Proper Generalized Decomposition based dynamic data driven inverse identification

David Gonzáles, Françoise Masson, Fabien Poulhaon, Adrien Leygue, Elías Cueto, Francisco Chinesta

Proper Generalized Decomposition based dynamic data driven inverse identification

D. González[a], F. Masson[a,b], F. Poulhaon[b], A. Leygue[b], E. Cueto[*,a], F. Chinesta[b]

[a] *Aragón Institute of Engineering Research (I3A), Universidad de Zaragoza, Spain.*
[b] *GeM UMR CNRS-ECN École Centrale de Nantes, France.*

Abstract

Dynamic Data-Driven Application Systems —DDDAS— appear as a new paradigm in the field of applied sciences and engineering, and in particular in simulation-based engineering sciences. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of systems and processes. One essential feature of DDDAS is the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically control the measurement process. DDDAS need accurate and fast simulation tools using if possible off-line computations to limit as much as possible the on-line computations. With this aim, efficient solvers can be constructed by introducing all the sources of variability as extra-coordinates in order to solve the model off-line only once. This way, its most general solution is obtained and therefore it can be then considered in on-line purposes. So to speak, we introduce a physics-based meta-modeling technique without the need for prior computer experiments. However, such models,

[☆]This work has been partially supported by the Spanish Ministry of Science and Innovation, through grant number CICYT-DPI2011-27778-C02-01.

[*]Corresponding author

Email addresses: gonzal@unizar.es (D. González), francoise.retat@ec-nantes.fr (F. Masson), Fabien.Poulhaon@eleves.ec-nantes.fr (F. Poulhaon), Adrien.Leygue@ec-nantes.fr (A. Leygue), ecueto@unizar.es (E. Cueto), Francisco.Chinesta@ec-nantes.fr (F. Chinesta)

that must be solved off-line, are defined in highly multidimensional spaces suffering the so-called curse of dimensionality. We proposed recently a technique, the Proper Generalized Decomposition —PGD—, able to circumvent the redoubtable curse of dimensionality. The marriage of DDDAS concepts and tools and PGD off-line computations could open unimaginable possibilities in the field of dynamic data-driven application systems. In this work we explore some possibilities in the context of on-line parameter estimation.

Contents

1. Introduction

1.1. Dynamic Data-Driven Application Systems —DDDAS

Traditionally, Simulation-based Engineering Sciences —SBES— relied on the use of static data inputs to perform the simulations. These data could

be parameters of the model(s) or boundary conditions, outputs at different time instants, etc., traditionally obtained through experiments. The word static is intended to mean here that these data could not be modified during the simulation.

A new paradigm in the field of Applied Sciences and Engineering has emerged in the last decade. Dynamic Data-Driven Application Systems (DDDAS) constitute nowadays one of the most challenging applications of simulation-based Engineering Sciences. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of simulations. As defined by the U.S. National Science Foundation, "DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process" [6].

The term Dynamic Data-Driven Application System was coined by F. Darema in a NSF workshop on the topic in 2000. The document that initially put forth this initiative stated that DDDAS constitute application simulations that can dynamically accept and respond to *online* field data and measurements and/or control such measurements [7]. DDDAS include different constituent blocks [20]:

1. A set of (possibly) heterogeneous simulation models.
2. A system to handle data obtained from both static and dynamic sources.
3. Algorithms to efficiently predict system behavior by solving the models under the restrictions set by the data.
4. Software infrastructure to integrate the data, model predictions, control algorithms, etc.

Almost a decade after the establishment of the concept, the importance of the challenge is now better appreciated. As can be noticed, it deals with very different and transversal disciplines: from simulation techniques, numerical issues, control, modeling, software engineering, data management and telecommunications, among others. In a nutshell, the characteristics of DDDAS could be depicted as in Fig. 1. It identifies three different blocks of interactions: (i) the one between human systems and the simulation, (ii) the simulation interaction with the physical system and (iii) the simulation and the hardware/ data infrastructure. Physical systems operate at very different time scales. In the picture (adapted from the NSF workshop on DDDAS 2000 final report [6]) we have included some representative values: from 10^{-20} Hz

for cosmological systems to 10^{20} Hz for problems at the atomic scale. Humans, however, can be considered as systems operating at rates from 3 Hz to 500 Hz in haptic[2] devices for instance to transmit realistic touch sensations. A crucial aspect of DDDAS is that of real-time simulation. This means that the simulations must run at the same time (or faster) than the data are collected. While this is not always true (as in weather forecasting, for instance, where collected data are not usually incorporated to the simulations), most applications require different forms of real-time simulations. In haptic surgery simulators, for instance, the simulation result, i.e., forces acting on the surgical tool, must be translated to the peripheral device at a rate of 500 Hz, which is the frequency of the free hand oscillation. In other applications, such as some manufacturing processes, the time scales are much bigger, and therefore real-time simulations can last for seconds or minutes.

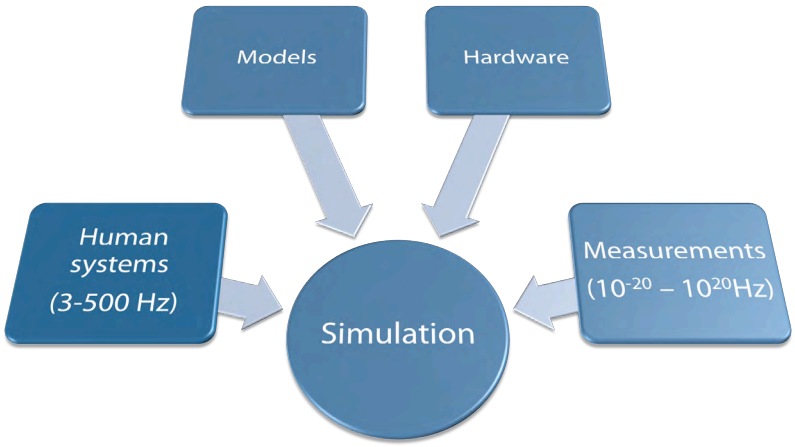


Figure 1: Structure of DDDAS and typical feedback rate among different constituents (Adapted from [6]).

As can be noticed from the introduction above, DDDAS can revolutionize

[2]Haptic devices are those able to translate to the user force reactions, such as, for instance, those used in surgical training.

the way in which simulation will be done in the next decades. No longer a single run of a simulation will be considered as a means of validating a design on the basis of a static data set.

The importance of DDDAS in the forthcoming decades can be noticed from the NSF Blue Ribbon Panel on Simulation Based Engineering Sciences report, that in 2006 included DDDAS as one of the five core issues or challenges in the field for the next decade (together with multiscale simulation, model validation and verification, handling large data and visualization).

While research on DDDAS should involve applications, mathematical and statistical algorithms, measurement systems, and computer systems software methods, our work focuses on the development of mathematical and statistical algorithms for the simulation within the framework of such a system. In brief, we intend to incorporate a new generation of simulation techniques into the field, allowing to perform faster simulations, able to cope with uncertainty, multiscale phenomena, inverse problems and many other features that will be discussed. This new generation of simulation techniques has received the name of Proper Generalized Decompositions —PGD— and has received an increasing level of attention by the SBES community. PGD was initially introduced for addressing multidimensional models encountered in science and engineering and was then extended to address general computational mechanics models.

1.2. Routes for circumventing the curse of dimensionality

Different techniques have been proposed for circumventing the curse of dimensionality[3] associated with high-dimensional models in science and engineering. Monte Carlo simulation is probably the most widely used. Its main drawback is the statistical noise, when magnitudes distinct to the moments of the distribution functions are computed. Another possibility lies in the use of sparse grids [8], within the deterministic framework, but they suffer also when the dimension of the space increases beyond a certain value (of about 20 dimensions).

To our knowledge there are few precedents of deterministic techniques able to circumvent efficiently the curse of dimensionality in highly multidimensional spaces. Hartree-Fock-based approaches are widely employed in

[3]The term *curse of dimensionality* refers to the exponential increase in number of degrees of freedom suffered by models defined in high-dimensional spaces being solved by mesh-based techniques like finite elements or volumes.

quantum chemistry [9].We proposed recently a technique based on a separated representation of the solution [11-12] that proceeds by expressing a generic multidimensional function $u(x_1, x_2, \ldots, x_N)$ as $u(x_1, x_2, \ldots, x_N) \approx \sum_{i=1}^{Q} F_i^1(x_1) \cdot \ldots \cdot F_i^N(x_N)$.

Remark 1. *In this expression the coordinates x_i denote any coordinate, scalar or vector, involving the physical space, the time or any other conformation coordinate (e.g. the conductivity in the example previously discussed).*

Thus, if M nodes are used to discretize each coordinate, the total number of unknowns involved in the solution is $Q \times N \times M$ instead of the M^N degrees of freedom involved in mesh based discretizations. We must recall that these functions are not "a priori" known, they are computed by introducing the separated representation approximation into the model's weak form and then solving the resulting non-linear problem. The interested reader can refer to [11] and the references therein for a detailed description of the numerical and algorithmic aspects. The construction of such approximation is called Proper Generalized Decomposition because this decomposition is not orthogonal but in many cases the number of terms in the finite sum is very close to the optimal decomposition obtained by applying the Proper Orthogonal Decomposition —POD- (or the Singular Value Decomposition —SVD-) on the model solution.

As can be noticed in the expression of the approximation separated representation the complexity scales linearly with the dimension of the space in which the model is defined, instead of the exponential growing characteristic of mesh based discretization strategies. In general, for many models, the number of terms Q in the finite sum is quite reduced (few tens). Thus, we can conclude about the generality of the separated representation, but its optimality depends on the solution features.

This kind of representation is not new. Similar approximations were considered many years ago in the context of Quantum Chemistry [9], the main difference lying in the constructor of the separated approximation. In the context of Computational Mechanics there is —to our knowledge— a unique precedent, the so-called radial approximation introduced by Pierre Ladeveze in the early eighties within the LATIN framework. At that time, Ladeveze looked for an efficient solver of non linear models, after noticing that in general these models involve a non-linear part that is local in the physical space, and a linear one (the one related to the structure equilibrium) that is obviously global, but linear. Thus, Pierre Ladeveze proposed a solution technique

for the decoupling of linear-global and non-linear-local problems. The time-dependent non-linearity related to complex thermomechanical behaviors is integrated locally at the integration points in the whole time interval. Then, with the behavior already computed at each position and time step, the equilibrium applies resulting in a global but linear problem defined in the whole time interval. To alleviate the calculation a separated space-time representation of the unknown fields is performed. Then the behavior is locally updated, and then the equilibrium. The process continues until reaching convergence, that allows verifying equilibrium and the constitutive equations.

This was the first ingredient of a wonderful and powerful numerical receipt, the so-called LATIN method. But another ingredient was needed for speeding up the solution of the space-time linear global model, and for this purpose, Ladeveze proposed expressing its solution in a space-time separated form [14], i.e.

$$u(\boldsymbol{x}, t) \approx \sum_{i=1}^{Q} X_i(\boldsymbol{x}) \cdot T_i(t).$$

If we look at the performance of such separated representations the outcome is for many models simply impressive as proved in Ladeveze's works [12] [14] [16] [15] as well as in [5]. If one considers a standard transient model defined in a 3D physical space, and if one considers P time steps, usual incremental strategies must solve P (in general non-linear) three-dimensional problems (do not forget that P can be millions!). However, if the radial approximation (space-time PGD) is considered, we should solve around $Q \cdot m$ 3D problems for computing the space functions $X_i(\boldsymbol{x})$ and $Q \cdot m$ 1D problems for computing the time functions $T_i(t)$ (m being the number of iterations needed for computing each term of the finite sum because of its non-linear nature). As $Q \cdot m \sim 100$ in many models the computing time savings can reach many orders of magnitude (millions and more!). The space-time separated decomposition has been extensively explored by Ladeveze's group in the context of multiscale analysis. The interested reader can refer to [14] and the references therein.

If we come back to the multidimensional models involving the physical space, the time, and a number of "exotic" extra-coordinates, the verdict is implacable: the multidimensional-PGD allows solving models never until now solved, suffering the so-called curse of dimensionality, and that were qualified many times as irresolvable. The PGD allows solving them, in some minutes, using a simple laptop!

The solution of physically multidimensional models encountered in quantum chemistry, the kinetic theory description of complex fluids and the chemical master equation were deeply described in some of our former works [4] [3] [10]. Parametric models where addressed in [22]. The thermal model considered in [17] contained 25 thermal parameters that were all considered as extra-coordinates. The coupling between local ODE and global PDE within a multiscale framework was analyzed in [21]. Finally, in [19], shape optimization was performed by assuming all the design parameters defining the domain geometry as extra-coordinates. Interested reader can refer to [11] and the references therein for a detailed description on the PGD. All these works opened unimaginable perspectives in the context of process and shape optimization and inverse identification that constitute a work in progress.

1.3. Structure of the paper

In this paper we focus on two different but intimately related problems, frequently appearing in the control of devices or industrial processes, but also in the control of epidemic diseases, for instance. These two problems are the identification of a parameter in a non-linear dynamical model from on-line measurements, on one hand, and the estimation of a boundary condition at a place where measurements are not possible from redundant information known in other parts of the domain boundary (the so-called Cauchy's problem). This scenario is found when controlling the temperature in the heat exchangers of nuclear plants for example, but it can also appear in many other industrial scenarios. We will consider the one-dimensional Cauchy problem where redundant boundary conditions—temperature and flux, namely— can be measured at one end of the interval, but the required boundary condition for the well-possessedness of the problem —temperature at the other end— is not available. Section 2 of the paper reviews these two problems and the associated PGD formulation. Section 3 shows the results for the parameter estimation problem, whereas section 4 includes the results for the estimation of boundary conditions of the Cauchy problem.

2. PGD based parametric solutions

In this paper we are only interested in addressing a new off-line dynamical system integration procedure that could be then used for on-line control purposes. For this reason at the present stage, we do not consider the inevitable noises related to model, control and measurements. Thus, in what follows,

the measurements corresponds to the solution of the dynamical systems for a given choice of model parameters, allowing circumventing controllability, observability of the inverse techniques issues.

2.1. Dynamic data-driven nonlinear parametric dynamical systems

In what follows we are focusing on a non-linear model usually encountered in the dynamic of populations and therefore in epidemiology modelling. In general the size of populations increases in direct relation to its size, but when this size becomes too large its growing rate decreases and sometimes inverses its tendency. The equation governing this kind of behaviour is known as the logistic equation, that writes:

$$\frac{du}{dt} = k \cdot u \cdot (u_\infty - u) \tag{1}$$

where $u = u(t)$, $t \in (0, t_{\max}]$ and whose initial condition is $u(t = 0) = u_g$ with $0 < u_g < u_\infty$.

The exact solution for a constant parameter k writes:

$$u(t) = \frac{u_\infty}{1 + \left(\frac{u_\infty}{u_g} - 1\right) e^{-k u_\infty t}}. \tag{2}$$

The parameter k is assumed unknown and taking values in the interval $k \in [k_{\min}, k_{\max}]$. The identification of the parameter is a key point in many applications as for example in the simulation and control of epidemic scenarios and also in the on-line adaptation of chemical kinetics governed by similar dynamical models. In order to identify such parameter different experimental measures are carried out at different times:

$$\begin{cases} \tilde{t}_1 \to \tilde{u}_1 \\ \tilde{t}_2 \to \tilde{u}_2 \\ \vdots \\ \tilde{t}_D \to \tilde{u}_D \end{cases} \tag{3}$$

The main challenge in the simulation and real-time control of the evolution of the field u is how to identify as fast as possible the best value of the parameter k assumed constant within each interval $(\tilde{t}_i, \tilde{t}_{i+1}]$, $i \in 1, \ldots, D-1$.

In what follows and without loss of generality we are assuming that intervals $(\tilde{t}_i, \tilde{t}_{i+1}]$ have the same length, that is, $\tilde{t}_{i+1} - \tilde{t}_i = \Delta$, $i = 1, \ldots, D-1$.

The unknown parameter is assumed taking values in the interval $[k_{\min}, k_{\max}]$ with $k_{\min} > 0$ and $k_{\max} > k_{\min}$. The most general solution allowing a complete knowledge of $u(t)$ under all possible circumstances lies in solving once and off-line, the problem:

Find $u(t, k, u^0)$ *such that*

$$\begin{cases} \frac{du}{dt} = k \cdot u \cdot (u_\infty - u), \;\; t \in (0, \Delta] \\ u(t=0) = u^0, \end{cases} \tag{4}$$

by assuming the initial condition u^0 and the model parameter k as extra-coordinates. Thus one should compute the three-dimensional solution $u(t, k, u^0)$, with $t \in (0, \Delta]$, $k \in [k_{\min}, k_{\max}]$ and $u^0 \in [u_g, u_\infty]$.

Remark 2. *The off-line computation of* $u(t, k)$*, with* $t \in (0, t_{max}]$*,* $k \in [k_{min}, k_{max}]$*, suffices only when the model parameter* k *is constant in* $(0, t_{max}]$*. We will prove this fact later on. Thus a 3D modeling involving both the model parameter and the initial condition as extra-coordinates is compulsory.*

The simplest way of solving Eq. (4) consists of defining discrete values of all the model coordinates: $(t_1, t_2, \ldots, t_P)$, $(k_1, k_2, \ldots, k_N)$ and, finally, $(u_1^0, u_2^0, \ldots, u_M^0)$. Without loss of generality we assume such discrete values uniformly distributed, i.e. $t_{i+1} - t_i = h_t$, $i = 1, \ldots, P-1$; $k_{i+1} - k_i = h_k$, with $i = 1, \ldots, N-1$ and, finally, $u_{i+1}^0 - u_i^0 = h_u$, with $i = 1, \ldots, M-1$. Now, Eq. (4) can be integrated by using a finite difference schema (the limitations of such discretization schema will be addressed later). We are considering the simplest strategy, a forward Euler schema, but more accurate schemes are available. Because Eq. (4) does not involve derivatives with respect to the extra-coordinates (the model parameter and the initial condition), the finite differences integration of Eq. (4) reduces to the solution of the $N \times M$ uncoupled time dependent ordinary differential equations:

$$\begin{cases} \frac{du^{r,s}}{dt} = k_r \cdot u^{r,s}(t) \cdot (u_\infty - u^{r,s}), \;\; t \in (0, \Delta] \\ u(t=0) = u_s^0, \end{cases}, \tag{5}$$

$\forall r, s \in [1, \ldots, N] \times [1, \ldots, M]$, and where $u^{r,s}(t) \equiv u(t; k_r, u_s^0)$. The finite differences integration of Eq. (5) by using a first order explicit schema writes: $\forall r, s, \in [1, \ldots, N] \times [1, \ldots, M]$,

$$\begin{cases} u_1^{r,s} = u_s^0 \\ u_i^{r,s} = u_{i-1}^{r,s} + h_t \cdot k_r \cdot u_{i-1}^{r,s} \cdot (u_\infty - u_{i-1}^{r,s}), \forall i \in [2, \ldots, P], \end{cases} \tag{6}$$

that allows to find $u_i^{r,s} = u(t_i, k_r, u_s^0) = u((i-1)\cdot h_t, k_{\min} + (r-1)\cdot h_k, u_g + (s-1)\cdot h_u)$.

We illustrate now the parameter identification from the only knowledge of $u_i^{r,s}$. We consider the restriction of $u(t, k, u^0)$ to $u(t = \Delta = \tilde{t}_1, k, u^0 = u_g)$ and we look for the intersection of such a solution with the experimental value $\tilde{u}_1$ that determines the best value of the parameter k in the interval $(0, \tilde{t}_1]$, $\tilde{k}_1$, as illustrated in Fig. 2.

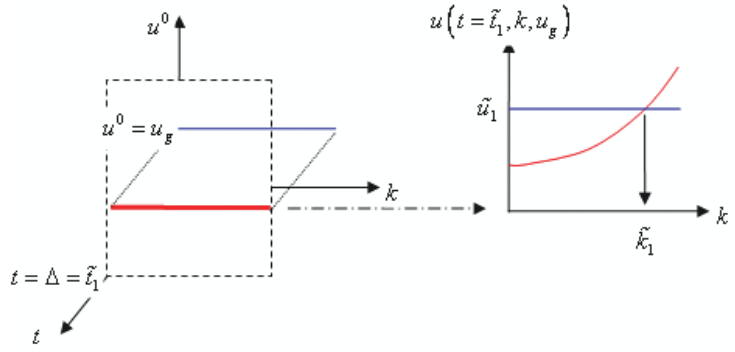


Figure 2: First step of the parameter k identification procedure.

Now, we consider the restriction of $u(t, k, u^0)$ to $u(t = \Delta, k, u^0 = \tilde{u}_1)$ and we look for the intersection of this solution with the experimental value $\tilde{u}_2$, point that determines the best value of the parameter k in the interval $[\tilde{t}_1, \tilde{t}_2]$, $\tilde{k}_2$, as shown in Fig. 3.

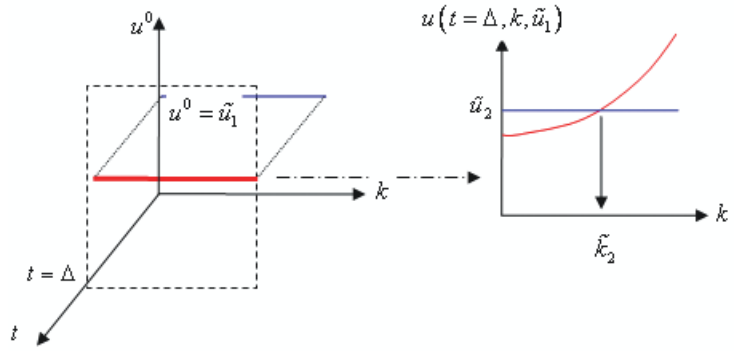


Figure 3: Second step of the parameter identification procedure.

We can notice that the integration in the whole time domain is performed from the only knowledge of $u_i^{r,s}$, despite the eventual variability of the model

parameter, that is computed once and off-line. Because we introduced as extra-coordinate the initial condition u_0 the integration can be adapted dynamically on-line to the experimental data coming from sensors, by adapting the value of the model parameter for reaching the recorded value, value that is considered as initial condition for the integration in the next time interval Δ.

As noticed previously, if instead of computing $u(t, k, u^0)$, with $t \in (0, \Delta]$, one computes $u(t, k)$ with $t \in (0, t_{\max}]$, from the given initial condition u_g, the value of k could be identified as soon as an experimental data is available, or taking the best compromise for fitting at the best all the experimental data available. However taking into account the change of the model parameter is not possible, and it implies the on-line integration of the model $u(t, k)$ in each interval $t \in (\tilde{t}_i, \tilde{t}_{i+1})$ taking the available data $\tilde{u}_i$ as initial condition. The drawback of such a procedure is the necessity to perform on-line integrations, fact that limits seriously its applicability in real time control and therefore in DDDAS.

Remark 3. *If the time step between two successive measurements Δ is constant we can easily ensure that the last integration time step P coincides with Δ . However, the data $\tilde{u}_i$ rarely coincides with a value of u^0_s. Different possibilities exist:*

1. *consider the nearest u^0_s to $\tilde{u}_i$;*
2. *consider the two neighbor values u^0_{s+1} and u^0_s allowing to define an interval to which $\tilde{k}_{i+1}$ belongs*
3. *consider the interpolated solution from $u(t = \Delta, k, u^0_{s+1})$ and $u(t = \Delta, k, u^0_s)$.*

In what follows we are considering the last alternative.

2.1.1. Proper Generalized Decomposition of Dynamical Systems

From now on we describe the application of Proper Generalized Decompositions to the simulation of dynamical systems governed by dynamical data. The method introduced above is obviously applicable by using any discretization techniques, such as finite differences, for instance. Indeed, it should work very well for the example shown in the next section. But if the number of dimensions of the model (parameters, boundary conditions, etc., considered as new dimensions of the problem, as explained before) increases, the only way to deal with them seems to be the PGD method here presented. We

keep, however, the number of dimensions of the problem restricted to three in order to make the development more clear, without loss of generality:

$$\begin{cases} \frac{du}{dt} = k \cdot u \cdot (u_\infty - u) \\ u(t=0) = u^0 \end{cases} \tag{7}$$

with $t \in I = (0, \Delta]$, $k \in \aleph = [k_{\min}, k_{\max}]$ and $u^0 \in \Im = [u_g, u_\infty]$.

As mentioned before, we intend to introduce k as well as the initial condition u^0 as additional coordinates in the problem. To introduce the initial condition in the ODE the following change of variables is suggested:

$$\hat{u} = u - u^0, \tag{8}$$

which, once introduced in Eq. (7) gives rise to

$$\begin{cases} \frac{d\hat{u}}{dt} = k(\hat{u} + u^0)(u_\infty - \hat{u} - u^0), \quad t \in (0, \Delta] \\ \hat{u}(t=0) = 0. \end{cases} \tag{9}$$

As a fundamental characteristic of PGD, it seeks the solution to Eq. (9) in the form:

$$\hat{u} \approx \sum_{i=1}^{Q} T_i(t) \cdot K_i(k) \cdot U_i(u^0). \tag{10}$$

Since the just made proper generalized decomposition constitutes an *a priori* method, i.e., it does not assume any particular form of the functions T_i, K_i or U_i, these must be generated during the execution of the proposed method. This gives rise to a non-linear problem that is described next. Within the iterative process inherent to non-linear problems, we assume that the approximation at iteration $n < Q$ is already achieved

$$\hat{u}^n = \sum_{i=1}^{n} T_i(t) \cdot K_i(k) \cdot U_i(u^0), \tag{11}$$

such that in the current iteration the functional product $T_{n+1}(t) \cdot K_{n+1}(k) \cdot U_{n+1}(u^0)$ is sought. With an eye towards the simplification of the notation, we will denote this product as $T_{n+1}(t) \cdot K_{n+1}(k) \cdot U_{n+1}(u^0) = R(t) \cdot S(k) \cdot W(u^0)$. Prior to solving this non-linear problem, a linearisation is mandatory. The simplest choice consists in a fixed-point iterative scheme along alternated directions. This scheme assumes $S(k)$ and $W(u^0)$ known and proceeds through

the determination of $R(t)$. With this function $R(t)$ just computed and with the previous $W(u^0)$ it proceeds by calculating $S(k)$ and, finally, with the just obtained $R(t)$ and $S(k)$, it determines $W(u^0)$. This procedure continues until convergence. Functions thus computed constitute the next term in the approximation, $T_{n+1}(t) = R(t)$, $K_{n+1}(k) = S(k)$ and $U_{n+1} = W(u^0)$.

All the mentioned steps are detailed in what follows:

Computation of $R(t)$ from $S(k)$ and $W(u^0)$. Consider now the linearized weak form of the problem (9) that consists of considering all the non-linear terms at the previous iteration, i.e. $\hat{u}^n$:

$$\int_{I\times\aleph\times\beth} \hat{u}^* \left(\frac{\partial \hat{u}}{\partial t} - k(\hat{u}^n + u^0)(u_\infty - u^0 - \hat{u}) \right) dt\, dk\, du^0 = 0, \qquad (12)$$

in which the trial and test functions, respectively, are

$$\hat{u}(t,k,u^0) = \sum_{i=1}^{n} T_i(t)\cdot K_i(k)\cdot U_i(u^0) + R(t)\cdot S(k)\cdot W(u^0) \qquad (13)$$

and

$$\hat{u}^*(t,k,u^0) = R^*(t)\cdot S(k)\cdot W(u^0) \qquad (14)$$

Introducing (13) and (14) in the weak form (12) we obtain

$$\int_{I\times\aleph\times\beth} R^*\cdot S\cdot W \left(\frac{dR}{dt}\cdot S\cdot W + k\cdot \left(\sum_{i=1}^{n} T_i\cdot K_i\cdot U_i + u^0 \right) R\cdot S\cdot W \right) dt\, dk\, du^0 =$$

$$= -\int_{I\times\aleph\times\beth} R^*\cdot S\cdot W \left(\sum_{i=1}^{n} \frac{dT_i}{dt}\cdot K_i\cdot U_i + \right.$$

$$\left. + k \left(\sum_{i=1}^{n} T_i\cdot K_i\cdot U_i + u^0 \right) \left(\sum_{i=1}^{n} T_i\cdot K_i\cdot U_i - u_\infty + u^0 \right) \right) dt\, dk\, du^0, \qquad (15)$$

Since all the functions depending on the parametric coordinate k and the initial condition u^0 are known, it is possible to perform the numerical integration of these functions along their respective domains $\aleph \times \beth$.

Eq. (15), after integration in $\aleph \times \beth$, represents the weak form of an ODE that defines the temporal evolution of the field $R(t)$. It could be solved by

any available discretization technique (SU, discontinuous Galerkin, ...). The strong form of Eq. (15), after integrating it in $\aleph \times \Im$, will be of the type

$$w_1 \cdot s_1 \cdot \frac{dR(t)}{dt} + w_2 \cdot s_2 \cdot R(t) + \left(\sum_{i=1}^{n} \alpha_i \beta_i T_i(t) \right) R(t) = f(t), \tag{16}$$

with

$$\begin{cases} w_1 = \int_{\Im} W^2 du^0 \\ w_2 = \int_{\Im} u^0 W^2 du^0 \\ s_1 = \int_{\aleph} S^2 dk \\ s_2 = \int_{\aleph} k \cdot S^2 dk \\ \alpha_i = \int_{\Im} W^2 U_i du^0 \\ \beta_i = \int_{\aleph} S^2 K_i k dk \end{cases} \tag{17}$$

and $f(t)$ the function that results from integrating of the right hand member in the $\Im$ and $\aleph$ intervals.

Computation of $S(k)$ from $R(t)$ and $W(u^0)$. In this case, the weighting function will be

$$\hat{u}^*(t, k, u^0) = S^*(k) \cdot R(t) \cdot W(u^0). \tag{18}$$

This gives rise to a weak form

$$\begin{aligned} \int_{I \times \aleph \times \Im} S^* \cdot R \cdot W \cdot \left(\frac{dR}{dt} \cdot S \cdot W + k \cdot \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i + u^0 \right) R \cdot S \cdot W \right) dt\, dk\, du^0 = \\ = - \int_{I \times \aleph \times \Im} S^* \cdot R \cdot W \cdot \left(\sum_{i=1}^{n} \frac{dT_i}{dt} \cdot K_i \cdot U_i + \right. \\ \left. + k \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i + u^0 \right) \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i - u_\infty + u^0 \right) \right) dt\, dk\, du^0, \end{aligned} \tag{19}$$

The neat difference between the problem defined by Eq (19) and that in Eq. (15) is that now no differential operator is involved. Thus the strong form resulting from the problem reads:

$$r_2 \cdot w_1 \cdot S(k) + r_1 \cdot w_2 \cdot k \cdot S(k) + \left(\sum_{i=1}^{n} \gamma_i \alpha_i K_i(k) \right) k S(k) = g(k), \tag{20}$$

with

$$\begin{cases} r_1 = \int_I R^2 dt \\ r_2 = \int_I R \cdot \frac{dR}{dt} \\ \gamma_i = \int_I T_i R^2 dt \end{cases} \tag{21}$$

and $g(k)$ the function that results after integration of the right hand member in the I and $\Im$ intervals. This represents an algebraic equation. Note that the introduction of parameters as additional spatial coordinates in the problem does not have a relevant effect in the computational cost of the resulting PGD approximation but it could induce the necessity of considering much more terms in the finite sums decomposition..

Computation of $W(u^0)$ from $R(t)$ and $S(k)$. In this last case the weighting function results:

$$\hat{u}(t, k, u^0) = W^*(u^0) \cdot R(t) \cdot S(k). \tag{22}$$

and therefore we arrive at a problem whose weak form reads

$$\begin{aligned} \int_{I\times\aleph\times\Im} W^*\cdot R\cdot S\cdot \left(\frac{dR}{dt} \cdot S \cdot W + k \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i + u^0 \right) \cdot R \cdot S \cdot W \right) dt\, dk\, du^0 = \\ = - \int_{I\times\aleph\times\Im} W^* \cdot R \cdot S \cdot \left(\sum_{i=1}^{n} \frac{dT_i}{dt} \cdot K_i \cdot U_i + \right. \\ \left. + k \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i + u^0 \right) \left(\sum_{i=1}^{n} T_i \cdot K_i \cdot U_i - u_\infty + u^0 \right) \right) dt\, dk\, du^0 \end{aligned} \tag{23}$$

Eq. (23) does not involve any differential operator again. The resulting strong form reads

$$(r_2 \cdot s_1 + r_1 \cdot s_2 \cdot u^0) \cdot W(u^0) + \left(\sum_{i=1}^{n} \gamma_i \beta_i U_i(u^0) \right) W(u^0) = h(u^0), \tag{24}$$

with $h(u^0)$ the function that results after integration of the right hand member in the I and $\aleph$ intervals. This results, again, in an algebraic equation. Convergence of these PGD approximations has been proved recently [18] [2] [1] for symmetric elliptic operators. See also [12] [13] for more details on the state of the art for PGD approximations and their underlying theory.

2.2. Estimation of unknown boundary conditions

As mentioned before, the second problem that frequently arises in industrial control settings is that of estimating unknown boundary conditions (operating conditions of some devices at places where measurements can not be performed). We exemplify it here by solving the transient heat equation on a 1D domain where redundant boundary conditions, temperature and thermal flux are known on one side, both fields in the other boundary being unknown. This problem is usually encountered in the control of thermal exchangers when it is not possible to introduce thermocouples inside a pipe (e.g. exchangers in nuclear plants) and the temperature of the fluid flowing inside must be inferred from redundant information known in the external pipe surface where temperature and heat flux can be experimentally measured.

For control purposes one should estimate accurately and in real time the unknown temperature, the one existing on the internal pipe surface in contact with the flowing fluid. One should identify temperature fluctuations occurring suddenly in order to take the pertinent control decisions. Thus, one needs accurate temperature estimations and these estimations should be obtained in real time.

This scenario defines an inverse problem that can be solved by applying a variety of well established techniques. Thus, at each time step an arbitrary trial temperature could be enforced at the internal boundary, then solving the heat equation from both boundary temperatures, one really measured and the other estimated, and then computing the heat flux in the boundary where it is known. By comparing the computed and experimentally measured heat fluxes the unknown boundary temperature could be updated by using an appropriate technique trying to minimize the difference between both fluxes. As soon as the unknown temperature is updated the thermal problem can be solved again and the procedure repeated until reaching the convergence, i.e. until obtaining a gap lower than a small enough value. After reaching convergence, one can move to the next time step in which the just described iteration procedure should be repeated. Other more sophisticated strategies exist, and even if there are no major conceptual difficulties, the real time constraint is not easy to address.

In what follows and using a similar strategy to the one that was used in the previous sections, we are describing a procedure able to address the identification inverse problem accurately and in real time. For the sake of clarity we consider a linear thermal model defined in a one-dimensional domain, however, the procedure that we propose can be easily extended for

considering more realistic scenarios, non linear behaviours in two or three dimensional geometries that constitutes a work in progress.

We consider the linear transient heat equation

$$\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = 0 \text{ in } \ (0, L) \times (0, t_{\max}] \tag{25}$$

with boundary conditions

$$\begin{cases} u(x,0) = u^0(x) \\ u(0,t) = u_g(t) \\ \frac{\partial u}{\partial x}(0,t) = q_g(t) \end{cases} \tag{26}$$

Actually, we want to perform inverse identification. Given a set of temperature values $\left\{u_g^i\right\}_{i=1}^D$ and heat fluxes $\left\{q_g^i\right\}_{i=1}^D$, coming from measurement devices placed on the left domain boundary $x = 0$, we must estimate the time evolution of the temperature at the right boundary $x = L$, denoted by $\theta(t)$, at least at the same times $\{\theta^i\}_{i=1}^D$. We assume that the measurements are performed at each time interval Δ.

As previously explained the Proper Generalized Decomposition allows computing a multidimensional solution of the thermal model where initial and boundary conditions could be considered as extra-coordinates. If the initial condition is parametrized by using a piece-wise linear finite element approximation defined from p nodes uniformly distributed in the interval $[0, L]$, $x_i,\ i = 1, \ldots, p$, the separated representation of the unknown temperature field reads:

$$u\left(x, t, u_g, u_2^0, \ldots, u_{p-1}^0, \theta\right) \approx \sum_{i=1}^{Q} X_i(x) \cdot T_i(t) \cdot U_i^1(u_g) \cdot U_i^2(u_2^0) \cdot \ldots \cdot U_i^{p-1}(u_{p-1}^0) \cdot U_i^p(\theta) \tag{27}$$

where $x \in (0, L),\ t \in (0, \Delta],\ u_i^0 \in [u_{min}, u_{max}],\ i = 2, \ldots, p-1,\ u_g \in [u_{min}, u_{max}]$ and $\theta \in [u_{min}, u_{max}]$.

This separated representation can be constructed by following the procedure previously described. As soon as it is available, the space derivative

involved in the heat flux can be computed

$$\frac{\partial u\left(x,t,u_g,u_2^0,\cdots,u_{p-1}^0,\theta\right)}{\partial x}\approx \sum_{i=1}^{Q}\frac{dX_i(x)}{dx}\cdot T_i(t)\cdot U_i^1(u_g)\cdot U_i^2(u_2^0)\ldots U_i^{p-1}(u_{p-1}^0)\cdot U_i^p(\theta) \quad (28)$$

If we assume known the temperature field at time $t = n\cdot\Delta$, at the following time step $n+1$, the temperature θ^{n+1} must be computed from the information known at $x=0$, u_g^{n+1} and q_g^{n+1}, and the temperature field at the previous time step. For this purpose we consider:

$$\left.\frac{\partial u\left(x,t,u_g,u_2^0,\cdots,u_{p-1}^0,\theta\right)}{\partial x}\right|_{\substack{x=0,t=\Delta,u_g=u_g^{n+1},u_2^0=u^n(x_2),\ldots,\\ u_{p-1}^0=u^n(x_{p-1}),\theta=\theta^{n+1}}} = q_g^{n+1} \quad (29)$$

that represents a single equation for determining the single unknown θ^{n+1}. In this equation $u^n(x_k)$, $k=2,\ldots,p-1$, comes from the final values at the previous time step according to:

$$u^n(x_k) = u(x=x_k,t=\Delta,u_g=u_g^n,u_2^0=u^{n-1}(x_2),\ldots,u_{p-1}^0=u^{n-1}(x_{p-1}),\theta=\theta^n) \quad (30)$$

where similar expressions apply for $u^{n-1}(x_k)$, $k=2,\ldots,p-1$, and so on.

In the previous analysis we make an important assumption, the one that assumes that the boundary conditions u_g^i and q_g^i are constant in the interval $[(i-1)\cdot\Delta, i\cdot\Delta]$, $i=2,\ldots,D$. Other possibilities exist, being the simplest one a linear evolution between each two consecutive measurements. In that case the separated representation of the temperature field reads $u\left(x,t,u_g,u_1^0,u_2^0,\ldots,u_{p-1}^0,u_p^0,\theta\right)$, where $x\in(0,L)$, $t\in(0,\Delta]$, $u_i^0\in[u_{min},u_{max}]$, $i=1,\ldots,p$, $u_g\in[u_{min},u_{max}]$ and $\theta\in[u_{min},u_{max}]$. After constructing the multidimensional separated representation the identification is performed according to:

$$\left.\frac{\partial u\left(x,t,u_g,u_1^0,u_2^0,\ldots,u_{p-1}^0,u_p^0,\theta\right)}{\partial x}\right|_{\substack{x=0,t=\Delta,u_g=u_g^{n+1},u_1^0=u_g^n,u_2^0=u^n(x_2),\ldots,\\ u_{p-1}^0=u^n(x_{p-1}),u_p^0=\theta^n,\theta=\theta^{n+1}}} = q_g^{n+1} \quad (31)$$

that represents again a single equation for determining the single unknown θ^{n+1}.

3. Numerical example: Dynamic Data Driven non-linear parametric dynamical system

In order to illustrate the just described procedure we are considering the dynamical model given by Eq. (1) where $u_\infty = 1$, $u_g = 0.2$, $t \in (0, 10]$ and the unknown model parameter taking values in the interval $k \in [0, 3]$.

The field u is measured at times $\tilde{t}_i = i \cdot \Delta$, $i = 1, \ldots, 10$ and $\Delta = 1$. Linear finite element meshes were chosen along each coordinate, with 200, 60 and 160 elements, respectively. We are considering two scenarios.

3.1. A first scenario

The first scenario concerns experimental measurements coming from the exact solution of Eq. (1) with $k = 0.7$ (the model parameter is assumed constant in the whole simulation time interval $(0, 10]$). In that case the measured values at times $\tilde{t} = 1, 2, \ldots, 10$ are given by $\tilde{u} = 0.335$, 0.503, 0.671, 0.804, 0.892, 0.943, 0.971, 0.985, 0.993, 0.996.

Figure 4 depicts the restriction of $u(t, k, u^0)$ at time $t = \Delta$, i.e. $u(t = \Delta, k, u^0)$. The numerical solution (right) shows the good accuracy obtained by PGD method by employing only one product of functions. This error (L2-norm) is $1.45 \cdot 10^{-4}$ for the mentioned single product of functions, shown in Fig. 5.

Remark 4. *It is well known that the forward Euler, finite difference discretization of the Verhulst logistic Eq. (1) gives rise to a recurrence relation very much like the logistic map. As it is well known, the logistic map presents chaotic behavior for values of the product $k \cdot \Delta$ bigger than 3.57, showing oscillations dependent of the initial condition for $k > 2$. In the simulations that follow chaotic or oscillatory behavior is avoided by a judicious choice of the time discretization.*

Figure 6 depicts the restriction of solution $u(t = \Delta, k, u_0)$ shown in Fig. 4 to the given initial condition $u^0 = u_g = 0.2$, that is: $u(t = \Delta, k, u^0 = 0.2)$. Since such value $u^0 = u_g$ does not coincide with a discrete value of the initial condition axis we consider the interpolated curve $u(t = \Delta, k, u_g) = \sum_{i=1}^{n} T_i(\Delta) \cdot K_i(k) \cdot U_i(u_g)$.

The resulting interpolated solution is depicted in cyan in Fig. 6 whereas the reference solution, $\tilde{u}_1 = 0.335$ is depicted in red. The intersection point of both curves defines the optimal model parameter k, $k = 0.701$, that in

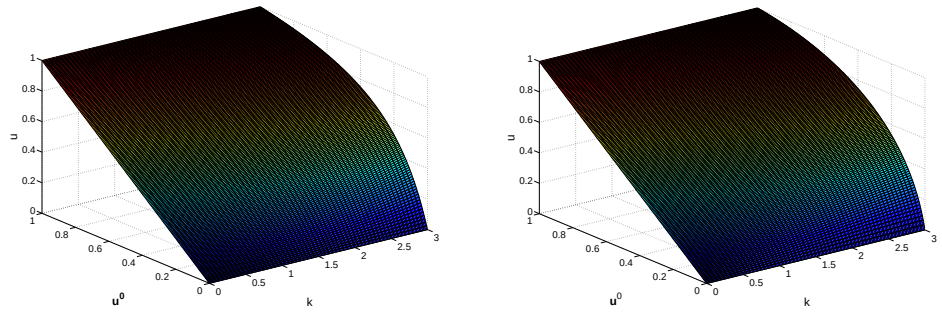


Figure 4: Solution $u(t = \Delta, k, u^0)$. Left, exact solution. Right, numerical one obtained by applying PGD.

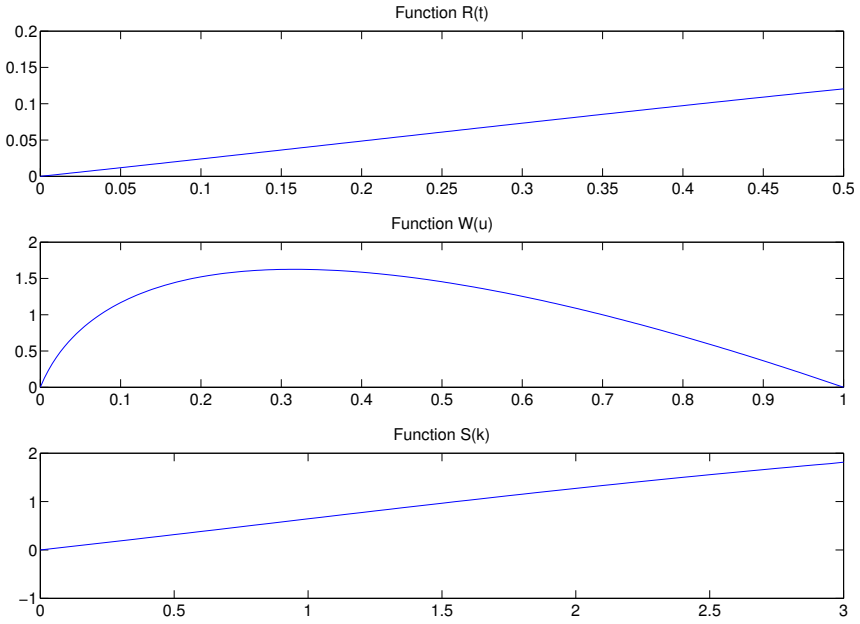


Figure 5: Functions employed for the numerical solution of the problem.

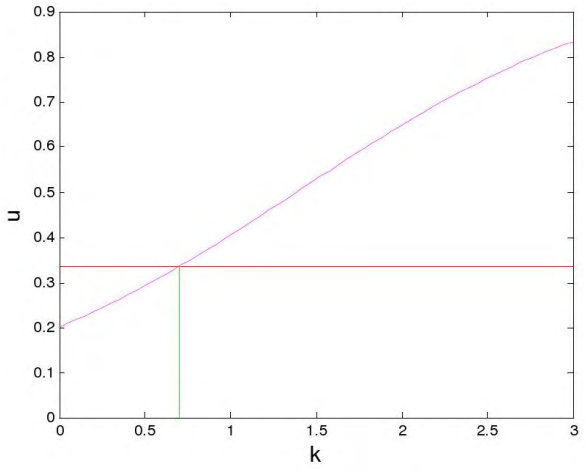


Figure 6: Solution $u(k) = u(t = \Delta, k, u^0 = 0.2)$.

this case is very close to the exact one $k = 0.7$ that served for calculating the different couples $(\tilde{t}_i, \tilde{u}_i)$. The procedure continues as described in the previous section. Table 1 groups the identified parameters.

3.2. A second scenario with random variation of the parameter

In what follows we are considering a second scenario slightly different. In the present case the parameter k is defined randomly in each time interval , according to the expression: $(\tilde{t}_i, \tilde{t}_{i+1})$, $i = 1, \ldots, D = 10$, with

$$k^{\text{exact}} = \varepsilon \cdot \mathcal{U}, \tag{32}$$

where $\varepsilon = 3$ and $\mathcal{U}$ denotes a uniform random variable defined in the interval $[0, 1]$. Table 2 shows the identified parameters. We can notice the high accuracy obtained by employing the on-line parameter identification procedure previously described, that only needed a single off-line solution of a model defined in a higher dimensional space, that was efficiently solved by using the Proper Generalized Decomposition.

4. Numerical example: Dynamic Data Driven Cauchy problem

In order to evaluate the reliability of the strategy introduced in section 2, two test cases are considered. For the first one, the temperature on the right boundary evolves linearly in time whereas in the second scenario a discontinuous step is considered.

$\tilde{t}$	$\tilde{u}$	k^{exact}	$\tilde{k}_{\text{ident}}$
1	0.335	0.7	0.701
2	0.503	0.7	0.700
3	0.671	0.7	0.699
4	0.804	0.7	0.699
5	0.892	0.7	0.699
6	0.943	0.7	0.699
7	0.871	0.7	0.699
8	0.985	0.7	0.699
9	0.993	0.7	0.699
10	0.996	0.7	0.699

Table 1: Parameter estimation for $k^{\text{exact}} = 0.7$.

$\tilde{t}$	$\tilde{u}$	k^{exact}	$\tilde{k}_{\text{ident}}$
1	0.742	2.444	2.445
2	0.978	2.717	2.703
3	0.984	0.381	0.381
4	0.999	2.740	2.722
5	0.999	1.897	1.893
6	0.999	0.293	0.294
7	0.999	0.835	0.838
8	0.999	1.641	1.639
9	0.999	2.873	2.858
10	0.999	2.895	2.880

Table 2: Parameter estimation for a parameter evolving randomly

4.1. Identifying a linear evolution on the unknown boundary condition

Let us assume firstly an initial constant value of the temperature, followed by a linear evolution in time at $x = L$, that constitutes the sought boundary condition, whereas at $x = 0$ we consider a convection-type boundary condition that allows solving the equation by using a finite difference technique. After solving it, the temperature derivative is computed at $x = 0$. The values of the temperature and its derivatives at $x = 0$ each $\Delta = 1$ are retained while we try to identify the profile at $x = L$. Firstly, it is assumed that the boundary conditions u_g^{n+1} and q_g^{n+1} are constant within the interval $[n \cdot \Delta, (n+1) \cdot \Delta]$, $n = 1, \cdots, D-1$. In the numerical experiments that follow the following values have been considered: $k = 0.1$ and $L = 1$.

At each Δ the temperature on the right boundary, θ, is identified as previously described. As soon as this temperature is calculated the temperature for $t \in (t_n, t_{n+1}] = (n \cdot \Delta, (n+1) \cdot \Delta]$ reads

$$u\left(x, t, u_g^{n+1}, u^n(x_2), \ldots, u^n(x_{p-1}), \theta^{n+1}\right) \tag{33}$$

Obviously, if we particularize it on the right boundary we obtain:

$$u\left(x = L, t, u_g^{n+1}, u^n(x_2), \ldots, u^n(x_{p-1}), \theta^{n+1}\right) = \theta^{n+1}, \quad \forall t \in [t_n, t_{n+1}] \tag{34}$$

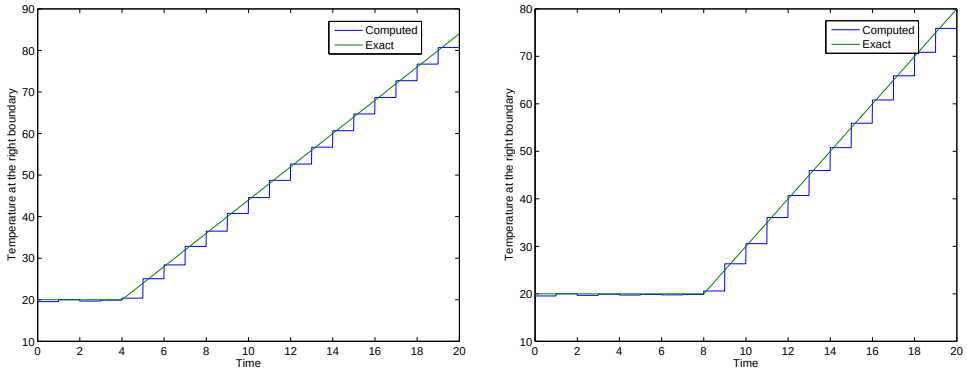
Thus one can expect a discontinuity of the identified temperature on the right boundary at each sampling time t_n, as noticed in all the solutions depicted in what follows.

Figure 7 depicts the identified evolution at $x = L$ that is compared with the exact one. We can appreciate the impact of the hypothesis of assuming constant boundary conditions within each time interval Δ for different values of the rate of increase of the temperature at $x = L$. The identified temperature follows the exact evolution but it underestimates the temperature values.

Obviously, by increasing the sampling frequency the accuracy is notably increased as Fig. 8 proves. The just referred issues disappear as soon as a piecewise linear evolution of the boundary conditions is assumed as illustrated in Fig. 9.

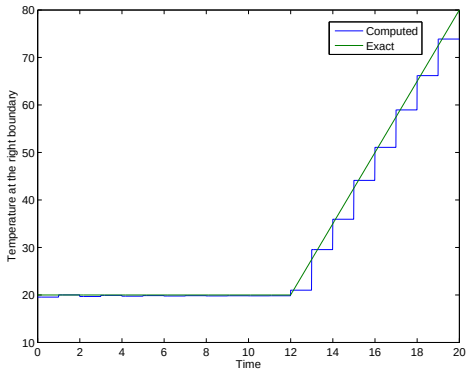
4.2. Identifying discontinuous evolutions in the unknown boundary condition

When considering a discontinuity in the temperature profile at the right boundary the accuracy of the identified temperature evolution could depend

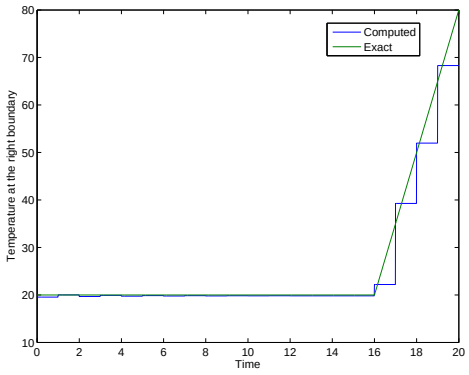


(a) slope = 4

(b) slope = 5

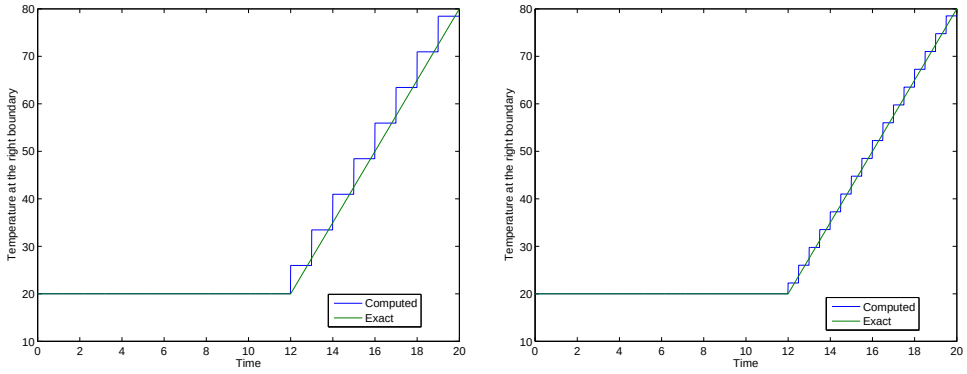


(c) slope = 7.5



(d) slope = 15

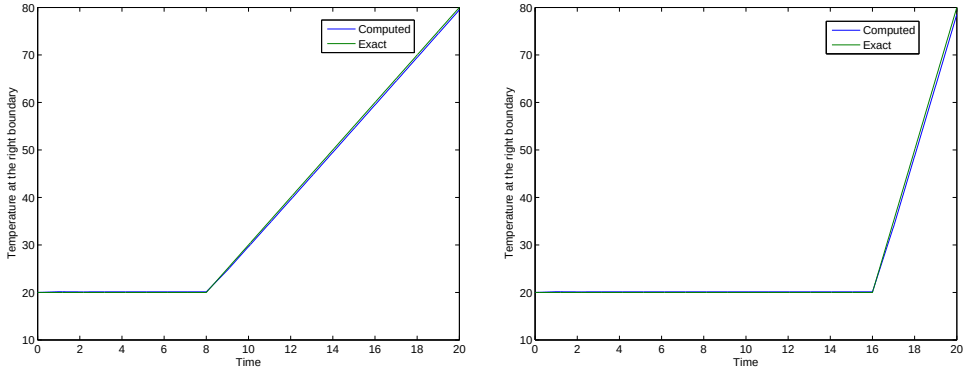
Figure 7: Identified solution for different rates of increase of the identified temperature



(a) f = 1Hz (b) f = 2Hz

(c) f = 5Hz (d) f = 10Hz

Figure 8: Influence of the sampling frequency.



(a) slope = 5 (b) slope = 15

Figure 9: Considering a linear evolution of the boundary conditions between two consecutive sampling times

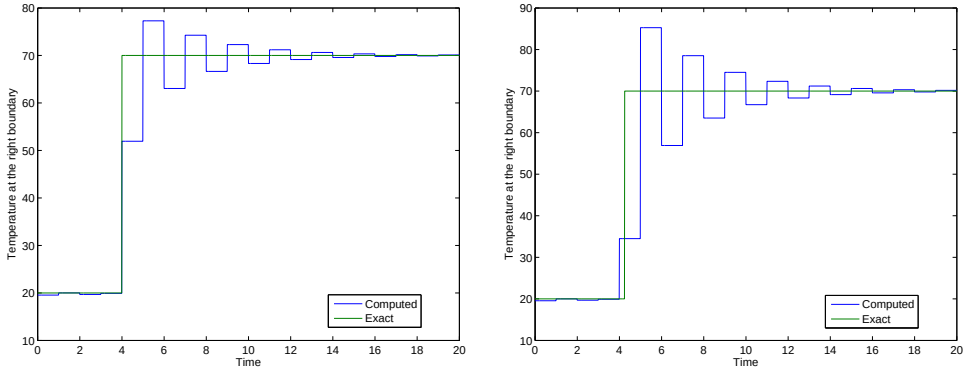
on several factors: (i) the material conductivity that controls the delay between the sudden increase of the temperature and its effects on the opposite boundary, (ii) the sampling frequency and (iii) the location of the discontinuity within the interval $T = \Delta$.

When considering a low diffusion coefficient the identification is quite poor because of the delay effects in the transfer of the information throughout the domain. The accuracy cannot be significantly improved by modifying the location of the discontinuity within the interval $T = \Delta$ as illustrated in Fig. 10 for $k = 0.1$.

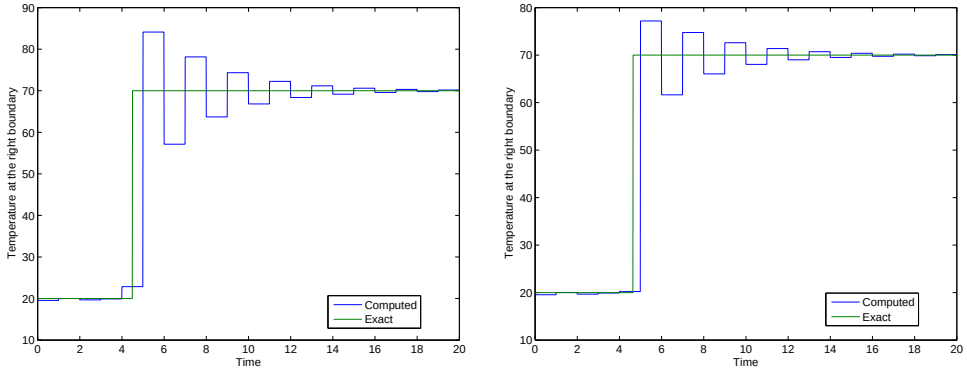
When increasing the thermal conductivity the identification is significantly improved and as expected the best identification is carried out when the discontinuity is located exactly in a sampling instant as shown in Fig. 11.

5. Conclusions

We have presented a method for efficient simulation in the context of DDDAS based upon the use of Proper Generalized Decompositions. The method is based on the construction of the solution for any value of the parameters considered in the equations. In this work two different, but frequently related, problems have been considered. These problems were the parameter estimation in a problem governed by dynamic equations and dynamic data, and the one of estimating boundary condition at places where

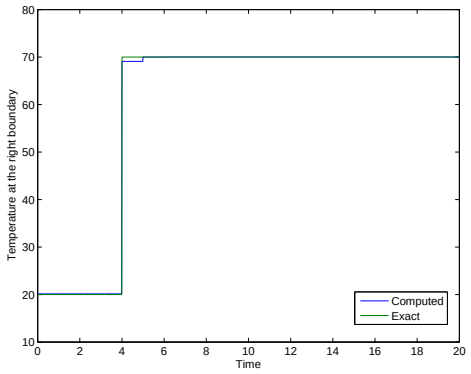


(a) 0 (b) $\frac{T}{4}$

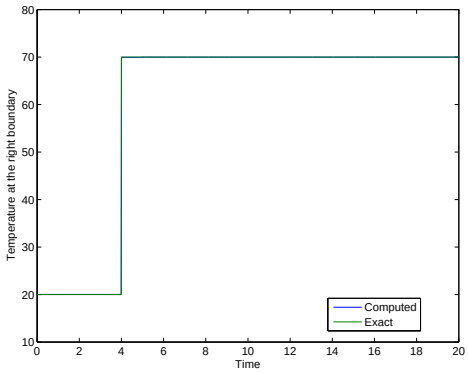


(c) $\frac{T}{2}$ (d) $\frac{3T}{4}$

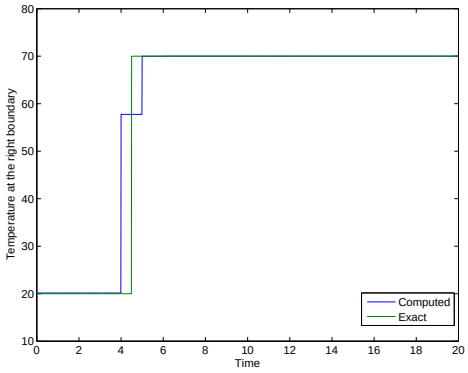
Figure 10: Identified solution for different positions of the discontinuity within the interval $T = \Delta$



(a) k = 0.5 / 0



(b) k = 1 / 0



(c) k = 0.5 / $\frac{T}{2}$

Figure 11: Influence of the conductivity and the location of the discontinuity

measurements are not possible. The method post-processes the solution in order to obtain on-line an accurate and fast solution of the problems, rather than simulating the evolution of the problem for any change in the parameters.

The results are still preliminary, but show encouraging properties in terms of accuracy and computational cost. The speed of calculation has not been deeply addressed yet, due to the very simple nature of the problems here considered, but previous applications of the authors in the field of DDDAS make us optimistic in this sense.

References

[1] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Computer Methods in Applied Mechanics and Engineering 199 (25-28) (2010) 1872 – 1880.
URL http://www.sciencedirect.com/science/article/pii/S0045782510000708

[2] A. Ammar, F. Chinesta, A. Falcó, On the convergence of a greedy rank-one update algorithm for a class of linear systems, Archives of Computational Methods in Engineering 17 (2010) 473–486, 10.1007/s11831-010-9048-z.
URL http://dx.doi.org/10.1007/s11831-010-9048-z

[3] A. Ammar, B. Mokdad, F. Chinesta, , R. Keunings., A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. part ii: transient simulation using space-time separated representations, J. Non-Newtonian Fluid Mech. 144 (2007) 98–121.

[4] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, J. Non-Newtonian Fluid Mech. 139 (2006) 153–176.

[5] A. Ammar, M. Normandin, F. Naim, D. González, E. Cueto, F. Chinesta, Non-incremental strategies based on separated representations: Applications in Computational Rheology, Communications in Mathematical Sciences 8 (3) (2010) 671–695.

[6] V. Authors, Final report. DDDAS workshop at Arlington, VA, Tech. rep., National Science Foundation (2000).

[7] V. Authors, Final report. DDDAS workshop at Arlington, VA, Tech. rep., National Science Foundation (2006).

[8] H.-J. Bungartz, M. Griebel, Sparse grids, Acta Numerica 13 (2004) 1–123.

[9] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. L. Bris, Y. Maday, Computational quantum chemistry: a primer, in: Handbook of Numerical Analysis, vol. X, 2003, pp. 3–270.

[10] F. Chinesta, A. Ammar, E. Cueto, On the use of Proper Generalized Decompositions for solving the multidimensional Chemical Master Equation, European Journal of Computational Mechanics 19 (2010) 53–64.

[11] F. Chinesta, A. Ammar, E. Cueto, Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models., Archives of Computational Methods in Engineering 17 (4) (2010) 327–350.

[12] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on Proper Generalized Decomposition., Archives of Computational Methods in Engineering 18 (2011) 395–404.

[13] D. González, A. Ammar, F. Chinesta, E. Cueto, Recent advances on the use of separated representations, International Journal for Numerical Methods in Engineering 81 (5) (2010) 637–659.
URL http://dx.doi.org/10.1002/nme.2710

[14] P. Ladeveze, Nonlinear Computational Structural Mechanics, Springer, N.Y., 1999.

[15] P. Ladeveze, J. Passieux, D. Neron, The LATIN multiscale computational method and the Proper Generalized Decomposition, COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 199 (21-22, SI) (2010) 1287–1296.

[16] P. Ladeveze, J.-C. Passieux, D. Neron, The latin multiscale computational method and the proper generalized decomposition, Computer

Methods in Applied Mechanics and Engineering 199 (21-22) (2010) 1287 – 1296.
URL http://www.sciencedirect.com/science/article/pii/S0045782509002643

[17] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin, F. Chinesta, Routes for the efficient non-concurrent homogenization of non-linear heterogeneous materials by using the proper generalized decomposition, Archives of Computational Methods in Engineering 17 (4) (2010) 373–391.

[18] C. Le Bris, T. Lelievre, Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, Constructive Approximation 30 (2009) 621–651, 10.1007/s00365-009-9071-1.
URL http://dx.doi.org/10.1007/s00365-009-9071-1

[19] A. Leygue, E. Verron, A first step towards the use of proper generalized decomposition methods for structural optimization, Archives of Computational Methods in Engineering 17 (4) (2010) 465–472.

[20] J. T. Oden, T. Belystchko, J. Fish, T. J. R. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, S. Yip, Simulation based Engineering Science: Revolutionizing Engineering Science through Simulation, Tech. rep., National Science Foundation blue ribbon panel on SBES (2006).

[21] E. Pruliere, A. Ammar, N. E. Kissi, F. Chinesta, Multiscale modelling of flows involving short fiber suspensions, Archives of Computational Methods in Engineering 16 (2009) 1–30.

[22] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of parametric models by using the proper generalized decomposition, Mathematics and Computer Simulation In press.