# An adaptive ROM approach for solving transfer equations

**Nicolas Verdon*** — **Cyrille Allery*** — **David Ryckelynck****
**Aziz Hamdouni***

* *LEPTAB, Université de La Rochelle, Avenue Michel Crépeau*
*F-17042 La Rochelle cedex 1*
** *LMSP, UMR CNRS-ENSAM-ESEM, Ecole Nationale Supérieure des Arts et Métiers*
*151 boulevard de l'Hôpital, F-75013 Paris*

ABSTRACT. *In this article, we present an adaptive method for solving transfer equations. The method consists in projecting the discretized problem on a basis we have defined in order to obtain a reduced model that can be quickly and accurately solved with classic numerical schemes. The originality of the methods stays in the way of the basis is constructed. At each iteration of computation, the basis is adapted: first the old basis is improved using a Karhunen-Loève decomposition whereas in a second phase the improved basis is expanded with Krylov vectors. The example we study is the one-dimension Burgers' equation. The results we obtained were compared to the Newton-Raphson method: whereas the accuracy is not better than the Newton-Raphson method, we show that the computationnal time is drastically reduced. In addition, the basis we obtain shows a great ability to represent the long-time dynamics of the system, as shown in the last part of the paper.*

RÉSUMÉ. *Dans cet article, nous présentons une méthode adaptative pour la résolution des équations de transfert. La méthode consiste à projeter le problème discrétisé sur une base que l'on a définie afin d'obtenir un modèle réduit qui peut être résolu rapidement et précisément avec des schémas numériques classiques. L'originalité de la méthode réside dans la façon dont la base est construite. A chaque itération de calcul, la base est adaptée : dans un premier temps l'ancienne base est améliorée à l'aide d'une décomposition de Karhunen-Loève tandis que, dans un second temps la base améliorée est enrichie avec des vecteurs de Krylov. La méthode a été testée sur le cas simple de l'équation de Burgers 1-D et les résultats ont été comparés à ceux obtenus avec la méthode de Newton-Raphson : la précision de la solution obtenue n'est pas meilleure mais le temps de calcul est considérablement réduit. De plus, nous montrons également la capacité de la base obtenue à décrire le comportement dynamique de la solution de l'équation de Burgers.*

KEYWORDS: *model reduction, Krylov subspace, proper orthogonal decomposition, low-order dynamical system, Burgers' equation.*

MOTS-CLÉS : *réduction de modèle, sous-espace de Krylov, décomposition orthogonale aux valeurs propres, système dynamique d'ordre faible, équation de Burgers.*

## 1. Introduction

In various fields of mechanics, fine discretizations employed to treat non-linear problems lead to very large non-linear systems to solve. Even if the solution one obtains is very accurate, solving these large systems is very computationally expensive that is why Reduced Order Models (ROM) are more and more used for this purpose. In the field of Computational Fluid Dynamics, two important categories of computational methods are very employed.

As a first category we find the *Krylov subspace methods* which are iterative methods for solving large linear systems $Ax = b$. They are projecting methods which consist in finding an approximation $x_m$ of the solution from an affine subspace $x_0 + \mathcal{K}_m$ where $\mathcal{K}_m$ is the Krylov subspace of dimension $m$. More details about the different classes of Krylov methods are given in (Saad, 1996) and (Van der Vorst, 2003). They also explain which method must be employed depending of the problem we have to treat. For example, for symmetric positive definite matrices, the Conjugate Gradient is the best method. For non-symmetric systems arising from discretizations of partial differential equations, it is common to use the General Minimal RESidual algorithm (GMRES) because it is known as the most robust one. But methods which combine non-linear outer and linear inner iterative procedure are also studied tu solve non-linear problems (Brown *et al.*, 1990; Knoll *et al.*, 2000). These methods, called *Newton-Krylov* methods, consist in mixing Newton algorithm and Krylov iterations at each Newton iterations, and are reviewed in (Knoll *et al.*, 2004).

Then, the second important method in CFD is a ROM called *Proper Orthogonal Decomposition*. Introduced in 1967 by Lumley (Lumley, 1967), it has been intensively used since the 80's in many applications such as optimization (Kunisch *et al.*, 1999), particle dispersion (Allery *et al.*, 2005) or even aeroelasticity (Beran *et al.*, 2004). This method, also known as the *Karhunen-Loève decomposition* in other domains, is based on the fact one can extract an optimal basis of the field. In order to obtain this basis, one need samples of the studied field, either numerical (DNS or LES), or experimental ones (PIV). The principle of the method is at the same time its most important disadvantage because one requires the knowledge of the flow over a short time interval to compute the POD basis. To get this sampling of solution, both numerical and experimental ressources are very expensive.

In order to avoid this limiting step, the method we propose consists in combining projections phases with improvements of the projection basis. Thus, one is assumed to get a representative basis without knowledge of the flow and at the same time, the algorithm provides us the solution of the problem. Furthermore, one can treat different configurations by changing control parameters such as the Reynolds number. Indeed, with a POD basis corresponding to a given configuration, one can not obtain solution when any of the control parameter is changed whereas with our algorithm, one is assumed to be able to adapt quickly our original basis to this change of parameter. Hence, with this basis one is supposed to be able to compute the solution for further times by constructing a dynamical system, as the same manner as we usually do with the POD basis. The methods have already been implemented in (Ryckelynck, 2005) in a symmetric case that is why the case we treat here is non-symmetric: the Burgers' equation, which contains the difficulty arising from the non-linear convective term. Its one-dimensional expression is recalled here:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} \tag{1}$$

Because one can find analytic solutions for certain boundary and initial conditions, this equation is a good first step for testing new numerical methods.

In the following sections of the article, we present the Proper Orthogonal Decomposition and the method we have implemented. Then, the results obtained for the one-dimensional Burgers' equation are compared to those calculated with the Newton-Raphson method and the POD. The influence of many parameters is tested before discussing the future improvements and applications of the method.

## 2. The Proper Orthogonal Decomposition

### 2.1. *Formulation of the problem*

The main aim of the POD (Proper Orthogonal Decomposition) is to find a set of orthogonal basis which describe as best as possible a random field. Let $u(x,t)$ be this random process in the bounded domain $\Omega$. We will now try to express $u$ as a decomposition of orthonormal spatial functions $\psi_n$ and the corresponding temporal coefficients $a_n$. The problem of the POD can be formulated in the sense that the seeking functions must represent the field as best as possible in a energetical sense, that is why we can express the following maximization problem:

$$\max_{\varphi \in H} \frac{\langle |(u,\varphi)|^2 \rangle}{(\varphi,\varphi)} = \frac{\langle |(u,\psi)|^2 \rangle}{(\psi,\psi)} \tag{2}$$

where $H$ is a Hilbert space, $\langle \bullet \rangle$ is a mean operator and $(\bullet, \bullet)$ is the scalar product on $H$, namely if we work in the space of square integrable functions $H = L^2(\Omega)$, we can define the associated scalar product as follows:

$$\forall f, g \in L^2(\Omega) \quad (f,g) = \int_\Omega f(x) \cdot g^*(x) dx \text{ où } x \in \Omega \tag{3}$$

where $g^*(x)$ represents the complex conjugate of $g(x)$. In our study, we will work in the real case.

The maximisation problem [2] leads, after some variational calculus (for further details, see (Holmes *et al.*, 1996)), to the following problem:

$$\int_\Omega R(x,x')\psi(x')dx' = \lambda\psi(x) \tag{4}$$

which is known as the *Fredholm equation*. In Equation [4], $R(x,x')$ is the *correlation tensor* and is defined as $R(x,x') = \langle u(x)u^*(x')\rangle$.

From Hilbert-Schmidt theory, it is well known that it exists a denumerable infinity of solutions of the Fredholm equation. Furthermore, because the correlation tensor is non-negative, these eigenvalues are semi-positive definite. Hence we can write each realization $u$ of the flow as its proper orthogonal decomposition:

$$u(x,t) = \sum_{i=1}^{\infty} a_i(t)\psi_i(x) \qquad [5]$$

where the temporal coefficients $\{a_i\}_{i=1,\cdots,\infty}$ verify:

$$\langle a_i(t)a_j^*(t)\rangle = \delta_{ij}\lambda_i \qquad [6]$$

This technique is widely used in many configurations because it has lots of advantages in practice and the most important are surely its optimality (in an energetic sense) and orthogonality.

### 2.2. *Snapshot POD*

Experimental samples provide more temporal informations than spatial ones, in opposition with samples arising from numerical computations. That is why, in order to minimize computational times for samples provided by fine numerical computations, Sirovich (Sirovich, 1987) proposed the *Snapshots method* in 1987. If we call $N$ the number of nodes of the discretization, the problem is of size $N \times N$ which can become very computationnaly expensive when increasing $N$. To avoid this problem, Sirovich considered that a sampling of $M$ realizations ($M \ll N$) of the flow is sufficient to describe the problem well.

Consider now the $M$ realizations of the flow. If $\boldsymbol{\psi}_i$ where $i = 1, \cdots, M$ is a mode of the decomposition we have:

$$\boldsymbol{\psi}_i = \sum_{k=1}^{M} a_k \mathbf{u}(x, t_k) \qquad [7]$$

Assuming that the ergodicity[1] condition is true, we can write the temporal mean as the statistical one and, recognizing the definition of the scalar product $(u^k, u^i) = \displaystyle\int_\Omega u^k(x')u^{i^*}(x')dx'$, we obtain the following eigenvalues problem to solve:

$$\frac{1}{M}\sum_{k=1}^{M}(u^k, u^i)a_k = \lambda a_i \text{ pour } i = 1, \cdots, M \qquad [8]$$

Hence, the temporal coefficients $a_k$ are obtained by solving the Equation [8] and the spatial modes are determined using [7].

[1] *i.e.* statistical and temporal means can permute.

## 3. The method of A Priori Reduction

In this Section, we present the adaptive Reduced Order Model method which was called *A Priori Reduction* method. *A priori* because we build a basis without knowledge of the solution before the computation, and *adaptive* because the basis is modified in order to be able to represent the solution.

### 3.1. *Discretization of the equation*

Every evolution problem of parabolic type can be discretized to be numerically solved. We write here symbolically the equation we have to solve:

$$\mathcal{F}_{t+\Delta t}(\mathbf{u_h}) = \mathcal{G}_t(\mathbf{u_h}) \qquad [9]$$

where $\mathbf{u_h}$ represents the discretised unknown, $\mathcal{F}$ and $\mathcal{G}$ represent operators which describe the parts of the discretized equation, respectively at the time $t + \Delta t$ and at the time $t$.

### 3.2. *Presentation of the method*

The method we have developped is separated into two distinct steps: a projection step which constitutes the reduction phase of the discretized system and the second step regroups the improvement and the expension of the projection basis in order to accelerate the convergence to the result. The method is hence adaptive because the basis is modified at each iteration of our computation. More precisely, we can decompose the algorithm into the following steps which will be described and discussed in detail in the following paragraphs:

1) *initialisation phase*: initialise the projection basis,

2) *reduction phase*: project the non-linear system of equations on the basis and solve the non-linear reduced system,

3) *resolution phase*: compute the residuals at each time step and check the convergence criterion, if the criterion is not validated, go to step 4,

4) *adaptive phase*: (a) Improve the quality of the basis if necessary with a Karhunen-Loève decomposition and (b) expand the number of basis vectors. Return to step 2.

#### 3.2.1. *Initialisation*

In this step, we must define the first space on which we will project our governing equations. The initial basis is the Krylov subspace of order $m$ associated to an initial residual vector $\mathbf{r}^0$, that is to say:

$$\psi^{(0)} = \mathcal{K}_m = \{\mathbf{r}^0, \mathbf{K}^0\mathbf{r}^0, \mathbf{K}^{0^2}\mathbf{r}^0, \cdots, \mathbf{K}^{0^{m-1}}\mathbf{r}^0\} \qquad [10]$$

where $\mathbf{K}^0$ indicates the jacobian matrix of the system at the initial time. In many works, it is common to take $m = 3$ but, for programming considerations, we have worked with Krylov subspaces of dimension one and the main results presented correspond to $m = 1$. It is obvious to remark we must avoid to take many basis vectors if those are too close to each other: the system would lead to a singular matrix and the computation would break down.

As explained in (Ryckelynck, 2005), the choice of the vector $\mathbf{r}^0$ for starting our computation has no fundamental importance for solving the problem, in other words we could choose an arbitrary vector for instance a constant vector, but we also observed that taking a good "physical" initial basis increase the convergence speed of the computation to the result. In all computations we have done, the first basis vector taken corresponds to the normalized initial condition vector. It is also good to precise that all basis vectors are normalized during a whole computation.

### 3.2.2. *Reduction phase*

The non-linear system [9], of size $N \times N$ where $N$ is the number of nodes in the discretization, can be relatively expensive to solve numerically if the mesh size is too fine. To reduce the number of equations to solve, the strategy of our method is based on Krylov methods for large sparse linear systems that is to say we project the system of equations – non-linear in our case – on a Krylov subspace or *modified* Krylov subspace, this will be explained in the following parts of the article. Let us now call $\boldsymbol{\psi}^{(k)}$ the basis corresponding to the projection subspace at an arbitrary iteration of computation $k$, and we call $n_k$ the number af basis vectors in $\boldsymbol{\psi}^{(k)}$. We suppose also that the velocity is known at the time $t$ and must be determined at $t + \Delta t$. If we project the discretized system onto this basis, we obtain:

$$\mathbf{u_h}_i^{(k)} = \sum_{j=1}^{n_k} \psi_{ij}^{(k)} \cdot \mathbf{a}_j^{(k)} \qquad [11]$$

where $\mathbf{u_h}_i = \mathbf{u_h}(x_i)$ and $\psi_{ij}^{(k)} = \psi_j^{(k)}(x_i)$. That is, in matricial notations:

$$\mathbf{u_h}^{(k)} = \boldsymbol{\psi}^{(k)} \cdot \mathbf{a}^{(k)} \qquad [12]$$

where $\boldsymbol{\psi}^{(k)}$ is the $N \times n_k$ matrix representing the basis, $\mathbf{u_h}^{(k)} = \mathbf{u}^{(k)}$ (the subscript is now ommitting when no confusion is possible) represents the vector of the discretized velocity and $\mathbf{a}^{(k)}$ the vector of the reduced variables. We can write [9] as follows:

$$\mathcal{F}_{t+\Delta t}(\boldsymbol{\psi}^{(k)} \cdot \mathbf{a}^{(k)}) = \mathcal{G}_t(\mathbf{u}^{(k)}) \qquad [13]$$

As we said, it is assumed that the solution is known at the time $t$ that is why we can write $\mathcal{G}_t(\mathbf{u}^{(k)})$ as a known second member $\mathcal{S}$:

$$\mathcal{F}_{t+\Delta t}(\boldsymbol{\psi}^{(k)} \cdot \mathbf{a}^{(k)}) = \mathcal{S} \qquad [14]$$

Then, by multiplying [13] by $\boldsymbol{\psi}^{(k)^T}$ where $^T$ is the transpose operator, we obtain the following reduced system, in matricial notations:

$$\boldsymbol{\psi}^{(k)^T} \cdot \mathcal{F}_{t+\Delta t}(\boldsymbol{\psi}^{(k)} \cdot \mathbf{a}^{(k)}) = \boldsymbol{\psi}^{(k)^T} \cdot \mathcal{S} \qquad [15]$$

which can be more symbolically expressed as the following non-linear reduced system of equations:

$$\mathcal{H}(\mathbf{a}^{(k)}) = 0 \qquad [16]$$

where $\mathbf{a}$ is the new unknown of the problem, of reduced dimension $n_k$.

### 3.2.3. *Resolution and reconstruction*

After projecting the global equation, we must solve the reduced system of non-linear equations at each time step. To do this, we use here a classical Newton-Raphson scheme, which is robust and very efficient for the size of the reduced system we considered in our computations. The Newton-Raphson scheme provides us the solution $\mathbf{a}$ of the reduced system at the time $t + \Delta t$. The global solution $\mathbf{u}$, at the time $t + \Delta t$, is then reconstructed using the relation [12]. Hence, we advance in time using the calculated solution as the known velocity in the second member of [13] which is again projected on $\psi_k$.

### 3.2.4. *Convergence criterion*

After an iteration of computation, one obtains a solution $\mathbf{u}$ for each time step. To check the quality of our method, we have chosen a classical convergence criterion, namely we consider that the calculation has converged if the difference between two solutions at two successive iterations, in the sense of the $L^2-$norm, is less than the value we have fixed.

In practice, between the iterations $k$ and $k+1$, the following inegality must be true to have a converged solution for each time step:

$$||\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}||_{L^2} < \varepsilon \qquad [17]$$

where $\varepsilon$ is the convergence criterion.

If the criterion is satisfied, the computation is stopped and the results are saved; if not, the basis is adapted as explained in the following section before performing a new iteration of projection and resolution.

### 3.2.5. *Adaptive phase*

If the convergence criterion is not validated, that means that the basis we use for the projection at the iteration $k$ is not able to represent well enough the solution. Hence, we have to adapt this basis before performing the next projection. The first step in this adaptive procedure is what was called in (Ryckelynck, 2005) the *improvement phase*. During the $k^{th}$ iteration, we store the $M_t$ vectors $\mathbf{a}^{(k)}$ – of dimension $n_k$ – for each time step $t_l$, $l = 1, \cdots, M_t$ and we perform a Karhunen-Loève decomposition over this set of $M_t$ realisations to extract the most important informations about the solution. In other words, the eigenvalues problem to solve at the iteration $k$ is the following:

$$\mathbf{C}^{(k)}\mathbf{\Phi}^{(k)} = \lambda\mathbf{\Phi}^{(k)} \qquad [18]$$

where $\mathbf{C}_{ij}^{(k)} = \sum_{l=1}^{M_t} a_i^{(k)}(t_l) a_j^{(k)}(t_l)^T$ is the $n_k \times n_k$ covariance matrix of our problem.

Then, we obtain $n_k$ eigenvalues that we sort as follows:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_{n_k} \qquad [19]$$

In order to keep only the most relevant informations provided by the set of vectors $\mathbf{a}$, only the significant $\eta$ first eigenvalues are taken into account, in practice $\eta$ is such as:

$$\lambda_\eta \geq \mu_{KL}\lambda_1 \qquad \text{and} \qquad \lambda_{\eta+1} < \mu_{KL}\lambda_1 \qquad [20]$$

We can also define a *selection matrix* $\mathbf{V}_k$, composed of the eigenvectors corresponding to the $\eta$ selected eigenvalues:

$$\mathbf{V}^{(\mathbf{k})} = [\mathbf{\Phi}_1^{(k)}, \mathbf{\Phi}_2^{(k)}, \text{‘} \cdots, \mathbf{\Phi}_\eta^{(k)}] \qquad [21]$$

The improved basis $\widetilde{\boldsymbol{\psi}^{(k)}}$ is then defined as follows:

$$\widetilde{\boldsymbol{\psi}^{(k)}} = \boldsymbol{\psi}^{(k)} \cdot \mathbf{V} \qquad [22]$$

In the second step, the basis is expanded to take into account the effects of the residuals computed during the calculation.

To do this, we construct the Krylov susbspace associated to a residual $\mathbf{r}^{(k)}$. Obviously, the problem consists in selecting the residual which will be at the origin of this basis. Many choices are possible, but the more physically is to choose the residual $\mathbf{r}^{t_e}$ at the time step corresponding to the time $t_e$ whose its $L_2$-norm exceeds a value $\mu_R$ fixed by the user. Hence, we are able to compute the additionnal basis:

$$\psi_{exp}^{(k)} = \{\mathbf{r}^{t_e}, \mathbf{K}^{t_e}\mathbf{r}^{t_e}, \mathbf{K}^{{t_e}^2}\mathbf{r}^{t_e}, \cdots, \mathbf{K}^{{t_e}^{m-1}}\mathbf{r}^{t_e}\} \qquad [23]$$

where $m$ is always the order of the Krylov subspace.

Finally, the projection basis obtained after the adaptive procedure, what we called the *modified* Krylov subspace is the concatenation of the improved basis and the expanded one:

$$\psi^{(k+1)} = \{\widetilde{\boldsymbol{\psi}^{(k)}}, \boldsymbol{\psi}_{exp}^{(k)}\} \qquad [24]$$

## 4. Low-order dynamical system

When the convergence of the of the computation is reached, one have a basis which describes the solution of Burgers' equation over a time interval supposed to be long enough to be representative of the evolution of the considered system, but short in comparison with the total time interval we want to compute. Hence, we use the decomposition of the flow we have obtained during the reduction procedure to perform a Galerkin projection of the equation on the corresponding basis.

### *Galerkin projection*

The decomposition of the field over the computed basis can be written as:

$$u(x,t) = \sum_{i=1}^{N} a_i(t)\psi_i(x) \qquad [25]$$

Using the expression [25] of the velocity, the Burgers' Equation [1] can be rewritten as follows:

$$\sum_{i=1}^{N} \frac{da_i}{dt}\psi_i = \nu \sum_{i=1}^{N} a_i \frac{d^2\psi_i}{dx^2} - \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \psi_i \frac{d\psi_j}{dx} \qquad [26]$$

At this point we can precise that the modes in the basis we have calculated are normed but not necessarily orthogonal. To ensure the orthogonality of the basis vectors, we performed a Gram-Schmidt algorithm to obtain an orthonormal basis. After projecting this equation on this basis, and because the basis vectors are orthonormal, we obtain :

$$\frac{da_i}{dt} = \nu \sum_{j=1}^{N} B_{ij} a_j + \sum_{j=1}^{N}\sum_{k=1}^{N} C_{ijk} a_j a_k \text{ for } i = 1, \cdots, N \qquad [27]$$

where we introduced $B_{ij} = \left(\psi_i, \frac{d^2\psi_j}{dx^2}\right)$ and $C_{ijk} = -\left(\psi_i, \psi_j \frac{d\psi_k}{dx}\right)$.

Then, we just need to compute $B_{ij}$ and $C_{ijk}$ just once to solve this system of differential equations over the time interval we desire.

## 5. Numerical results

In this part, we present the results obtained with the A Priori Reduction (APR) method on the example of the one-dimension Burgers' equation.

### 5.1. *Test case*

The method was implemented to solve the Burgers' equation which is recalled here:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x} \qquad [28]$$

Under the following initial and boundary conditions:

$$\begin{cases} u(x,0) = \sin(\pi x) & \text{for} \quad 0 < x < 1 \\ u(0,t) = u(1,t) = 0 & \text{for} \quad t > 0 \end{cases} \qquad [29]$$

We can express an analytical solution in terms of Fourier series as follows:

$$u(x,t) = 2\pi\nu \frac{\sum_{n=1}^{\infty} a_n e^{-n^2\pi^2\nu t} n \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n e^{-n^2\pi^2\nu t} \cos(n\pi x)} \qquad [30]$$

where the Fourier coefficients are defined by the following expressions:

$$\begin{aligned} a_0 &= \int_0^1 \exp\{-(2\pi\nu)^{-1}[1-\cos(\pi x)]\}dx \\ a_n &= 2\int_0^1 \exp\{-(2\pi\nu)^{-1}[1-\cos(\pi x)]\}\cos(n\pi x)dx \end{aligned}$$

The Burgers' equation was discretized using a finite difference upwind scheme. All the solutions were computed in time with the Crank-Nicholson scheme, the parameters for the residual selection $\mu_R$ and for the selection of the relevant eigenvalues $\mu_{KL}$ were respectely fixed for the test case at the following values:

$$\mu_R = 10^{-6} \quad \text{and} \quad \mu_{KL} = 10^{-10} \qquad [31]$$

At last, we fixed the convergence criterion at $\varepsilon = 10^{-5}$.

### 5.2. *Results and discussion*

As a first test, we computed the solution over $0.1s$ with a number of nodes $N_x = 50$ and a time step fixed at $\Delta t = 10^{-4}s$. The Figure 1 shows the solution obtained with the A Priori Reduction method in comparison with the analytical solution at the times $t = 0.05s$ and $t = 0.1s$. The method is then able to compute a solution which seems to be qualitatively quite good.
In order to check more precisely the accuracy of the method, we have computed solutions for a number of discretization nodes $N_x$ from 50 to 200, and calculated the error with the analytic solution, in the sense of the $L^2-$norm, defined as follows:

$$e = \max_{t\in T} ||u_{APR}(t) - u_{an}(t)||_{L^2} \qquad [32]$$

Table 1 compares the errors with the analytic solution for the A Priori Reduction (APR) and the Newton-Raphson method applied to the global system (GNR). As a first inreresting result, we observe that the APR method has an accuracy of the same order as the NR one which is what we expected. Indeed, the goal of our method is to find *a priori* a basis, which is able to accurately represent the solution of the transfer equations, but at the same time with a small computational effort.

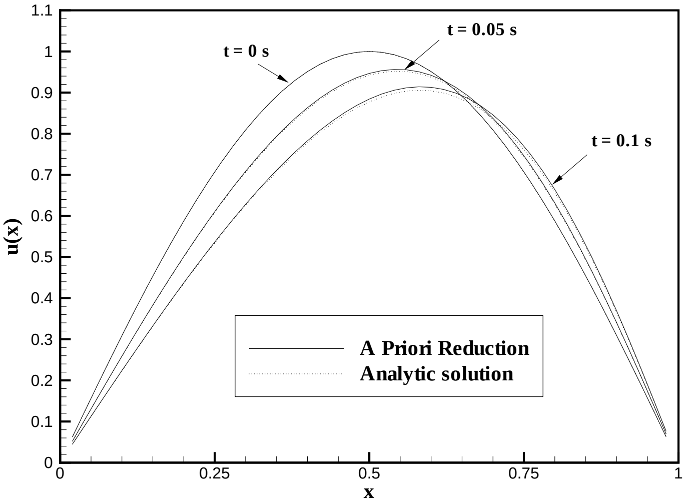

**Figure 1.** *Solution of Burgers' equation at different times for* $\nu = 0.1$, $N_x = 50$

**Table 1.** *Comparison of the precision of the method with Newton-Raphson in the sense of the* $L^2$*–norm* ($\times 10^{-2}$)

| $N_x$ | error APR | error GNR |
|---|---|---|
| 50 | 0.543 | 0.538 |
| 60 | 0.451 | 0.446 |
| 70 | 0.386 | 0.381 |
| 80 | 0.337 | 0.332 |
| 90 | 0.3 | 0.295 |
| 100 | 0.269 | 0.264 |
| 150 | 0.179 | 0.174 |
| 200 | 0.134 | 0.129 |

That is why we show in Figure 2 the evolution of the computational times between the APR and NR methods when increasing the number of nodes. For small

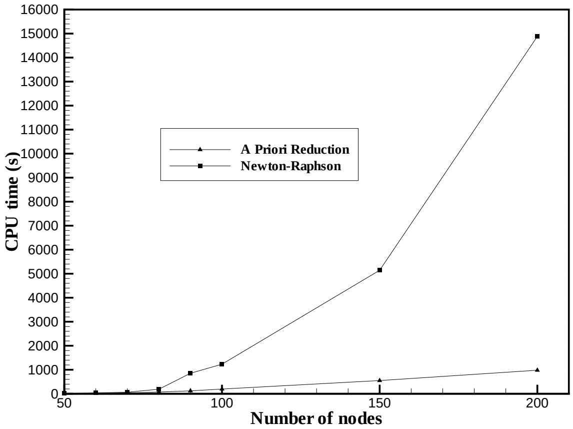

**Figure 2.** *Comparison of CPU time between Newton-Raphson and the A Priori Reduction method for* $Nx = 50, \cdots, 200$ *over* $0.1s$

discretizations, computational times are of the same order but, from $Nx = 60$, when we increase the number of nodes, the difference between the computational time necessary for each method grows very fast. Then, the APR method is less computationally expensive and for example, a APR computation requires 15 times less time than the corresponding NR one for $N_x = 200$ which represents a very important gain of time for a one-dimension problem.

To ensure the quality of the method we have tried to compute the solution for longer times, namely over $3s$.

The solution we obtain is represented in Figure 3 compared to the analytic solution for any values of $t$. In addition, Figure 4 shows the corresponding error. We can see that the maximum error is obtained around $t \approx 0.394s$.

In Figure 4 for $t$ from $0s$ to $0.394s$, we can remark that the error increases when the convective effects are dominant, whereas after $t = 0.394s$, the diffusion become more important and the error decreases asymptotically to zero. Finally, we were able to obtain accurate solutions of the Burgers' equation even for long times with a few computational effort, but computing the solution over $3s$ for $N_x = 100$ and with a time step equal to $\Delta t = 10^{-4}$ requires $2980.17s$ of computational time which is always too expensive.

But the interesting part of the method is that it provides us a basis of the solution which is supposed to be able, like POD modes, to describe the long-term dynamics of the system. Hence, we have constructed the dynamical system associated to the modes provided by APR method. Then, the dynamical system is solved over times

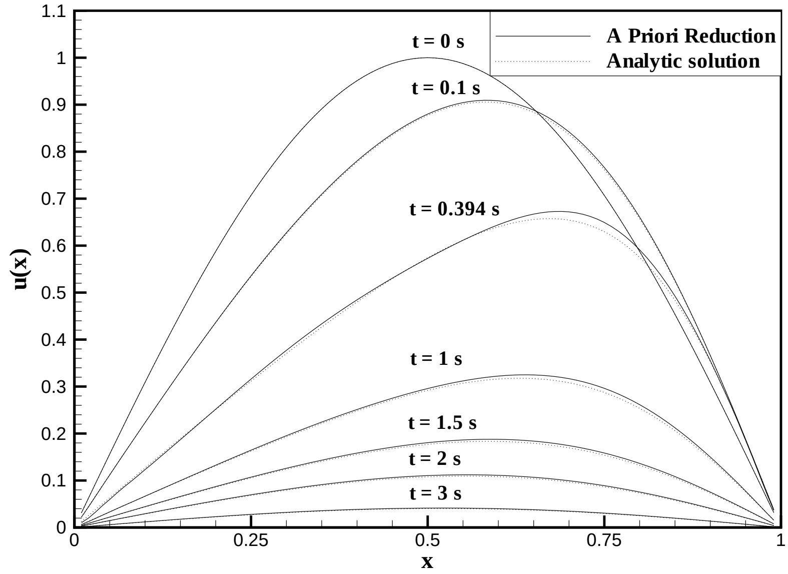

**Figure 3.** *Solution of Burgers' equation at different times for* $\nu = 0.1$, $N_x = 100$

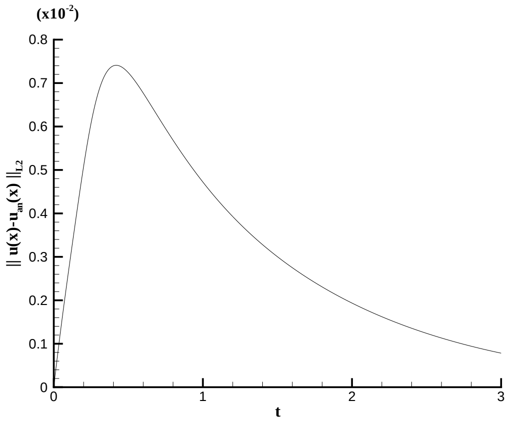

**Figure 4.** $L^2-$*norm of the difference between the APR solution and the analytic solution*

longer than the sample time interval, namely we solved the dynamical system over $3s$ whereas the sample interval was chosen equal to $0.1s$.

Because solving the dynamical system consists in solving a small set of partial differential equations, even with a very small time step, the solution is computed very fastly. Then, we have chosen $\Delta t = 10^{-4}$ and the results show that the APR basis is able to predict accurately the solution for $t \in [0s, 3s]$.

For this test case, we found a maxium error of around $0.23.10^{-3}$ at $t = 0.35s$ and the computational time necessary to get the solution for the whole time interval is $6s$, which is a considerable gain of time.

In order to check the quality of the APR modes, we have computed a POD basis over $T = 0.1s$ with datas from the analytic solution and we have constructed the coresponding dynamical system.

Then the dynamical system is again solved with the same parameters as before over $t = 3s$. Figure 5 compares the modes from the APR method and from the POD whereas the results in terms of $L^2$ error are reported in Figure 6.

We easily remark that the three first modes are quite identical, whereas the fourth and fifth – even if there are not identical – have the same structure. Without any mathematical proof yet, we can just say that the modes obtained with the APR method tend to be indentical to the POD modes. It is an interesting result because the POD basis is the most optimal basis, and with the APR method, one can also construct a very energetic basis without prerequisite knowledge of the flow.

We observe in Figure 6 that the POD associated with a dynamical system is the most accurate method we present. However, to compute the POD basis, we need a sample of solution over a small intervall whcih is very computationaly expensive. The APR method associated with the dynamical system is also a very fast and accurate solution for the numerical resolution of the Burgers' equation up to $3s$. Indeed, with a time step of $\Delta t = 10^{-4}$ we have computed the APR basis in $189s$ and obtained the solution over $t = 3s$ in $6s$ with the dynamical system constructed from the APR basis, that is the whole computation last around $200s$ whereas one needed $1230s$ with the NR method to solve just $0.1s$ of the flow.

As a last result, we present how the APR method is able to adapt its basis to a change in the control parameters. Indeed, we have changed the viscosity from $\nu = 0.1$ to $\nu = 0.05$ and computed the solution over $0.1s$ with our method taking the basis obtained for $\nu = 0.1$ as initial basis for the new computation.

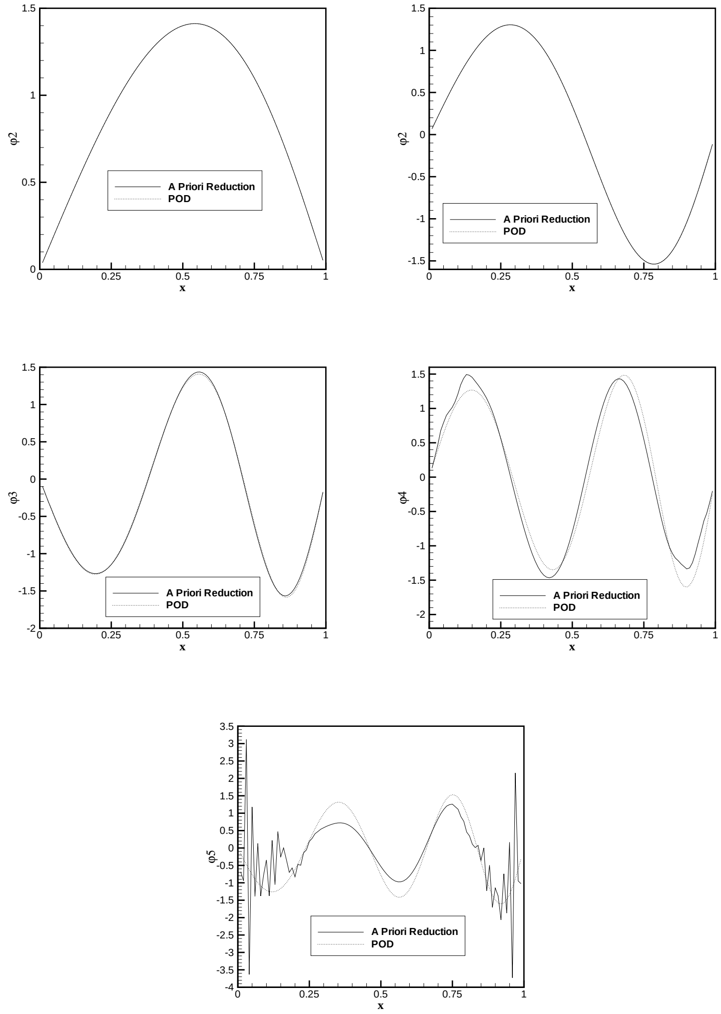

**Figure 5.** *Comparison between the modes obtained with the APR method and those from the POD for* $N_x = 100$

In just 6 iterations and for a computational time of around $15s$, the APR method is able to reach the convergence and the basis is modified in consequence. Furthermore, the maximum error obtained for $t = 0.1s$ is around $0.29 \cdot 10^{-2}$ which is very satisfaying.

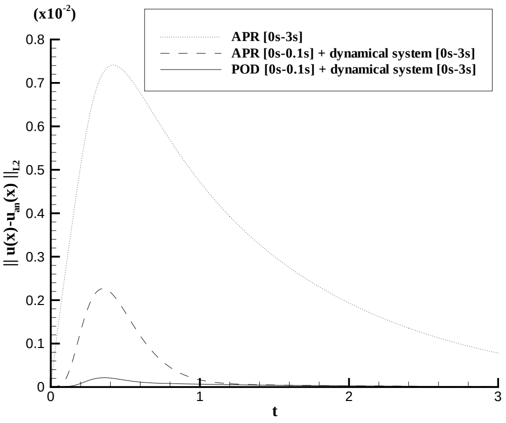

**Figure 6.** *$L^2$–norm of the difference between the APR solution and the analytic solution*

## 6. Conclusions and perspectives

In this article, we have presented an original numerical method for solving transfer equations which consists in a *a priori* construction of a basis of the solution.

The algorithm was tested on the Burgers' equation and it has proved its effenciency in terms of accuracy and computational time. Furthermore, the obtained basis was used to build a dynamical system for solving the equation for longer times. The basis has shown the ability to describe well the long-term behavior of the flow, which seems to be natural when comparing the APR modes with the POD ones. The modes are indeed very similar to each other, at least for the three first modes, and the APR basis is supposed to have more or less the same energetical properties as the POD which explains the efficiency of the basis.

Further tests are planed to test the method on other configurations and future works will be dedicated to adapt the algorithm to solve the two-dimension Navier-Sokes equations for standard benchmarks.

## 7. References

Allery C., Beghein C., Hamdouni A., " Applying proper orthogonal decomposition to the computation of particle dispersion in a two-dimensional ventilated cavity", *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, n° 8, p. 907-920, December, 2005.

Beran P. S., Lucia D. J., Pettit C. L., " Reduced Order Modeling of Limit-Cycle Oscillation for Aeroelastic Systems", *Journal of Fluids and Structures*, vol. 19, p. 575-590, 2004.

Brown P. N., Saad Y., “ Hybrid Krylov methods for nonlinear systems of equations”, *SIAM J. Sci. Stat. Comput.*, vol. 11, n° 3, p. 450-481, 1990.

Holmes P., Lumley J. L., Berkooz G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996.

Knoll D. A., Keyes D. E., “ Jacobian-free Newton-Krylov methods a survey of approaches and applications”, *Journal of Computational Physics*, vol. 193, p. 357-397, 2004.

Knoll D. A., Mousseau V. A., “ On Newton-Krylov Multigrid Methods for the Incompressible Navier-Stokes Equations”, *Journal of Computational Physics*, vol. 163, n° 1, p. 262-267, September, 2000.

Kunisch K., Volkwein S., “ Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition”, *Journal of Optimization Theory and Applications*, vol. 102, n° 2, p. 347-371, August, 1999.

Lumley J. L., “ The structure of inhomogeneous turbulent flows”, *Atmospheric Turbulence and Radio Wave Propagation*, vol. In A.M. Yaglom and Tararsky, p. 166-178, 1967.

Ryckelynck D., “ A priori hyperreduction method: an adaptive approach”, *Journal of Computational Physics*, vol. 202, p. 346-366, 2005.

Saad Y., *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.

Sirovich L., “ Turbulence and the dynamics of coherent structures, Part 1: Coherent structures. Part 2: Symmetries and transformations. Part 3: Dynamics and scaling”, *Quartely of Applied Mechanics*, vol. 45, p. 561-590, 1987.

Van der Vorst H. A., *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, 2003.