Proper Generalized Decomposition based dynamic data-driven control of thermal processes

Chady Ghnatios, Françoise Masson, Antonio Huerta, Adrien Leygue, Elías Cueto, Francisco Chinesta

Proper Generalized Decomposition based dynamic data-driven control of thermal processes ☆

Ch. Ghnatios [a], F. Masson [a], A. Huerta [b], A. Leygue [a], E. Cueto [c,*], F. Chinesta [a]

[a] EADS Corporate Foundation International Chair, GEM, UMR CNRS-Centrale Nantes, 1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France
[b] Laboratori de Calcul Numeric, Universitat Politècnica de Catalunya, Jordi Girona, 1 Campus Nord, C2, E-08034 Barcelona, Spain
[c] Aragon Institute of Engineering Research, Universidad de Zaragoza, María de Luna, 7, E-50018 Zaragoza, Spain

Dynamic Data-Driven Application Systems—DDDAS—appear as a new paradigm in the field of applied sci-ences and engineering, and in particular in Simulation-based Engineering Sciences. By DDDAS we mean a set of techniques that allow to link simulation tools with measurement devices for real-time control of systems and processes. In this paper a novel simulation technique is developed with an eye towards its employ in the field of DDDAS. The main novelty of this technique relies in the consideration of param-eters of the model as new dimensions in the parametric space. Such models often live in highly multidi-mensional spaces suffering the so-called curse of dimensionality. To avoid this problem related to mesh-based techniques, in this work an approach based upon the Proper Generalized Decomposition—PGD—is developed, which is able to circumvent the redoubtable curse of dimensionality. The approach thus developed is composed by a marriage of DDDAS concepts and a combination of PGD "off-line" com-putations, linked to "on-line" post-processing. In this work we explore some possibilities in the context of process control, malfunctioning identification and system reconfiguration in real time, showing the potentialities of the technique in real engineering contexts.

1. Introduction: 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 here to mean 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 SBES. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of simulations and applications. As defined by the US 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" [37].

The term Dynamic Data-Driven Application System was coined by Darema in a NSF workshop on the topic in 2000 [36]. 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. This synergistic and symbiotic feedback control loop among applications, simulations, and measurements is a novel technical direction that can open new domains in the capabilities of simulations with a high potential pay-off, and create applications with new and enhanced capabilities. It has the potential to transform the way science and engineering are done, and induces a major beneficial impact in the way many functions in our society are conducted, such as manufacturing, commerce, transportation, hazard prediction/management, and medicine, to name a few" [14].

The importance of DDDAS in the forthcoming decades can be noticed from the NSF Blue Ribbon Panel on SBES report [33], 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). This panel concluded that "Dynamic Data-Driven Application Systems will rewrite the book on the validation and

☆ This work has been partially supported by the Spanish Ministry of Science and Innovation, through grant number CICYT-DPI2011-27778-C02-01/02.

* Corresponding author.
E-mail addresses: antonio.huerta@upc.es (A. Huerta), ecueto@unizar.es (E. Cueto), Francisco.Chinesta@ec-nantes.fr (F. Chinesta).

verification of computer predictions" and that "research is needed to effectively use and integrate data-intensive computing systems, ubiquitous sensors and high-resolution detectors, imaging devices, and other data-gathering storage and distribution devices, and to develop methodologies and theoretical frameworks for their integration into simulation systems" [33]. Moreover, the NSF believes that "...The DDDAS community needs to reach a critical mass both in terms of numbers of investigators, and in terms of the depth, breadth and maturity of constituent technologies..." [37].

A DDDAS includes different constituent blocks:

(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 behaviour 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 better appreciated. As can be noticed, it deals with very different and transversal disciplines: from simulation techniques, numerical issues, control, modelling, software engineering, data management and telecommunications, among others. The three different blocks of interactions concern: (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: from 10^{-20} Hz for cosmological systems to 10^{20} Hz for problems at the atomic scales. Humans, however, can be considered as a system operating at rates from 3 Hz to 500 Hz in haptic 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 data are collected. While this is not always true (as in weather forecasting, for instance, where collected data are usually incorporated to the simulations after long time periods), 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.

As can be noticed from the introduction above, DDDAS can revolutionize 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 way of validating a design on the basis of a static data set [33].

While research on DDDAS should involve applications, mathematical and statistical algorithms, measurement systems, and computer systems software methods, see for instance [16,17,21,28,29], 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 Decomposition—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 (see [1,2] and the references therein) and was then extended to address general computational mechanics models [10]. We are revisiting the motivation and the key ideas of such technique in the next sections.

1.1. When the solution of many direct problems is needed

An important issue encountered in DDDAS, related to process control and optimization, inverse analysis, etc., lies in the necessity of solving many direct problems. Thus, for example, process optimization implies the definition of a cost function and the search of optimum process parameters, which minimize the cost function. In most engineering optimization problems the solution of the model is the most expensive step. Real-time computations with zero-order optimization techniques can not be envisioned except for very particular cases. The computation of sensitivity matrices and adjoint approaches also hampers fast computations. Moreover, global minima are only ensured under severe conditions, which are not (or cannot be) verified in problems of engineering interest. There are many strategies for updating the set of design parameters and the interested reader can find most of them in books focusing on optimization procedures. Our interest here is not the discussion on optimization strategies, but pointing out that standard optimization strategies need numerous direct solutions of the problem that represents the process, one solution for each tentative choice of the process parameters, plus those required for sensitivity.

As we discussed in the previous paragraphs, the solution of the model is a tricky task that demands important computational resources and usually implies extremely large computing times. Usual optimization procedures are inapplicable under real-time constraints because they need numerous solutions. The same issues are encountered when dealing with inverse analysis in which material or process parameters are expected to be identified from numerical simulation, by looking for the unknown parameters such that the computed fields agree in minute with the ones measured experimentally. However, some previous references exist on the treatment of problems that require extensive solution procedures for different parameter values. The interested reader can consult, for instance [6,7,20].

1.2. Towards generalized parametric modelling

One possibility for solving many problems very fast consists of using some kind of model order reduction based on the use of reduced bases [18,34]. In these works authors proved the capabilities of performing real time simulation even using light-computing devices, as smartphones for example. The tricky point in such approaches is the construction of such reduced bases and the way of adapting them when the system explores regions far from the ones considered in the construction of the reduced model. Solutions to this issue exist and others are been developed to fulfil with real time requirements.

Multidimensionality offers an alternative getaway to avoid too many direct solutions. In our opinion it could represent a new paradigm in computational mechanics. For the sake of clarity, the use of multidimensional modelling in an academic physical problem is illustrated and motivated.

Imagine for example that we are interested in solving the heat equation but the material's thermal conductivity is not known, because it has a stochastic nature or simply because prior to solve the thermal model it is necessary to measure it experimentally. Three possibilities arise: (i) wait to know the conductivity before solving the heat equation (a conservative solution); (ii) solve the equation for many values of the conductivity (a sort of Monte Carlo method); or (iii) solve the heat equation only once for any value of the conductivity.

Obviously the third alternative is the most appealing one. To compute this quite general solution it suffices to introduce the conductivity as an extra independent coordinate, taking values in a certain interval and playing a similar role as standard space and

time coordinates. Note, that there is no need to have derivatives involving this extra-coordinate. Thus, by solving only once the resulting multidimensional thermal model, the most general solution is computed; that is, a solution that produces at each physical point and instant the value of the temperature for any value of the thermal conductivity.

This procedure works very well, as will be proven later. Note that it can be extended to introduce many other extra-coordinates: source term, initial conditions, boundary conditions and even the domain geometry. Thus, moving loads in structural mechanics, geometrical parameters in shape optimization, material parameters in material characterization, boundary conditions in inverse analysis or process optimization, etc., can be treated as extra-coordinates to compute *off-line* multidimensional parametric solutions that could then be used *on-line*, running in real time. These general solutions computed off-line could be introduced in very light computing devices, as for example smartphones, opening an unimaginable field of applications that Fig. 1 caricatures. This methodology constitutes in our opinion a new paradigm of real-time simulation.

The procedure outlined above introduces a major difficulty. If unknown parameters are considered as new dimensions of the problem, we will rapidly attain a high-dimensional model. Problems defined in high dimensional spaces are well known in the literature for the so-called *curse of dimensionality* [24]. This curse is related to the exponential growth of the number of degrees of freedom if such a multidimensional model is discretized by standard mesh-based discretization techniques.

To illustrate the technique proposed here to circumvent this curse of dimensionality, consider a problem defined in a space of dimension d for the unknown field $u(x_1,\dots,c,x_d)$. Here, the coordinates x_i denote any coordinate (scalar or vectorial) related to physical space, time or any model parameter. We thus seek a solution for each $(x_1,\dots,x_d)\in\Omega_1\times\cdots b\times\Omega_d$ in separated forms:

$$u(x_1,\dots,x_d)\approx\sum_{i=1}^{N}F_i^1(x_1)\dots F_i^d(x_d). \tag{1}$$

The Proper Generalized Decomposition (PGD) (the interested reader can find a complete state of the art with all up to now published papers in [13]) consists in the construction of a separated representation of the model solution from N functional products involving each a number d of functions $F_i^j(x_j),\ j=1,\dots c,d$, that are unknown *a priori*. This approximation is constructed by successive enrichment up to a prescribed accuracy, whereby each functional product is determined in sequence. At a particular enrichment step $n+1$, the functions $F_i^j(x_j)$ are known for $i\leqslant n$ from the previous steps, and one must compute the new product involving the d unknown functions $F_{n+1}^j(x_j),\ j=1,\dots c,d$. This is achieved by invoking the weak form of the problem under consideration. The resulting formulation is non-linear, which implies that iterations are needed at each enrichment step. By using a fixed-point alternating-directions strategy a low-dimensional problem can thus be defined in Ω_j for each of the d functions $F_{n+1}^j(x_j)$.

If M degrees of freedom are used to discretize each coordinate, the total number of PGD unknowns is $N\cdot M\cdot d$ instead of the M^d degrees of freedom involved in standard mesh-based discretizations. Moreover, all numerical experiments carried out to date with the PGD show that the number of terms N required to obtain an accurate solution is not a function of the problem dimension d, but rather depends on the regularity of the solution, very much like Singular Value Decomposition methods. The PGD thus avoids the exponential complexity with respect to the problem dimension.

In many applications studied to date, N is found to be as small as a few tens. Moreover, the approximation converges towards the solution associated with the complete tensor product of the approximation bases considered for each Ω_j. Thus, we can be confident on the generality of the separated representation (1). When an exact solution of a particular problem can be represented with enough accuracy by a reduced number of functional products, the PGD approximation is suitable. If the solution is a strictly non-separable function, the PGD solver proceeds to enrich the approximation until including all the elements of the functional space, i.e. the M^d functions involved in the full tensor product of the approximation bases considered for each Ω_j.

The foundations and some recent advances in the proper generalized decomposition based discretization strategies can be found in [1–12,19–32,35] and the references therein.

1.3. Paper outline

Section 2 illustrates the construction of a separated, PGD, representation of the solution of a parametric boundary valued problem. The off-line solution construction procedure is analysed and thoroughly described. Then, Section 3 simulates a breakdown scenario that implies (i) identifying the malfunctioning device and (ii) the process reconfiguration. Both tasks should be performed as fast as possible and using the lightest possible computational resources (e.g. a smartphone, for instance) for real-time decision making. This is particularly crucial when the device is operated by personnel without a technical background. These on-line calculations on light computing platforms are described in Section 4.

2. Proper Generalized Decomposition for a parametric model of a material flowing in a heated die

In this section the main ideas related to casting the model into a multidimensional framework, followed by process optimization,

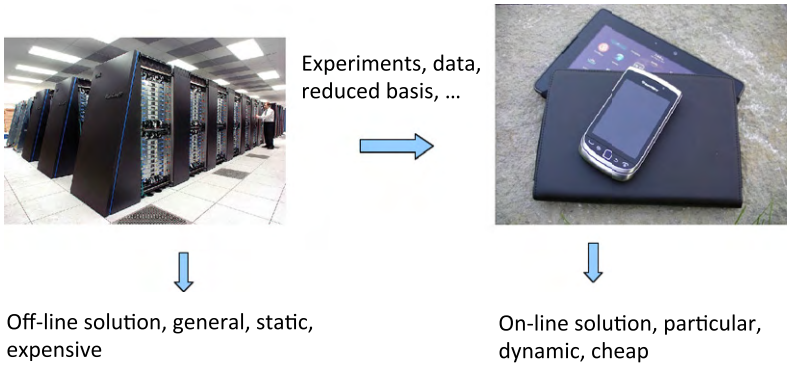

Fig. 1. "Off-line" solution of a general enough parametric model and "on-line" particularization of such a general solution in a particular context. <http://es.wikipedia.org/wiki/Archivo:UPM-CeSViMa-SupercomputadorMagerit. jpg>.

are introduced. For the sake of clarity in what follows we consider the thermal model related to a material flowing into a heated die. Despite the apparent simplicity, the strategy here described can be extended to address more complex scenarios.

The 2D thermal process is sketched in Fig. 2. The material flows with a velocity $\mathbf{v}$ inside a die Ω of length L and width H. The temperature of the material at the die entrance is u_0. The die is equipped with two heating devices of lengths L_1 and L_2 respectively, whose temperatures θ_1 and θ_2, respectively, can range within an interval $[\theta_{min}, \theta_{max}]$.

The steady state temperature field $u(\mathbf{x})$ in any point of the die $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$ can be obtained from the solution of the 2D heat transfer equation that involves advection and diffusion mechanics as well as a possible source term Q. The velocity field is everywhere unidirectional, i.e. $\mathbf{v}^T = (0, v)$. The steady-state heat transfer equation thus reduces to:

$$\rho c\left(v\frac{\partial u}{\partial x}\right) = k\Delta u + Q, \tag{2}$$

where k is the thermal conductivity, ρ is the density and c is the specific heat.

2.1. Building-up the parametric solution in the framework of a multidimensional model

The die is equipped with two heating devices as depicted in Fig. 2 whose temperatures constitute the process parameters to be optimized and, eventually, controlled. For the sake of simplicity the internal heat generation Q is assumed constant, as well as the velocity v and the inlet temperature u_0.

Different values of prescribed temperatures at both heating devices can be considered. The resulting 2D heat transfer equation can be then solved. As noted earlier, optimization or inverse identification will require many direct solutions or, as named in the introduction, static data computations. Obviously, when the number of the process parameters involved in the model is increased, standard approaches fail to compute optimal solutions in a reasonable time. Thus, for a large number of process parameters, real-time computations are precluded and, moreover, performing "on-line" optimization or inverse analysis is a challenging issue.

The method proposed here consists of introducing both process parameters, i.e. temperatures of the heating devices, θ_1 and θ_2, as extra coordinates.

Remark 1. If some of these desired extra-coordinates are fields depending on other coordinates instead independent parameters (temperatures of the heating devices evolving in time, source term evolving in time and/or space etc.) prior to introduce them as extra-coordinates one should parametrize such evolutions in an appropriate manner, and finally introduce the coefficients involved in those parametrizations as extra-coordinates as considered in [25].

Other parameters such as ρ, c, v, k, ..., c, can be set as extra-coordinates as well. The temperature field can be thus computed at each point and for any possible value of the temperatures θ_1 and θ_2. As soon as this multidimensional solution $u(x,y,\theta_1,\theta_2)$ is available, it is possible to particularize it for any value of the process parameters without the necessity of further executions of the code. Thus, optimization procedures can proceed from the only knowledge of an "off-line" pre-computed parametric solution.

To circumvent the curse of dimensionality related to the high dimensional space in which the temperature field $u(x,y,\theta_1,\theta_2)$ is defined—which we retain to be four-dimensional for the ease of exposition—we consider a separated representation of that field:

$$u(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^{N} F_i(x, y)\Theta_i^1(\theta_1)\Theta_i^2(\theta_2), \tag{3}$$

where all the functions involved in such separated representation are computed by applying the Proper Generalized Decomposition technique, described below.

Remark 2. Because the geometrical simplicity of Ω that can be written as $\Omega = \Omega_x \times \Omega_y$, we could consider a fully separated representation of the unknown field that now writes:

$$u(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \cdot \Theta_i^1(\theta_1) \cdot \Theta_i^2(\theta_2). \tag{4}$$

This fully separated representation can be also applied in complex domains as proved in [19], however when the physical domain is complex the most natural representation is the one given in Eq. (3).

The prescribed essential boundary conditions write:

$$\begin{cases} u(x = 0, y, \theta_1, \theta_2) = u_0, \\ u(x \in I_1, y = 0 \text{ or } y = H, \theta_1, \theta_2) = \theta_1, \\ u(x \in I_2, y = 0 \text{ or } y = H, \theta_1, \theta_2) = \theta_2, \\ \frac{\partial u}{\partial x}(x = L, y, \theta_1, \theta_2) = 0, \end{cases} \tag{5}$$

where I_1 and I_2 are the intervals of the x coordinate where the heating devices of length L_1 and L_2 respectively are defined. A null heat flux is assumed in the remaining part of the domain boundary. Thus, the temperature field u depends on four different coordinates, the two space coordinates (x,y) and the two temperatures prescribed in both regions on the die wall. Parameters θ_1 and θ_2 now take values in the intervals $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively.

The case of essential (Dirichlet) boundary conditions as parameters of the model deserves some comments. Non-homogeneous essential boundary conditions in PGD methods are usually treated by means of a simple change of variable

$$u = \psi + z, \tag{6}$$

where ψ is a function verifying essential boundary conditions. This leads to a problem in the z variable with homogeneous boundary conditions. Efficient construction of ψ functions in the framework of PGD approximations has been deeply analysed in [19]. We refer the interested reader to this reference for further details. This function ψ can, for the problems addressed here, be expressed in separated form

$$\psi(x, y, \theta_1, \theta_2) = \sum_{i=1}^{3} F_i(x, y)\Theta_i^1(\theta_1)\Theta_i^2(\theta_2), \tag{7}$$

where each functional product is used to impose initial conditions and the two non-homogeneous essential boundary conditions, one for each heater position. These functions are depicted in Fig. 3

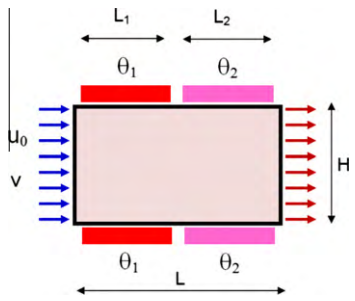

Fig. 2. Thermal process consisting of two heating devices located on the die walls where the temperature is enforced to the values θ_1 and θ_2, respectively.

The resulting PGD approximation reads

$$u(x,y,\theta_1,\theta_2) \approx \psi(x,y,\theta_1,\theta_2) + \sum_{i=4}^{N} F_i(x,y)\Theta_i^1(\theta_1)\Theta_i^2(\theta_2). \quad (8)$$

Functions F_i, Θ_i^1 and Θ_i^2, $i = 4,\dots c,N$, are determined by solving a sequence of non-linear problems with homogeneous boundary conditions, as described below. Assuming that the first n functional products of Eq. (8) have already been computed, we look for the $n+1$ term:

$$u^{n+1}(x,y,\theta_1,\theta_2) = \sum_{i=1}^{n} F_i(x,y)\Theta_i^1(\theta_1)\Theta_i^2(\theta_2) + R(x,y)S(\theta_1)T(\theta_2) \quad (9)$$

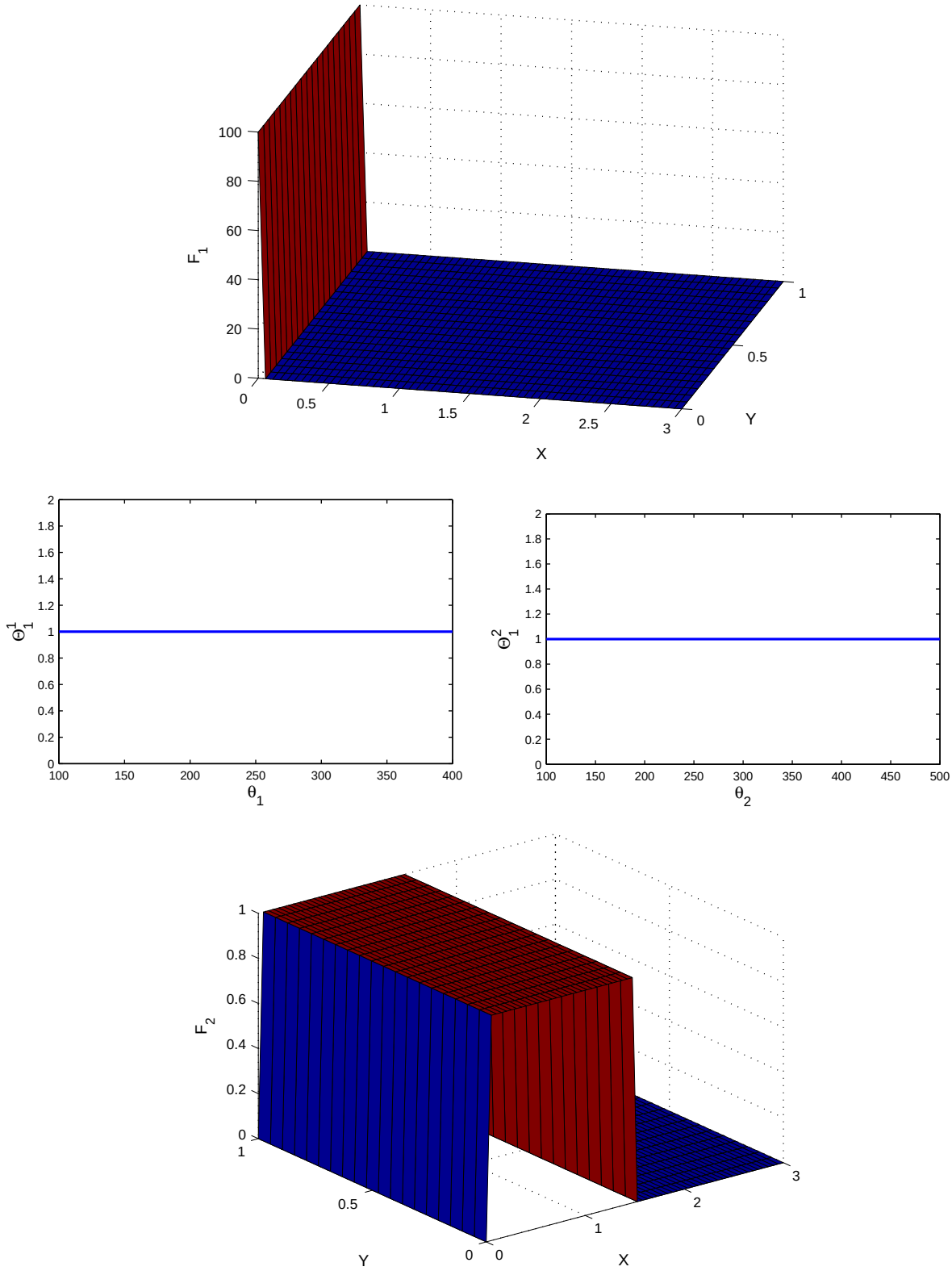

Fig. 3. Functions $F_i(x,y)$, $\Theta_i^1(\theta_1)$ and $\Theta_i^2(\theta_2)$ used to approximate initial and boundary conditions.

or, equivalently,

$$u^{n+1}(x,y,\theta_1,\theta_2) = u^n(x,y,\theta_1,\theta_2) + R(x,y)S(\theta_1)T(\theta_2). \tag{10}$$

Note that in the weak form associated to (2) the unknown functions R, S, and T are determined and the test function are

$$u^*(x,y,\theta_1,\theta_2) = R^*(x,y)S(\theta_1)T(\theta_2) + R(x,y)S^*(\theta_1)T(\theta_2) + R(x,y)S(\theta_1)T^*(\theta_2). \tag{11}$$

This approach allows us to determine the unknown functions $R(x,y)$, $S(\theta_1)$ and $T(\theta_2)$ in an alternating directions fixed-point algorithm. In fact, we proceed by determining sequentially each one of these functions, as described below, until reaching convergence. For more details we refer the interested reader to [10].

2.1.1. Determining $R(x,y)$ assuming $S(\theta_1)$ and $T(\theta_2)$ known

First, the weak form associated to (2) and boundary conditions defined in (5) must be determined. It reads

$$\int_{\Omega\times\mathcal{I}_1\times\mathcal{I}_2} \left(u^*\rho c\left(v\frac{\partial u}{\partial x}\right) + \nabla u^* k \nabla u\right) d\Omega d\theta_1 d\theta_2 = \int_{\Omega\times\mathcal{I}_1\times\mathcal{I}_2} u^* Q \; d\Omega d\theta_1 d\theta_2, \tag{12}$$

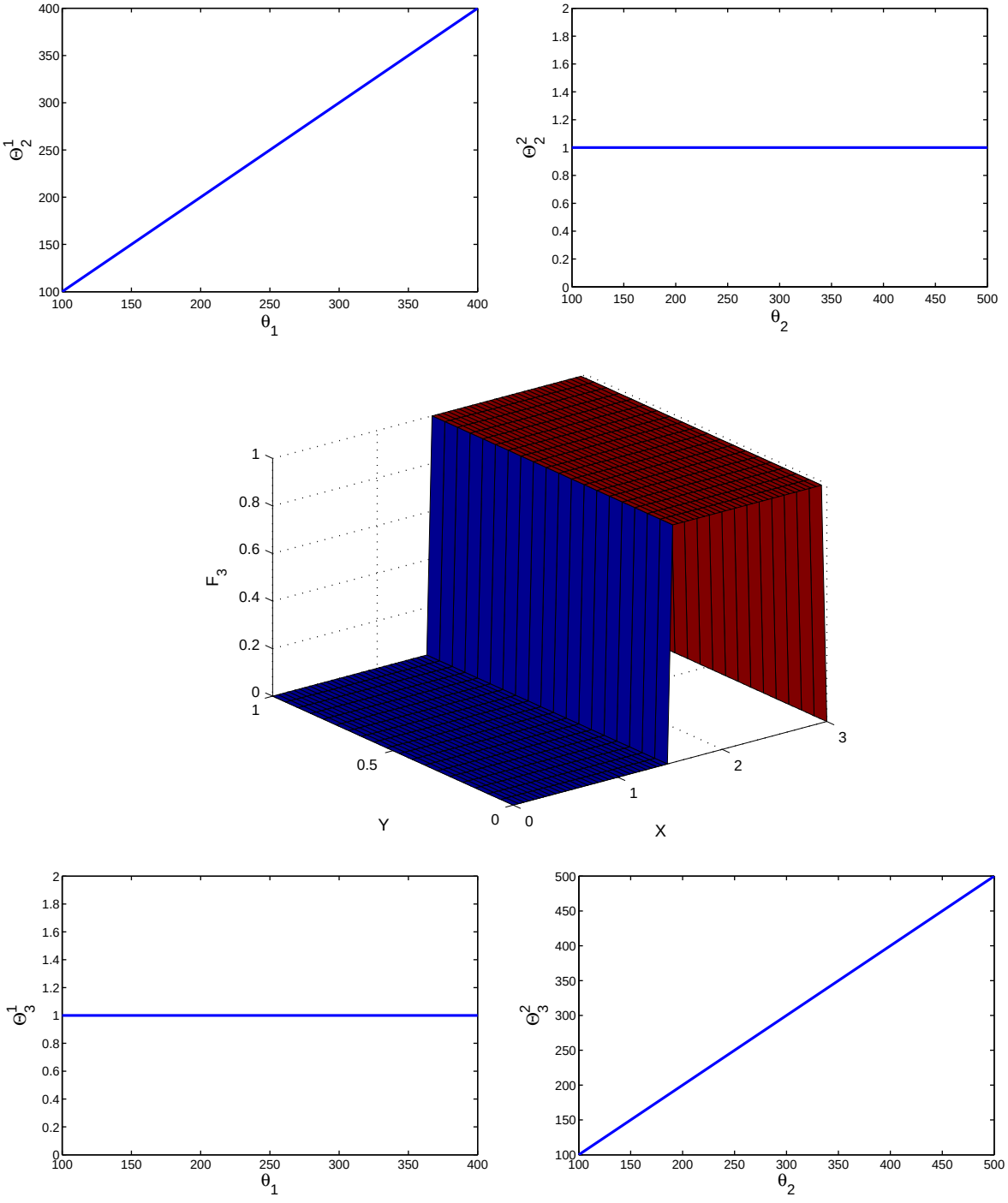

Fig. 3 (*continued*)

where the integration by parts, to reduce the order of derivation, has only been used for the spatial domain, Ω. Then, since an alternating direction approach is used to solve this problem, see the definition of the test function in (11), and inasmuch as S and T are assumed known, (11) reduces to:

$$u^*(x,y,\theta_1,\theta_2) = R^*(x,y)S(\theta_1)T(\theta_2).$$

For the sake of clarity, in what follows functional dependencies of each function are omitted. Approximation u^{n+1}, see (10), is substituted in (12). Thus, only R, S, and T must be determined. Now, as noted earlier, S and T are assumed known, and, consequently the unknown is R and the test function, defined above, induces the following problem: find R for all R^* (in the appropriate functional spaces) such that

$$\int_{\Omega\times\mathcal{I}_1\times\mathcal{I}_2} S^2T^2\left(\rho c v R^*\frac{\partial R}{\partial x} + k\nabla R^*\nabla R\right)d\Omega d\theta_1 d\theta_2 = r(R^*ST, u^n), \quad (13)$$

where $r(\cdot,\cdot)$ the weak residual, see (12), evaluated for $u^* = R^*ST$ at iteration n, i.e. $u = u^n$,

$$r(u^*,u) = \int_{\Omega\times\mathcal{I}_1\times\mathcal{I}_2}\left(u^*\rho c\left(v\frac{\partial u}{\partial x}\right) + \nabla u^* k\nabla u - u^*Q\right)d\Omega d\theta_1 d\theta_2.$$

Since functions involving the parametric coordinates θ_1 and θ_2 are assumed known in the present step, integrals in the parametric domain $\mathcal{I}_1\times\mathcal{I}_2$ can be calculated, leading to a problem defined in the two-dimensional space domain Ω whose solution is precisely the searched unknown function $R(x,y)$. This domain can be discretized, of course, by standard finite elements. Note however, that the resulting problem is a convection–diffusion equation along the x coordinate, which eventually may need some form of stabilization.

2.1.2. Determining $S(\theta_1)$ assuming $R(x,y)$ and $T(\theta_2)$ known

In this case, the test function reads

$$u^*(x,y,\theta_1,\theta_2) = R(x,y)S^*(\theta_1)T(\theta_2)$$

and the weak problem, see (12), becomes: find S for all S^* (in the appropriate functional spaces) such that

$$\int_{\Omega\times\mathcal{I}_1\times\mathcal{I}_2} T^2\left(\rho c v R\frac{\partial R}{\partial x} + k\nabla R\nabla R\right)S^*Sd\Omega d\theta_1 d\theta_2 = r(RS^*T, u^n). \quad (14)$$

Again, integrations involving the domain $\Omega\times\mathcal{I}_2$ can be performed, leading to a one-dimensional algebraic problem defined in $\mathcal{I}_1$, because the model does not contain derivatives with respect to the extra-coordinate θ_1. Its solution allows computing the function $S(\theta_1)$.

The calculation of function $T(\theta_2)$ can be performed in a similar way.

Remark 3. In the present case we assumed a linear model, however many problems encountered in science and engineering are in fact non-linear. For example the heat equation here addressed becomes non linear as soon as dependences of thermal parameters on the temperature field are taken into account.

In the non-linear case additional linearization techniques are required. Application of standard linearization strategies within the PGD framework were addressed in [3,35].

A better way to address strongly non-linear models consists in using the LATIN technique [22]. However its extension to the case of multidimensional parametric modelling is not straightforward.

2.2. "Off-line" optimization procedure

Optimization procedures look for optimal parameters minimizing an appropriate single or multi objective cost function (sometimes subjected to many constraints). In this work we consider a simple scenario, in which the cost function only involves the coldest thermal history of an imaginary material particle traversing the die, it is expressed as:

$$\mathcal{C}(\theta_1,\theta_2) = \frac{1}{2}\left(\int_0^L u\left(x,\frac{H}{2},\theta_1,\theta_2\right)dx - \beta\right)^2, \quad (15)$$

where β denotes the optimal value of the thermal history able to ensure a certain material transformation. Values lower than β imply that the material has not received the necessary amount of heat, whereas values higher than β imply an unnecessary extra-heating.

Now, optimal process parameters θ_1^{opt} and θ_2^{opt} must be calculated by minimizing the cost function. There exist many techniques for such minimization. The interested reader can refer to any book on optimization. Many of them proceed by evaluating the gradient of the cost function and then moving on that direction. The gradi-

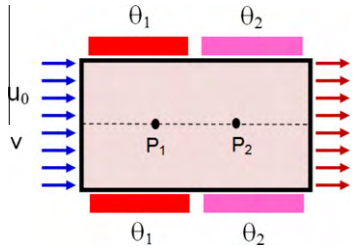

Fig. 5. Location of the thermocouples $\mathbf{P}_1$ and $\mathbf{P}_2$ whose measurements serve for controlling the process, identify malfunctioning devices and reconfigure the system after a breakdown.

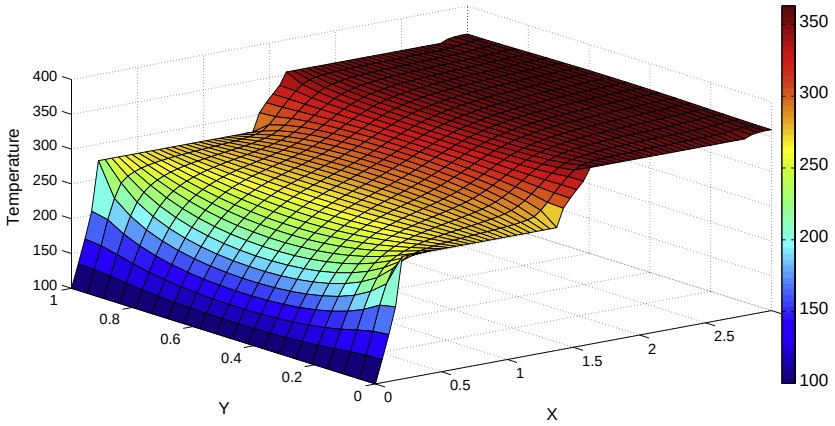

Fig. 4. Multidimensional solution particularized for the optimal temperature of both heating devices: $u(x,y,\theta_1^{\text{opt}} = 275.8, \theta_2^{\text{opt}} = 353.4)$.

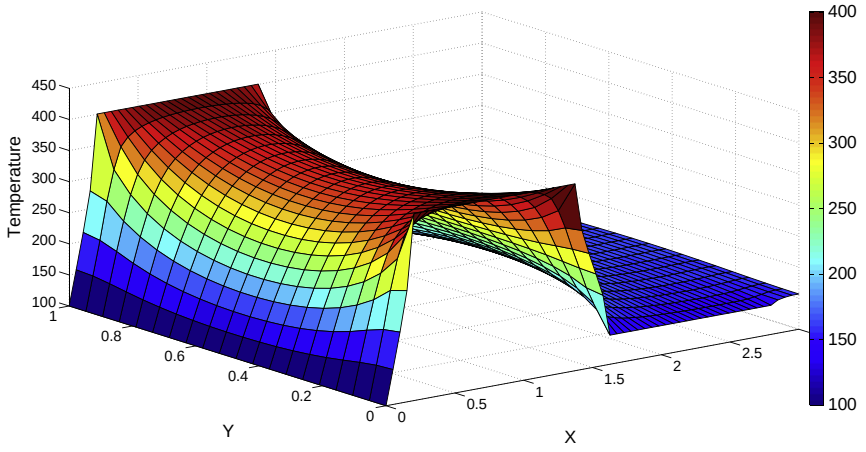

Fig. 6. Multidimensional solution particularized for the optimal temperature after reconfiguring the system: $u(x, y, \theta_1^*, \theta_2^{\text{est}})$.

ent computation involves the necessity of performing first derivatives of the cost function with respect to the process parameters. Other techniques involve the calculation of second derivatives. To this end, one should calculate the derivatives of the problem solution with respect to the optimization parameters.

It is important to note that separated representations of the process parameters drastically simplifies this task because as the solution depends explicitly on the parameters its derivation is straightforward, namely,

$$\frac{\partial u}{\partial \theta_1}(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^{N} F_i(x, y) \frac{\partial \Theta_i^1}{\partial \theta_1}(\theta_1) \Theta_i^2(\theta_2)$$

and

$$\frac{\partial u}{\partial \theta_2}(x, y, \theta_1, \theta_2) \approx \sum_{i=1}^{N} F_i(x, y) \Theta_i^1(\theta_1) \frac{\partial \Theta_i^2}{\partial \theta_2}(\theta_2).$$

Note that second derivatives are also similarly obtained. The calculation of the solution derivatives is a tricky point when proceeding from standard discretization techniques because the parametric dependency of the solution is, in general, not explicit.

Moreover, the separated rank-N representation of the solution, see (3), further simplifies the expression of the cost function. That is, substituting (3) in (15), induces

$$\mathcal{C}(\theta_1, \theta_2) = \frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right)^2, \tag{16}$$

where $\alpha_i = \int_0^L F_i(x, \frac{H}{2}) dx$, and the different derivatives of the cost function becomes:

$$\begin{cases} \frac{\partial \mathcal{C}}{\partial \theta_1}(\theta_1, \theta_2) = \left(\sum_{i=1}^{N} \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right) \left(\sum_{i=1}^{N} \alpha_i \frac{\partial \Theta_i^1}{\partial \theta_1}(\theta_1) \Theta_i^2(\theta_2) \right), \\ \frac{\partial \mathcal{C}}{\partial \theta_2}(\theta_1, \theta_2) = \left(\sum_{i=1}^{N} \alpha_i \Theta_i^1(\theta_1) \Theta_i^2(\theta_2) - \beta \right) \left(\sum_{i=1}^{N} \alpha_i \Theta_i^1(\theta_1) \frac{\partial \Theta_i^2}{\partial \theta_2}(\theta_2) \right). \end{cases}$$

In the simulations carried out in what follows, the minimization of the cost function was performed by using a Levenberg–Marquardt algorithm, see [15] for further details.

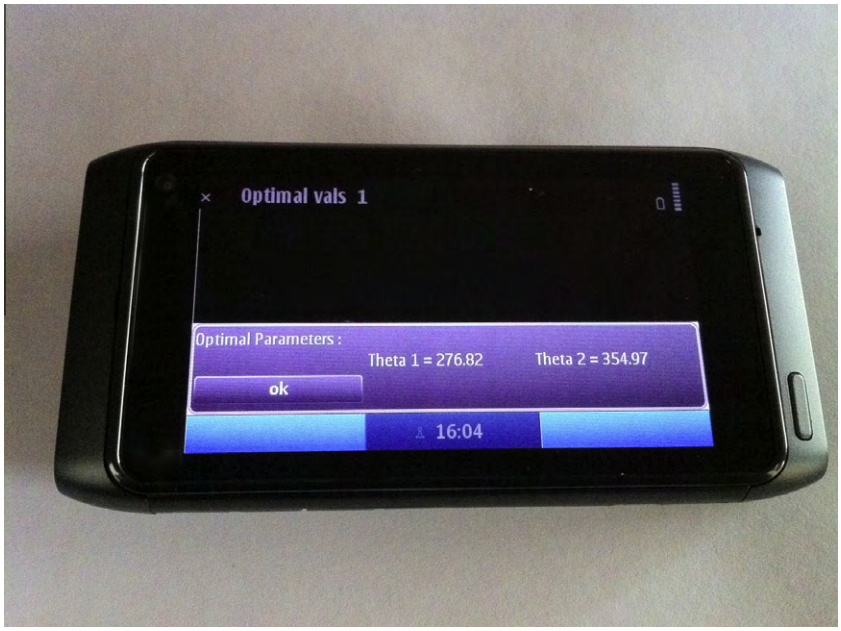

Fig. 7. Optimal temperatures of both heating devices.

2.3. Defining the process control

Once the optimal parameters θ_1^{opt} and θ_2^{opt} are determined, the general solution, see (3), can be evaluated for those optimal values of the process parameters, namely

$$u(x, y, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) \approx \sum_{i=1}^{N} F_i(x, y)\Theta_i^1(\theta_1^{\text{opt}})\Theta_i^2(\theta_2^{\text{opt}})$$

to obtain the temperature field everywhere in the domain Ω.

For the sake of clarity, in what follows, we illustrate the dynamic data driven thermal model trough a numerical example without physical relevance. We consider the thermal model described above, see Fig. 2, with the following values of the different parameters (units are omitted because all are defined in the metric system): $\rho = 1$, $c = 1$, $k = 1$, $Q = 50$, $v = 1$, $u_0 = 100$, $L = 3$, and $H = 1$. The first heating device acts in the interval $I_1 = [0.2, 1.4]$ whereas the second one is defined by $I_2 = [1.6, 2.8]$, both having the same length $L_1 = L_2 = 1.2$.

To solve the parametric model one needs to approximate the functions $F_i(\mathbf{x})$ involved in the solution's separated representation (3), as well as functions $\Theta_i^1(\theta_1)$ and $\Theta_i^2(\theta_2)$. Space functions are approximated by using a finite element mesh composed of 4-nodes quadrilateral elements, on a uniform nodal distribution composed of 60×20 nodes in the x and y directions, respectively. Functions depending on the process parameters θ_1 and θ_2 are approximated by two uniform 1D linear finite meshes (300 nodes each) over

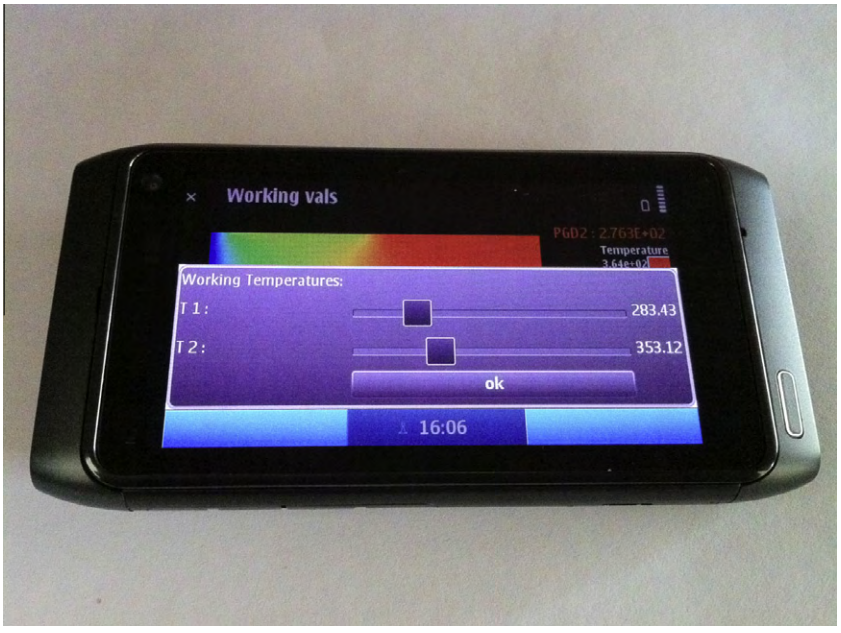

Fig. 8. Optimal temperatures at positions $\mathbf{P}_1$ and $\mathbf{P}_2$ when both heaters work at the optimal conditions.

Fig. 9. Simulating a failure by considering a temperature at position $\mathbf{P}_2$ different of the optimal one.

the interval of variation of these parameters ($\mathcal{I}_1 = [100, 400]$ and $\mathcal{I}_2 = [100, 500]$).

The parametric solution $u(\mathbf{x}, \theta_1, \theta_2)$ is computed by using the Proper Generalized Decomposition strategy illustrated in Section 2.1. This solution implies 42 functional products in the sum, that is $N = 42$ in (3). From this general solution we compute the optimal process parameters θ_1^{opt} and θ_2^{opt} with respect to the cost function introduced in (15) where $\beta = 897$. The convergence of the Levenberg–Marquardt algorithm is reached in only 4 iterations, being the optimal values $\theta_1^{\text{opt}} = 275.8$ and $\theta_2^{\text{opt}} = 353.4$. Fig. 4 depicts the resulting temperature field related to the optimal process parameters.

Remark 4. Up to now, the convergence criterion that we use concerns the norm of the residuals. We have considered in our former works more sophisticated error estimators, as the one based on quantities of interest [4].

Error estimators based on quantities of interest are preferred because they are more adapted to the outputs considered in control strategies. However, at present we have not extended these estimators to the multidimensional parametric modelling. Thus, prior to consider any parametric solution computed off-line in on-line procedures we should proceed to verification and validation.

In order to control the process we could imagine two thermocouples located at two points on the axis of symmetry, namely $\mathbf{P}_1 = (1, 0.5)$ and $\mathbf{P}_2 = (2, 0.5)$, see also Fig. 5. Obviously, for negligible validation and verification errors, the "on-line" measurements $\tilde{u}_1$ and $\tilde{u}_2$ will give values coincident, or almost, with the predictions of our model. These predictions are easily computed from the separate representation of the solution, i.e.

$$u^{\text{opt}}(\mathbf{P}_1) = u(1, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) = 286.6,$$
$$u^{\text{opt}}(\mathbf{P}_2) = u(2, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{opt}}) = 352.$$

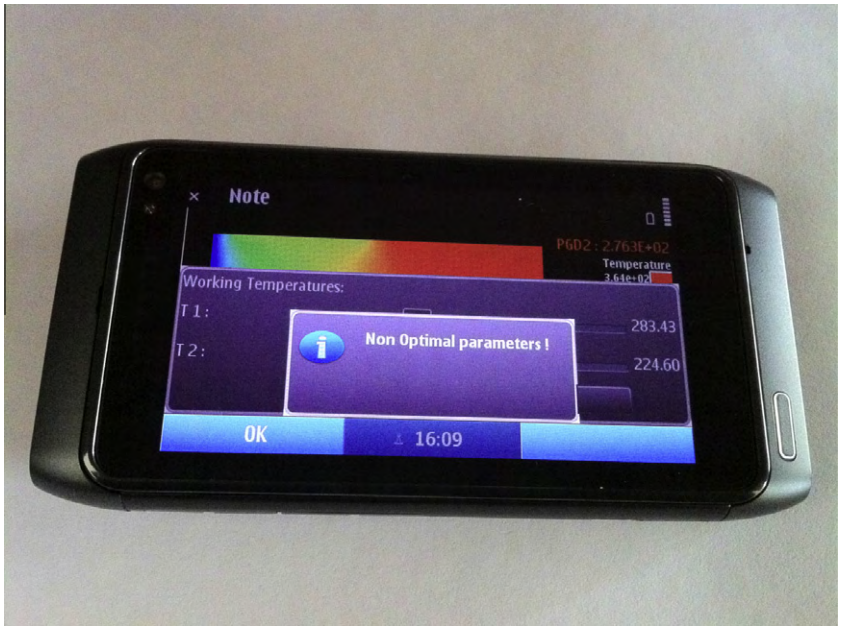

Fig. 10. Malfunction alert.

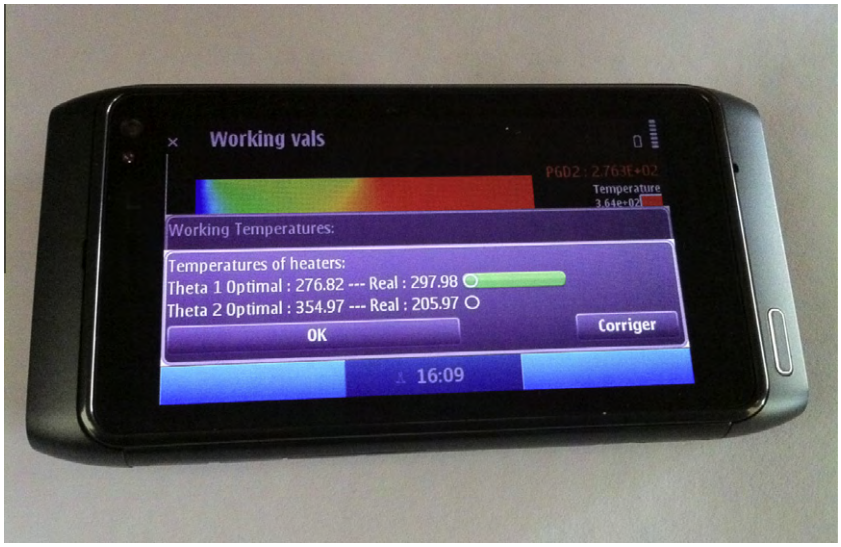

Fig. 11. Identification of the real process parameters, i.e. the real temperatures of both heating devices.

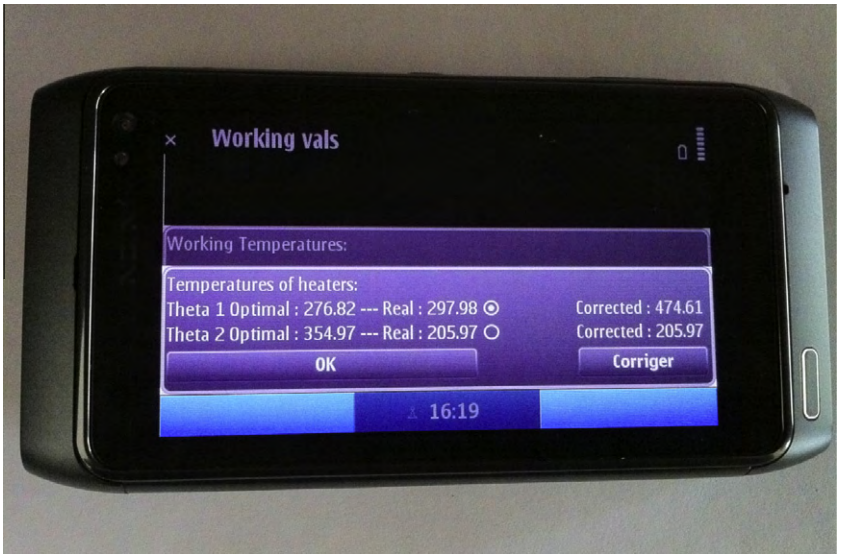

Fig. 12. System reconfiguration by adjusting the temperature of the first heating device.

Fig. 13. New optimal temperature field related to the reconfigured heating system.

Under these circumstances the process can be considered working in optimal conditions, the one related to the optimal temperatures of both heating devices.

However, if the heating devices do not run optimally the thermocouple measures $\tilde{u}_1$ and $\tilde{u}_2$ will differ from the predicted optimal conditions $u^{\text{opt}}(\mathbf{P}_1)$ and $u^{\text{opt}}(\mathbf{P}_2)$. Under these circumstances we could infer a breakdown in the system. Thus, checking the state of the system seems quite easy, however, the most important point is the identification of the malfunction and the reconfiguration of the system taking into account the real state of the system. These questions are addressed in the next section.

3. Simulating a breakdown scenario

In this section we simulate a breakdown scenario, whose solution requires both: (i) identifying the device malfunction and (ii) the process reconfiguration. Both tasks should be performed as fast as possible and using the lightest computational resources (e.g. a smartphone) in order to be able to do real-time decision making independently of the expertise of the personnel in charge of the process.

3.1. "On-line" inverse analysis

A possible scenario of a breakdown is a malfunction of the second heating device. For illustration purposes, we will assume that it only applies a fraction of the desired temperature. That is, the device is only able to prescribe a temperature of $\theta_2^{\text{brk}} = 0.4\theta_2^{\text{opt}} = 141.4$, instead of the optimal one (the slight gap is due to the fact of considering few iterations in order to fulfill real time requirements). Under these circumstances, both thermocouples will indicate temperatures at $\mathbf{P}_1$ and $\mathbf{P}_2$ equal to

$$\begin{aligned} u^{\text{brk}}(\mathbf{P}_1) &= u(1, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{brk}}) = 254, \\ u^{\text{brk}}(\mathbf{P}_2) &= u(2, 0.5, \theta_1^{\text{opt}}, \theta_2^{\text{brk}}) = 165. \end{aligned} \tag{17}$$

These values are obtained from the representation of the general solution, see (3), because validation and verification errors are assumed negligible.

To reproduce the practical scenario, it is assumed that a decision must be taken with the only information of the thermocouple temperatures, namely (17). That is, it is not known which device is malfunctioning and, of course, that the real prescribed temperature is θ_2^{brk}. In fact, the only available data is (17) which clearly indicates

that $u^{opt}(\mathbf{P}_1) \neq u^{brk}(\mathbf{P}_1)$ and $u^{opt}(\mathbf{P}_2) \neq u^{brk}(\mathbf{P}_2)$. Under these circumstances, the system should be reconfigured to ensure that the process continues working.

The first step is to determine which are the actual working temperatures. Thus, an inverse analysis is required. The following step, see the next subsection, will be to reconfigure the process in order to impose the desired thermal history, see (15). Obviously, both steps require a swift resolution if real-time decision are required. That is "on-line" computations are designed.

The "on-line" inverse analysis minimizes the following least-squares problem

$$\widetilde{\mathcal{C}}(\theta_1,\theta_2) = \frac{1}{2}\sum_{i=1}^{2}\left(u^{brk}(\mathbf{P}_i) - u(x_i, 0.5, \theta_1, \theta_2)\right)^2, \tag{18}$$

where x_i for $i = 1, 2$ are the coordinates of the points at which the thermocouples are located. The Levenberg–Marquardt algorithm, see [15], reaches convergence after three iterations and the estimated temperatures for both heating devices are $\theta_1^{est} = 261$ and $\theta_2^{est} = 146$, agree with the considered scenario ($\theta_1^{opt} = 275.8$, $\theta_2^{brk} = 141.4$). The inverse identification runs very fast and it only involves slight calculations, so it could be performed "on-line" on a very light computing devices, such as a smartphone.

3.2. "On-line" process reconfiguration

Finally, the process is reconfigured to impose the desired thermal history, see (15). Obviously, there are many possibilities and strategies. Here, since the second heating device is the one not giving the desired optimal heating. The action consists in keeping the second heating device in its present state, i.e. $\theta_2^{est} = 146$ and looking for the optimal value of θ_1^* minimizing the cost function (15) for a fixed and known $\theta_2 = \theta_2^{est} = 146$, namely

$$\mathcal{C}(\theta_1) = \frac{1}{2}\left(\int_0^L u\left(x, \frac{H}{2}, \theta_1, \theta_2^{est}\right) dx - \beta\right)^2.$$

In three interactions of the Levenberg–Marquardt algorithm convergence is attained to the value $\theta_1^* = 400$. Fig. 6 depicts the resulting temperature field related to the new optimized process parameters $\theta_1^* = 400$ and $\theta_2^{est} = 146$.

4. Parametric solution post-processing by using light computing platforms

As soon as the parametric solution $u(x,y,\theta_1,\theta_2)$ has been computed only once and off-line, the subsequent processes, i.e. optimization, control and system reconfiguration, only involve very light calculations that could be performed on-line and using mobile computing platforms, like smartphones.

To illustrate this capability, we assume that the parametric solution previously considered has already been computed and that it is available in a separated form. Now this solution can be introduced in a smartphone that will perform all the on-line calculations described in the previous sections. Only the most significative modes of the separated representation are retained in order to speed-up the computations. In our applications we considered a Nokia platform, with 256 MB of RAM, 16 GB of internal memory, a 680 MHz ARM 11 CPU and with Symbian 3 as operating system.

Figs. 7–13 illustrate all the scenarios analyzed in the previous sections. Fig. 7 shows the output for the optimal values of both heating devices $\theta_1^{opt} = 276.82$ and $\theta_2^{opt} = 354.97$. They are slightly different from the ones previously computed because as just argued the parametric solution introduced into the smartphone has been restricted to the most significant modes of the off-line parametric solution to alleviate the on-line calculations performed by the smartphone.

The temperatures at the thermocouple positions $\mathbf{P}_1 = (1, 0.5)$ and $\mathbf{P}_2 = (2, 0.5)$ are then computed for the optimal temperatures of both heating devices. Fig. 8 shows both temperatures $u^{opt}(\mathbf{P}_1) = 283.43$ and $u^{opt}(\mathbf{P}_2) = 353.12$.

A failure scenario is then simulated by considering that the temperatures at positions $\mathbf{P}_1$ and $\mathbf{P}_2$ are not the optimal ones. In fact we consider the scenario defined by $u^{brk}(\mathbf{P}_1) = 283.43$ and $u^{brk}(\mathbf{P}_2) = 224.60$ illustrated in Fig. 9. The system identifies the malfunction and displays an alert message as shown in Fig. 10. The system then identifies the real temperatures of both heating devices, by applying the inverse strategy previously discussed. The real temperatures of both heating devices are identified as $\theta_1^{est} = 297.98$ and $\theta_2^{est} = 205.97$ instead of the optimal ones $\theta_1^{opt} = 276.82$ and $\theta_2^{opt} = 354.97$ as depicted in Fig. 11.

We can decide to reconfigure the system by choosing to change the temperature of one (or eventually both) heating devices. In our case we decide to calculate the new temperature of the first heating device able to ensure optimal process conditions. For this purpose we select the first heater as shown in Fig. 11. The system then recomputes on-line the optimal temperature of the first heating device: $\theta_1^* = 474.61$ as shown in Fig. 12. The new temperature field corresponding to the new operational conditions $u(x, y, \theta_1^* = 474.61, \theta_2^{est} = 205.97)$ is illustrated in Fig. 13.

5. Conclusions

This work presents a first attempt of applying dynamic data driven simulation for controlling industrial processes whose modelling involves complex linear or non-linear partial differential equations. For the sake of simplicity we addressed in the present work the linear case, the non-linear one constituting a work in progress.

The procedure that we propose combines heavy "off-line" calculations for solving the partial differential model associated to the industrial process by introducing all the sources of variability or process parameters as extra-coordinates. Thus, the resulting model becomes multidimensional making the use of the well experienced mesh-based discretization techniques impossible. However, the use of separated representations within the Proper Generalized Decomposition framework allows us to circumvent the curse of dimensionality that multidimensional models suffer.

As soon as this parametric solution is available, one could proceed to optimize the process, still "off-line", by calculating the optimal process parameters in order to minimize an appropriate cost function.

However, the system's response faced to a breakdown should be computed "on-line" and as fast as possible. Separated representation of the solution built-up by applying the PGD-based solver allows computing explicitly the derivatives of the solution with respect to the process parameters, making the fast calculation of minimization strategies possible. Thus, malfunctioning devices can be identified "on-line" and the systems reconfigured by making some light calculations that we could perform using for example a simple smartphone.

This paper constitutes a first attempt of solving complex models by combining "off-line" and "on-line" computations in the framework of Proper Generalized Decompositions, in scenarios needing real time responses. In our opinion the possibility of computing parametric solutions that are then used by simply post-processing opens an unimaginable number of potential applications. For instance, uncertainty in the measurements (that has not been considered here), and that constitutes an essential ingredient in DDDAS, could eventually be efficiently treated in the PGD framework by considering uncertain parameters as additional dimen-

sions. Worst-case scenarios could be computed in a similar way than the here proposed optimization procedures. This constitutes our current effort of research and will be published elsewhere.

In the numerical, academic, examples here addressed "off-line" calculations needed two minutes of computing time (using matlab, a standard laptop and a non optimized simulation code) whereas all the "on-line" calculations were performed in 0.0015 s. The examples here addressed are too simple to be conclusive, but at least, they prove the pertinence of the proposed approach.

References

[1] 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.

[2] 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.

[3] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta, Non-incremental strategies based on separated representations: applications in computational rheology, Communications in Mathematical Sciences 8/3 (2010) 671–695.

[4] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1872–1880.

[5] A. Ammar, M. Normandin, F. Chinesta, Solving parametric complex fluids models in rheometric flows, J. Non-Newtonian Fluid Mech. 165 (23–24) (2010) 1588–1601.

[6] P.-A. Boucard, P. Ladeveze, Une application de la méthode LATIN au calcul multirésolution de structures non linéaires, Rev. Eur. Eléments Finis. 8 (1999) 903–920 (In french).

[7] P.-A. Boucard, S. Buytet, P.-A. Guidault, A multiscale strategy for structural optimization, Int. J. Numer. Methods Engrg. 78 (1) (2009) 101–126.

[8] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Comput. Methods Appl. Mech. Engrg. 199 (25-28) (2010) 1872–1880., doi:10.1016/j.cma.2010.02.01. ISSN 0045-7825.

[9] F. Chinesta, A. Ammar, P. Joyot, The nanometric and micrometric scales of the structure and mechanics of materials revisited: an introduction to the challenges of fully deterministic numerical descriptions, Int. J. Multiscale Comput. Engrg. 6/3 (2008) 91–213.

[10] F. Chinesta, A. Ammar, E. Cueto, Recent advances in the use of the proper generalized decomposition for solving multidimensional models, Arch. Comput. Methods Engrg. 17 (4) (2010) 327–350.

[11] F. Chinesta, A. Ammar, E. Cueto, Proper generalized decomposition of multiscale models, Int. J. Numer. Methods Engrg 83 (8–9) (2010) 1114–1132.

[12] F. Chinesta, A. Ammar, E. Cueto, On the use of proper generalized decompositions for solving the multidimensional chemical master equation, Eur. J. Comput. Mech. 19 (2010) 53–64.

[13] F. Chinesta, P. Ladeveze, E. Cueto. A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Engrg., (in press). doi:10.1007/s11831-011-9064-7.

[14] F. Darema, Engineering/scientific and commercial applications: differences, similarities, and future evolution, in: Proceedings of the Second Hellenic European Conference on Mathematics and Informatics, HERMIS, vol. 1, 1994, pp. 367–374.

[15] J.E. Dennis Jr., R.B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, Classics in Applied Mathematics, 16, Corrected reprint of the 1983 original, Society for Industrial and Applied Mathematics (SIAM), PA, 1996.

[16] John Michopoulos, Panagiota Tsompanopoulou, Elias N. Houstis, Charbel Farhat, Michel. Lesoinne, John R. Rice, Anupam Joshi, On a data-driven environment for multiphysics applications, Future Generat. Comput. Syst. 21 (6) (2005) 953–968.

[17] C. Farhat, J.G. Michopoulos, F.K. Chang, L.J. Guibas, A.J. Lew, Towards a dynamic data driven system for structural and material health monitoring, in: V.N. Alexandrov, G.D. Alexandrov, P.M.A. van Albada, J. Sloot, J. Dongarra (Eds.), Lecture Notes in Computer Science, vol. 3993, Springer-Verlag, 2006, pp. 456–464.

[18] D. Amsallem, J. Cortial, C. Farhat, Toward real-time CFD-based aeroelastic computations using a database of reduced-order information, AIAA J. 48 (2010) 2029–2037.

[19] D. Gonzalez, A. Ammar, F. Chinesta, E. Cueto, Recent advances in the use of separated representations, Int. J. Numer. Methods Engrg. 81 (5) (2010) 637–659.

[20] C. Heyberger, P.-A. Boucard, D. Neron, Multiparametric analysis within the proper generalized decomposition framework, Comput. Mech. (2011), doi:10.1007/s00466-011-0646-x.

[21] Vipul Hingne, Anupam Joshi, Elias Houstis, John Michopoulos, On the Grid and Sensor Networks, IEEE/ACM International Workshop on Grid Computing, p. 166, Fourth International Workshop on Grid Computing, 2003.

[22] P. Ladeveze, Nonlinear Computational Structural Mechanics, Springer, NY, 1999.

[23] P. Ladeveze, J.-C. Passieux, D. Neron, The LATIN multiscale computational method and the proper generalized decomposition, Comput. Methods Appl. Mech. Engrg. 199 (21–22) (2010) 1287–1296.

[24] R.B. Laughlin, David. Pines, The theory of everything, Proc. Natl. Acad. Sci. 97 (1) (2000) 28–31.

[25] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin, F. Chinesta, Routes for efficient computational homogenization of non-linear materials using the proper generalized decomposition, Arch. Comput. Methods Engrg. 17 (4) (2010) 373–391.

[26] C. Le Bris, T. Lelièvre, Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, Construct. Approx., doi:10.1007/s00365-009-9071-1.

[27] G.M. Leonenko, T.N. Phillips, On the solution of the Fokker–Planck equation using a high-order reduced basis approximation, Comput. Methods Appl. Mech. Engrg. 199 (1–4) (2009) 58–168.

[28] John G. Michopoulos, Panayota Tsompanopoulou, Elias N. Houstis, Anupam Joshi, Sasikanth Avancha, Haiping Zhang,Towards agent-based grid-enabled and sensor-driven fire dynamics simulation harnessed over bluetooth and Wi-Fi devices, in: ASME Conference Proceedings 2005, vol. 111, 2005. doi:10.1115/detc2005-84770.

[29] John Michopoulos, Panagiota Tsompanopoulou, Elias N. Houstis, Anupam Joshi, Agent-based simulation of data-driven fire propagation dynamics, in: International Conference on Computational Science, 2004, pp. 732–739.

[30] A. Nouy, P. Ladeveze, Multiscale computational strategy with time and space homogenization: a radial-type approximation technique for solving microproblems, Int. J. Multiscale Comput. Engrg. 170 (2) (2004).

[31] A. Nouy, Recent developments in spectral stochastic methods for the solution of stochastic partial differential equations, Arch. Comput. Methods Engrg. 16 (3) (2009) 251–285.

[32] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high-dimensional stochastic problems, Arch. Comput. Methods Engrg. 17 (4) (2010) 403–434.

[33] J.T. Oden, T. Belytschko, 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, NSF Blue Ribbon Panel on SBES, 2006.

[34] J.L. Eftang, D.J. Knezevic, A.T. Patera, An hp certified reduced basis method for parametrized parabolic partial differential equations, Math. Comput. Model. Dynam. Syst. 17 (4) (2011) 395–422.

[35] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of parametric models by using the proper generalized decomposition, Math. Comput. Simulat. 81 (4) (2010) 791–810.

[36] DDDAS Workshop 2000 Final Report, Arlington, VA, USA, Technical report, National Science Foundation, 2000.

[37] DDDAS Workshop 2006 Final Report, Arlington, VA, USA, Technical report, National Science Foundation, 2006.