Mathematical Modelling and Numerical Analysis Will be set by the publisher
Modélisation Mathématique et Analyse Numérique

CERTIFIED REDUCED-BASIS SOLUTIONS OF VISCOUS BURGERS EQUATION PARAMETRIZED BY INITIAL AND BOUNDARY VALUES

Alexandre Janon[1], Maëlle Nodet[1] and Clémentine Prieur[1]

Abstract. We present a reduced basis offline/online procedure for viscous Burgers initial boundary value problem, enabling efficient approximate computation of the solutions of this equation for parametrized viscosity and initial and boundary value data. This procedure comes with a fast-evaluated rigorous error bound certifying the approximation procedure. Our numerical experiments show significant computational savings, as well as efficiency of the error bound.

1991 Mathematics Subject Classification. 35K20, 35K55, 65M15, 65M60.

The dates will be set by the publisher.

Introduction

This paper is set in the context of sensitivity analysis and uncertainty analysis in geophysical models. Such models typically involve a wide range of parameters, such as: source terms (climatic forcings, heat/wind/matter fluxes), boundary conditions (forcings, open boundaries), and the initial state of the system.
Their study generally leads to parametrized partial differential equations (PDEs). These equations often involve poorly-known parameters. Therefore, it is important to be able to measure the impact of a given parameter on the quality of the solution, and also to identify the "sensitive" parameters, that is, the parameters for which a small variation implies a large variation of the model solution. Due to their ability to perform global sensitivity analyses for nonlinear models, stochastic tools [13, 23] are rapidly expanding. These methods require "many queries," that is solving the parametrized PDE for a large (say, thousands) number of values of the parameters. When analytic solution to the PDE is not known (as it is often the case), one has to use a numerical method (such as finite difference or finite element) to compute an approximate value of the solution. Such methods lead to computer codes that could take a large time to produce an accurate-enough approximation — for a single value of the parameter. Having the "many-query" problem in mind, it is crucial to design a procedure that solves the equation for several values of the parameter faster than the naïve approach of calling the numerical code for each required instance of the parameter.
The reduced basis (RB) method is such a procedure; we split the overall computation into two successive parts: one part, the *offline* phase, makes use of the standard, computationally intensive numerical procedure used to solve the PDE to gather "knowledge" about solutions of the latter; and the other one, the *online* phase, where we rely on data collected during the offline phase to compute, for each desired instance of the parameter, a good approximation of the solution, for a per-instance cost that is orders of magnitude smaller than the cost

Keywords and phrases: Reduced-basis methods, parametrized PDEs, nonlinear PDEs, Burgers equation

[1] Joseph Fourier University, LJK/MOISE, BP 53, 38041 Grenoble Cedex, France ; e-mail: `alexandre.janon@imag.fr` & `maelle.nodet@inria.fr` & `clementine.prieur@imag.fr`.

of one run of the standard numerical code. The advantage is that, for a sufficiently large number of online evaluations, the fixed cost of the offline phase will be strongly dominated by the reduction in the marginal cost provided by the online procedure. This cost reduction is made possible by the fact that, in most cases, the desired solutions of the PDE, for all the considered values of the parameter, lie in some manifold of functions that is "close" to a low-dimensional linear subspace. One goal of the offline phase is to find such a suitable subspace, so that the online procedure can look for the solution of the PDE as an element of the subspace — so as to reduce the number of degrees of freedom and thus the computational cost. One interesting feature of the RB approach is that it comes with an *online error bound*, that is a (provably) certified, natural norm, fast-computed (i.e. almost of the same complexity of the online phase) upper bound of the distance between the solution provided by the online phase (called the *reduced*, or *online*) solution and the one given by the standard, expensive numerical procedure (called the *full* or *reference* solution). This "certified RB" framework has been developed for *affinely* parametrized second-order elliptic linear PDEs in [19]. It has been extended to nonlinear, non-affinely parametrized, parabolic PDEs, see e.g. [11], [10] and applied to problems such as steady incompressible Navier-Stokes [25]. Moreover theoretical work has been done to ensure *a priori* convergence of the RB procedure [4].

In this paper, we are interested in the RB reduction of the time-dependent viscous Burgers equation (which will serve as a "test case" for the "real" equations modelling geophysical fluids we are interested in). Papers [12] and [22] extend certified RB methodology to linear initial-boundary value problems. The case of homogeneous Dirichlet boundary conditions, zero initial value and fixed (*i.e.*, not parametrized) source term has been treated in [26], [18] ; in these works, the only parameter was the viscosity coefficient. Parametrization of initial and Dirichlet boundary conditions (treated using a conversion to a homogeneous Dirichlet problem) has been done in [16], for a general multidimensional quadratically nonlinear equation. Our methodology allows parametrization of the viscosity, the initial state, the source term and and of the boundary conditions. Compared to the works cited above, we use a weak (penalization) treatment of the Dirichlet boundary conditions. We will see that this weak treatment is more favorable in terms of both computation and storage complexities. Besides, our paper features a new error bound, which has shown to be, in all the testcases we performed, much more efficient than the existing bound. In particular, we will show that our bound exhibits improved sharpness for low viscosities, where [18] pointed out the moderate efficiency of the presented *a posteriori* bound.

This paper is organised as follows: in the first part, we introduce the viscous Burgers equation, and present a standard numerical procedure used to solve it; in the second part, we expose our offline/online reduction procedure; in the third part, we develop a certified online error bound; finally in the fourth part we validate and discuss our results based on numerical experiments.

1. Model

In this section, we describe the model we are interested in. Subsection 1.1 introduces the viscous Burgers equation, while Subsection 1.2 presents the "full" numerical procedure on which our reduction procedure, described in Section 2, relies on.

1.1. Equation

We are interested in u, function of space $x \in [0;1]$ and time $t \in [0;T]$ (for $T > 0$), with regularity: $u \in C^1\left([0,T], H^1(]0,1[)\right)$, satisfying the *viscous Burgers equation*:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}(u^2) - \nu \frac{\partial^2 u}{\partial x^2} = f \tag{1.1}$$

where $\nu \in \mathbf{R}_*^+$ ($\mathbf{R}$ denotes the set of real numbers, $\mathbf{R}_*^+$ the set of positive real numbers) is the *viscosity*, and $f \in C^0\left([0,T], L^2(]0,1[)\right)$ is the *source term*.

For u to be well-defined, we also prescribe initial values $u_0 \in H^1(]0,1[)$:

$$u(t=0,x) = u_0(x) \ \ \forall x \in [0;1] \tag{1.2}$$

and boundary values $b_0, b_1 \in C^0([0,T])$:

$$\begin{cases} u(t,x=0) = b_0(t) \\ u(t,x=1) = b_1(t) \end{cases} \qquad \forall t \in [0;T] \tag{1.3}$$

Where b_0, b_1 and u_0 are given functions, supposed to satisfy *compatibility conditions*:

$$u_0(0) = b_0(0) \quad \text{and} \quad u_0(1) = b_1(0) \tag{1.4}$$

This problem can be analyzed by means of the Cole-Hopf substitution (see [14] for instance), which turns (1.1) into the heat equation, leading to an integral representation of u.

1.2. **Numerical resolution**

We now describe the "expensive" numerical resolution of the problem described above that will serve as our reference for the reduction procedure described in the next section. We proceed in two steps: space discretization in paragraph 1.2.1 and time discretization in paragraph 1.2.2.

1.2.1. *Space discretization*

For space discretization, we use a $\mathbf{P}^1$ finite element procedure with weak (penalty) setting of the Dirichlet boundary conditions (1.3).
We first have to write the weak formulation of our PDE ; to do so, we multiply (1.1) by a function $v \in H^1(]0;1[)$ and integrate over $]0;1[$:

$$\int_0^1 \frac{\partial u}{\partial t}(t,x)v(x)\mathrm{d}x + \frac{1}{2}\int_0^1 \frac{\partial (u^2)}{\partial x}(t,x)v(x)\mathrm{d}x \\ - \nu \int_0^1 \frac{\partial^2 u}{\partial x^2}(t,x)v(x)\mathrm{d}x = \int_0^1 f(t,x)v(x)\mathrm{d}x \ \ \forall v \in H^1(]0;1[) \ \forall t \in [0;T] \tag{1.5}$$

Next, we integrate by parts the second and the third integral appearing in the left hand side of the previous equation:

$$\int_0^1 \frac{\partial (u^2)}{\partial x}(t,x)v(x)\mathrm{d}x = -\int_0^1 u^2(t,x)\frac{\partial v}{\partial x}(x)\mathrm{d}x + \left[u^2(t,\cdot)v\right]_0^1$$
$$\int_0^1 \frac{\partial^2 u}{\partial x^2}(t,x)v(x)\mathrm{d}x = -\int_0^1 \frac{\partial u}{\partial x}(t,x)\frac{\partial v}{\partial x}(x)\mathrm{d}x + \left[\frac{\partial u}{\partial x}(t,\cdot)v\right]_0^1$$

Inserting this into (1.5), we get:

$$\int_0^1 \frac{\partial u}{\partial t}(t,x)v(x)\mathrm{d}x - \frac{1}{2}\int_0^1 u^2(t,x)\frac{\partial v}{\partial x}(x)\mathrm{d}x + \nu\int_0^1 \frac{\partial u}{\partial x}(t,x)\frac{\partial v}{\partial x}(x)\mathrm{d}x \\ + \frac{1}{2}\left[u^2(t,\cdot)v\right]_0^1 - \nu\left[\frac{\partial u}{\partial x}(t,\cdot)v\right]_0^1 = \int_0^1 f(t,x)v(x)\mathrm{d}x \ \ \forall v \in H^1(]0;1[) \ \forall t \in [0;T] \tag{1.6}$$

To get rid of the two boundary terms arising in the integrations by parts, one usually restricts v to satisfy $v(0) = v(1) = 0$ so as to make the boundary terms disappear; the Dirichlet boundary conditions (1.3) are then incorporated "outside" of the weak formulation. However, the reduction framework we are to expose later requires the boundary conditions to be ensured by the weak formulation itself. The Dirichlet penalty method, presented in [2], is a way of doing so, at the expense of a slight approximation error. This method entails replacement of boundary conditions (1.3) with the following conditions:

$$\begin{cases} -\dfrac{1}{2}u^2(t,x=0)+\nu\dfrac{\partial u}{\partial x}(t,x=0)=P\left(u(t,x=0)-b_0(t)\right) \\ \dfrac{1}{2}u^2(t,x=1)-\nu\dfrac{\partial u}{\partial x}(t,x=1)=P\left(u(t,x=1)-b_1(t)\right) \end{cases} \quad \forall t\in[0;T] \tag{1.7}$$

with a fixed *penalization constant* $P>0$.
The intuitive idea underlying (1.7) is that it can clearly be rewritten as:

$$\begin{cases} u(t,x=0)=b_0(t)+\dfrac{1}{P}\left(-\dfrac{1}{2}u^2(t,x=0)+\nu\dfrac{\partial u}{\partial x}(t,x=0)\right) \\ u(t,x=1)=b_1(t)+\dfrac{1}{P}\left(\dfrac{1}{2}u^2(t,x=1)-\nu\dfrac{\partial u}{\partial x}(t,x=1)\right) \end{cases} \quad \forall t\in[0;T]$$

so that (1.3) is asymptotically verified for $P\to+\infty$. The reader can refer to [3] for rigorous *a priori* error estimates when using Dirichlet penalty in the linear elliptic case.
In practice, we can check if our approximation is sufficiently accurate by means of the following *a posteriori* procedure: we take for P some large value (typically $P=10^7$), we compute (numerically) an approximate solution u_d, using the procedure we are currently describing, and we check if an indicator of the amount of failure in verification of (1.3) is small enough; such an indicator can be, for instance:

$$\varepsilon_b=\sup_t\left[\max\left(|u_d(t,x=0)-b_0(t)|,|u_d(t,x=1)-b_1(t)|\right)\right] \tag{1.8}$$

where the supremum is taken over all discrete time steps. If this indicator is larger than a prescribed tolerance, then P has to be increased. Our numerical results in Section 4 will assert this condition. We can then invoke the well-posedness of the boundary/initial value problem (specifically, continuous dependence on the boundary values) to ensure that the solution of (1.1), (1.2) and (1.7) will be close to the solution of (1.1), (1.2) and (1.3). This reasoning is analogous to the one made when omitting the approximation made when replacing exact boundary values by their discretized counterparts.
Going back to our weak formulation, we multiply the first line of (1.7) by $v(0)$, the second one by $v(1)$ and add up these two equations. We get that:

$$\frac{1}{2}\left[u^2(t,\cdot)v\right]_0^1 \quad - \quad \nu\left[\frac{\partial u}{\partial x}(t,\cdot)v\right]_0^1 \quad = \quad P\left[\left(u(t,x=0)v(0)-b_0(t)v(0)\right)+\left(u(t,x=1)v(1)-b_1(t)v(1)\right)\right]$$

Putting it back into (1.6) and isolating the terms not involving u on the right-hand side yields the following weak formulation:

$$\int_0^1\frac{\partial u}{\partial t}v-\frac{1}{2}\int_0^1 u^2\frac{\partial v}{\partial x}+\nu\int_0^1\frac{\partial u}{\partial x}\frac{\partial v}{\partial x}+P\left(u(t,x=0)v(0)+u(t,x=1)v(1)\right)$$
$$=\int_0^1 f(t,\cdot)v+P\left(b_0(t)v(0)+b_1(t)v(1)\right)\quad\forall v\in H^1(]0;1[)\ \forall t\in[0;T] \tag{1.9}$$

that is:

$$\Big\langle \frac{\partial u}{\partial t}(t,\cdot), v \Big\rangle + c(u(t,\cdot), u(t,\cdot), v) + \nu a(u(t,\cdot), v) + B(u(t,\cdot), v) \\ = \ell(v,t) + b_0(t)\beta_0(v) + b_1(t)\beta_1(v) \;\; \forall v \in H^1(]0;1[), \forall t \in [0,T] \quad (1.10)$$

by introducing the following notations (for all $v, w, z \in H^1(]0;1[)$, $t \in [0;T]$):

$$\langle w, v \rangle = \int_0^1 wv \qquad a(w,v) = \int_0^1 \frac{\partial w}{\partial x}\frac{\partial v}{\partial x}$$

$$B(w,v) = P\left(w(0)v(0) + w(1)v(1)\right) \qquad \ell(v,t) = \int_0^1 f(t,\cdot)v$$

$$\beta_0(v) = Pv(0) \qquad \beta_1(v) = Pv(1)$$

and:

$$c(w,v,z) = -\frac{1}{2}\int_0^1 wv\frac{\partial z}{\partial x}$$

for every w, v and z in $H^1(]0;1[)$ for which $wv\dfrac{\partial z}{\partial x} \in L^1(]0;1[)$.

The weak formulation is then discretized with Lagrange $\mathbf{P}^1$ finite elements (see [21]) by choosing some integer $\mathcal{N}$ and considering a uniform subdivision of $[0;1]$ with $\mathcal{N}+1$ nodes: $\{x_i\}_{i=0,\dots,\mathcal{N}}$ and, for each $i = 0,\dots,\mathcal{N}$ we denote by ϕ_i the piecewise-affine "hat" function whose value is 1 on x_i and 0 on every other nodes.

We denote by X the linear subspace of $H^1(]0;1[)$ spanned by $\{\phi_i\}_{i=0,\dots,\mathcal{N}}$. We also set $\|\cdot\|$ to be the $L^2(]0;1[)$ norm.

Every $\psi \in X$ can be written as $\psi = \displaystyle\sum_{j=0}^{\mathcal{N}} \psi_j\phi_j$, and we have $\psi(x_i) = \psi_i$ for all $i = 0,\dots,\mathcal{N}$. This justifies that π defined below is a projection of $H^1(]0;1[)$ onto X:

$$\pi : \begin{cases} H^1(]0;1[) \to X \\ \qquad \psi \mapsto \displaystyle\sum_{j=0}^{\mathcal{N}} \psi(x_j)\phi_j \end{cases}$$

The space discretization of our problem is the following: for all $t \in [0;T]$, find $u(t,\cdot) \in X$ so that :

$$\begin{cases} u(t=0,\cdot) = \pi(u_0) \\ \Big\langle \dfrac{\partial u}{\partial t}(t,\cdot), v \Big\rangle + c(u(t,\cdot), u(t,\cdot), v) + \nu a(u(t,\cdot), v) + B(u(t,\cdot), v) \\ \qquad = \ell_\pi(v,t) + b_0(t)\beta_0(v) + b_1(t)\beta_1(v) \;\; \forall v \in X \end{cases} \quad (1.11)$$

where

$$\ell_\pi(v,t) = \int_0^1 \pi\left(f(t,\cdot)\right) v$$

Note that u now stands for a discrete solution, while it was used to designate an analytic solution before.

1.2.2. *Time discretization*

We now discretize (1.11) in time using the backward Euler scheme: we choose a timestep $\Delta t > 0$ and consider an uniform subdivision of $[0;T]$: $\{t_k = k\Delta t\}_{k=0,\dots,\mathcal{T}}$ where $\mathcal{T} = \dfrac{T}{\Delta t}$.

Our fully discrete problem is: for $k = 0, \ldots, \mathcal{T}$, find $u^k \in X$, approximation of $u(t_k, \cdot)$, satisfying:

$$u^0 = \pi(u_0) \tag{1.12a}$$

and:

$$\left\langle \frac{u^k - u^{k-1}}{\Delta t}, v \right\rangle + c(u^k, u^k, v) + \nu a(u^k, v) + B(u^k, v) \\ = \ell_\pi(v, t_k) + b_0(t_k)\beta_0(v) + b_1(t_k)\beta_1(v) \;\; \forall v \in X \; \forall k = 1, \ldots, \mathcal{T} \tag{1.12b}$$

We sequentially compute $\{u^k\}_{k=0,\ldots,\mathcal{T}}$ in the following way: u^0 comes straightforwardly from (1.12a), and for $k = 1, \ldots, \mathcal{T}$, u^k depends on u^{k-1} through (1.12b), which can be rewritten:

$$\frac{1}{\Delta t}\langle u^k, v\rangle + c(u^k, u^k, v) + \nu a(u^k, v) + B(u^k, v) \\ = \frac{1}{\Delta t}\langle u^{k-1}, v\rangle + \ell_\pi(v, t_k) + b_0(t_k)\beta_0(v) + b_1(t_k)\beta_1(v) \;\; \forall v \in X \tag{1.13}$$

We can now expand our unknown $u^k \in X$ on the $\{\phi_j\}_j$ basis: $u^k = \sum_{j=1}^{\mathcal{N}} u_j^k \phi_j$, and the vector $\left(u_j^k\right)_j$ becomes our new unknown.
Moreover, it is sufficient for (linear-in-v) relation (1.13) to be satisfied for all v in a basis of X, namely for $v = \phi_i$, $\forall i = 0, \ldots, \mathcal{N}$.
So (1.13) can be rewritten as a nonlinear (due to the nonlinearity in $c(u^k, u^k, v)$) system of $\mathcal{N}+1$ equations (one for each instantation $v = \phi_i$) involving $\left(u_j^k\right)_{j=0,\ldots,\mathcal{N}}$. This nonlinear system is solved using Newton iterations: starting with an initial guess $\overline{u^k}$, one looks for $\delta = \sum_{j=1}^{\mathcal{N}} \delta_j \phi_j$ so that $u^k = \overline{u^k} + \delta$ satisfies the linearization near $\delta = 0$ of (1.13) for $v = \phi_i$, $i = 0, \ldots, \mathcal{N}$, that is to say:

$$\frac{1}{\Delta t}\langle \overline{u^k} + \delta, \phi_i\rangle + c(\overline{u^k}, \overline{u^k}, \phi_i) + 2c(\overline{u^k}, \delta, \phi_i) + \nu a(\overline{u^k} + \delta, \phi_i) + B(\overline{u^k} + \delta, \phi_i) \\ = \frac{1}{\Delta t}\langle u^{k-1}, \phi_i\rangle + \ell_\pi(\phi_i, t_k) + b_0(t_k)\beta_0(\phi_i) + b_1(t_k)\beta_1(\phi_i) \;\; \forall i = 0, \ldots, \mathcal{N} \tag{1.14}$$

System (1.14) is a $(\mathcal{N}+1) \times (\mathcal{N}+1)$ linear system involving $(\delta_j)_{j=0,\mathcal{N}}$. Once solved for δ, one can test for convergence of the Newton iteration: if the norm of δ is smaller than a prescribed precision, then we stop here, produce u^k and get to the next time step $k+1$; otherwise we do one more Newton step, this time using u^k as initial guess $\overline{u^k}$.
We can note that the linear system to be solved at each Newton step is not symmetric but is sparse. Due to this sparsity, its solution (using an iterative method such as BICGSTAB or GMRES) takes about $O\left(\mathcal{N}^2\right)$ operations in the worst case. One can take advantage of the tridiagonal structure of the matrix (which is present in the one-dimensional case, since ϕ_i and ϕ_j have no common support if $|j - i| > 1$) and use Thomas' algorithm [24] to solve the system with $O(\mathcal{N})$ operations.

2. Reduction procedure

In this section, we show the offline/online procedure announced in the introduction to produce reduced basis solutions of the problem formed by (1.1), (1.2) and (1.3), based on the "full basis" numerical method presented above. We begin by a description of our parameters in Subsection 2.1. Our offline/online reduction procedure is described subsequently, in Subsection 2.2.

2.1. **Parameters**

We parametrize u_0, b_0, b_1 and f as:

$$b_0(t) = b_{0m} + \sum_{l=1}^{n(b_0)} A_l^{b_0} \Phi_l^{b_0}(t) \qquad b_1(t) = b_{1m} + \sum_{l=1}^{n(b_1)} A_l^{b_1} \Phi_l^{b_1}(t)$$

$$f(t,x) = f_m + \sum_{l=1}^{n_T(f)} \sum_{p=1}^{n_S(f)} A_{lp}^{f} \Phi_l^{fT}(t) \Phi_p^{fS}(x) \qquad u_0(x) = u_{0m} + \sum_{l=1}^{n(u_0)} A_l^{u_0} \Phi_l^{u_0}(x)$$

The functions involved in these linear combinations Φ^{b_0}, $\Phi_l^{b_1}$, Φ_l^{fT}, Φ_p^{fS} and $\Phi_l^{u_0}$ as well as the number of terms in the decompositions $n(b_0)$, $n(b_1)$, $n_T(f)$, $n_S(f)$ and $n(u_0)$ are fixed, while our parameters, namely: viscosity ν, coefficients b_{0m}, b_{1m}, f_m and u_{0m}, and $\left(A_l^{b_0}\right)_{l=1,\dots,n(b_0)}$, $\left(A_l^{b_1}\right)_{l=1,\dots,n(b_1)}$, $\left(A_{lp}^{f}\right)_{l=1,\dots,n_T(f);p=1,\dots,n_S(f)}$ and $(A_l^{u_0})_{l=1,\dots,n(u_0)}$ live in some Cartesian product of intervals $\mathcal{P}'$, subset of $\mathbf{R}^{1+4+n(b_0)+n(b_1)+n_T(f)n_S(f)+n(u_0)}$. Note that m stands for *mean* and is not a "numerical" index.
The compatibility condition (1.4) constraints b_{0m} and b_{1m} as functions of the other parameters:

$$b_{0m} = u_{0m} \quad \text{and} \quad b_{1m} = u_{0m} + \sum_{l=1}^{n(u_0)} A_l^{u_0} \Phi_l^{u_0}(x)$$

so that our "compliant" parameters actually belong to $\mathcal{P}$ defined by:

$$\mathcal{P} = \Big\{ \mu = \big(\nu, b_{0m}, A_1^{b_0}, \dots, A_{n(b_0)}^{b_0}, b_{1m}, A_1^{b_1}, \dots, A_{n(b_1)}^{b_0}, f_m, A_{11}^{f}, A_{12}^{f}, \dots, A_{1,n_S(f)}^{f}, \\ A_{2,1}^{f}, \dots, A_{2,n_S(f)}^{f}, \dots, A_{n_T(f),n_S(f)}^{f}, u_{0m}, A_1^{u_0}, \dots, A_{n(u_0)}^{u_0}\big) \in \mathcal{P}' \text{ satisfying (1.4)} \Big\} \tag{2.1}$$

2.2. **Offline/online procedure**

The key heuristic [19] for RB approximation of the (linear) parametrized variational problem is the following: given $\mu \in \mathcal{P}$,

$$\text{find } u(\mu) \in X \text{ so that } A(u(\mu), v; \mu) = L(v; \mu),\ \forall v \in X$$

is to choose a parameter-independent family $\mathcal{R}$ of linearly independent functions in X — with $\#\mathcal{R} \ll \dim X$, to achieve computational economy — and then, given an instance of the parameter, to search for the reduced solution:

$$\widetilde{u}(\mu) \in \operatorname{Span}\mathcal{R},\ \text{so that } A(\widetilde{u}(\mu), v; \mu) = L(v; \mu)\ \forall v \in \operatorname{Span}\mathcal{R}$$

Let us apply this idea on problem (1.12). We rely on the procedure described in [18], modified to allow parametrization of initial condition, boundary data and source term. To simplify notations, we do not explicitly write dependence of u and $\widetilde{u}$ on μ.
We suppose that a reduced basis $\mathcal{R} = \{\zeta_1, \dots, \zeta_N\}$ has been chosen (see Subsection 2.3 for one way to do so); we define $\widetilde{X} = \operatorname{Span}\mathcal{R}$ and we look for $\{\widetilde{u}^k\}_{k=0,\dots,\mathcal{T}} \subset \widetilde{X}$ satisfying:

$$\widetilde{u}^0 = \widetilde{\pi}\left(\pi(u_0)\right) \tag{2.2a}$$

and:

$$\left\langle \frac{\widetilde{u}^k - \widetilde{u}^{k-1}}{\Delta t}, v \right\rangle + c(\widetilde{u}^k, \widetilde{u}^k, v) + \nu a(\widetilde{u}^k, v) + B(\widetilde{u}^k, v) \\ = \ell_\pi(v, t_k) + b_0(t_k)\beta_0(v) + b_1(t_k)\beta_1(v)\ \ \forall v \in \widetilde{X}\ \forall k = 1, \dots, \mathcal{T} \tag{2.2b}$$

where $\widetilde{\pi}$ is the orthogonal projection from X onto $\widetilde{X}$, with respect to the standard L^2 inner product on X: $\langle w, v\rangle = \int_0^1 wv$.
The offline/online procedure for computation of $\widetilde{u}^0$ will come easily from our parametrization of u_0 in Subsection 2.1: since the constant function $\mathbf{1} = \sum_{j=0}^{\mathcal{N}} \phi_j$ belongs to X, we have:

$$\widetilde{u}^0 = u_{0m}\widetilde{\pi}(\mathbf{1}) + \sum_{l=1}^{n(u_0)} A_l^{u_0}\, \widetilde{\pi}\left(\pi\left(\Phi_l^{u_0}(\cdot)\right)\right) \tag{2.3}$$

We now discuss computation of $\widetilde{u}^k$ from $\widetilde{u}^{k-1}$ for $k = 1, \dots, \mathcal{T}$. We are willing to proceed with Newton steps as for the solution of (1.12b). We denote, for $k = 1, \dots, \mathcal{T}$, by

$$\widetilde{u}^{k-1} = \sum_{j=1}^{N} u_j^{k-1}\zeta_j \quad ; \quad \overline{\widetilde{u}^k} = \sum_{j=1}^{N} \overline{u_j^k}\zeta_j$$

respectively, the reduced solution at time t_{k-1}, and previous guess for the reduced solution at time t_k.
Our procedure relies on the following proposition:

Proposition 2.1. (1) *The Newton increment* $\delta = \sum_{j=1}^{N} \delta_j\zeta_j$ *satisfies the following equations:*

$$\begin{aligned}\sum_{j=1}^{N} \delta_j \left\{ \frac{\langle \zeta_j, \zeta_i\rangle}{\Delta t} + 2\sum_{j'=1}^{N} \overline{u_{j'}^k} c(\zeta_{j'}, \zeta_j, \zeta_i) + \nu a(\zeta_j, \zeta_i) + B(\zeta_j, \zeta_i) \right\} \\ = \sum_{j=1}^{N} u_j^{k-1} \frac{\langle \zeta_j, \zeta_i\rangle}{\Delta t} + \ell_\pi(\zeta_i, t_k) + b_0(t)\beta_0(\zeta_i) + b_1(t)\beta_1(\zeta_i) \\ - \sum_{j=1}^{N} \overline{u_j^k} \left(\frac{\langle \zeta_j, \zeta_i\rangle}{\Delta t} + \sum_{j'=1}^{N} \overline{u_{j'}^k} c(\zeta_{j'}, \zeta_j, \zeta_i) + \nu a(\zeta_j, \zeta_i) + B(\zeta_j, \zeta_i) \right) \quad \forall i = 1, \dots, N\end{aligned} \tag{2.4}$$

(2) *We have:*

$$\ell_\pi(\zeta_i, t_k) = f_m \int_0^1 \zeta_i + \sum_{l=1}^{n_T(f)} \sum_{p=1}^{n_S(f)} A_{lp}^f \Phi_l^{fT}(t_k) \int_0^1 \pi\left(\Phi_p^{fS}(\cdot)\right) \zeta_i \tag{2.5}$$

for all $i = 1, \dots, N$ *and* $k = 1, \dots, \mathcal{T}$.

Proof. (1) Equation (2.2b) for $\widetilde{u}^k = \overline{\widetilde{u}^k} + \delta$ linearized near $\delta = 0$, for $v = \zeta_i$, $\forall i = 1, \dots, N$ is:

$$\begin{aligned}\frac{1}{\Delta t}\langle \overline{\widetilde{u}^k} + \delta, \zeta_i\rangle + c(\overline{\widetilde{u}^k}, \overline{\widetilde{u}^k}, \zeta_i) + 2c(\overline{\widetilde{u}^k}, \delta, \zeta_i) + \nu a(\overline{\widetilde{u}^k} + \delta, \zeta_i) + B(\overline{\widetilde{u}^k} + \delta, \zeta_i) \\ = \frac{1}{\Delta t}\langle \widetilde{u}^{k-1}, \zeta_i\rangle + \ell_\pi(\zeta_i, t_k) + b_0(t)\beta_0(\zeta_i) + b_1(t)\beta_1(\zeta_i) \quad \forall i = 1, \dots, N\end{aligned}$$

Rewriting this equation using expansions of $\overline{\widetilde{u}^k}$ and δ in $\mathcal{R}$ and linearity of $\langle\cdot,\cdot\rangle$, c, a and B with respect to their first argument, and putting all $(\delta_j)_j$-dependent terms on the left-hand side give the announced equation.

(2) is a direct consequence of the parametrization of f given in Subsection 2.1.

□

The following Proposition 2.2 justifies the well-definedness of the Newton iteration, for an orthonormal reduced basis $\{\zeta_1, \dots, \zeta_N\}$. In practice, the Gram-Schmidt process can always be used to ensure this condition.

Proposition 2.2. *Suppose that $\{\zeta_1, \dots, \zeta_N\}$ is orthonormal with respect to $\langle\cdot,\cdot\rangle$. Then, for Δt small enough, i.e. $\Delta t < \Delta t^* (\nu, \{\zeta_i\}_{i=1,\dots,N})$, and initial guess $\overline{\widetilde{u}^k}$ sufficiently close to $\widetilde{u}^k$, the Newton iteration (2.4) is well defined and converges (quadratically) to $\widetilde{u}^k$.*

Proof. Iteration (2.4) is a Newton iteration for solving $F(x) = \alpha$, for appropriate α and F given by:

$$F(x) = \sum_{i=1}^{N} \left(\frac{\langle x, \zeta_i \rangle}{\Delta t} + c(x, x, \zeta_i) + \nu a(x, \zeta_i) + B(x, \zeta_i) \right) \zeta_i$$

We apply the result stated pp. 353–355 of [21] and pp. 362–367 of [6]. To do so, we have to check that, for Δt small enough:

(1) the differential of F at $\widetilde{u}^k$, denoted by $DF(\widetilde{u}^k)$, is invertible;
(2) $DF(\cdot)$ is Lipschitz-continuous *i.e.* there exist $L > 0$ so that

$$\forall x, x' \in X, \quad \forall v \in X, \quad \|DF(x) \cdot v - DF(x') \cdot v\| \le L \|x - x'\| \|v\|$$

It is easy to check that the matrix of $DF(\widetilde{u}^k)$ in the reduced basis $\{\zeta_1, \dots, \zeta_N\}$ is diagonally dominant, hence invertible, for:

$$\Delta t < \Delta t^* := \min_{i=1,\dots,N} \frac{1}{\sum_{j=1}^{N} |2c(\widetilde{u}^k, \zeta_j, \zeta_i) + \nu a(\zeta_j, \zeta_i) + B(\zeta_j, \zeta_i)|}$$

and (2) is a straightforward computation. □

We put our offline/online procedure in Algorithm 1. Let us make some remarks about the complexity of the above online algorithm, in contrast with the "full basis" one described in Section 1.2:

Remark 2.3. (1) The most computationally demanding step is (2) (c), since it involves resolution of a (nonsymmetric, dense) $N \times N$ linear system; the "full basis" counterpart solves $(\mathcal{N} + 1) \times (\mathcal{N} + 1)$ nonsymmetric tridiagonal system. Thus significant computational savings are expected for $N \ll \mathcal{N}$.

(2) Thanks to our parametrization of $f(t, \cdot)$, all integrals over $[0; 1]$ in equation (2.4) can be precomputed during the offline phase, yielding a $\mathcal{N}$-*independent* online phase. This means that one can in principle choose *arbitrary* high precision on the full model *without* impact on the marginal cost of evaluation of an online solution. This "$\mathcal{N}$-independence" property shall be required of every complexity of any online procedure. One should also note that our online procedure does not produce "nodal" values $\widetilde{u}^k(x_i)$, $i = 0, \dots, \mathcal{N}$ (as this would clearly violate the $\mathcal{N}$-independence), but rather the components of $\widetilde{u}^k$ in the reduced basis.

(3) Taken independently, the number of parameters $n(u_0)$, $n(b_0)$, $n(b_1)$, $n_S(f)$ and $n_T(f)$, as well as the number of timesteps $\mathcal{T}$ have a linear impact on the online complexity. Moreover, due to the double sum in (2.5), the online complexity is proportional to $n_S(f) n_T(f)$. We note that an advantage of treating the Dirichlet boundary condition weakly, as we do in this paper, is that $n_T(f)$ does not get increased by functions of $n(b_0)$ or $n(b_1)$. This is a clear advantage of our method: when boundary conditions are treated by returning to an homogeneous Dirichlet problem, as in [16], $n(b_0) + n(b_1)$ terms are added in the parametrization of f.

- *offline:*
  1. choose a parameter-, and time-independent reduced basis $\{\zeta_1, \dots, \zeta_N\}$ (see Section 2.3)
  2. compute and store the following parameter-independent functions of $\widetilde{X}$ and scalars, for all $i, j, j' = 1, \dots, N$, $l = 1, \dots, n(u_0)$, $p = 1, \dots, n_S(f)$:

$$\begin{array}{cc} \widetilde{\pi}(\mathbf{1}) & \widetilde{\pi}\left(\pi\left(\Phi_l^{u_0}(\cdot)\right)\right) \\ \langle \zeta_j, \zeta_i \rangle & a(\zeta_j, \zeta_i) \\ c(\zeta_{j'}, \zeta_j, \zeta_i) & B(\zeta_j, \zeta_i) \\ \beta_0(\zeta_i) & \beta_1(\zeta_i) \\ \int_0^1 \zeta_i & \int_0^1 \pi\left(\Phi_p^{fS}(\cdot)\right) \zeta_i \end{array}$$

- *online:*
  1. assemble $\widetilde{u}^0$ as the linear combination (2.3) ;
  2. for $k = 1, \dots, \mathcal{T}$:
     - (a) set up an initial guess $\overline{\widetilde{u}^k} = \sum_{j=1}^N \overline{u_j^k} \zeta_j$;
     - (b) compute and store $\ell_\pi(\zeta_i, t_k)$ by using (2.5) ;
     - (c) look for $\delta = \sum_{j=1}^N \delta_j \zeta_j$ by solving the linear system (2.4) ;
     - (d) set $\widetilde{u}^k \leftarrow \overline{\widetilde{u}^k} + \delta$;
     - (e) if $\|\delta\|$ is small enough:
       - (i) output $\widetilde{u}^k$
       - (ii) set $k = k + 1$
     - (f) else:
       - (i) update the guess : $\overline{\widetilde{u}^k} \leftarrow \widetilde{u}^k$, i.e. $\widetilde{u}_j^k \leftarrow u_j^k \; \forall j = 1, \dots, N$
       - (ii) go back to (c)

Algorithm 1: offline/online procedure

2.3. Choice of the reduced basis

In this subsection, we describe different ways of choosing a pertinent reduced basis $\{\zeta_1, \dots, \zeta_N\}$. These lead to three different bases fitting into the certified (that is to say, the three admit the same procedure for online error bound we describe in Section 3) reduced basis framework.
The first is based on proper orthogonal decomposition (POD), the second is based on a "greedy" selection algorithm. The third is an hybridation of POD and greedy. These methods are standard in the literature.

Notation

The two procedures described below involve computation of the reference solution for different instances of the parameter, so we should use special notations, local to this section, to emphasize the dependence of the reference solution on the parameters. We define a parametrized solution u by:

$$u : \begin{cases} \{1, \dots, \mathcal{T}\} \times \mathcal{P} \to X \\ \qquad (k, \mu) \mapsto u(k, \mu) = \widetilde{u}^k \text{ satisfying (1.12b) for } \mu \text{ as parameter} \end{cases}$$

2.3.1. *POD-driven procedure*

We denote by $\{u_j^k(\mu)\}_j$ the coordinates of $u(k,\mu)$ in the basis $\{\phi_0,\dots,\phi_{\mathcal{N}}\}$:

$$u(k,\mu)=\sum_{j=1}^{\mathcal{N}} u_j^k(\mu)\phi_j$$

In the POD-based procedure (see [5]) of the reduced basis choice, we choose a finite-sized parameter sample $\Xi\subset\mathcal{P}$, compute the reference solutions $u(k,\mu)$ for all $k=1,\dots,\mathcal{T}$ and all $\mu\in\Xi$, and form the *snapshots matrix* containing the components of these solutions in our basis $\{\phi_0,\dots,\phi_{\mathcal{N}}\}$:

$$M=\begin{pmatrix} u_0^0(\mu_1) & u_0^1(\mu_1) & \cdots & u_0^{\mathcal{T}}(\mu_1) & u_0^0(\mu_2) & \cdots & \cdots & u_0^{\mathcal{T}}(\mu_S)\\ u_1^0(\mu_1) & u_1^1(\mu_1) & \cdots & u_1^{\mathcal{T}}(\mu_1) & u_1^0(\mu_2) & \cdots & \cdots & u_1^{\mathcal{T}}(\mu_S)\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ u_{\mathcal{N}}^0(\mu_1) & u_{\mathcal{N}}^1(\mu_1) & \cdots & u_{\mathcal{N}}^{\mathcal{T}}(\mu_1) & u_{\mathcal{N}}^0(\mu_2) & \cdots & \cdots & u_{\mathcal{N}}^{\mathcal{T}}(\mu_S) \end{pmatrix}$$

where $\Xi=\{\mu_1,\dots,\mu_S\}$.
One can check that M has $\mathcal{N}+1$ rows and $S(\mathcal{T}+1)$ columns.
To finish, we choose the size $N<S(\mathcal{T}+1)$ of the desired reduced basis, we form the $S(\mathcal{T}+1)\times S(\mathcal{T}+1)$ non-negative symmetric matrix $M^T\Omega M$, where Ω is the matrix of our inner product $\langle\cdot,\cdot\rangle$, to find $z_1,\dots,z_N$ the N leading nonzero eigenvectors of this matrix (that is, the ones associated with the N largest eigenvalues, counting repeatedly possible nonsimple eigenvalues), and, for $i=1,\dots,N$, the coordinates of ζ_i with respect to $\{\phi_0,\dots,\phi_{\mathcal{N}}\}$ are given by:

$$\frac{1}{\|Mz_i\|}Mz_i \tag{2.6}$$

2.3.2. *"Local" greedy selection procedure*

The *greedy* basis selection algorithm (cf. [9,11]) is the following:

Parameter: N, the desired size of the reduced basis.

(1) Choose a finite-sized, random, large sample of parameters $\Xi\subset\mathcal{P}$.
(2) Choose $\mu_1\in\mathcal{P}$ and $k_1\in\{0,\dots,\mathcal{T}\}$ at random, and set

$$\zeta_1=\frac{u(\mu_1,k_1)}{\|u(\mu_1,k_1)\|}$$

(3) Repeat, for n from 2 to N:
   (a) Find:

$$(\mu_n,k_n)=\operatorname*{argmax}_{(\mu,k)\in\Xi\times\{0,\dots,\mathcal{T}\}}\varepsilon^*(\mu,k)$$

   where $\varepsilon^*(\mu,k)$ is a (fastly evaluated) estimator of the RB error $\|u(\mu,k)-\widetilde{u}(\mu,k)\|$, where $\widetilde{u}(\mu,k)$ stands for the RB approximation to $u(\mu,k)$ using $\{\zeta_1,\dots,\zeta_{n-1}\}$ as reduced basis (see below).
   (b) Compute $\zeta_n^*=u(\mu_n,k_n)$.
   (c) Using one step of the (stabilized) Gram-Schmidt algorithm, find $\zeta_n\in\operatorname{Span}\{\zeta_1,\dots,\zeta_{n-1},\zeta_n^*\}$ so that $\{\zeta_1,\dots,\zeta_{n-1},\zeta_n\}$ is an orthonormal family of $(X,\langle\cdot,\cdot\rangle)$.

Algorithm 2: Greedy basis selection

The "greedy" name for this algorithm comes from the fact that the algorithm chooses, at each step of the repeat loop, the "best possible" time and parameter tuple to the reduced basis, that is the one for which the RB approximation error is estimated to be the worst.

Let's now discuss the choice for the error indicator ε^*. A natural candidate would be the online error bound ε described in Section 3. However, as we shall see in the next section, the bound for timestep t_k is a compound of the "propagation" of the error made in the previous timesteps $t_0, t_1, \dots, t_{k-1}$ (the ε_{k-1} term) and the "local error" just introduced at the k-th time discretization. Hence, this error estimator ε_k has a natural tendency to grow (exponentially) with k. Thus using it as error indicator ε^* will favor times k_n near final time $\mathcal{T}$ to be chosen at step 3.(a) of the algorithm below. Such choices are suboptimal, because including them in the reduced basis will not fix this exponential growth problem which is inherent to our approximation. Instead we use, as in [11], a purely *local*-at-t_k error indicator, that is: the computable error bound described in Section 3 when taking $\varepsilon_{k-1} = 0$.
It has been noted in the literature that the greedy procedure can "stall" (i.e. select vectors for addition in the basis which have no effect in decreasing the error bounds). The author in [9] provides a "back-up procedure" in such a case. In our experiments, however, we have not noticed any "stalling" of the greedy procedure, maybe because of our sharper error bound.

2.3.3. *POD-Greedy procedure*

We can also make use of the POD-Greedy procedure [12] in Algorithm 3. This procedure aims to combine the advantages of the greedy and the POD based procedure; it has also been proposed so as to overcome the "stalling" of the greedy procedure. Again, an error indicator ε^* has to be chosen. We do as in [17], where the

Parameter: P_1.

(1) Choose a finite-sized, random, large sample of parameters $\Xi \subset \mathcal{P}$.
(2) Choose $\mu_1 \in \mathcal{P}$ at random, and choose as the current reduced basis $\mathcal{B}$ an orthonormalized basis of $\text{Span}\{u(\mu_1, t_0), u(\mu_1, t_1), \dots, u(\mu_1, \mathcal{T})\}$.
(3) Repeat, until the desired number of items in the basis is reached:
   (a) Find:
$$\mu^* = \operatorname*{argmax}_{\mu \in \Xi} \varepsilon^*(\mu)$$
   where $\varepsilon^*(\mu)$ is a (fastly evaluated) estimator of the RB error $\|u(\mu) - \widetilde{u}(\mu)\|$, where $\widetilde{u}(\mu)$ stands for the RB approximation to $u(\mu)$ using $\mathcal{B}$ as reduced basis.
   (b) Append to $\mathcal{B}$ the P_1 leading POD modes of $\{u^{\text{proj}}(\mu^*, t_0), u^{\text{proj}}(\mu^*, t_1), \dots, u^{\text{proj}}(\mu^*, \mathcal{T})\}$, where the proj superscript denotes projection on the orthogonal complement of $\text{Span}(\mathcal{B})$.
(4) Output $\mathcal{B} = \{\zeta_1, \dots, \zeta_{\#\mathcal{B}}\}$ as reduced basis.

Algorithm 3: POD-Greedy basis selection

authors take the online error bound at final time.

2.3.4. *Expansion of the basis by initial data modes*

In case they did not get chosen by the POD or greedy algorithm, a classical strategy is to initialize the basis selection algorithms by taking in the reduced basis the constant function $\mathbf{1}$, and the functions $\Phi_l^{u_0}(\cdot)$ for $l = 1, \dots, n(u_0)$. This may increase the size of the reduced basis (and thus online computation times) but ensures that $\widetilde{u}^0 = u_0$ (i.e. initial error is zero). Such an enrichment can possibly be a good move, as the error gets accumulated and amplified throughout the time iterations, zero initial error will certainly reduce the (estimated, as well as actual) RB approximation error.

3. Error bound

In this section, we derive a parameter and time dependent online error bound ε^k (for $k = 1, \dots, \mathcal{T}$) satisfying: $\left\|u_e^k - \widetilde{u}^k\right\| \le \varepsilon^k$, where $\|v\| = \left(\int_0^1 v^2\right)^{1/2}$, and u_e^k is our reference "truth" solution in X satisfying the fully discretized PDE with strong Dirichlet enforcement, that is:

$$\begin{cases} \dfrac{\langle u_e^k - u_e^{k-1}, v\rangle}{\Delta t} + c(u_e^k, u_e^k, v) + \nu a(u_e^k, v) = \ell_\pi(v, t_k) \quad \forall v \in X_0 \\ u_e^k(0) = b_0(t_k) \\ u_e^k(1) = b_1(t_k) \end{cases} \tag{3.1}$$

where X_0 is the "homogeneous" subspace of X:

$$X_0 = \{v \in X \text{ st. } v(0) = v(1) = 0\}.$$

Our error bound should be precise enough (i.e., not overestimating the actual error $\left\|u_e^k - \widetilde{u}^k\right\|$ too much, and approaching zero as N increases) and online-efficient (that is, admit an offline/online computation procedure with an $\mathcal{N}$-independent online complexity).
We notice that our error bound measures the error between the reduced and the reference solution; it does not reflect the discretization error made when replacing the actual analytical solution of the Burgers equation with its numerical approximation (3.1). This is consistent with the fact that RB methods relies strongly on the existence of a high-fidelity numerical approximation of the analytical solution by a discrete one, hence regarded as "truth".
Subsection 3.1 deals with the derivation of the error bound; this error involves quantities whose computation is detailed in Subsections 3.2 and 3.3.

3.1. **Error bound**

The sketch of the derivation of our error bound is the following: we first give a "theoretical" error bound in Theorem 3.1; we then replace the uncomputable quantities appearing in this bound by their computable surrogates in paragraph 3.1.2.

3.1.1. *Theoretical error bound*

Notation. We suppose that the convergence tests appearing in the Newton iterations performed in Section 1.2.2 and Section 2.2 are sufficiently demanding so as to neglect the errors due to the iterative solution of the nonlinear systems (1.13) and (2.2b).
We now set up some notations : first the error at time t_k:

$$e_k = \begin{cases} u_e^k - \widetilde{u}^k & \text{if } k > 0 \\ \pi(u_0) - \widetilde{\pi}\left(\pi(u_0)\right) & \text{if } k = 0 \end{cases}$$

the residual form r_k, for $v \in X_0$:

$$r_k(v) = \ell_\pi(v, t_k) - \frac{1}{\Delta t}\langle \widetilde{u}^k - \widetilde{u}^{k-1}, v\rangle - c(\widetilde{u}^k, \widetilde{u}^k, v) - \nu a(\widetilde{u}^k, v) \tag{3.2}$$

and the "X_0-norm" of the residual:

$$\|r_k\|_0 = \sup_{v \in X_0, \|v\|=1} r_k(v)$$

We introduce :

$$\psi_k(v, w) = 2c(\widetilde{u}^k, v, w) + \nu a(v, w)$$

and the so-called *stability constants*:

$$C_k = \inf_{v \in X_0, \|v\|=1} \psi_k(v, v) \tag{3.3}$$

To finish, we define:

$$\eta_k = |e_k(0)|\, \|\phi_0\| + |e_k(1)|\, \|\phi_\mathcal{N}\| \quad ; \quad \sigma_k = 2\, |C_k|\, \eta_k \quad ; \quad \mathcal{E} = \sup_{v \in X_0, \|v\|=1} v\left(\frac{1}{\mathcal{N}}\right)$$

($\mathcal{E}$ is finite because X_0 is finite dimensional), and, finally:

$$f_k = \mathcal{E}\big(\, |e_k(0)|\, |\psi_k(\phi_0, \phi_1)| + |e_k(1)|\, |\psi_k(\phi_\mathcal{N}, \phi_{\mathcal{N}-1})|\, \big)$$

$$\xi_k^\mathcal{A} = \frac{\mathcal{E}^2(|e_k(0)| + |e_k(1)|)}{3} \quad ; \quad \xi_k^\mathcal{B} = \frac{5}{3}\mathcal{E}\left(e_k(0)^2 + e_k(1)^2\right) \quad ; \quad \xi_k^\gamma = \frac{|e_k(0)|^3 + |e_k(1)|^3}{3}$$

We will also make use of the standard notation:

$$[x]_- = \max(-x, 0) \quad \forall x \in \mathbf{R}$$

The theoretical foundation for our error bound is the following theorem. As the technique developed in [18] did not fit our problem (because of the weak Dirichlet treatment, whose advantage has been shown in Remark 2.3 point (3)), we developed a new strategy for obtaining this error bound.

Theorem 3.1. *If:*

$$\frac{1}{\Delta t} + C_k - \xi_k^\mathcal{A} > 0 \ \forall k = 1, \dots, \mathcal{T} \tag{3.4}$$

then the norm of the error $\|e_k\|$ satisfies:

$$\|e_k\| \le \begin{cases} \dfrac{\mathcal{B}_k + \sqrt{\mathcal{D}_k}}{2\mathcal{A}_k} & \text{if } \mathcal{D}_k \ge 0 \\ \dfrac{\mathcal{B}_k}{\mathcal{A}_k} & \text{if } \mathcal{D}_k < 0 \end{cases}$$

with:

$$\mathcal{A}_k = \frac{1}{\Delta t} + C_k - \xi_k^\mathcal{A} \quad ; \quad \mathcal{B}_k = \frac{2\eta_k + \|e_{k-1}\| + \mathcal{E}\langle\phi_0, \phi_1\rangle(|e_k(0)| + |e_k(1)|)}{\Delta t} + \sigma_k + f_k + \|r_k\|_0 + \xi_k^\mathcal{B}$$

$$\gamma_k = \frac{\eta_k\, \|e_{k-1}\| + \mathcal{E}\eta_k\langle\phi_0, \phi_1\rangle(|e_k(0)| + |e_k(1)|)}{\Delta t} + \eta_k f_k + [C_k]_-\, \eta_k^2 + \frac{1}{6}\left|e_k(1)^3 - e_k(0)^3\right| + \xi_k^\gamma + \|r_k\|_0\, \eta_k$$

and:

$$\mathcal{D}_k = (\mathcal{B}_k)^2 + 4\mathcal{A}_k\gamma_k.$$

The proof of Theorem 3.1 is presented in the appendix.

3.1.2. *Computable error bound.*

We now find an efficiently computable (that is, with an offline/online decomposition, with a complexity of the online part independent of $\mathcal{N}$) error bound ε_k derived from the one described above; to do so we discuss each of the ingredients appearing in its expression.

- Computation of the norm of the initial error $\|e_0\|$ is addressed in the next Section 3.2; the one of $\|r_k\|_0$ is in Section 3.3.
- We have, for $w \in \{0, 1\}$:

$$e_k(w) = u_e^k(w) - \widetilde{u}^k(w) = b_w(t_k) - \widetilde{u}^k(w)$$

  so that $e_k(0)$ and $e_k(1)$ can be computed during the online phase.

- Similarly, the scalars $\|\phi_0\|$, $\psi_k(\phi_0,\phi_1)$ and $\psi_k(\phi_{\mathcal{N},\mathcal{N}-1})$ can straightforwardly be computed online.
- The "continuity constant" $\mathcal{E}$ can be computed offline and stored by solving the optimization problem defining it. Thus η_k, f_k, $\xi_k^{\mathcal{A}}$, $\xi_k^{\mathcal{B}}$ and ξ_k^{γ} can be computed online.
- The exact value of C_k could be found by solving a generalized eigenvalue problem on X: C_k is the smallest $\lambda \in \mathbf{R}$ so that there exists $z \in X_0$, $\|z\| = 1$ satisfying:

$$\psi_k^{Sym}(z,v) = \lambda\langle z, v\rangle \ \ \forall v \in X_0$$

  with ψ_k^{Sym} the symmetric bilinear form defined by:

$$\psi_k^{Sym}(w,v) = \nu a(w,v) + c(\widetilde{u}^k, w, v) + c(\widetilde{u}^k, v, w) \ \ \forall w, v \in X_0$$

  The cost of doing so is prohibitive as it is an increasing function of $\dim X = \mathcal{N}+1$. Instead, we will see in Section 3.4 how to compute lower and upper bounds C_k^{inf} and C_k^{sup}: $C_k^{inf} \leq C_k \leq C_k^{sup}$.
- We can then compute the following lower and upper bounds for $\mathcal{A}_k$:

$$\mathcal{A}_k^{inf} = \frac{1}{\Delta t} + C_k^{inf} - \xi_k^{\mathcal{A}} \qquad ; \qquad \mathcal{A}_k^{sup} = \frac{1}{\Delta t} + C_k^{sup} - \xi_k^{\mathcal{A}}$$

  and the hypothesis (3.4) is ensured by checking that $\mathcal{A}_k^{inf} > 0$.
- We can also compute an upper bound of σ_k:

$$\sigma_k^{sup} = 2\eta_k \max\left(|C_k^{sup}|, \left|C_k^{inf}\right|\right)$$

- To compute an upper bound of $\mathcal{B}_k$, we need to replace the preceding error norm $\|e_{k-1}\|$ which is (except for $k=1$) not exactly computable, with the online upper bound $\varepsilon_{k-1} \geq \|e_{k-1}\|$ at the preceding time step:

$$\mathcal{B}_k^{sup} = \frac{2\eta_k + \varepsilon_{k-1} + \mathcal{E}\langle\phi_0,\phi_1\rangle(|e_k(0)| + |e_k(1)|)}{\Delta t} + \sigma_k^{sup} + f_k + \|r_k\|_0 + \xi_k^{\mathcal{B}}$$

- And γ_k gets replaced by its upper bound γ_k^{sup}:

$$\gamma_k^{sup} = \frac{\eta_k\varepsilon_{k-1} + \mathcal{E}\eta_k\langle\phi_0,\phi_1\rangle(|e_k(0)| + |e_k(1)|)}{\Delta t} + \eta_k f_k + \left[C_k^{inf}\right]_- \eta_k^2 + \frac{1}{6}\left|e_k(1)^3 - e_k(0)^3\right| + \xi_k^{\gamma} + \|r_k\|_0\,\eta_k$$

- We finally compute an upper bound for $\mathcal{D}_k$:

$$\mathcal{D}_k^{sup} = \begin{cases} (\mathcal{B}_k^{sup})^2 + 4\mathcal{A}_k^{sup}\gamma_k^{sup} & \text{if } \gamma_k^{sup} \geq 0 \\ (\mathcal{B}_k^{sup})^2 + 4\mathcal{A}_k^{inf}\gamma_k^{sup} & \text{if } \gamma_k^{sup} < 0. \end{cases}$$

  and our "computable" error bound is then:

$$\begin{cases} \dfrac{\mathcal{B}_k^{sup} + \sqrt{\mathcal{D}_k^{sup}}}{2\mathcal{A}_k^{inf}} & \text{if } \mathcal{D}_k^{sup} \geq 0 \\ \dfrac{\mathcal{B}_k^{sup}}{\mathcal{A}_k^{inf}} & \text{if } \mathcal{D}_k^{sup} < 0. \end{cases}$$

The remainder of the section consists in the description of computation of the four left-out quantities $\|e_0\|$, $\|r_k\|_0$, C_k^{sup} and C_k^{inf}.

3.2. **Initial error**

The present subsection deals with efficient computation of the $\|e_0\|$ term in the computable error bound described in paragraph 3.1.2. We denote by $\mathbf{H}$ the Gram matrix of the family:

$$\left\{\mathbf{1}-\widetilde{\pi}(\mathbf{1}), \pi\left(\Phi_1^{u_0}(\cdot)\right)-\widetilde{\pi}\left(\pi\left(\Phi_1^{u_0}(\cdot)\right)\right), \ldots, \pi\left(\Phi_{n(u_0)}^{u_0}(\cdot)\right)-\widetilde{\pi}\left(\pi\left(\Phi_{n(u_0)}^{u_0}(\cdot)\right)\right)\right\}$$

that is, $\mathbf{H}$ is the $(1+n(u_0)) \times (1+n(u_0))$ symmetric matrix of all the inner products between two any of the above vectors and by $\mathbf{e_0}$ the vector containing the components of e_0 with respect to the family above, *i.e.*:

$$\mathbf{e_0}=\left(u_{0m}, A_1^{u_0}, \ldots, A_{n(u_0)}^{u_0}\right)^T$$

We have:

Lemma 3.2. *The norm of the initial error* $\|e_0\|$ *is given by:*

$$||e_0|| = \sqrt{\mathbf{e_0}^T\mathbf{H}\mathbf{e_0}} \tag{3.5}$$

Proof. Parametrization 2.1 gives:

$$e_0=\pi(u_0)-\widetilde{\pi}\left(\pi(u_0)\right)=u_{0m}\left(\mathbf{1}-\widetilde{\pi}(\mathbf{1})\right)+\sum_{l=1}^{n(u_0)} A_l^{u_0}\left(\pi\left(\Phi_l^{u_0}(\cdot)\right)-\widetilde{\pi}\left(\pi\left(\Phi_l^{u_0}(\cdot)\right)\right)\right)$$

and the result follows from the expansion of $\|e_0\|^2$ when e_0 is replaced by the expression above. □

This formula allows us to compute the time and parameter-independent Gram matrix $\mathbf{H}$ during the offline phase, and, during the online phase, to assemble the $(1+n(u_0))$-vector $\mathbf{e_0}$ and to perform (3.5) to get $||e_0||$ with an online cost dependent only of $n(u_0)$.

3.3. **Norm of the residual**

We now present the computation of the $\|r_k\|_0$ term in the computable error bound of Section 3.1.2.

- Let the expansions of the reduced solutions with respect to the reduced basis be:

$$\widetilde{u}^p=\sum_{j=1}^N u_j^p \zeta_j \quad \text{for } p \in \{k-1, k\}$$

- Let $\mathbf{G}$ be the $(1+n_S(f)+2N+N^2) \times (1+n_S(f)+2N+N^2)$-sized Gram matrix of

$$\{\Gamma^{int}, \Gamma_1^{fS}, \ldots, \Gamma_{n_S(f)}^{fS}, \Gamma_1^{\langle\rangle}, \ldots, \Gamma_N^{\langle\rangle}, \Gamma_{1,1}^c, \Gamma_{1,2}^c, \ldots, \Gamma_{1,N}^c, \Gamma_{2,1}^c, \ldots, \Gamma_{2,N}^c, \ldots, \Gamma_{N,N}^c, \Gamma_1^a, \ldots, \Gamma_N^a\}$$

where $\Gamma^{int}, \Gamma_p^{fS}, \Gamma_j^{\langle\rangle}, \Gamma_{j,j'}^c, \Gamma_j^a \in X_0$ $(p = 1, \dots, n_S(f); j, j' = 1, \dots, N)$ satisfy:

$$\begin{aligned}
\langle \Gamma^{int}, v\rangle &= \int_0^1 v && \forall v \in X_0 \\
\langle \Gamma_p^{fS}, v\rangle &= \int_0^1 \pi\left(\Phi_p^{fS}(\cdot)\right) v && \forall v \in X_0 \\
\langle \Gamma_j^{\langle\rangle}, v\rangle &= \langle \zeta_j, v\rangle && \forall v \in X_0 \\
\langle \Gamma_{j,j'}^c, v\rangle &= c(\zeta_j, \zeta_{j'}, v) && \forall v \in X_0 \\
\langle \Gamma_j^a, v\rangle &= a(\zeta_j, v) && \forall v \in X_0
\end{aligned}$$

(Those Γ's exist by virtue of the Riesz representation theorem).

- Let $\boldsymbol{\rho}_k$ be the following vector:

$$\boldsymbol{\rho}_k = \Big(f_m, \sum_{l=1}^{n_T(f)} A_{l,j}^f \Phi_l^{fT}(t_k)\ (j = 1, \dots, n_S(f)), -\frac{1}{\Delta t}(u_j^k - u_j^{k-1})\ (j = 1, \dots, N), \\ u_j^k u_l^k\ (j, l = 1, \dots, N), \nu u_j^k\ (j = 1, \dots, N)\Big)^T$$

Lemma 3.3. *We have:*

$$\|r_k\|_0 = \sqrt{\boldsymbol{\rho}_k^T \mathbf{G} \boldsymbol{\rho}_k} \tag{3.6}$$

Proof. From the Riesz representation theorem, there exists a unique $\rho_k \in X_0$ so that

$$\langle \rho_k, v\rangle = r_k(v) \ \ \forall v \in X_0 \tag{3.7}$$

and we have: $\|r_k\|_0 = \|\rho_k\|$.
From (3.7) and the definition of r_k (3.2), we have that ρ_k is defined uniquely by:

$$\langle \rho_k, v\rangle = \ell_\pi(v, t_k) - \frac{1}{\Delta t}\langle \widetilde{u}^k - \widetilde{u}^{k-1}, v\rangle - c(\widetilde{u}^k, \widetilde{u}^k, v) - \nu a(\widetilde{u}^k, v) \forall v \in X_0$$

because $\beta_0(v) = \beta_1(v) = B(\cdot, v) = 0$ for all $v \in X_0$.
Using parametrization (2.5) of $\ell_\pi(\cdot, t_k)$, we get that (3.7) is equivalent to:

$$\langle \rho_k, v\rangle = f_m \int_0^1 v + \sum_{l=1}^{n_T(f)} \sum_{p=1}^{n_S(f)} A_{lp}^f \Phi_l^{fT}(t_k) \int_0^1 \pi\left(\Phi_p^{fS}(\cdot)\right) v - \frac{1}{\Delta t}\sum_{j=1}^N (u_j^k - u_j^{k-1})\langle \zeta_j, v\rangle \\ - \sum_{j=1}^N u_j^k \left(\sum_{j'=1}^N u_{j'}^k c(\zeta_j, \zeta_{j'}, v) + \nu a(\zeta_j, v)\right) \ \forall v \in X_0 \tag{3.8}$$

By the superposition principle, ρ_k can be written as the linear combination:

$$\rho_k \ = \ f_m \Gamma^{int} + \sum_{p=1}^{n_S(f)} \left(\sum_{l=1}^{n_T(f)} A_{lp}^f \Phi_l^{fT}(t_k)\right) \Gamma_p^{fS} - \frac{1}{\Delta t}\sum_{j=1}^N (u_j^k - u_j^{k-1}) \Gamma_j^{\langle\rangle} - \sum_{j=1}^N \sum_{j'=1}^N u_j^k u_{j'}^k \Gamma_{j,j'}^c - \sum_{j=1}^N \nu u_j^k \Gamma_j^a$$

Thus $\boldsymbol{\rho}_k$ contains the components of ρ_k with respect to the family whose $\mathbf{G}$ is the Gram matrix, and so:

$$\|r_k\|_0 = \|\rho_k\| = \sqrt{\boldsymbol{\rho}_k^T \mathbf{G} \boldsymbol{\rho}_k}. \tag{3.9}$$

$\square$

The offline/online decomposition for computation of $\|r_k\|_0$ is as follows: in the offline phase, we compute the Γ's vectors, and compute and store their Gram matrix $\mathbf{G}$. In the online phase, we compute $\boldsymbol{\rho}_k$ and compute $\|r_k\|_0$ using (3.6). Note that one can reduce offline and online computational burden, as well as storage requirement, by noticing that $\Gamma^c_{j,j'} = \Gamma^c_{j',j}$ for all j, j'.
The cost of computation (and storage) of ρ_k and $\|\rho_k\|$ asymptotically dominates the cost of the online phase. This cost is in $O\left(\left(n_S(f)n_T(f) + N^2\right)^2\right)$. Again, we see that our weak Dirichlet treatment, as $n_T(f)$ remains independent of $n(b_0)$ and $n(b_1)$, allows for a better complexity of the online phase.

3.4. Lower and upper bounds on stability constant

To find lower and upper bounds on C_k efficiently in order to use them in our computable error bound of Section 3.1.2, we turn to the successive constraints method (SCM) [15,18]. Here we present the application to our case for the sake of self-containedness. Our difference is the use of the metric (3.10) during the constraint-selection phase.

Notation. As in Section 2.3, we will need to handle reduced solutions of several values of the parameter tuple $\mu \in \mathcal{P}$ (see (2.1)), for different timesteps $k = 1, \ldots, \mathcal{T}$. We thus define an application that is the "reduced" counterpart of u defined in Section 2.3:

$$\widetilde{u} : \begin{cases} \{1, \ldots, \mathcal{T}\} \times \mathcal{P} \to \widetilde{X}_0 \\ \qquad (k, \mu) \mapsto \widetilde{u}(k, \mu) = \widetilde{u}^k \text{ satisfying (2.2b) for } \mu \text{ as parameter} \end{cases}$$

We make C_k depend explicitly on μ by defining:

$$C_k(\mu) = \inf_{v \in X_0, \|v\|=1} \left[2c(\widetilde{u}(k,\mu), v, v) + \nu(\mu) a(v, v)\right]$$

SCM lower bound. We now proceed to the derivation of the SCM lower bound of $C_k(\mu)$. We use the RB expansion: $\widetilde{u}(k, \mu) = \sum_{j=1}^{N} u_j^k(\mu)\zeta_j$ to rewrite $C_k(\mu)$ as:

$$\begin{aligned} C_k(\mu) = \inf_{v \in X_0, \|v\|=1} \left[\sum_{j=1}^{N} 2u_j^k(\mu) c(\zeta_j, v, v) + \nu a(v, v)\right] &= \inf_{y=(y_1,\ldots,y_{N+1}) \in \mathcal{Y}} \left[\sum_{j=1}^{N} 2u_j^k(\mu) y_j + \nu y_{N+1}\right] \\ &= \inf_{y \in \mathcal{Y}} \mathcal{J}(\mu, k, y) \end{aligned}$$

where:

$$\mathcal{Y} = \left\{y = (y_1, \ldots, y_{N+1}) \in \mathbf{R}^{N+1} | \exists v \in X_0, \|v\| = 1 \text{ s.t. } y_j = c(\zeta_j, v, v) \forall j = 1, \ldots, N, y_{N+1} = a(v, v)\right\}$$

and:

$$\mathcal{J}(\mu, k, y) = 2\sum_{j=1}^{N} u_j^k(\mu) y_j + \nu y_{N+1}$$

We define, for a given "constraint subset" $\mathcal{C} \subset \{1, \ldots, \mathcal{T}\} \times P$ that will be chosen later:

- $\widetilde{\mathcal{Y}} = \left\{ (y_1, \ldots, y_{N+1}) \in \prod_{i=1}^{N+1} \left[\sigma_i^{min}; \sigma_i^{max}\right] | \mathcal{J}(\mu', k', y) \geq C_{k'}(\mu'),\ \forall (\mu', k') \in \mathcal{S}(\mu, k) \right\}$ with $\mathcal{S}(\mu, k)$ standing for the set of the M points in $\mathcal{C}$ that are closest to (μ, k) with respect to this metric:

$$d\big((\mu, k), (\mu', k')\big) = \sum_{i=1}^{\dim \mathcal{P}} \left(\frac{\mu^i - {\mu'}^i}{\mu^i_{min} - \mu^i_{max}} \right)^2 + \left(\frac{k - k'}{\mathcal{T}} \right)^2 \tag{3.10}$$

where $\left(\mu^1, \ldots, \mu^{\dim \mathcal{P}}\right)$ are the coordinates of $\mu \in \mathcal{P}$, and:

$$\mu^i_{min} = \min_{\mu \in \mathcal{P}} \mu^i, \qquad \mu^i_{max} = \max_{\mu \in \mathcal{P}} \mu^i$$

for $i = 1, \ldots, \dim \mathcal{P}$ (here $\dim \mathcal{P} = 1 + 2 + n(b_0) + n(b_1) + n_T(f) n_S(f) + n(u_0)$ is the number of parameters).

The metric defined in (3.10) quantifies proximity of two parameter-time tuples, with appropriate weighting so as to account for scaling differences between the parameters.
- We further define:

$$\sigma_i^{min} = \inf_{v \in X_0, \|v\|=1} c(\zeta_i, v, v), \quad \forall i = 1, \ldots, N \qquad \sigma_{N+1}^{min} = \inf_{v \in X_0, \|v\|=1} a(v, v)$$
$$\sigma_i^{max} = \sup_{v \in X_0, \|v\|=1} c(\zeta_i, v, v), \quad \forall i = 1, \ldots, N \qquad \sigma_{N+1}^{max} = \sup_{v \in X_0, \|v\|=1} a(v, v)$$

The SCM lower bound is then given by the following lemma:

Lemma 3.4 (Proposition 1 in [15]). *For every $\mathcal{C} \subset \{1, \ldots, \mathcal{T}\} \times P$ and $M \in N$, and every $k = 1, \ldots, \mathcal{T}$, a lower bound for $C_k(\mu)$ is given by:*

$$C_k^{inf}(\mu) = \inf_{y \in \widetilde{\mathcal{Y}}} [\mathcal{J}(\mu, k, y)] \tag{3.11}$$

An algorithm for choosing a constraint subset $\mathcal{C}$ will be given after the description of the SCM upper bound and the SCM offline/online procedure.

SCM upper bound. We define:

$$\widetilde{\mathcal{Y}}^{up} = \{y^*(k_i, \mu_i)\ ;\ i = 1, \ldots, I\ ;\ (k_i, \mu_i) \in \mathcal{C}\}$$

where:

$$\mathcal{C} = \{(k_1, \mu_1), (k_2, \mu_2), \ldots, (k_I, \mu_I)\}$$

and:

$$y^*(k_i, \mu_i) = \operatorname*{arginf}_{y \in \mathcal{Y}} [\mathcal{J}(\mu_i, k_i, y)] \quad (i = 1, \ldots, I)$$

Lemma 3.5 (Proposition 1 in [15]). *For every $k = 1, \ldots, \mathcal{T}$, an upper bound for C_k is given by:*

$$C_k^{sup}(\mu) = \inf_{y \in \widetilde{\mathcal{Y}}^{up}} \mathcal{J}(\mu, k, y) \tag{3.12}$$

SCM offline/online procedure. Relying on Lemmas 3.4 and 3.5, our offline/online procedure for computing C_k^{inf} and C_k^{sup} reads: In the lower bound online phase, the optimization problem required to solve is a *linear programming* problem (LP) with $N+1$ variables and $N+1+M$ constraints (M one-sided inequalities and $N+1$ two-sided). There are algorithms, such as the simplex algorithm (see [20] for instance), which solve such optimization problems under (on average) polynomial complexity with respect to the number of variables and number of constraints, even if they can be exponential in the worst cases. What matters here is this complexity

- offline:
  (1) choose M and constraint set $\mathcal{C}$ (see next paragraph) ;
  (2) compute and store σ_i^{min} and σ_i^{max} $(i = 1, \ldots, N+1)$ by solving a generalized eigenproblem on X_0 ;
  (3) for each $(k', \mu') \in \mathcal{C}$:
    (a) solve a generalized eigenproblem on X_0 to find $C_{k'}(\mu')$ (and store it);
    (b) let $w \in X_0$ be a unit eigenvector of the above eigenproblem; compute and store (in $\widetilde{\mathcal{Y}}^{up}$) $y^*(k', \mu')$ using:

$$y^*(k', \mu')_j = c(\zeta_j, w, w) \quad (j = 1, \ldots, N)$$
$$y^*(k', \mu')_{N+1} = a(w, w)$$

- online:
  – for the lower bound:
    (1) assemble and solve optimization problem (3.11) ;
  – for the upper bound: test one-by-one each element of $\widetilde{\mathcal{Y}}^{up}$ to solve (3.12).

Algorithm 4: SCM offline/online

is independent of $\mathcal{N}$. The upper bound online phase has a complexity depending linearly on the cardinality of the reasonably-sized $\mathcal{C}$ and on N.
"Greedy" constraint set selection. To choose $\mathcal{C}$ in Algorithm 4, step 1, we can use the greedy constraint set selection Algorithm 5. The repeat loop in this algorithm can be stopped either when $\#\mathcal{C}$ has reached a

(1) choose $M \in N$;
(2) initialize $\mathcal{C} = \{(k_1, \mu_1)\}$ with arbitrary $k_1 \in \{1, \ldots, \mathcal{T}\}$ and $\mu_1 \in \mathcal{P}$;
(3) choose a rather large, finite-sized sample $\Xi \subset \{1, \ldots, \mathcal{T}\} \times \mathcal{P}$;
(4) repeat:
  - using the "current" $\mathcal{C}$ to compute C^{sup} and C^{inf}, append:

$$(k^*, \mu^*) = \underset{(k,\mu)\in\Xi}{\operatorname{argmax}} \frac{\exp\left(C_k^{sup}(\mu)\right) - \exp\left(C_k^{inf}(\mu)\right)}{\exp\left(C_k^{sup}(\mu)\right)} \qquad \text{to } \mathcal{C}.$$

Algorithm 5: Greedy constraint set selection

maximal value, or when the "relative exponential sharpness" indicator:

$$\max_{(k,\mu)\in\Xi} \frac{\exp\left(C_k^{sup}(\mu)\right) - \exp\left(C_k^{inf}(\mu)\right)}{\exp\left(C_k^{sup}(\mu)\right)}$$

gets less than a desired precision. We use a measure of the difference between the exponentials so as to account for the "exponential" effect of the stability constants on the error bounds [18].
As in the greedy algorithm for basis selection described at Section 2.3.2, this algorithm makes, at each step, the "best possible" choice, that is the value of the parameter and time for which the bounds computed using the current constraint set are the less sharp.
A last remark we can give on the algorithm is about the trade-off in the choice of M: whatever M is, we always get a certified bound on $C_k(\mu)$, but increasing M will improve sharpness of this bound, at the expense of an increase in online computation time.

4. Numerical results

We now present some numerical results obtained with the methodology described above. We implemented it in a software package written in C++, using GNU OpenMP [8] as threading library, ARPACK [1] for eigenvalues computation and GLPK [7] as linear programming problems solver.
For all the experiments above, the convergence test for Newton iterations when solving (1.13) and (2.2b) was the following: $\|\delta\|^2 \leq 3 \times 10^{-16}$. The penalization constant used was $P = 10^7$.
We also took $M = \#\mathcal{C} = 10$ as parameters for the SCM procedure.
The Φ functions appearing in the parametrizations of u_0, b_0, b_1 and f are chosen to be sine functions with fixed known angular velocity. More specifically, we suppose that:

$$b_0(t) = b_{0m} + \sum_{l=1}^{n(b_0)} A_l^{b_0} \sin(\omega_l^{b_0} t) \qquad b_1(t) = b_{1m} + \sum_{l=1}^{n(b_1)} A_l^{b_1} \sin(\omega_l^{b_1} t)$$

$$f(t,x) = f_m + \sum_{l=1}^{n_T(f)} \sum_{p=1}^{n_S(f)} A_{lp}^f \sin(\omega_l^{fT} t) \sin(\omega_p^{fS} x) \qquad u_0(x) = u_{0m} + \sum_{l=1}^{n(u_0)} A_l^{u_0} \sin(\omega_l^{u_0} x)$$

4.1. Reference solutions

Figure 1 shows an example of the reference solution of (1.1), (1.2) and (1.7) every 10 timesteps. The parameters were:

$$\begin{aligned} \mathcal{N} &= 40 & \Delta t &= .02 \\ T &= 2 & \nu &= 1 \\ b_0(t) &= 1 & b_1(t) &= 1.28224 \\ f(t,x) &= 1 & u_0(x) &= 1 + 2\sin(3x) \end{aligned} \tag{4.1}$$

Figure 2 does the same with a lower viscosity. The parameters were:

$$\begin{aligned} \mathcal{N} &= 40 & \Delta t &= .002 \\ T &= 2 & \nu &= .1 \\ b_0(t) &= 1 & b_1(t) &= 1.28224 \\ f(t,x) &= 1 & u_0(x) &= 1 + 2\sin(3x) \end{aligned} \tag{4.2}$$

Figures 1 and 2 show the solution u of the viscous Burgers' equation plotted as functions of space x, for various times t, respectively for the parameter set (4.1) and (4.2).
We have checked that the *a posteriori* indicator of boundary error (1.8) gets no higher than 6×10^{-7} in both cases.

4.2. Reduced solutions

Computational economy

To show the substantial time savings provided by the reduced basis approximation, we compute the reduced solution for the parameters set given by (4.1), with $\mathcal{N} = 60$. The full solution (with $\mathcal{N} = 60$) takes 0.26s CPU time to be produced (when using Thomas' algorithm for tridiagonal matrices inversion).
We use the POD-driven basis selection procedure to select the $N = 7$ leading POD modes (using $S = 30$ snapshots). The resulting basis (with the functions sorted by increasing magnitude of eigenvalues) is shown in Figure 3. We did not make use of the "expansion" procedure described in Section 2.3.4. The overall CPU time for the offline phase was 6.36s.
We used fixed parameters $n(b_0) = n(b_1) = n_S(f) = n_T(f) = n(u_0) = 1$ and parameter ranges as shown in Table 1.

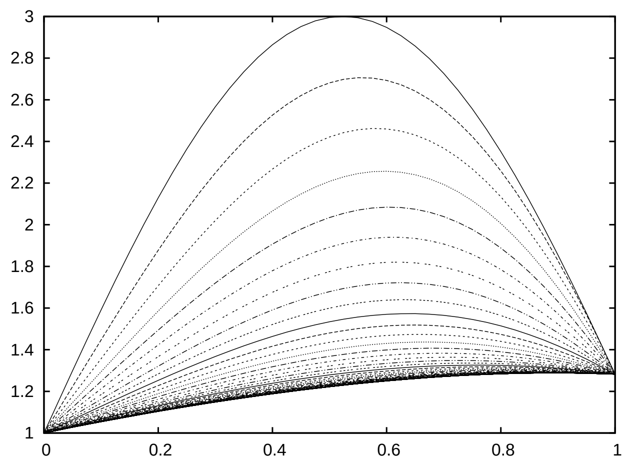

FIGURE 1. Full solution with high viscosity $\nu = 1$. Plots of the solution u of equation (1.1), (1.2), (1.7) as a function of space x, for various times t ranging from $t = 0$ to $t = 2$ (the bottom lines correspond to high times). We use the parameters defined in (4.1).

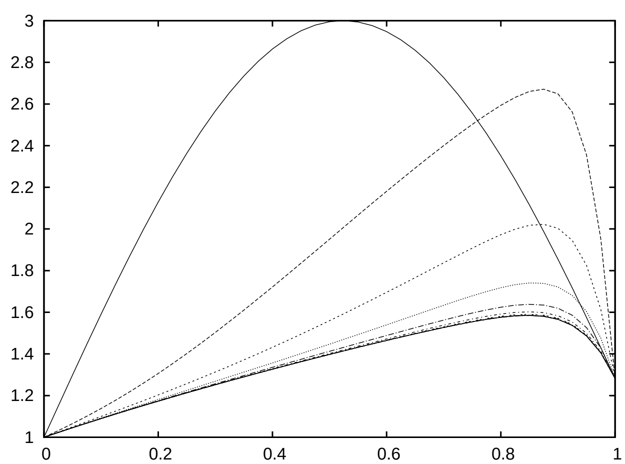

FIGURE 2. Full solution with low viscosity $\nu = .1$. Plots of the solution u of (1.1), (1.2), (1.7) as a function of space x, for various times t ranging from $t = 0$ to $t = 2$ (the bottom lines correspond to high times). We use the parameters defined in (4.2).

We then used this basis to compute the reduced solution for a particular (randomly chosen) instance of the parameters. The reduced solution was computed in 0.04s, *including* the time necessary for the online error bound computation, shown in Figure 4 (solid line). Our procedure reduces the marginal cost to 15% of the original cost, yet providing a certified L^2 relative error of less than 1‰.

Error bound estimation

Still using the preceding POD basis and instance of the parameters, we compared the online error bound with the actual error, for the same parameter set as above. The result is shown in Figure 4. We see that our error bound is quite sharp, especially when it follows the decrease in the actual error near $t = 1.3$. We also checked

Parameter	Min.	Max.	Parameter	Min.	Max.
ν	.8	1.2	$A_1^{u_0}$	1.1	3
$A_1^{b_0}$	.9	1.2	$\omega_1^{b_0}$	1	1
$A_1^{b_1}$	.9	1.2	$\omega_1^{b_1}$	1	1
f_m	0	2	ω_1^{fT}	2	2
$A_{1,1}^f$	0.7	1.3	ω_1^{fS}	2	2
u_{0m}	0	1	$\omega_1^{u_0}$	3	3

TABLE 1. Ranges of the different parameters in the benchmark of Section 4.2.

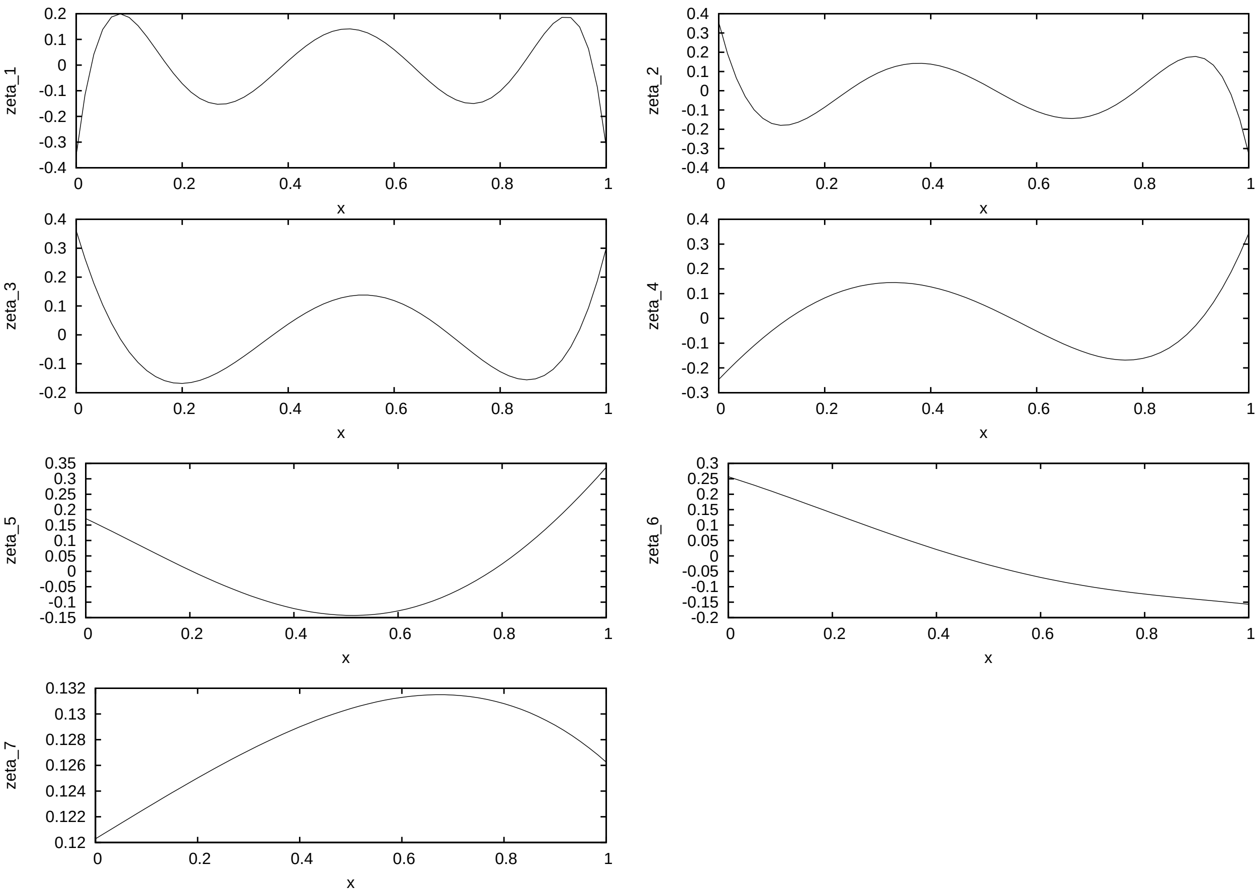

FIGURE 3. POD reduced basis: plots, as functions of space, of the 7 leading POD modes, i.e. the ζ_i $(i = 1, \ldots, 7)$ defined by (2.6), with z_i $(i = 1, \ldots, 7)$ the leading eigenvectors of $M^T \Omega M$. The modes are sorted (from top to bottom, and from left to right) by increasing magnitude of eigenvalues. Parameter ranges for snapshot sampling are those in Table 1.

for the quality of the SCM procedure, by comparing the actual stability constant with the lower bound provided by SCM (Figure 5).

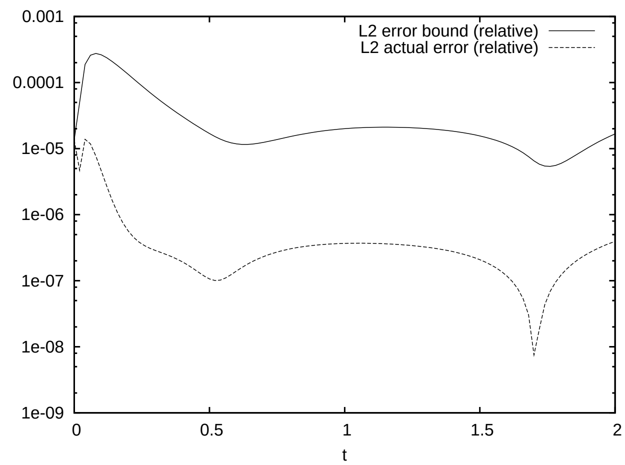

FIGURE 4. Relative L^2 online error bound and actual error. We plot in solid line $\dfrac{\varepsilon_k}{\|\widetilde{u}^k\|}$, and in dashed line $\dfrac{\|u^k - \widetilde{u}^k\|}{\|\widetilde{u}^k\|}$ as functions of $t = k\Delta t$ for $k = 1, \dots, \mathcal{T}$.

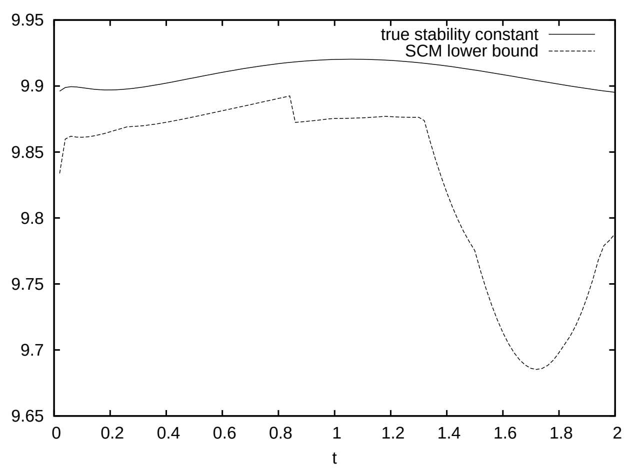

FIGURE 5. True stability constant C_k defined by (3.3) and SCM lower bound C_k^{inf} defined by (3.11) as functions of time. We use $M = \#\mathcal{C} = 10$ as SCM parameters.

4.3. **Convergence benchmarks**

In order to compare our three basis selection procedures (POD, greedy and POD-Greedy), we have made "convergence benchmarks", i.e. representations of the maximal and mean (estimated) error over all timesteps, and over a sample of 100 parameters as functions of the size of the reduced basis. The same sample of benchmark parameters is used throughout all the procedure.
Comparison of greedy (with $\#\Xi = 100$) and POD (with $S = 60$) procedures for $\mathcal{N} = 40$, $T = 2$, $\Delta t = .02$, $n(b_0) = n(b_1) = n_S(f) = n_T(f) = n(u_0) = 0$, with parameters f_m and ν fixed to unity, and varying $u_{0m} \in [0, 1]$

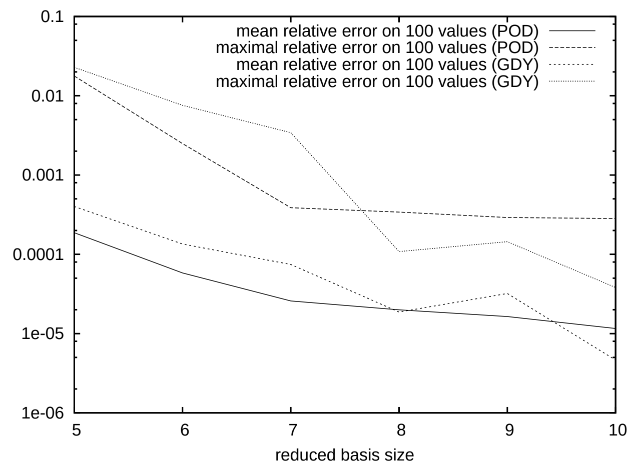

FIGURE 6. Convergence benchmark 1. We plot (on a logarithmic scale) maximal and mean relative online error bounds over a (uniform) random sample of 100 initial values $u_{0m} \in [0, 1]$, as functions of the reduced basis size N, when the reduced basis is chosen using POD-based procedure (POD) or greedy procedure (GDY). Fixed parameters are $\nu = 1$ and $f_m = 1$.

(and thus initial boundary values b_{0m} and b_{1m}, moving accordingly to compatibility conditions (1.4)) is shown in Figure 6.

The benchmarking process for greedy took 19.5s of CPU time, the one for POD took 14.98s. The online cost, depending only on the size of the reduced basis, is the same regardless of how the reduced basis has been chosen. We also see the fast (exponential) convergence of error bound towards zero as N increases.

Another benchmark was then made, with the same data, except that $\nu = .1$ and $\Delta t = .002$. The result is visualized in Figure 7. The POD benchmarking process took 349 s of CPU time, the POD-Greedy, 282 s, and the Greedy, 470 s. We notice that a smaller viscosity leads to degraded precision of our RB approximation. The resulting final basis selected by the POD-Greedy is displayed in Figure 8.

The POD-Greedy algorithm is run using $P_1 = 2$ and initialized using the full time-discrete trajectory (hence, 200 vectors) for one parameter value. Using the initial data as initialization for the POD-Greedy algorithm and using $P_1 = 1$ may give better overall performance of POD-Greedy (at the expense of an increased offline computation time), as the sample of error bounds is updated at every step of the algorithm (instead of every other step for $P_1 = 2$).

4.4. **Effect of mesh refinement and penalization constant**

To check for the robustness of our bound, we studied the influence of $\mathcal{N}$ (the number of "full" spatial discretization points) and P on the magnitude and the sharpness of our error bound. We ran the same benchmark as in Section 4.2, but with different $\mathcal{N}$ or P.

We did the test with refined meshes ($\mathcal{N} = 200$, $\mathcal{N} = 800$) and we obtained a similar error bound profile; we conclude that the sharpness of our error bound is quite insensitive to mesh refinement.

In Figure 9, we visualize the actual error and the error bound for various values of P (with $\mathcal{N} = 60$). We see that the error bound is tighter and sharper for high values of the penalization constant P. This can easily be explained by the fact that, as $P \to +\infty$, the errors at the boundary $e_k(0)$ and $e_k(1)$ vanish, hence modifying all the terms in the error bound (i.e., decreasing η_k, σ_k, f_k, $\xi_k^{\mathcal{A}}$, $\xi_k^{\mathcal{B}}$, ξ_k^{γ}, $\mathcal{B}_k$, γ_k and increasing $\mathcal{A}_k$) where these errors appear.

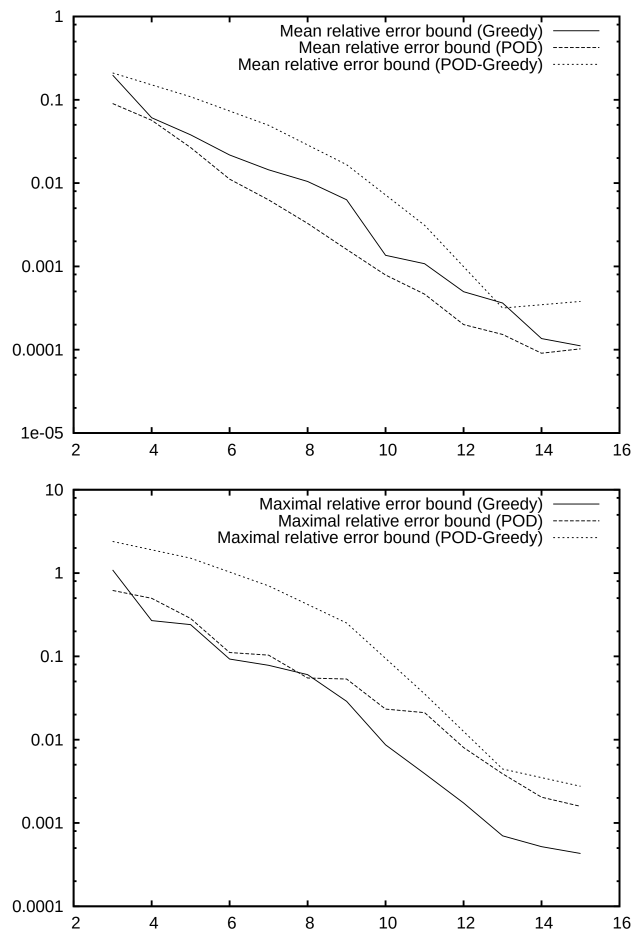

Figure 7. Convergence benchmark 2. We plot (on a logarithmic scale) maximal (bottom) and mean (top) online error bounds over a (uniform) random sample of 100 initial values $u_{0m} \in [0,1]$, as functions of the reduced basis size N, when reduced basis is chosen using POD-based procedure with $S = 90$, Greedy, or POD-Greedy procedure with $P_1 = 2$. Fixed parameters are $\nu = 0.1$ and $f_m = 1$.

4.5. **Comparison with existing bound**

We compared our error bound with the bound described in [18], for $\mathcal{N} = 60$, $\Delta t = .02$, $T = 2$, fixed $u_0 = b_0 = b_1 = 0$, $f = 1$ and variable $\nu \in [0.1; 1]$. The reduced basis is found by POD with $S = 90$. Figure 10 shows that our bound is clearly better than the existing reference bound.
Besides, we have performed a benchmark over a fixed sample of 100 random values of ν uniformly chosen in $[0.1; 1]$. For our bound, the mean error bound is 0.00076, the maximum error bound is 0.02, while for the bound of [18], we obtain 0.0041 for the mean bound and 0.25 for the maximum.
It is easy to see that, when the boundary conditions are fixed to zero, the recurrence formula for the error bound reduces to:

$$\|e_k\| \leq \frac{\|e_{k-1}\| + \Delta t \, \|r_k\|_0}{1 + C_k \Delta t}$$

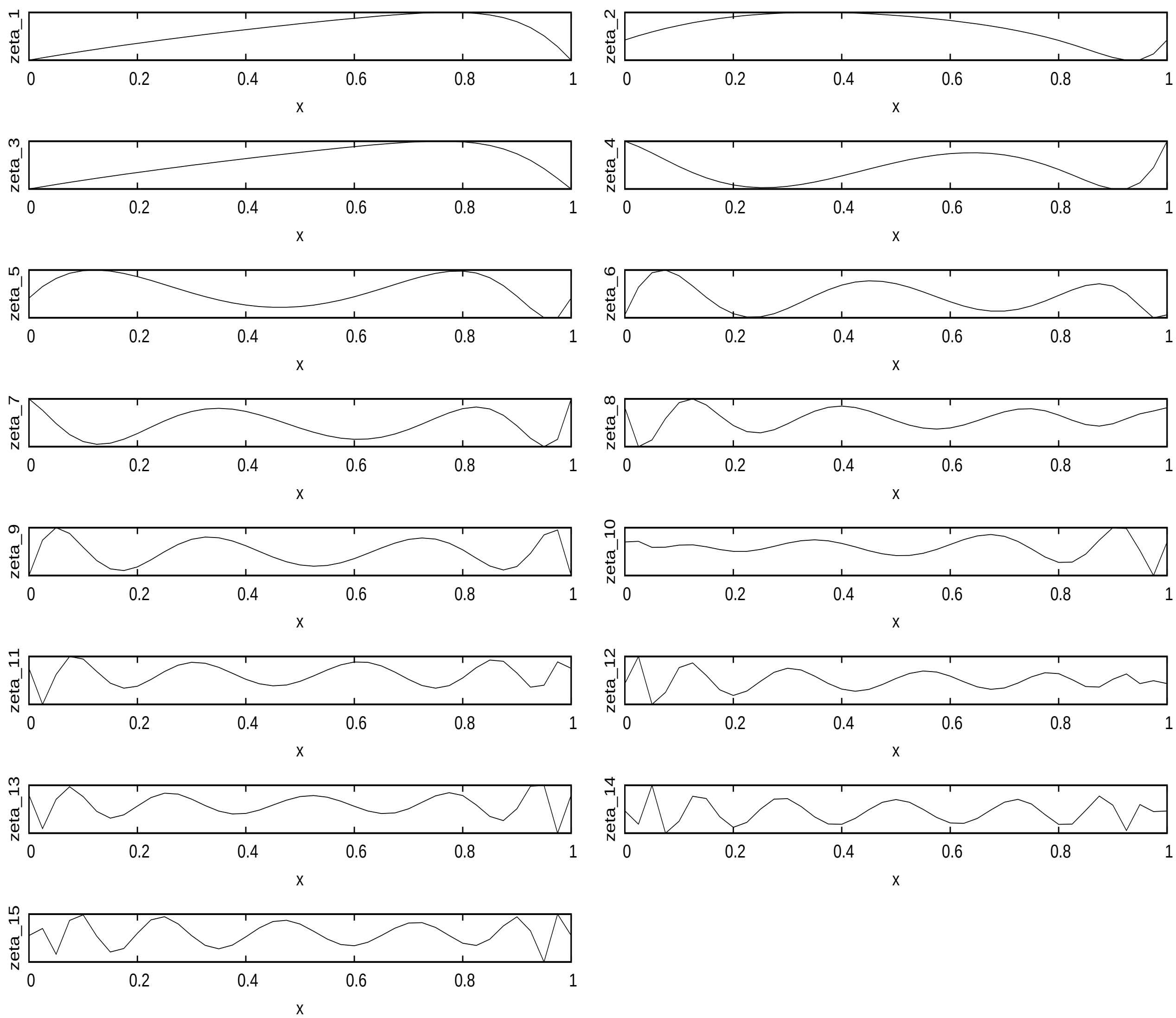

FIGURE 8. Reduced basis selected at the final step of the POD-Greedy procedure carried out for the previous benchmark (Figure 7). Basis elements are plotted as functions of space, and are sorted (from top to bottom, left to right) in order of selection in the greedy procedure.

while the error bound described in [18] has the following expression:

$$\|e_k\| \leq \sqrt{\frac{\|e_{k-1}\|^2 + \frac{\Delta t}{\nu}\|r_k\|_0^2}{1 + \widetilde{C_k}\Delta t}}$$

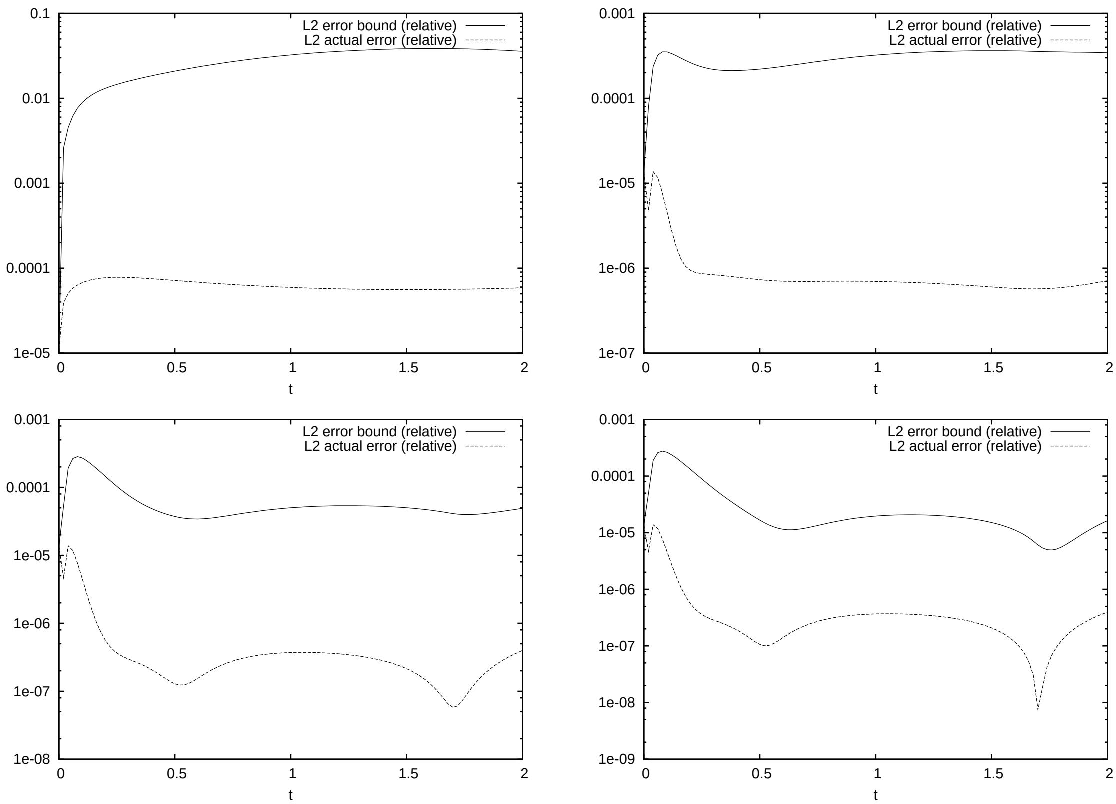

FIGURE 9. Relative L^2 online error bound and actual error. We plot in solid line $\frac{\varepsilon_k}{\|\widetilde{u}^k\|}$, and in dashed line $\frac{\|u^k - \widetilde{u}^k\|}{\|\widetilde{u}^k\|}$ as functions of $t = k\Delta t$ for $k = 1, \ldots, \mathcal{T}$, and $\mathcal{N} = 60$ and, from left to right and top to bottom, $P = 10^2$, $P = 10^4$, $P = 10^5$ and $P = 10^{12}$.

where the modified stability constant $\widetilde{C_k}$ reads:

$$\widetilde{C_k} = \inf_{v \in X_0} \frac{4c(\widetilde{u}^k, v, v) + \nu a(v, v)}{\|v\|^2}$$

We recall that our stability constant C_k is given by:

$$C_k = \inf_{v \in X_0, \|v\|=1} \left[2c(\widetilde{u}^k, v, v) + \nu a(v, v) \right]$$

The ν-dependence of the bound can explain why our bound is better, especially for small values of ν. Besides, the derivation of our error bound makes lesser use of inequalities (for instance, we do not make use of Young's inequality at the beginning of the proof, each inequality used is a potential source of optimality loss) and keeps treating more terms.

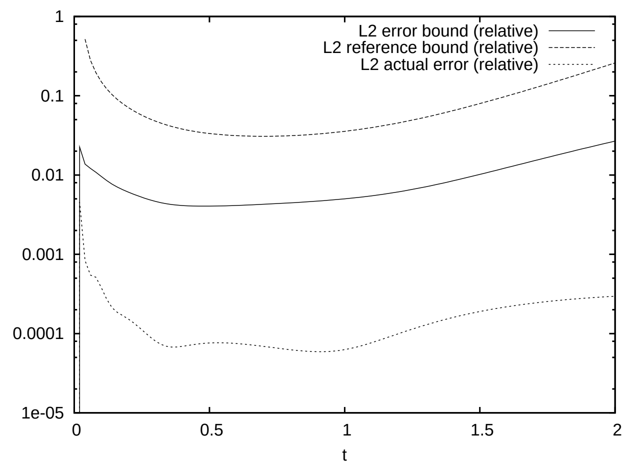

FIGURE 10. Comparison with the existing error bound for reduced basis Burgers equation (reference bound). We plot the actual error, the existing (reference) error bound and our error bound as functions of time. We took $\nu = 0.1$.

CONCLUSION

We have presented a certified procedure for low marginal cost approximate resolution of the viscous Burgers equation with parametrized viscosity, as well as initial and boundary value data. This procedure makes use of a reduced basis offline/online procedure for a penalized weak formulation, an efficiently computed error bound in natural L^2 norm (made possible by the successive constraints method (SCM)), and three procedures at hand for choosing a basis to expand reduced solutions in.

Our procedure becomes less useful when the ratio time/viscosity increases, as this degradates the stability constant C_k. Another limitation of our method, for one willing to use it for large times, is that the online procedure complexity still depends on the temporal discretization step. However, our numerical experiments show a substantial decrease in marginal cost when using reduced basis approximation, as well as efficiency (both in terms of sharpness and computation time) of the provided error bound for moderate viscosities. This decrease in the cost is made possible by the fact that online procedure has a complexity that is independent from the number of spatial discretization points.

APPENDIX A. PROOF OF THEOREM 3.1

Proof. Subtracting left-hand side of (2.2b) from both sides of relation (3.1) yields that for $k = 1, \ldots, \mathcal{T}$, the error at time t_k: $e_k = u_e^k - \widetilde{u}^k$ satisfies, for every $v \in X_0$:

$$\frac{1}{\Delta t}(\langle e_k, v\rangle - \langle e_{k-1}, v\rangle) + c(u_e^k, u_e^k, v) - c(\widetilde{u}^k, \widetilde{u}^k, v) + \nu a(e_k, v) = r_k(v) \tag{A.1}$$

We write:

$$e_k = e_k(0)\phi_0 + e_k(1)\phi_{\mathcal{N}} + e_k^z \tag{A.2}$$

with $e_k^z \in X_0 = \{v \in X \text{ st. } v(0) = v(1) = 0\}$. And, by applying (A.1) with $v = e_k^z$:

$$\frac{1}{\Delta t}(\langle e_k, e_k^z\rangle - \langle e_{k-1}, e_k^z\rangle) + c(u_e^k, u_e^k, e_k^z) - c(\widetilde{u}^k, \widetilde{u}^k, e_k^z) + \nu a(e_k, e_k^z) = r_k(e_k^z) \tag{A.3}$$

Since:

$$\begin{aligned}\langle e_k, e_k^z\rangle &= e_k(0)\langle \phi_0, e_k^z\rangle + e_k(1)\langle \phi_{\mathcal{N}}, e_k^z\rangle + \|e_k^z\|^2 \\ &= \|e_k^z\|^2 + e_k(0)e_k^z\left(\frac{1}{\mathcal{N}}\right)\langle \phi_0, \phi_1\rangle + e_k(1)e_k^z\left(1-\frac{1}{\mathcal{N}}\right)\langle \phi_{\mathcal{N}-1}, \phi_{\mathcal{N}}\rangle\end{aligned}$$

and that, for every $v \in X$:

$$\begin{aligned}c(u_e^k, u_e^k, v) - c(\widetilde{u}^k, \widetilde{u}^k, v) &= -\frac{1}{2}\int_0^1 \left((u_e^k)^2 - (\widetilde{u}^k)^2\right)\frac{\partial v}{\partial x} \\ &= -\frac{1}{2}\int_0^1 \left(u_e^k - \widetilde{u}^k\right)\left(u_e^k + \widetilde{u}^k\right)\frac{\partial v}{\partial x} \\ &= -\frac{1}{2}\int_0^1 e_k\left(\widetilde{u}^k + \widetilde{u}^k + e_k\right)\frac{\partial v}{\partial x} \\ &= 2c(\widetilde{u}^k, e_k, v) + c(e_k, e_k, v)\end{aligned}$$

We have that (A.3) implies:

$$\begin{aligned}\frac{1}{\Delta t}(\|e_k^z\|^2 - \langle e_{k-1}, e_k^z\rangle) + \psi_k(e_k, e_k^z) + \frac{1}{\Delta t}\left(e_k(0)e_k^z\left(\frac{1}{\mathcal{N}}\right)\langle \phi_0, \phi_1\rangle + e_k(1)e_k^z\left(1-\frac{1}{\mathcal{N}}\right)\langle \phi_{\mathcal{N}}, \phi_{\mathcal{N}-1}\rangle\right) \\ = r_k(e_k^z) - c(e_k, e_k, e_k^z) \qquad \text{(A.4)}\end{aligned}$$

We are now willing to find a lower bound for the left-hand side of (A.4) and an upper bound for its right-hand side.

Lower bound for LHS. From Cauchy-Schwarz inequality:

$$-\langle e_{k-1}, e_k^z\rangle \quad \geq \quad -\|e_{k-1}\|\,\|e_k^z\|$$

and, by the triangle inequality:

$$\|e_k\| - \eta_k \leq \|e_k^z\| \leq \|e_k\| + \eta_k \qquad \text{(A.5)}$$

because $\eta_k = |e_k(0)|\,\|\phi_0\| + |e_k(1)|\,\|\phi_{\mathcal{N}}\|$.

So:

$$-\langle e_{k-1}, e_k^z\rangle \quad \geq \quad -\|e_{k-1}\|\left(\|e_k\| + \eta_k\right) \qquad \text{(A.6)}$$

We also have:

$$\|e_k^z\|^2 \geq \|e_k\|^2 + \|e_k - e_k^z\|^2 - 2\,\|e_k\|\,\|e_k - e_k^z\| \geq \|e_k\|^2 - 2\,\|e_k\|\,\eta_k \qquad \text{(A.7)}$$

because of Cauchy-Schwarz and triangle inequalities.

Besides,

$$0 \leq \|e_k^z\| \leq \|e_k^z - e_k\| + \|e_k\| \leq \eta_k + \|e_k\|$$

so that:

$$\|e_k^z\|^2 \leq \left(\|e_k\| + \eta_k\right)^2 \qquad \text{(A.8)}$$

Using the bilinearity of ψ_k, we have:

$$\psi_k(e_k, e_k^z) = \psi_k(e_k^z, e_k^z) + e_k(0)\psi_k(\phi_0, \phi_1)e_k^z\left(\frac{1}{\mathcal{N}}\right) + e_k(1)\psi_k(\phi_{\mathcal{N}}, \phi_{\mathcal{N}-1})e_k^z\left(1-\frac{1}{\mathcal{N}}\right) \qquad \text{(A.9)}$$

because $\psi_k(\phi_0, \phi_{\mathcal{N}}) = \psi_k(\phi_{\mathcal{N}}, \phi_0) = 0$, since ϕ_0 and $\phi_{\mathcal{N}}$ have no common support, $\psi_k(\phi_0, \phi_j) = 0$ for $j > 1$, and $\psi_k(\phi_{\mathcal{N}}, \phi_j) = 0$ for $j < \mathcal{N} - 1$.

From the definition of the stability constant C_k:

$$\psi_k(e_k^z, e_k^z) \geq C_k \left\| e_k^z \right\|^2$$

So that, thanks to (A.7) and (A.8),

$$\psi_k(e_k^z, e_k^z) \geq \begin{cases} C_k \left\| e_k \right\|^2 - 2\eta_k C_k \left\| e_k \right\| & \text{if } C_k \geq 0 \\ C_k \left\| e_k \right\|^2 + 2\eta_k C_k \left\| e_k \right\| + C_k \eta_k^2 & \text{if } C_k \leq 0 \end{cases}$$

That is

$$\psi_k(e_k^z, e_k^z) \geq C_k \left\| e_k \right\|^2 - \sigma_k \left\| e_k \right\| - [C_k]_- \, \eta_k^2 \tag{A.10}$$

from the definition of σ_k.
We have:

$$\left| e_k^z \left(\frac{1}{\mathcal{N}} \right) \right| \leq \mathcal{E} \left\| e_k^z \right\| \leq \mathcal{E} \left\| e_k \right\| + \mathcal{E} \eta_k \tag{A.11}$$

and by symmetry:

$$\left| e_k^z \left(1 - \frac{1}{\mathcal{N}} \right) \right| \leq \mathcal{E} \left\| e_k^z \right\| \leq \mathcal{E} \left\| e_k \right\| + \mathcal{E} \eta_k \tag{A.12}$$

so that, combining (A.11), (A.12) and introducing f_k:

$$\left| e_k(0) \psi_k(\phi_0, \phi_1) e_k^z \left(\frac{1}{\mathcal{N}} \right) + e_k(1) \psi_k(\phi_{\mathcal{N}}, \phi_{\mathcal{N}-1}) e_k^z \left(1 - \frac{1}{\mathcal{N}} \right) \right| \leq \left\| e_k \right\| f_k + \eta_k f_k$$

Now we can say that:

$$e_k(0) \psi_k(\phi_0, \phi_1) e_k^z \left(\frac{1}{\mathcal{N}} \right) + e_k(1) \psi_k(\phi_{\mathcal{N}}, \phi_{\mathcal{N}-1}) e_k^z \left(1 - \frac{1}{\mathcal{N}} \right) \geq - \left\| e_k \right\| f_k - \eta_k f_k \tag{A.13}$$

And we also have:

$$\begin{aligned} \left| e_k(0) e_k^z \left(\frac{1}{\mathcal{N}} \right) \langle \phi_0, \phi_1 \rangle + e_k(1) e_k^z \left(1 - \frac{1}{\mathcal{N}} \right) \langle \phi_{\mathcal{N}-1}, \phi_{\mathcal{N}} \rangle \right| &\leq |e_k(0)| \left(\mathcal{E} \left\| e_k \right\| + \mathcal{E} \eta_k \right) \langle \phi_0, \phi_1 \rangle \\ &\quad + |e_k(1)| \left(\mathcal{E} \left\| e_k \right\| + \mathcal{E} \eta_k \right) \langle \phi_{\mathcal{N}}, \phi_{\mathcal{N}-1} \rangle \\ &= \mathcal{E} (|e_k(0)| \langle \phi_0, \phi_1 \rangle + |e_k(1)| \langle \phi_{\mathcal{N}}, \phi_{\mathcal{N}-1} \rangle) \left\| e_k \right\| \\ &\quad + \mathcal{E} \eta_k (|e_k(0)| \langle \phi_0, \phi_1 \rangle + |e_k(1)| \langle \phi_{\mathcal{N}}, \phi_{\mathcal{N}-1} \rangle) \\ &= \mathcal{E} \langle \phi_0, \phi_1 \rangle (|e_k(0)| + |e_k(1)|) \left\| e_k \right\| \\ &\quad + \mathcal{E} \eta_k \langle \phi_0, \phi_1 \rangle (|e_k(0)| + |e_k(1)|) \end{aligned} \tag{A.14}$$

since $\langle \phi_0, \phi_1 \rangle = \langle \phi_{\mathcal{N}}, \phi_{\mathcal{N}-1} \rangle$ by symmetry.
Thus, thanks to (A.7), (A.6), (A.9), (A.10), (A.13) and (A.14), the left-hand side of (A.4) is greater than:

$$\begin{aligned} &\left(\frac{1}{\Delta t} + C_k \right) \left\| e_k \right\|^2 - \left(\frac{2\eta_k + \left\| e_{k-1} \right\| + \mathcal{E} \langle \phi_0, \phi_1 \rangle (|e_k(0)| + |e_k(1)|)}{\Delta t} + \sigma_k + f_k \right) \left\| e_k \right\| \\ &\qquad - \frac{\eta_k \left\| e_{k-1} \right\| + \mathcal{E} \eta_k \langle \phi_0, \phi_1 \rangle (|e_k(0)| + |e_k(1)|)}{\Delta t} - [C_k]_- \, \eta_k^2 - \eta_k f_k \end{aligned} \tag{A.15}$$

Upper bound for RHS. We have:

$$c(e_k, e_k, e_k^z) = c(e_k, e_k, e_k) - e_k(0) c(e_k, e_k, \phi_0) - e_k(1) c(e_k, e_k, \phi_{\mathcal{N}})$$

but:

$$\begin{aligned} c(e_k, e_k, e_k) &= -\frac{1}{2}\int_0^1 e_k^2 \frac{\partial e_k}{\partial x} \\ &= -\frac{1}{6}\int_0^1 \frac{\partial\left[(e_k)^3\right]}{\partial x} \\ &= -\frac{1}{6}\left((e_k(1))^3 - (e_k(0))^3\right) \end{aligned}$$

So:

$$\begin{aligned} c(e_k, e_k, e_k^z) &= -\frac{1}{6}\left((e_k(1))^3 - (e_k(0))^3\right) - \left(e_k(0)\int_0^1 e_k^2 \frac{\partial \phi_0}{\partial x} + e_k(1)\int_0^1 e_k^2 \frac{\partial \phi_{\mathcal{N}}}{\partial x}\right) \\ &= -\frac{1}{6}\left((e_k(1))^3 - (e_k(0))^3\right) + \mathcal{N}\left(e_k(0)\int_0^{1/\mathcal{N}} e_k^2 - e_k(1)\int_{1-1/\mathcal{N}}^1 e_k^2\right) \end{aligned} \tag{A.16}$$

Since, for all $x \in \left[0; \dfrac{1}{\mathcal{N}}\right]$,

$$e_k(x) = e_k(0) + \mathcal{N}\left(e_k\left(\frac{1}{\mathcal{N}}\right) - e_k(0)\right)x$$

we have, thanks to $\left|e_k\left(\dfrac{1}{\mathcal{N}}\right)\right| \leq \mathcal{E}\left\|e_k\right\|$:

$$\begin{aligned} \left|\mathcal{N} \times e_k(0)\int_0^{1/\mathcal{N}} e_k^2\right| &\leq \mathcal{N}\frac{|e_k(0)|}{\mathcal{N}}\left(\left|e_k\left(\frac{1}{\mathcal{N}}\right)\right||e_k(0)| + \frac{e_k\left(\frac{1}{\mathcal{N}}\right)^2 + e_k(0)^2}{3} + \frac{2\left|e_k\left(\frac{1}{\mathcal{N}}\right)e_k(0)\right|}{3}\right) \\ &\leq |e_k(0)|\left(\mathcal{E}\left\|e_k\right\||e_k(0)| + \frac{\mathcal{E}^2\left\|e_k\right\|^2}{3} + \frac{e_k(0)^2}{3} + \frac{2\,|e_k(0)|}{3}\mathcal{E}\left\|e_k\right\|\right) \\ &\leq \frac{\mathcal{E}^2\,|e_k(0)|}{3}\left\|e_k\right\|^2 + \frac{5}{3}|e_k(0)|^2\,\mathcal{E}\left\|e_k\right\| + \frac{|e_k(0)|^3}{3} \end{aligned}$$

As a similar computation can be worked out for $\left|\mathcal{N} \times e_k(1)\displaystyle\int_{1-1/\mathcal{N}}^1 e_k^2\right|$, we have:

$$-\left(e_k(0)\int_0^1 e_k^2\frac{\partial\phi_0}{\partial x} + e_k(1)\int_0^1 e_k^2\frac{\partial\phi_{\mathcal{N}}}{\partial x}\right) \leq \xi_k^{\mathcal{A}}\left\|e_k\right\|^2 + \xi_k^{\mathcal{B}}\left\|e_k\right\| + \xi_k^{\gamma}$$

We also have, thanks to (A.5):

$$|r_k(e_k^z)| \leq \|r_k\|_0\,\|e_k^z\| \leq \|r_k\|_0\,\|e_k\| + \|r_k\|_0\,\eta_k$$

where:

$$\|r_k\|_0 = \sup_{v\in X_0, \|v\|=1} r_k(v)$$

Hence, the right-hand side of (A.4) is less than:

$$\xi_k^{\mathcal{A}}\left\|e_k\right\|^2 + (\|r_k\|_0 + \xi_k^{\mathcal{B}})\left\|e_k\right\| + \frac{1}{6}\left|e_k(1)^3 - e_k(0)^3\right| + \xi_k^{\gamma} + \|r_k\|_0\,\eta_k$$

Conclusion. Now (A.4) implies, thanks to (A.15), and (A.16):

$$\mathcal{A}_k \|e_k\|^2 - \mathcal{B}_k \|e_k\| - \gamma_k \leq 0 \tag{A.17}$$

Viewing left-hand side of (A.17) as a (convex, thanks to our hypothesis (3.4)) quadratic function Q of $\|e_k\|$, whose discriminant is $\mathcal{D}_k$, equation (A.17) implies that, if $\mathcal{D}_k \geq 0$, $\|e_k\|$ is smaller than the greatest real root of Q, that is:

$$\|e_k\| \leq \frac{\mathcal{B}_k + \sqrt{\mathcal{D}_k}}{2\mathcal{A}_k}$$

If $\mathcal{D}_k < 0$, then necessarily $\gamma_k < 0$ (as $\mathcal{A}_k$ is positive). Hence (A.17) implies:

$$\mathcal{A}_k \|e_k\|^2 - \mathcal{B}_k \|e_k\| \leq 0$$

and so:

$$\|e_k\| \leq \frac{\mathcal{B}_k}{\mathcal{A}_k}. \qquad \square$$

We would like to thank the anonymous referees, whose careful reading and comments have helped to greatly improve this paper. This work has been partially supported by the French National Research Agency (ANR) through COSINUS program (project COSTA-BRAVA n° ANR-09-COSI-015).

References

[1] ARPACK: Arnoldi Package. http://www.caam.rice.edu/software/ARPACK/.
[2] I. Babuska. The finite element method with penalty. *Math. Comp*, 27(122):221–228, 1973.
[3] J.W. Barrett and C.M. Elliott. Finite element approximation of the Dirichlet problem using the boundary penalty method. *Numerische Mathematik*, 49(4):343–366, 1986.
[4] A. Buffa, Y. Maday, A.T. Patera, C. Prud'homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis. *Mathematical Modelling and Numerical Analysis*, 2009.
[5] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.
[6] V. Girault and P.A. Raviart. Finite element methods for Navier-Stokes equations, volume 5 of Springer Series in Computational Mathematics, 1986.
[7] GLPK: GNU Linear Programming Kit. http://www.gnu.org/software/glpk/.
[8] GOMP: An OpenMP implementation for GCC. http://gcc.gnu.org/projects/gomp/.
[9] M.A. Grepl. *Reduced-Basis Approximation and A Posteriori Error Estimation for Parabolic Partial Differential Equations.* PhD thesis, Massachusetts Institute of Technology, 2005.
[10] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis*, 41(3):575–605, 2007.
[11] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *Mathematical Modelling and Numerical Analysis*, 39(1):157–181, 2005.
[12] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.
[13] J.C. Helton, J.D. Johnson, C.J. Sallaberry, and C.B. Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 91(10-11):1175–1209, 2006.
[14] E. Hopf. The partial differential equation $u_t + uu_x = \mu_{xx}$. *Communications on Pure and Applied Mathematics*, 3(3):201–230, 1950.
[15] D.B.P. Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathematique*, 345(8):473–478, 2007.
[16] N. Jung, B. Haasdonk, and D. Kroner. Reduced Basis Method for quadratically nonlinear transport equations. *International Journal of Computing Science and Mathematics*, 2(4):334–353, 2009.
[17] D.J. Knezevic and A.T. Patera. A certified reduced basis method for the Fokker-Planck equation of dilute polymeric fluids: FENE dumbbells in extensional flow. *SIAM Journal on Scientific Computing*, 32(2):793–817, 2010.
[18] N.C. Nguyen, G. Rozza, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. *Calcolo*, 46(3):157–185, 2009.
[19] N.C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. *Handbook of Materials Modeling*, pages 1523–1558, 2005.

[20] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Verlag, 1999.
[21] A.M. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer, 2008.
[22] D.V. Rovas, L. Machiels, and Y. Maday. Reduced-basis output bound methods for parabolic problems. *IMA journal of numerical analysis*, 26(3):423, 2006.
[23] A. Saltelli, K. Chan, and E.M. Scott. *Sensitivity analysis*. Wiley: New York, 2000.
[24] J.C. Strikwerda. *Finite difference schemes and partial differential equations*. Society for Industrial Mathematics, 2004.
[25] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.
[26] K. Veroy, C. Prud'homme, and A.T. Patera. Reduced-basis approximation of the viscous Burgers equation: rigorous a posteriori error bounds. *Comptes Rendus Mathematique*, 337(9):619–624, 2003.