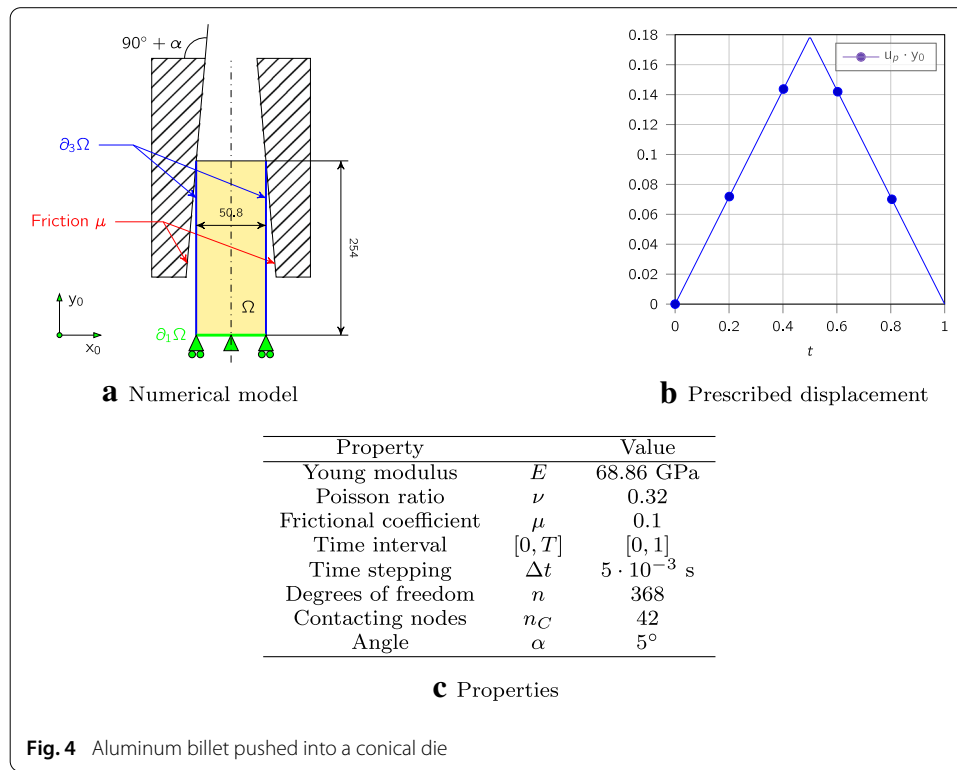
Given a problem formulated by PDEs and denoted by \mathcal{P} , one aims at finding its solution \mathcal{S} on the domain Ω . This domain is spatial or temporal or parametric (a space-time-parameter domain). Model reduction methods rely on several expectations on \mathcal{S} :

- Reducibility: \mathcal{S} can be represented on a low-dimensional basis, i.e. \mathcal{S} can be written accurately (up to a certain level) with a linear combination of a few vectors
- Dominant trends (scale separability): some vectors of the basis (which are supposed to be the first ones) are highly contributory to generate \mathcal{S} whereas other ones are less important. These vectors depict the different scales of the problem [4].

Assuming \mathcal{S} is known, its SVD decomposition can be computed. Then, the set of the p first vectors is the optimal basis of size p for \mathcal{S} thanks to the Eckart and Young Theorem [5]. In other words, these p first vectors are the most contributory ones in the solution, considering the Frobenius norm. One has to make a difficult compromise between having a small p and preserving a good accuracy for basis (i.e. relevancy of the generated subspace).

The obtained basis can be used within the a posteriori approach to generate Reduced Order Models (ROMs). Indeed, a first approach consists in projecting \mathcal{P} (Galerkin projection) into the spanned subspace [6–8]. Secondly, this basis can be considered as a filter for data due to the basis truncation. Indeed, noise is expected to be generated by the high order SVD modes. Therefore the basis can be used to generate surrogate models relying on regression methods (ARMA, ARIMA processes [9,10]), time series analysis [11], etc. These resulting models are expected to be easy to use and computationally efficient. The quality of the snapshots depends highly on the initial chosen vectors and the process to generate the model; error criterion could be difficult to exhibit.

A widespread a priori approach is the Proper Generalized Decomposition (PGD) [12–14]. This approach aims at finding \mathcal{S} directly into a separated form or low rank expansion



as in equation (1) without prior knowledge. PGD solvers are incremental processes which consists in enriching progressively a low rank expansion to make an iterated solution \mathcal{S}_i more accurate. The ideal PGD solver should be able to find each vectors of the SVD decomposition of \mathcal{S} , i.e. the first iterated vector is the most contributory one of \mathcal{S} , then the second, etc. Basically, PGD does not prescribe orthogonality for left or right vectors of the low rank expansion. In practice, such a condition is often applied for a sake of numerical efficiency. In practice, the low rank expansion generated by the PGD is generally not the optimal none. Nevertheless, as the computational effort is concentrated on rank-1 tensors, a great amount of computations and memory can be spared.

We propose to illustrate the previously described algorithm into a space-time PGD solver aiming at solving a frictional contact solid mechanic quasi-static problem. We suggest to embed into a PGD iteration, one iteration of Algorithm 3 in order to compress progressively the iterated low rank expansion. Doing so, one can expect to make it close to the optimal one and stem inflation of iterated expansion [15].

Quasi-static frictional contact problems

Reference problem

To exemplify the suggested algorithm, an extrusion of an elastic aluminum billet into a rigid conical die is simulated (Fig. 4). This problem [16,17] is investigated assuming small perturbations even if such hypothesis is not ensured. The finite element method is used and the solid is meshed with 2D quadrangular elements. A displacement is prescribed in such a way that the billet is pushed into, then extracted from, a conical die. Signorini's conditions and Coulomb's Law are considered for the frictional contact laws.

The large time increment method

To solve this problem, the non-linear LArge Time INcrement (LATIN) solver is used [18]. This method, close to augmented Lagrangian methods, is well-known for its ability to solve difficult non-linear and time-dependent large problems with a global time-space approach (non-linear material [19], contact problems [4, 15, 20], large displacement [21], transient dynamics [22, 23], fracture mechanics [24, 25]...). The non-incremental LATIN method was proposed as a commitment of three principles, which are, for the elastic frictional contact problems:

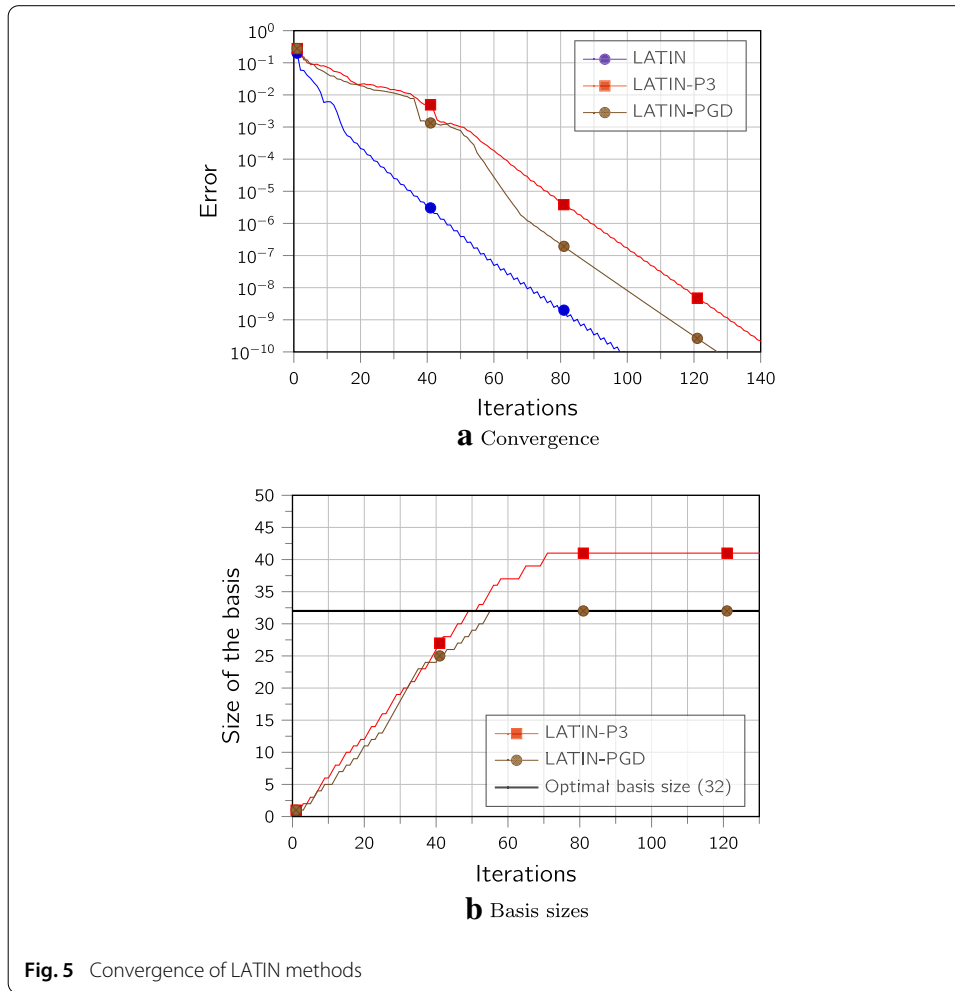
- (P1) **Separation of the linear and non-linear behaviors.** We denote by \mathbf{u} the displacement field over $\Omega \times [0, T]$ and λ the contact force field over $\partial_3\Omega \times [0, T]$. \mathcal{A} denotes the set of solutions $\mathbf{s} = (\mathbf{u}, \lambda)$ satisfying linear constitutive law, kinematic admissibility and static admissibility. These are defined on the whole space-time domain $\Omega \times [0, T]$. Γ denotes the set of solutions $\hat{\mathbf{s}} = (\hat{\mathbf{v}}, \hat{\lambda})$ verifying frictional contact conditions and are defined locally at the contacting interface and on the whole time interval $\partial_3\Omega \times [0, T]$. The solution of the problem is $\mathbf{s} \in \mathcal{A} \cap \Gamma$.
- (P2) **A two-staged iterative algorithm.** The solution of the problem is searched with the construction of two sequences of approximations belonging alternatively to \mathcal{A} and Γ . At the i th iteration, the local stage consists in finding $\hat{\mathbf{s}}_i = (\hat{\mathbf{v}}_i, \hat{\lambda}_i) \in \Gamma$ with a search direction $(\hat{\mathbf{s}}_i - \mathbf{s}_{i-1}) = (\hat{\mathbf{v}}_i - \mathbf{v}_{i-1}, \hat{\lambda}_i - \lambda_{i-1}) \in \mathbf{E}^+$. Note that $\mathbf{s}_{i-1} = (\mathbf{v}_{i-1}, \lambda_{i-1})$ is known from the previous iteration. Then, the global stage consists in finding $\mathbf{s}_i = (\mathbf{v}_i, \lambda_i) \in \mathcal{A}$ with another search direction $(\mathbf{s}_i - \hat{\mathbf{s}}_i) = (\mathbf{v}_i - \hat{\mathbf{v}}_i, \lambda_i - \hat{\lambda}_i) \in \mathbf{E}^-$. Note that $\hat{\mathbf{s}}_i = (\hat{\mathbf{v}}_i, \hat{\lambda}_i)$ is known from the previous local stage.
- (P3) **Radial approximation or space-time separation.** Unknown fields are represented as a sum of products between a space function and a time function to limit memory usage. An orthonormality condition is prescribed for space modes (i.e. left vectors).

For certain cases and for a sake of simplicity, the LATIN method can be formulated without the space-time separation (i.e. the solution is not sought into a low rank approximation). In this case several similarities can be stated with augmented Lagrangian methods [26]. All in all, the LATIN method for frictional contact problems consists in global / local strategy whose global stage does not require matrix re-factorization (stiffness operator remains constant along LATIN iterations, symmetric and definite positive) and local stage is explicit (no iterations are required to handle the non-linear behavior at the contacting boundary). As a consequence, comparisons between LATIN and Newton solvers is not an easy task as the number of iterations is not a good performance indicator for possible comparison. Only CPU measures seem a good approach for that purpose.

Numerical results

We consider the LATIN method (including only the first and second principle) as the reference non-linear solver.

Then, the LATIN method including the third principle (LATIN-P3) and the LATIN method including the third principle and the suggested algorithm are compared (LATIN-PGD). On Fig. 5 the convergence and evolution of basis sizes are depicted. Convergence plots show that for a given level of accuracy the LATIN method needs less iterations than LATIN-P3 and LATIN-PGD. Nonetheless, an iteration of the LATIN method is



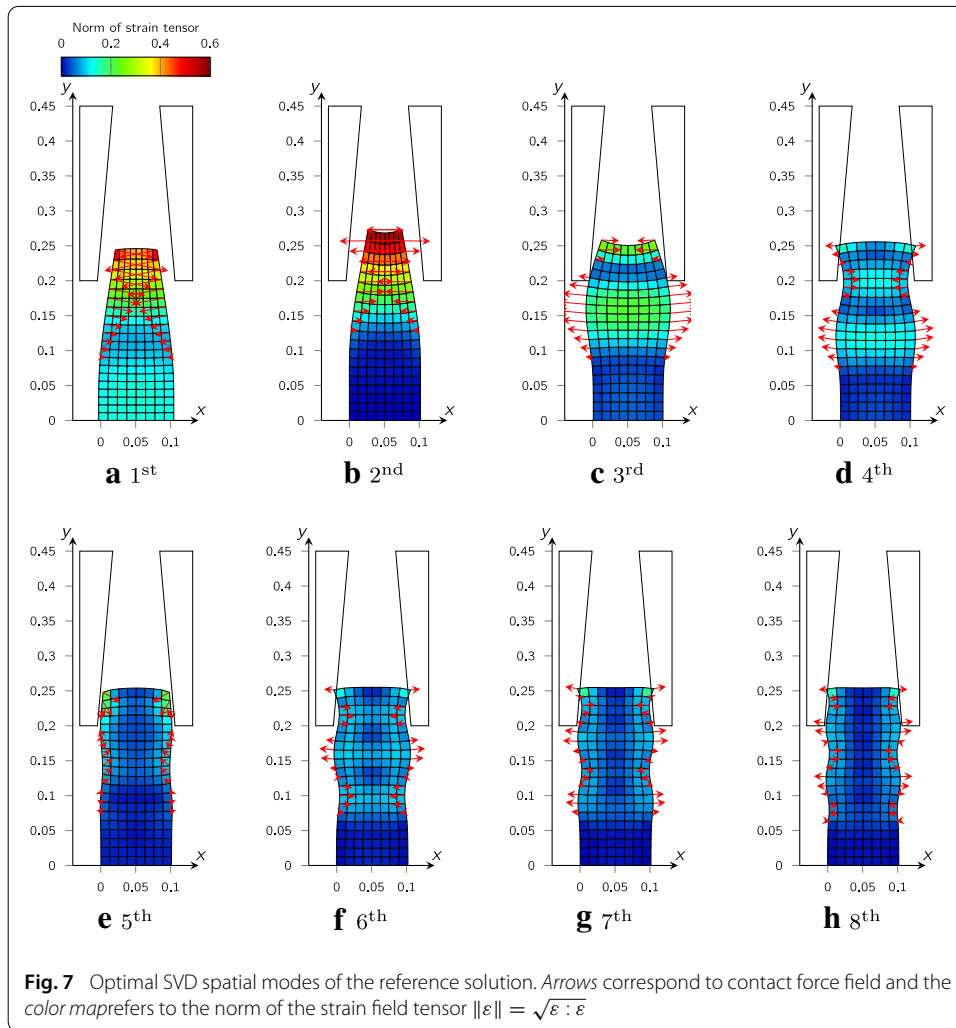
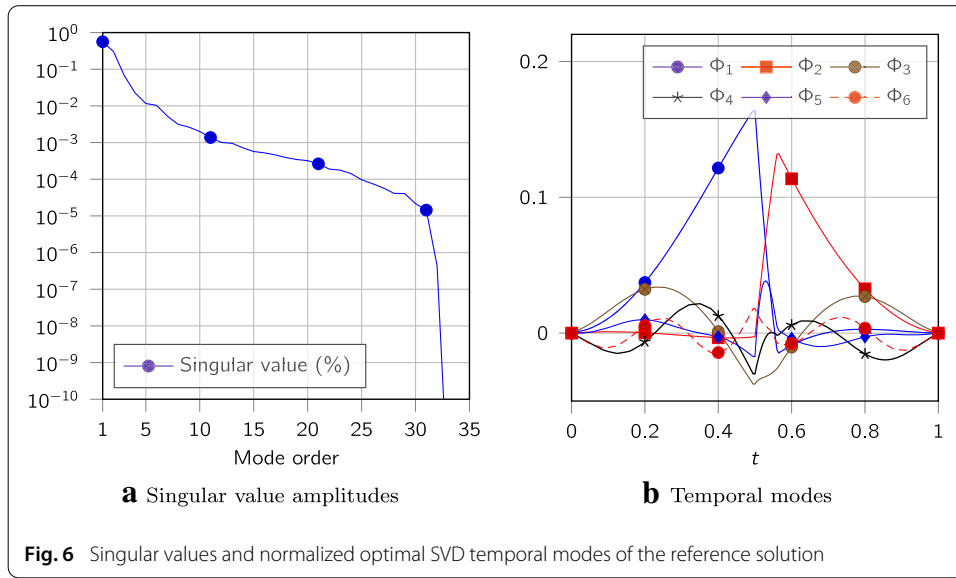
more computationally consuming than LATIN-P3 or LATIN-PGD. Such a behavior is not surprising as the space-time separation is not prescribed for the LATIN method, leading to a better accuracy of the iterated solutions.

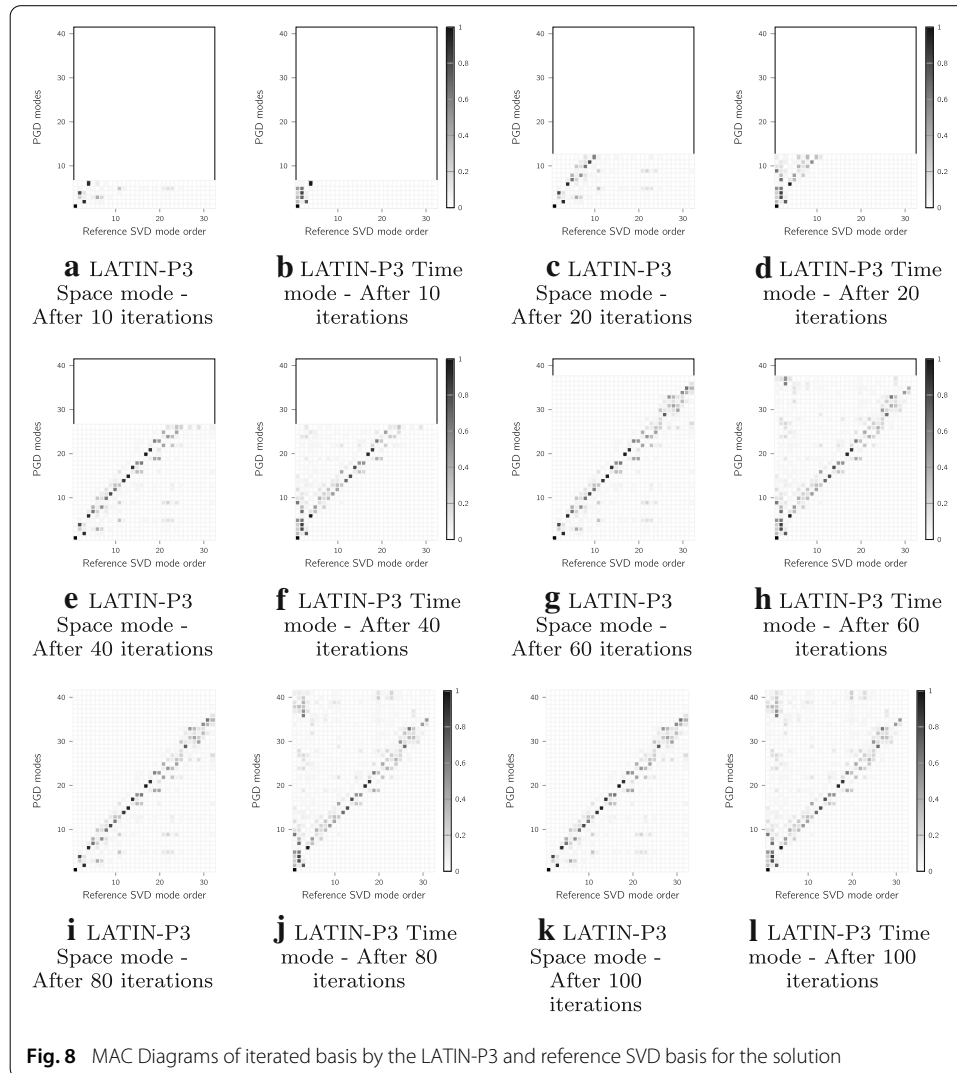
As far as accuracy is concerned, LATIN-P3 and LATIN-PGD shows similar performances on a first group of iterations. Then, convergence rate of LATIN-PGD accelerates until being stabilized to an asymptotic convergence rate (which is reached by the LATIN-P3 next the first group of iterations). Basis sizes evolution is interesting: SVD analysis of the reference solution shows that the full space-time solution is optimally generated with 32 modes (see Figs. 6, 7). From a quantitative point of view, the LATIN-P3 generates a low rank approximation for the solution that overshoots the optimal basis size whereas the LATIN-PGD generate a basis which size fits the optimal.

Given two sets of vectors of same dimension $(\mathbf{X}_i)_1^p$ and $(\mathbf{Y}_i)_1^q$, the MAC (modal assurance criterion) matrix [27] denoted by \mathbf{M} whose entries are:

$$M_{ij} = \frac{|\mathbf{X}_i^T \mathbf{Y}_j|^2}{\|\mathbf{X}_i\|^2 \|\mathbf{Y}_j\|^2} \in [0, 1] \quad (21)$$

for $1 \leq i \leq p$ and $1 \leq j \leq q$. The coefficient M_{ij} measures the correlation between modes \mathbf{X}_i and modes \mathbf{Y}_j . If $M_{ij} = 1$, then \mathbf{X}_i and \mathbf{Y}_j are colinear (highly correlated). On the contrary, $M_{ij} = 0$ means that \mathbf{X}_i and \mathbf{Y}_j are orthogonal.

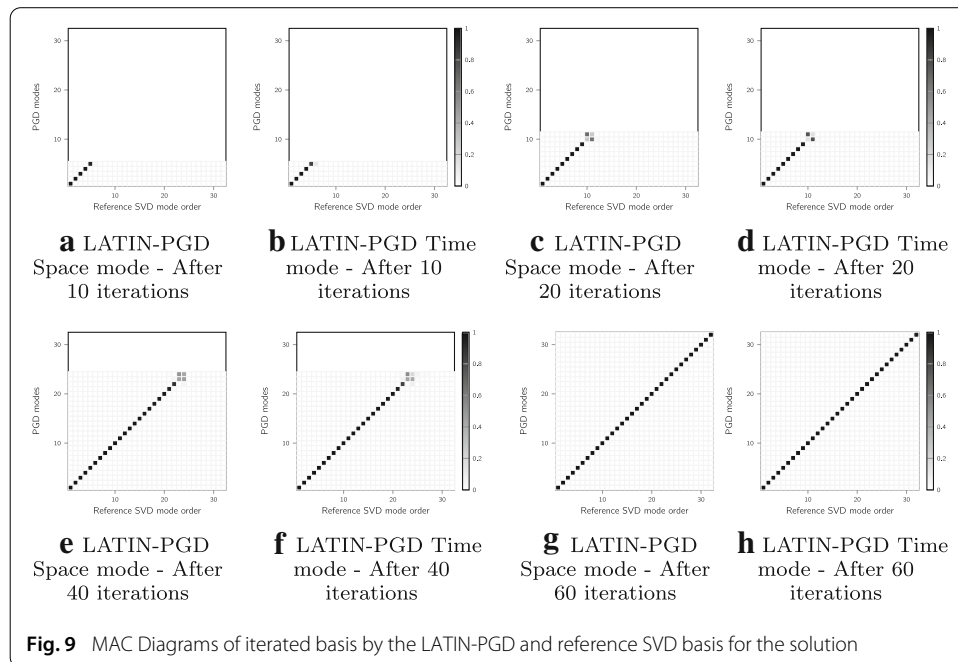




On Figs. 8, 9, MAC matrices are plotted to assess the quality of iterated basis for both methods. LATIN-P3 catches roughly the trends of the solution. But optimal vectors are obviously not computed. On the other hand, the LATIN-PGD computes painlessly dominant trends and iterated vector are very close to SVD optimal vectors of the solution. Even if a given iterated vector is not the most suited one (in regard to the converged solution), it is quickly corrected through next iterations. The LATIN-PGD computes nearly the solution of the numerical problem into its optimal SVD expansion. Additional numerical experiments confirm that the combination of the proposed algorithm with a LATIN-P3 method achieves a strong solver to design quasi-optimal basis for the solution with a reduced computational effort. The basis enriching strategy allowed by the PDG is completed with an on-the-fly compression strategy provided by the proposed algorithm.

Conclusion

In this paper, an iterative SVD algorithm is proposed. It relies on rotations around subspace which compress a given low rank approximation to its SVD form. Different strategies can be proposed as far as rotations are concerned (selection, order, simultaneity ...) provided



that appropriate conditions are fulfilled. Nonetheless, its interest does not rely on SVD expansions but on quasi-optimal bases which are expected to be close to. Indeed, the proposed algorithm feature is to provide such quasi-optimal bases after a few iterations. This efficiency depends on low rank expansion characteristics (ratios of right vector norms).

It provides an interesting tool for basis enrichment strategies. Usually, reduced order modeling techniques do not require a computationally expensive optimal basis (i.e. quasi-optimal is enough). These enrichment strategies can be embedded into PGD methods as shown herein. But a posteriori or SVD approaches within big data framework could also be concerned. Indeed, to design a relevant basis upon which ROMs or surrogate models are built, snapshots are stored and an associated generated basis has to be updated. The basis update can be expensive if one considers the optimal basis. Using the proposed algorithm makes a compromise by refreshing cheaply the basis but weakens the optimality property. Moreover, the suggested algorithm enables to consider specific inner product and to control the quasi-optimality.

An interesting extension of such an algorithm could be designed for higher order rank one tensors. This extension could be useful for PGD multiparametric studies and may converge to recent works concerning the High Order SVD (HOSVD) or similar tensor decomposition [28–32].

Authors' contributions

The three authors contributed to the implementation of the suggested quasi-optimal LATIN-PGD method. DD and AGi participated to the development of mathematical proves and numerical studies of the suggested compression algorithm. AGi has drafted the manuscript. DD and AGr have supervised the different studies and the corrections of the draft. All authors read and approved the final manuscript.

Author details

¹ANSYS France, 11 avenue Albert Einstein, 69100 Villeurbanne, France, ²LaMCoS, CNRS UMR 5259, 18-20 rue des Sciences, 69621 Villeurbanne Cedex, France, ³Institut universitaire de France, 1, rue Descartes, 75231 Paris Cedex 05, France.

Received: 12 December 2015 Accepted: 1 April 2016

Published online: 25 April 2016

References

- Brand M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl.* 2006;415(1):20–30. doi:[10.1016/j.laa.2005.07.021](#).
- Bunch JR, Nielsen CP. Updating the singular value decomposition. *Numerische Mathematik.* 1978;31(2):111–29. doi:[10.1007/BF01397471](#).
- Golub GH, Van Loan CF. *Matrix computations*, vol. 3. Baltimore: John Hopkins University Press; 2012.
- Giacoma A, Dureisseix D, Gravouil A, Rochette M. A multiscale large time increment/fas algorithm with time-space model reduction for frictional contact problems. *Int J Numer Methods in Eng.* 2014;97(3):207–30. doi:[10.1002/nme.4590](#).
- Eckart C, Young G. The approximation of one matrix by another of lower rank. *Psychometrika.* 1936;1:211–8.
- Carlberg K, Farhat C, Cortial J, Amsallem D. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J Comput Phys.* 2013;242:623–47. doi:[10.1016/j.jcp.2013.02.028](#).
- Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *Int J Numer Methods Eng.* 2012;92(10):891–916. doi:[10.1002/nme.4371](#).
- Amsallem D, Cortial J, Farhat C. Towards real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information. *AIAA J.* 2010;48(9):2029–37. doi:[10.2514/1.J050233](#).
- Li WK, McLeod AI. Distribution of the residual autocorrelations in multivariate arma time series models. *J R Stat Soc B.* 1981;43(2):231–9.
- Asteriou D, Hall SG. *ARIMA models and the Box-Jenkins methodology*. 2nd ed. New York: Palgrave MacMillan; 2011. p. 266–85.
- Box GEP, Jenkins GM. *Time series analysis: forecasting and control*. 3rd ed. Englewood Cliffs: Prentice Hall; 1994.
- Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *J Non Newtonian Fluid Mech.* 2006;139(3):153–76. doi:[10.1016/j.jnnfm.2006.07.007](#).
- Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids: Part II: Transient simulation using space-time separated representations. *J Non Newtonian Fluid Mech.* 2007;144(2–3):98–121. doi:[10.1016/j.jnnfm.2007.03.009](#).
- Chinesta F, Keunings R, Leygue A. *The proper generalized decomposition for advanced numerical simulations*. Heidelberg: Springer; 2014.
- Giacoma A, Dureisseix D, Gravouil A, Rochette M. Toward an optimal a priori reduced basis strategy for frictional contact problems with LATIN solver. *Comput Methods Appl Mech Eng.* 2015;283:1357–81. doi:[10.1016/j.cma.2014.09.005](#).
- Kikuchi N, Oden JT. *Contact problems in elasticity: a study of variational inequalities and finite element methods*. Stud Appl Numer Math. 1988. doi:[10.1137/1.9781611970845](#).
- Laursen TA. *Formulation and treatment of frictional contact problems using finite elements*. PhD thesis, Stanford University. 1992.
- Ladevèze P. *Nonlinear computational structural methods: new approaches and non-incremental methods of calculation*. New York: Springer; 1999.
- Relun N, Néron D, Boucard P-A. A model reduction technique based on the pgd for elastic-viscoplastic computational analysis. *Comput Mech.* 2013;51(1):83–92. doi:[10.1007/s00466-012-0706-x](#).
- Champaney L, Cognard J-Y, Ladevèze P. Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions. *Comput Struct.* 1999;73:249–66. doi:[10.1016/S0045-7949\(98\)00285-5](#).
- Boucard P-A, Ladevèze P, Poss M, Rougée P. A nonincremental approach for large displacement problems. *Comput Struct.* 1997;64(1–4):499–508. doi:[10.1016/S0045-7949\(96\)00165-4](#).
- Odièvre D, Boucard P-A, Gatuignt F. A parallel, multiscale domain decomposition method for the transient dynamic analysis of assemblies with friction. *Comput Methods Appl Mech Eng.* 2010;199(21–22):1297–306. doi:[10.1016/j.cma.2009.07.014](#).
- Boucinha L, Gravouil A, Ammar A. Space-time proper generalized decompositions for the resolution of transient elastodynamic models. *Comput Methods Appl Mech Eng.* 2013;255:67–88. doi:[10.1016/j.cma.2012.11.003](#).
- Ribeaucourt R, Baïetto-Dubourg M-C, Gravouil A. A new fatigue frictional contact crack propagation model with the coupled X-FEM/LATIN method. *Comput Methods Appl Mech Eng.* 2007;196:3230–47. doi:[10.1016/j.cma.2007.03.004](#).
- Trollé B, Gravouil A, Baïetto M-C, Nguyen-Tajan TML. Optimization of a stabilized X-FEM formulation for frictional cracks. *Finite Elem Anal Des.* 2012;59:18–27. doi:[10.1016/j.finel.2012.04.010](#).
- Alart P, Dureisseix D, Renouf M. Using nonsmooth analysis for numerical simulation of contact mechanics. *Nonsmooth mechanics and analysis: theoretical and numerical advances*. *Advances in Mechanics and Mathematics*, vol 12. Kluwer Academic Press; 2005. p. 195–207. doi:[10.1007/0-387-29195-4_17](#).
- Allemang RJ. The modal assurance criterion—twenty years of use and abuse. *Sound and vibration magazine.* 2003;37(8):14–23.
- Modesto D, Zlotnik S, Huerta A. Proper generalized decomposition for parameterized helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. *Comput Methods Appl Mech Eng.* 2015;295:127–49. doi:[10.1016/j.cma.2015.03.026](#).
- Kolda TG. Orthogonal tensor decompositions. *SIAM J Matrix Anal Appl.* 2001;23(1):243–55. doi:[10.1137/S0895479800368354](#).
- De Lathauwer L, De Moor B, Vandewalle J. On the best rank-1 and rank- (R_1, R_2, \dots, R_n) approximation of higher-order tensors. *SIAM J Matrix Anal Appl.* 2000;21(4):1324–42. doi:[10.1137/S0895479898346995](#).
- De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM J Matrix Anal Appl.* 2000;21(4):1253–78. doi:[10.1137/S0895479896305696](#).
- Luo D, Ding C, Huang H. Are tensor decomposition solutions unique? On the global convergence HOSVD and ParaFac algorithms. In: Huang J, Cao L, Srivastava J, editors. *Advances in knowledge discovery and data mining. Lecture notes in computer science*, vol. 6634. Heidelberg: Springer; 2011. pp. 148–159. doi:[10.1007/978-3-642-20841-6_13](#).