Model reduction, data-based and advanced discretization in computational mechanics

Reduced-order model of optimal temperature control for the automated fibre placement process

Nicolas Bur [a],[*], Pierre Joyot [b], Pierre Villon [c]

[a] *ESTIA-Recherche, Technopôle Izarbel, 64210 Bidart, France*
[b] *ESTIA, CNRS, I2M Bordeaux, Technopôle Izarbel, 64210 Bidart, France*
[c] *Université de technologie de Compiègne, Département "Génie des systèmes mécaniques", Centre de recherches de Royallieu, CS 60319, 60203 Compiègne cedex, France*

A R T I C L E   I N F O

A B S T R A C T

Industrialising manufacturing processes for aeronautic composite parts is a challenging issue. Among the existing techniques, the Automated Fibre Placement (AFP) is a promising one, since it allows the making of large and complex pieces with good productivity and repeatability. However, in order to ensure the regulatory requirements, the process must be controlled efficiently. In this paper, we propose the off-line computation of a parametric solution to a minimisation problem subject to heat equation. To solve this saddle-point problem with the so-called PGD method, we considered using Uzawa's technique or the Ideal Minimal Residual-based formulation, the aim being real-time control of the heat source within the AFP process.

© 2018 Académie des sciences. Published by Elsevier Masson SAS. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Automated Fibre Placement (AFP) is one of the main technologies employed today to manufacture advanced composite laminates from unidirectional prepregs [1]. This technique consists in laying and welding tapes of prepregs, building a laminate with more or less complex geometry, as depicted in Fig. 1.

In the 1990s, numerical models were proposed [2,3], with crude assumptions far from real process conditions. Improvements were made in [4,5].

In [6], even if the model itself was more relevant (3d domain, anisotropic material, inter-ply interfaces...), the numerical method was novel. To achieve a global thermo-mechanical process modelling, the numerical strategy proposed was based on the Proper Generalized Decomposition (PGD) [7,8]. This method uses a separated representation of the unknown field, reducing the computational complexity of the system. A key asset of this technique is its ability to introduce parameters (from process, from material, even from geometry...) as extra-coordinates into the model, without incurring the *curse of dimensionality*. Thus, in a single computation we have access to a multi-parametric virtual chart providing all possible solutions for each combination of the considered parameters [9–13].

Then, the computational vademecum can be exploited *on-line* for process control or process optimisation purposes. Indeed, within the AFP we want to efficiently control the heating power: tapes have to be heated enough to ensure the melting of the matrix coating the fibres and the cohesion with the previously laid tapes, while not exceeding a threshold

---

* Corresponding author.
  E-mail addresses: n.bur@estia.fr (N. Bur), p.joyot@estia.fr (P. Joyot), pierre.villon@utc.fr (P. Villon).
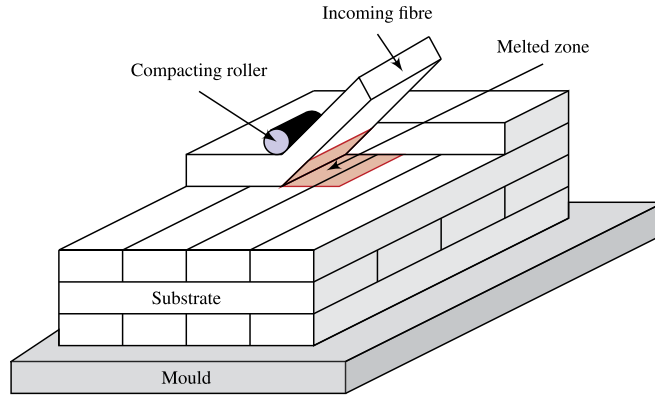
Fig. 1. Process sketch.

from which material burns, for a given process velocity. Indeed, both reticulation and thermal degradation depend on temperature and on the time spent at that temperature [14].

Therefore, we took advantage of the PGD to build *off-line* static and transient virtual charts in order to determine the best power associated with a draping velocity profile [15].

However, in those simulations, solutions were computed from equations provided by underlying physics of the studied phenomena, the optimisation being carried out as post-process. We propose here to compute directly the solution to an optimisation problem in order to get the separated representations of both the field and the control to obtain it. That is to say that the optimisation is made directly *off-line*, reducing the cost of the post-process and improving the real-time control of the AFP.

Within the next section, we present the equations governing the phenomenon under consideration. In order to have a reference solution, the system is solved by a standard finite element method (FEM). Thereafter, section 3 shortly presents the so-called PGD, and we take benefit of this technique in section 4 to build virtual charts to be used for control. Section 5 focuses on the writing and solving of the optimal system. We improve the obtained results by applying an iterative scheme in section 6. To circumvent the drawback of this additional loop, we also tried a modified PGD-based solver in section 7. Lastly section 8 addresses few conclusions and perspectives.

2. Process modelling

Let us consider a unidirectional stacking of layers constituting a homogeneous domain Ω, depicted in Fig. 2. This picture also makes explicit the meaning of some geometrical parameters.

We assume the heat source acts only on a part of the upper boundary, denoted by Γ_P in Fig. 2. With the assumption of a steady state, within a large piece, a Dirichlet condition is set on the boundary Γ_D, homogeneous for the sake of simplicity. On the remaining boundaries, gathered under the unique name Γ_N, we consider homogeneous Neumann conditions, once again for the sake of simplicity.
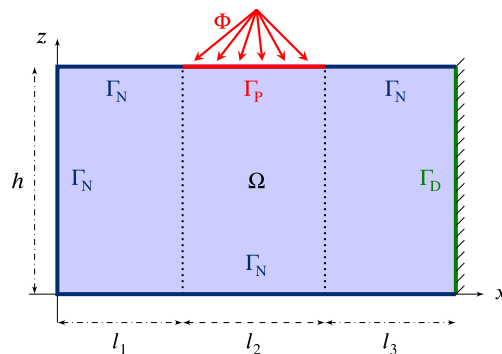

Fig. 2. Domain of study.

Table 1
Sets of parameters used.

Simulation	$V\ [\mathrm{m\,s^{-1}}]$	$P_w\ [\mathrm{W}]$	$h\ [\mathrm{m}]$
0	10^{-3}	600	0.01
1	10^{-2}	1900	0.003
2	10^{-1}	6000	0.0015
3	1	19000	0.0005

This leads to the convection–diffusion equation

$$\begin{cases} -\mathrm{div}\,(\mathbf{K}\nabla u) + \rho C_p \mathsf{V} \cdot \nabla u = 0 & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D \\ \mathbf{K}\nabla u \cdot \vec{n} = 0 & \text{on } \Gamma_N \\ \mathbf{K}\nabla u \cdot \vec{n} = \Phi & \text{on } \Gamma_P \end{cases} \tag{1}$$

with $\mathbf{K} = \begin{pmatrix} k_\parallel & 0 \\ 0 & k_\perp \end{pmatrix}$ and $\mathsf{V} = \begin{pmatrix} v \\ 0 \end{pmatrix}$.

The associated weak form writes

$$\begin{cases} \text{Find } u(\mathbf{x}) \in H^1(\Omega), \text{ verifying} \\ \forall u^\star \in H^1(\Omega), \text{ with } u(\mathbf{x} \in \Gamma_D) = u^\star(\mathbf{x} \in \Gamma_D) = 0 \\ \int_\Omega \mathbf{K}\nabla u \cdot \nabla u^\star \, \mathrm{d}\omega + \int_\Omega \rho C_p \mathsf{V} \cdot \nabla u u^\star \, \mathrm{d}\omega = \int_{\Gamma_P} \Phi u^\star \, \mathrm{d}\gamma \end{cases} \tag{2}$$

This system can obviously be solved in a classical way with a standard FEM. However, since we want a reference solution to later compare with results from other methods, we have to implement comparable systems. Consequently, in order to compare solution in separated representation, we take advantage of the tensor product method introduced by R.E. Lynche, J.R. Rice, and D.H. Thomas [16,17]. The discrete form of Eq. (2) involves matrices that can be expressed in terms of tensor products of lower-order matrices, leading to a computational gain. Authors also claim that there are no numerical instabilities with this method.

For more details on the properties of tensor products, the interested reader can refer to [18] and [19].

Expressing Eq. (2) with Kronecker products and then solving it with a standard FEM is what we call *K-FEM*. The aim of this artefact is to use shape functions from tensor product, so that involved matrices are the same as in the PGD formulation.

To illustrate our point, we consider four draping speeds, changing accordingly, not only the height h of the domain (since the slower the velocity is, the deeper the heat propagates), but also the power P_w of the source taken as a Gaussian function defined on Γ_P by Eq. (3). The sets of parameters are provided in Table 1. These values of P_w are chosen in order to reach a maximal temperature between 350 K and 400 K on Γ_P.

$$\begin{cases} \Phi(x) = \dfrac{P_\mathrm{w}}{S_\mathrm{ray}} \left(1 + \left(\dfrac{x - \mu}{r} \right)^4 - 2 \left(\dfrac{x - \mu}{r} \right)^2 \right) \\ S_\mathrm{ray}: \text{heated surface} \\ \mu = l_1 + \dfrac{l_2}{2}: \text{centre of the Gaussian} \\ r = \dfrac{l_2}{2}: \text{radius of the Gaussian} \end{cases} \tag{3}$$

Temperature fields computed from the four simulations are given in Fig. 3. As the dimensions of the domain depend on the simulation (see Table 1), these fields are plotted on a normalized domain.

In a former work (see [20]), we had to use a stabilisation technique with the standard FEM on this problem, since the convection term becomes dominant with an increasing velocity.

3. PGD at a glance

Consider a problem defined in a space of dimension d for the unknown field $u(x_1, \cdots, x_d)$. Here, the coordinates x_i denote any usual coordinate (scalar or vectorial) related to physical space, time, or conformation space in microscopic descriptions [7], for example, but they could also include model parameters such as boundary conditions or material parameters.

We seek a solution for $u(x_1, \ldots, x_d) \in \Omega_1 \times \ldots \times \Omega_d$. The PGD yields an approximate solution in the separated form

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{N} X_i^1(x_1) \cdot \ldots \cdot X_i^d(x_d) = \sum_{i=1}^{N} \prod_{j=1}^{d} X_i^j(x_j) \tag{4}$$

(a) Simulation 0

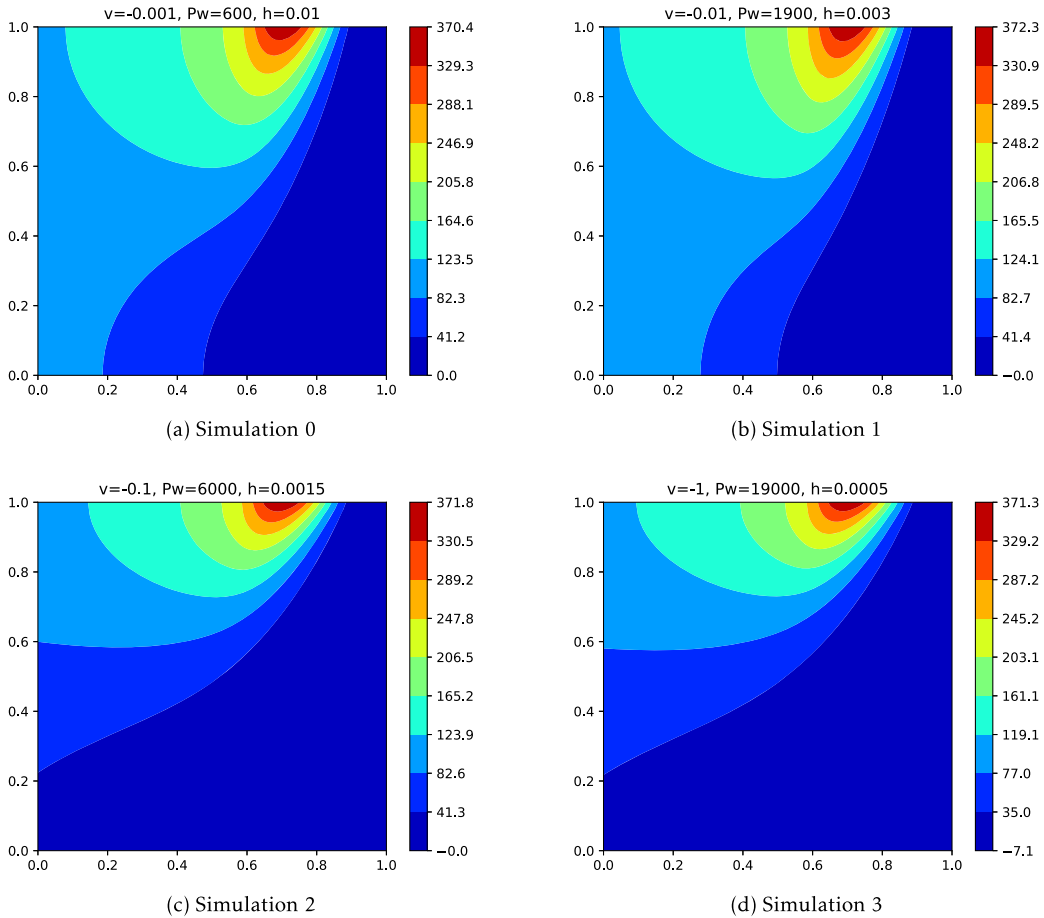(b) Simulation 1

(c) Simulation 2

(d) Simulation 3

Fig. 3. Reference temperature fields.

The PGD approximation is thus a sum of N functional products involving each a number d of functions $X_i^j(x_j)$ that are unknown *a priori*. It is constructed by successive enrichment, whereby each functional product is determined in sequence. At a particular enrichment step $n + 1$, the functions $X_i^j(x_j)$ are known for $i \leq n$ from the previous steps, and one must compute the new product involving the d unknown functions $X_{n+1}^j(x_j)$. This is achieved by invoking the weak form of the problem under consideration. The resulting problem is non-linear, which implies that iterations are needed at each enrichment step. A low-dimensional problem can thus be defined in Ω_j for each of the d functions $X_{n+1}^j(x_j)$.

If $\mathcal{M}$ nodes are used to discretise each coordinate, the total number of PGD unknowns is $N \times \mathcal{M} \times d$ instead of the $\mathcal{M}^d$ degrees of freedom involved in standard mesh-based discretisations.

In the case of a field depending on the physical space $\mathbf{x} \in \Omega_{\mathbf{x}} \subset \mathbb{R}^3$, the time $t \in \mathcal{I}_t \subset \mathbb{R}_+$ and Q parameters $p^1, \ldots, p^Q$, where $p^j \in \Omega_{p^j}$, with $j = 1, \ldots, Q$, the solution is sought under the separated form

$$u(\mathbf{x}, t, \theta^1, \ldots, \theta^Q) \approx \sum_{i=1}^{N} \left(X_i(\mathbf{x}) T_i(t) \prod_{j=1}^{Q} \Theta_i^j(\theta^j) \right) \tag{5}$$

As soon as this solution is available, after solving the multidimensional model within the PGD framework, we can have access to any possible solution.

PGD solution procedures have already been extensively described and successfully used in a plethora of applications. The interested reader can refer to the reviews [21–23] as well as to the primer [24] that describes the practical issues related to its computational implementation.

4. Controlling the AFP process

In order to ensure the quality of the produced composite parts, the thermoplastic has to be melted at an optimal temperature denoted u_{opt}. To reach it, the heat source must be drastically controlled. The numerical simulation is a great
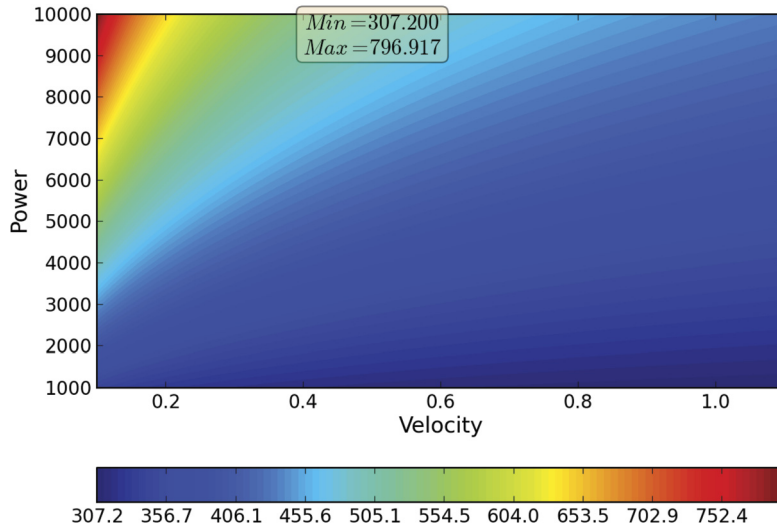
Fig. 4. Highest temperature as a function of the laser power and the line velocity: $u(P_w, v)$.
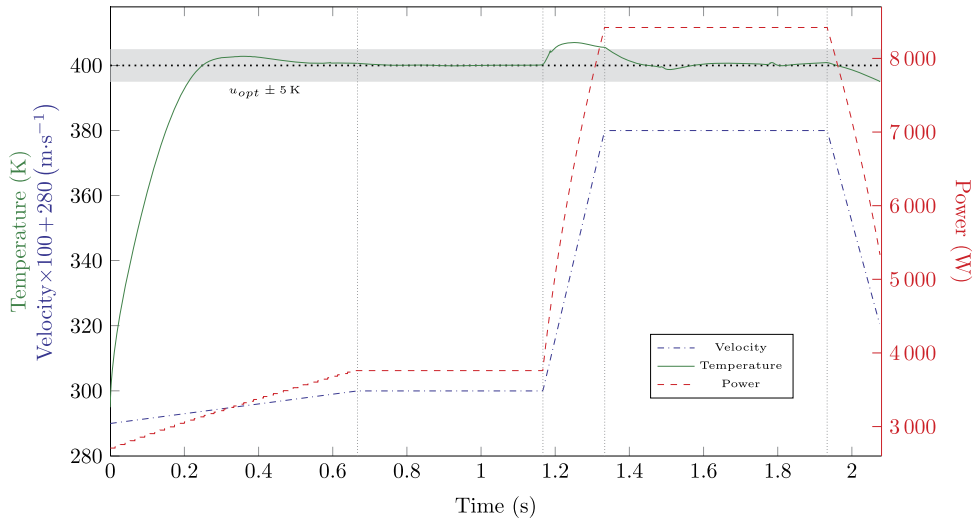

Fig. 5. Heating control from the parametric steady-state solution.

tool to avoid experiments: instead of the expensive loop of test/error, we can take advantage of a numerical model. However, this expensive loop can also be implemented with a time-consuming model: the realistic try is simply replaced by its numerical equivalent. We perform a plethora of standard finite element computations depending on the set of parameters under consideration. These results are then stored in a kind of database.

To circumvent this issue, we benefit from the ability of the PGD to deal with extra-coordinates. That way, we are able to build a multi-parametric chart in a single computation, providing the solution for any value of the parameters in given domains, instead of running a computation for each possible combination.

For instance, we can construct a multi-parametric solution giving the temperature at any point in the domain (x, z), for any power of the source P_w and for any laying velocity v, i.e. $u = u(x, z, P_w, v)$.

Within the PGD framework the separated representation reads

$$u(x, z, P_w, v) \approx \sum_{i=1}^{N} X_i(x) \cdot Z_i(z) \cdot \mathcal{P}_i(P_w) \cdot \mathcal{V}_i(v) \tag{6}$$

Denoting by u_n the rank-n solution, the approximation u_{n+1} results from

$$(X_{n+1}(x), Z_{n+1}(z), \mathcal{P}_{n+1}(P_w), \mathcal{V}_{n+1}(v)) = \underset{(X,Z,\mathcal{P},\mathcal{V})}{\operatorname{argmin}} \|\mathbb{L}^h (u_n + X(x) \cdot Z(z) \cdot \mathcal{P}(P_w) \cdot \mathcal{V}(v)) - f\| \tag{7}$$
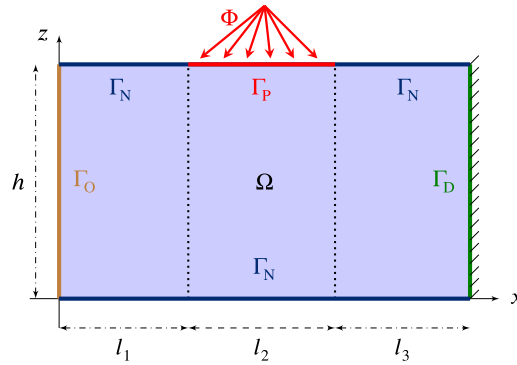
Fig. 6. New domain of study.

Table 2
Mapping table.

Matrix	Integral	Description
$\mathbb{K}^u$	$\int_\Omega \mathbf{K}\nabla u \cdot \nabla u^\star \, d\omega$	Stiffness matrix for u
$\mathbb{A}^u$	$\rho C_p \int_\Omega \mathbf{V} \cdot \nabla u u^\star \, d\omega$	Convection term matrix for u
$\mathbb{B}_P^{up}$	$\int_{\Gamma_P} u p^\star \, d\gamma$	Coupling matrix on Γ_P
$\mathbb{K}^p$	$\int_\Omega \mathbf{K}\nabla p \cdot \nabla p^\star \, d\omega$	Stiffness matrix for p
$\mathbb{A}^p$	$\rho C_p \int_\Omega \mathbf{V} \cdot \nabla p p^\star \, d\omega$	Convection term matrix for p
$\mathbb{B}_O^p$	$\rho C_p v \int_{\Gamma_O} p p^\star \, d\gamma$	Boundary condition on Γ_O for p
$\mathbb{B}_P^p u_d$	$\int_{\Gamma_P} u_d p^\star \, d\gamma$	Right hand side term for p

where $\mathbb{L}^h$ is the discretised operator from Eq. (2). The enrichment is computed using a fixed-point algorithm, where the minimisation of the above expression is carried out in turn on each of the four functions as previously described. The interested reader can refer to the primer [24] for the details on the separated representation constructor.

If one is interested by ensuring the target temperature u_{opt} at the hottest point $(x_{\mathrm{opt}}, z_{\mathrm{opt}})$ (which depends on the process parameters and can be determined by post-processing the PGD result), it suffices to extract from $u(x, z, P_{\mathrm{w}}, v)$ the following parametric solutions:

$$u\left(x = x_{\mathrm{opt}}, z = z_{\mathrm{opt}}, P_{\mathrm{w}}, v\right) \tag{8}$$

Fig. 4 depicts $u(P_{\mathrm{w}}, v)$, that is, the highest temperature, as a function of the heating power and the line velocity.

As soon as the target temperature u_{opt} is selected, it suffices extracting from $u(P_{\mathrm{w}}, v)$ the curve $u(P_{\mathrm{w}}, v) = u_{\mathrm{opt}}$, which corresponds to an isothermal curve.

Such static parametric solutions can be used for controlling transient regimes. We consider the typical velocity–time profile shown in Fig. 5 (blue curve), and we want to adjust the laser power in order to ensure a constant target temperature $u_{\mathrm{opt}} = 400$ K. Using the parametric solution $u(P_{\mathrm{w}}, v)$, we get an ideal power profile depicted by the red curve.

To check the efficiency of our method, this varying heat source is then used within a transient solver. The temperature at the point of interest $(x_{\mathrm{opt}}, z_{\mathrm{opt}})$ is represented by the green curve.

Thus, despite the fact of using a parametric solution computed under the stationary constraint, the control seems quite good because, when considering both inputs (the laser power and the line velocity), the solution to the thermal problem computed using a transient solver (green curve) remains very close to the target temperature.

We also showed in a former work how to improve these results using transient charts [15], especially when acceleration is quite important.

5. Setting up the optimal system

Another way to control the AFP process is to use optimal control theory. Within the AFP process, a heat flux provided by a laser melts the thermoplastic. The difficulty consists in determining the best power of the laser to reach an optimal temperature to melt the thermoplastic enough whilst avoiding degrading it.

Thus we consider the following cost-function J to be minimised, since the flux is applied only on Γ_P, part of the boundary

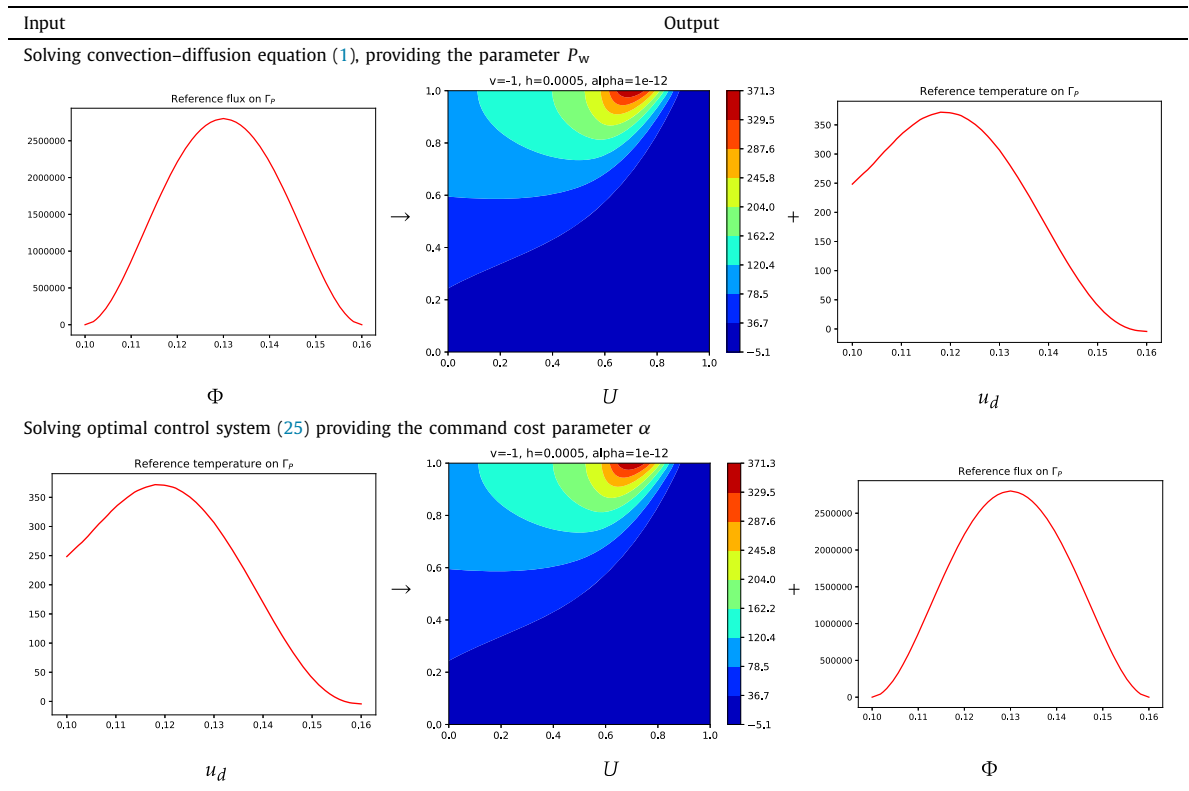**Table 3**
Solving the state equation and the control system.

Input	Output

Solving convection–diffusion equation (1), providing the parameter P_w

$$\Phi \qquad\qquad U \qquad\qquad u_d$$

Solving optimal control system (25) providing the command cost parameter α

$$u_d \qquad\qquad U \qquad\qquad \Phi$$

Table 4
Sets of parameters used for control.

Simulation	$V\,[\mathrm{m\,s^{-1}}]$	α	$h\,[\mathrm{m}]$
0	10^{-3}	1×10^{-9}	0.01
1	10^{-2}	1×10^{-9}	0.003
2	10^{-1}	1×10^{-12}	0.0015
3	1	1×10^{-12}	0.0005

$$J(u, \Phi) = \frac{1}{2} \int\limits_{\Gamma_P} (u - u_d)^2 + \alpha \Phi^2 \, \mathrm{d}\gamma \tag{9}$$

where α is the command cost parameter. In addition, u and Φ are linked through the next state convection–diffusion equation

$$\begin{cases} -\mathrm{div}\,(\mathbf{K}\nabla u) + \rho C_p \mathsf{V} \cdot \nabla u = 0 & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_D \\ \mathbf{K}\nabla u \cdot \vec{n} = 0 & \text{on } \Gamma_N \\ \mathbf{K}\nabla u \cdot \vec{n} = \Phi & \text{on } \Gamma_P \end{cases} \tag{10}$$

That way Φ is used as control and we want to reach u_d on boundary Γ_P. The domain of study is depicted in Fig. 2. Denoting by p the adjoint parameter, the corresponding Lagrangian writes ([25])

$$\mathcal{L}\,(u, \Phi, p) = J(u, \Phi) + \int\limits_{\Omega} p\left(-\mathrm{div}\,(\mathbf{K}\nabla u) + \rho C_p \mathsf{V} \cdot \nabla u\right) \mathrm{d}\omega \tag{11}$$

(a) Simulation 0

(b) Simulation 1

(c) Simulation 2

(d) Simulation 3
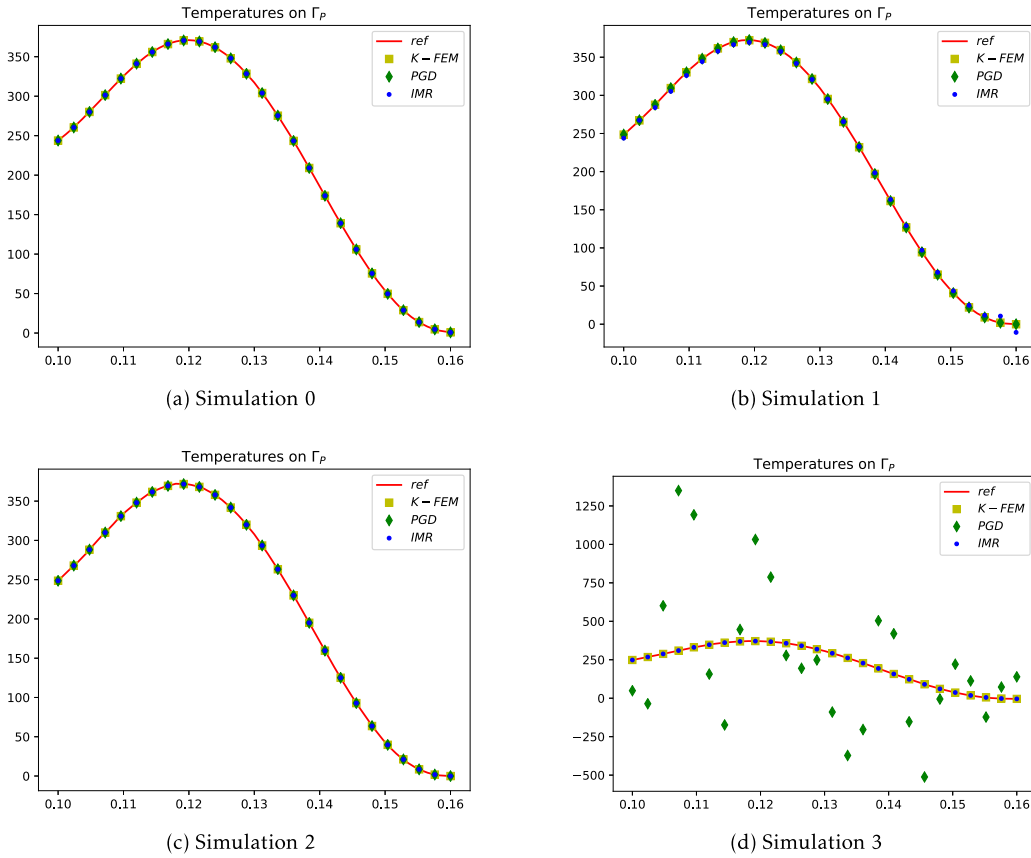
Fig. 7. Temperature on Γ_P.

Applying the Green–Ostrogradski's theorem, we get

$$\int_\Omega p\left(-\mathrm{div}\left(\mathbf{K}\nabla u\right)\right)\mathrm{d}\omega = \int_\Omega \nabla p\mathbf{K}\nabla u\,\mathrm{d}\omega - \int_\Gamma p\mathbf{K}\nabla u\cdot\vec{n}\,\mathrm{d}\gamma$$

$$= \int_\Omega u\left(-\mathrm{div}\mathbf{K}\nabla p\right)\mathrm{d}\omega + \int_\Gamma u\mathbf{K}\nabla p\cdot\vec{n}\,\mathrm{d}\gamma - \int_\Gamma p\mathbf{K}\nabla u\cdot\vec{n}\,\mathrm{d}\gamma$$

$$= \int_\Omega u\left(-\mathrm{div}\mathbf{K}\nabla p\right)\mathrm{d}\omega + \int_{\Gamma\backslash\Gamma_D} u\mathbf{K}\nabla p\cdot\vec{n}\,\mathrm{d}\gamma - \int_{\Gamma_D} p\mathbf{K}\nabla u\cdot\vec{n}\,\mathrm{d}\gamma - \int_{\Gamma_P} p\Phi\,\mathrm{d}\gamma$$

$$\rho C_p\int_\Omega p\mathsf{V}\cdot\nabla u\,\mathrm{d}\omega = -\rho C_p\int_\Omega u\nabla p\cdot\mathsf{V}\,\mathrm{d}\omega + \rho C_p\int_\Gamma pu\mathsf{V}\cdot\vec{n}\,\mathrm{d}\gamma$$

$$= -\rho C_p\int_\Omega u\nabla p\cdot\mathsf{V}\,\mathrm{d}\omega - \rho C_p\nu\int_{\Gamma_O} pu\,\mathrm{d}\gamma$$

Due to this last equation, we have to split the boundary Γ_N as depicted in Fig. 6, where we introduce the new boundary Γ_O.

To find a stationary point of the Lagrangian, we set

$$\begin{cases} \partial_u\mathcal{L}\left(u,\Phi,p\right)=0 & \text{(12)}\\ \partial_\Phi\mathcal{L}\left(u,\Phi,p\right)=0 & \text{(13)}\\ \partial_p\mathcal{L}\left(u,\Phi,p\right)=0 & \text{(14)} \end{cases}$$

(a) Simulation 0

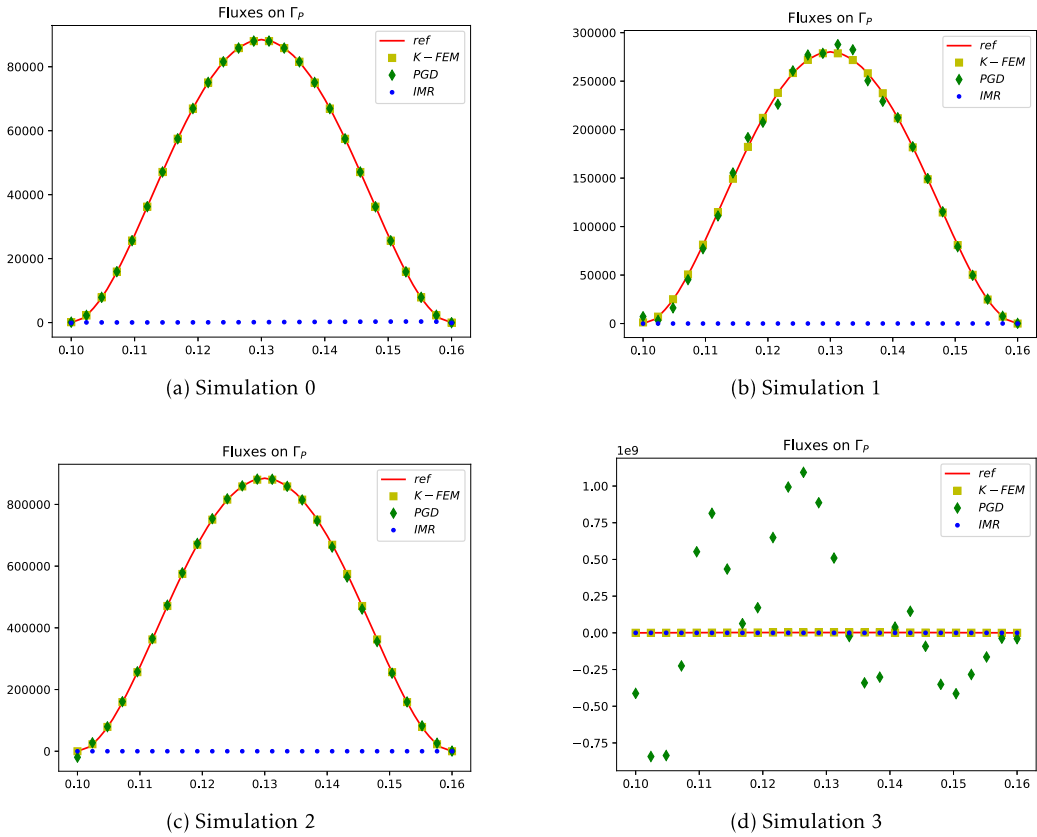(b) Simulation 1

(c) Simulation 2

(d) Simulation 3

Fig. 8. Flux on Γ_P.

By definition, $\partial_u \mathcal{L}(u, \Phi, p)$ is written as $\lim_{\lambda \to 0} \frac{1}{\lambda} (\mathcal{L}(u + \lambda \delta u, \Phi, p) - \mathcal{L}(u, \Phi, p))$, for any δu. Thus

$$\partial_u \mathcal{L}(u, \Phi, p) = \int_\Omega \delta u \left(-\text{div}(\mathbf{K} \nabla p)\right) \, d\omega - \rho C_p \int_\Omega \delta u \nabla p \cdot \mathsf{V} \, d\omega$$

$$+ \int_{\Gamma \setminus \Gamma_D} \delta u \mathbf{K} \nabla p \cdot \vec{n} \, d\gamma - \int_{\Gamma_D} p \mathbf{K} \nabla \delta u \cdot \vec{n} \, d\gamma - \rho C_p v \int_{\Gamma_O} p \delta u \, d\gamma + \int_{\Gamma_P} (u - u_d) \delta u \, d\gamma \quad (15)$$

Since this equation has to verify $\partial_u \mathcal{L}(u, \Phi, p) = 0$ for any δ_u, with different assumptions on this last term, it is possible to retrieve a strong formulation on p:

$$\begin{cases} -\text{div}(\mathbf{K} \nabla p) - \rho C_p \mathsf{V} \cdot \nabla p = 0 & \text{in } \Omega \\ p = 0 & \text{on } \Gamma_D \\ \mathbf{K} \nabla p \cdot \vec{n} = 0 & \text{on } \Gamma_N \\ \mathbf{K} \nabla p \cdot \vec{n} = \rho C_p v p & \text{on } \Gamma_O \\ \mathbf{K} \nabla u \cdot \vec{n} = -(u - u_d) & \text{on } \Gamma_P \end{cases} \quad (16)$$

Similarly,

$$\partial_\Phi \mathcal{L}(u, \Phi, p) = \lim_{\lambda \to 0} \frac{1}{\lambda} (\mathcal{L}(u, \Phi + \lambda \delta \Phi, p) - \mathcal{L}(u, \Phi, p)), \qquad \forall \delta \Phi$$

$$= \int_{\Gamma_P} (\alpha \Phi - p) \delta \Phi \, d\gamma \quad (17)$$

which leads to

$$p = \alpha \Phi \qquad \text{on } \Gamma_P \quad (18)$$

Table 5
Summary of key values.

Field	Simulation			
	0	1	2	3
$v\,[\mathrm{m\,s^{-1}}]$	0.001	0.01	0.1	1
$P_{\mathrm{w}}\,[\mathrm{W}]$	600	1897	6000	18970
$h\,[\mathrm{m}]$	0.01	0.003	0.0015	0.0005
Reference – K-FEM				
$U_{\max}\,[\mathrm{K}]$	371.13	372.28	372.14	371.77
$\Phi_{\max}\,[\mathrm{W}]$	88495.57	280235.99	884955.75	2802359.88
Ctrl – K-FEM				
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.12 (0.003)	372.28 (0)	372.14 (0)	371.77 (0)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	88494.46 (0.001)	280233.1 (0.001)	884955.95 (0)	2805516.8 (0.113)
Ctrl – K-FEM Uzawa				
iterations Uzawa	20	50	60	60
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.27 (0.038)	371.39 (0.239)	374.43 (0.615)	369.19 (0.694)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	88496.45 (0.001)	278023.92 (0.789)	864382.42 (2.325)	2762923.79 (1.407)
Ctrl – PGD Galerkin				
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.12 (0.003)	372.53 (0.067)	372.14 (0)	1459.59 (292.606)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	88490.81 (0.005)	290787.3 (3.765)	886127.75 (0.132)	1108779311.18 (39465.9)
Ctrl – PGD Galerkin Uzawa				
iterations Uzawa	20	50	60	60
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.72 (0.159)	373.59 (0.352)	403.53 (8.435)	348.57 (6.24)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	89022.12 (0.595)	278998.68 (0.442)	858910.0 (2.943)	2396624.61 (14.478)
Ctrl – PGD IMR				
iterations IMR	50	50	50	50
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.13 (0)	369.54 (0.736)	372.13 (0.003)	371.77 (0)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	333.69 (99.623)	63.17 (99.977)	38.54 (99.996)	9.77 (100)
Ctrl – PGD IMR Uzawa				
iterations Uzawa	20	50	60	60
iterations IMR	50	50	50	50
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	371.43 (0.081)	372.2 (0.021)	374.4 (0.607)	368.76 (0.81)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	88560.92 (0.074)	278838.63 (0.499)	864410.58 (2.322)	2763309.57 (1.393)
Ctrl – PGD IMR-P				
iterations IMR	50	50	50	50
$U_{\max}\,[\mathrm{K}]\ (\varepsilon_U\,[\%])$	370.9 (0.062)	370.31 (0.529)	371.76 (0.102)	368.61 (0.85)
$\Phi_{\max}\,[\mathrm{W}]\ (\varepsilon_\Phi\,[\%])$	88466.35 (0.033)	278822.02 (0.505)	878749.58 (0.701)	2734782.95 (2.411)

Lastly, writing the partial derivative of the Lagrangian according to p and setting it to 0 leads to the original state convection–diffusion equation (10).

Therefore, gathering all these equations gives the non-linear optimality system

$$
\begin{cases}
-\mathrm{div}\,(\mathbf{K}\nabla u) + \rho C_p \mathsf{V}\cdot\nabla u = 0 & \text{in } \Omega & \text{(a)}\\
-\mathrm{div}\,(\mathbf{K}\nabla p) - \rho C_p \mathsf{V}\cdot\nabla p = 0 & \text{in } \Omega & \text{(b)}\\
p = \alpha\Phi & \text{on } \Gamma_P & \text{(c)}\\
u = p = 0 & \text{on } \Gamma_D & \text{(d)}\\
\mathbf{K}\nabla u\cdot\vec{n} = \mathbf{K}\nabla p\cdot\vec{n} = 0 & \text{on } \Gamma_N & \text{(e)}\\
\mathbf{K}\nabla u\cdot\vec{n} = 0 & \text{on } \Gamma_O & \text{(f)}\\
\mathbf{K}\nabla p\cdot\vec{n} = \rho C_p v p & \text{on } \Gamma_O & \text{(g)}\\
\mathbf{K}\nabla u\cdot\vec{n} = \Phi & \text{on } \Gamma_P & \text{(h)}\\
\mathbf{K}\nabla p\cdot\vec{n} = -(u - u_d) & \text{on } \Gamma_P & \text{(i)}
\end{cases}
\tag{19}
$$

The interested reader can find more details in [25].

Given test-functions $u^\star$ and $p^\star$, the weak form of system (19) above reads

$$
\begin{cases}
\int_\Omega \mathbf{K}\nabla u\cdot\nabla u^\star\,\mathrm{d}\omega + \rho C_p \int_\Omega \mathsf{V}\cdot\nabla u u^\star\,\mathrm{d}\omega - \int_{\Gamma_P} \frac{1}{\alpha} p u^\star\,\mathrm{d}\gamma = 0 \quad \forall u^\star\\
\int_\Omega \mathbf{K}\nabla p\cdot\nabla p^\star\,\mathrm{d}\omega - \rho C_p \int_\Omega \mathsf{V}\cdot\nabla p p^\star\,\mathrm{d}\omega + \int_{\Gamma_P} u p^\star\,\mathrm{d}\gamma\\
\qquad - \rho C_p v \int_{\Gamma_O} p p^\star\,\mathrm{d}\gamma = \int_{\Gamma_P} u_d p^\star\,\mathrm{d}\gamma \quad \forall p^\star
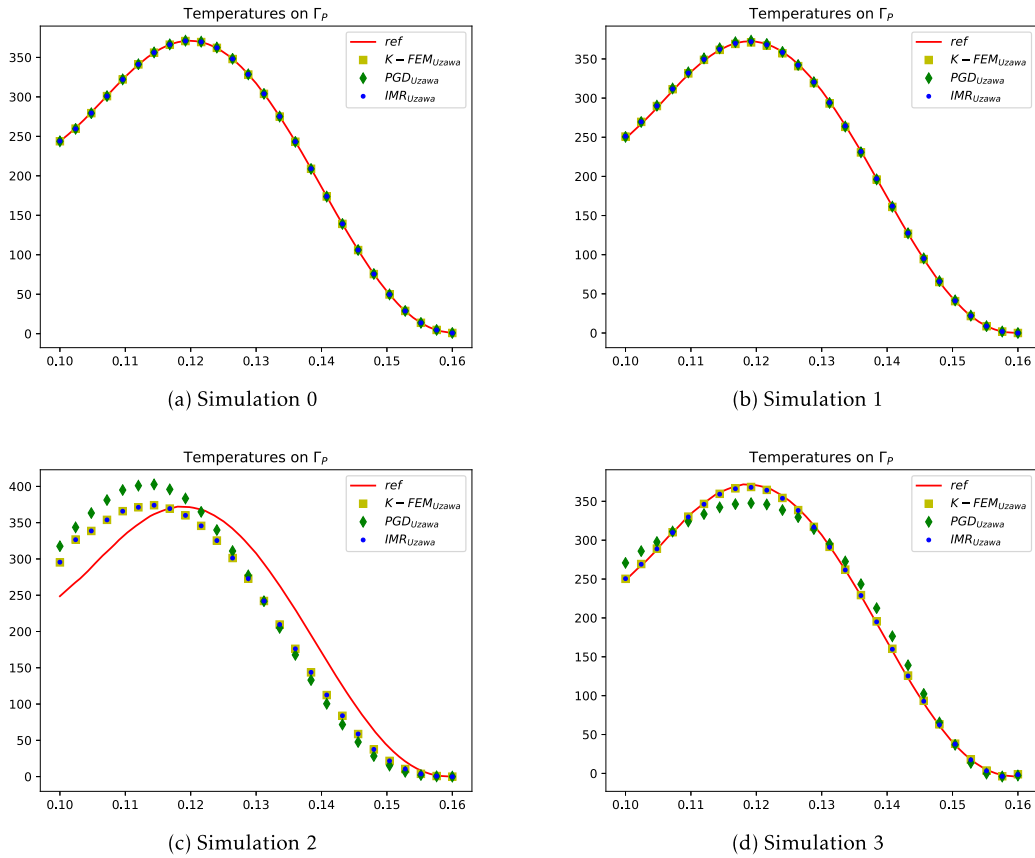\end{cases}
\tag{20}
$$

Fig. 9. Temperature on Γ_P with Uzawa.

In Eq. (20), fields u and p are coupled. The computation of such a problem can be achieved using mixed formulation, within a standard FEM as well as in the PGD framework as described thereafter.

The discrete form of the state variable $u\,(x, z)$ and the adjoint parameter $p\,(x, z)$ are expressed in tensor product form

$$U = \sum_{i}^{\infty} U_x^i \otimes U_z^i \quad \text{and} \quad P = \sum_{i}^{\infty} P_x^i \otimes P_z^i \tag{21}$$

The vector Ψ, which brings together nodal values of u and p, takes the form

$$\Psi = \begin{bmatrix} U \\ P \end{bmatrix} = \sum_{i}^{\infty} \begin{bmatrix} U_x^i \otimes U_z^i \\ P_x^i \otimes P_z^i \end{bmatrix} \tag{22}$$

The discretised weak form of Eq. (19) is written $\Psi^{\star\mathsf{T}} \mathsf{A} \Psi = \Psi^{\star\mathsf{T}} \mathsf{B}$, with the test function defined by

$$\Psi^{\star} = \begin{bmatrix} U_x^{\star} \otimes U_z + U_x \otimes U_z^{\star} \\ P_x^{\star} \otimes P_z + P_x \otimes P_z^{\star} \end{bmatrix} \tag{23}$$

Tensors A and B read

$$\mathsf{A} = \begin{bmatrix} \mathbb{K}^u + \mathbb{A}^u & \frac{-1}{\alpha} \mathbb{B}_P^{up} \\ \mathbb{B}_P^{up} & \mathbb{K}^p - \mathbb{A}^p - \mathbb{B}_O^p \end{bmatrix} \quad \text{and} \quad \mathsf{B} = \begin{bmatrix} 0 \\ \mathbb{B}_P^p u_d \end{bmatrix} \tag{24}$$

where the elements of these tensors are defined in Table 2.

However, as we really want to reach the desired state on Γ_P, we have to set α, the command cost parameter, at a very low value (of the order 1×10^{-10}, depending on the velocity). Consequently, tensor A is highly non-symmetric.

(a) Simulation 0

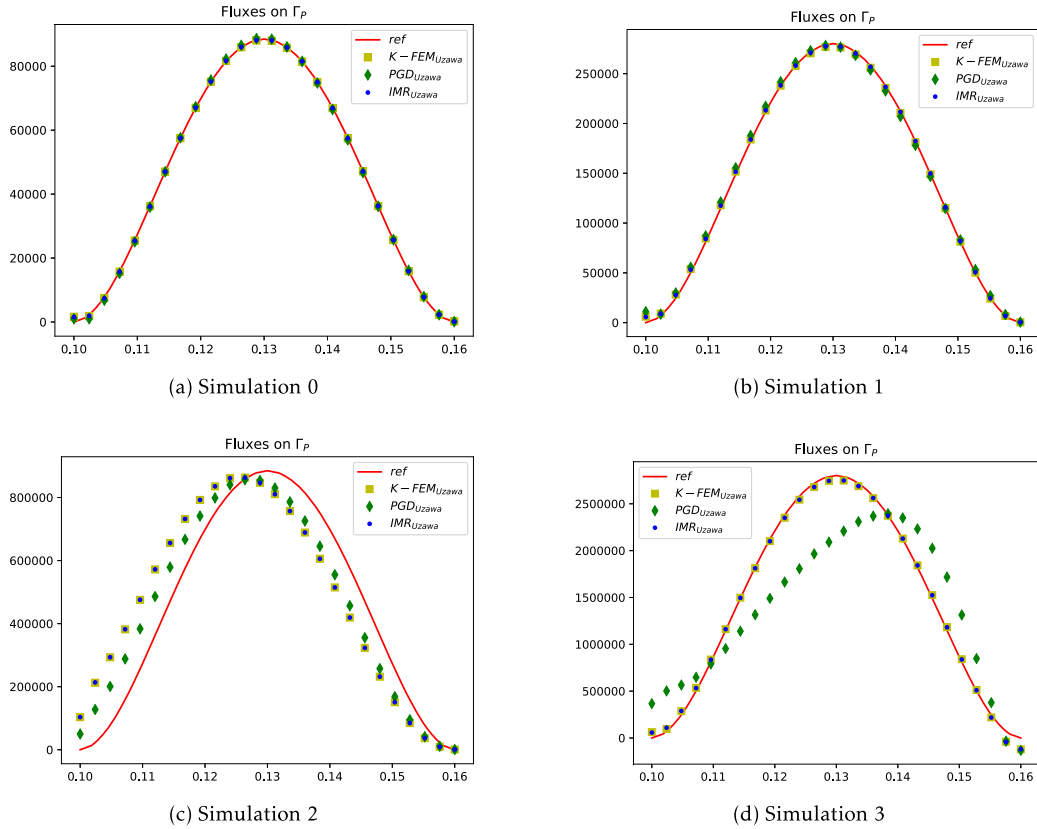(b) Simulation 1

(c) Simulation 2

(d) Simulation 3

Fig. 10. Flux on Γ_P with Uzawa.

In order to retrieve the classical form of the saddle point problem, we introduce $\psi = \frac{-p}{\alpha}$, leading to the weak form

$$\begin{cases} \int_\Omega \mathbf{K}\nabla u \cdot \nabla u^\star \, d\omega + \rho C_p \int_\Omega \mathbf{V} \cdot \nabla u u^\star \, d\omega + \int_{\Gamma_P} \psi u^\star \, d\gamma = 0 \quad \forall u^\star \\ -\alpha \int_\Omega \mathbf{K}\nabla \psi \cdot \nabla \psi^\star \, d\omega + \alpha\rho C_p \int_\Omega \mathbf{V} \cdot \nabla \psi \psi^\star \, d\omega + \int_{\Gamma_P} u \psi^\star \, d\gamma \\ \quad + \alpha\rho C_p v \int_{\Gamma_O} \psi \psi^\star \, d\gamma = \int_{\Gamma_P} u_d \psi^\star \, d\gamma \quad \forall \psi^\star \end{cases}$$ (25)

Adapting the integrals definitions in Table 2, this system can be written in the discretised form

$$\begin{bmatrix} U^{\star\mathsf{T}} & \psi^{\star\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbb{K}^u + \mathbb{A}^u & \mathbb{B}_P^{u\psi} \\ \mathbb{B}_P^{u\psi} & -\alpha\left(\mathbb{K}^\psi - \mathbb{A}^\psi - \mathbb{B}_O^\psi\right) \end{bmatrix} \begin{bmatrix} U \\ \psi \end{bmatrix} = \begin{bmatrix} U^{\star\mathsf{T}} & \psi^{\star\mathsf{T}} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbb{B}_P^\psi u_d \end{bmatrix}$$ (26)

In what follows, tensors A and B and vector Ψ are defined from this Eq. (26).

We use this last formulation to solve Eq. (19) with both K-FEM and PGD, for the four considered simulations.

In addition, we tried the Ideal minimal residual-based proper generalized decomposition (IMR) introduced in [26,27]. This solver computes the solution to $A\Psi = B$ iteratively, through the system given in Eq. (27).

$$\begin{cases} AA^\mathsf{T} y = B - A\Psi^k \\ \Psi^{k+1} = \Psi^k + A^\mathsf{T} y \end{cases}$$ (27)

Let us recall that Φ is initially provided as an input of the K-FEM solver and leads to reference solutions. From these results, we extract the temperature on boundary Γ_P defining the desired state u_d in Eq. (25).

Using these techniques, we solve Eq. (19) in order to obtain, not only the temperature field, but also the required flux Φ. This is illustrated in Table 3. On every temperature and flux plots, the abscissa corresponds to the boundary Γ_P.

Moreover, P_w is no longer an input parameter: the four simulations take their parameters in Table 4, where the indices of simulations refer to the same speed and height as in Table 1.

We gather in Fig. 7 (respectively Fig. 8) the solutions (resp. the fluxes) on boundary Γ_P for each considered simulation.

We report in Table 5 some key values compared to references. In this table, fields ε_U and ε_Φ give the relative errors on the maximum temperature and maximum power, expressed in percent and computed as
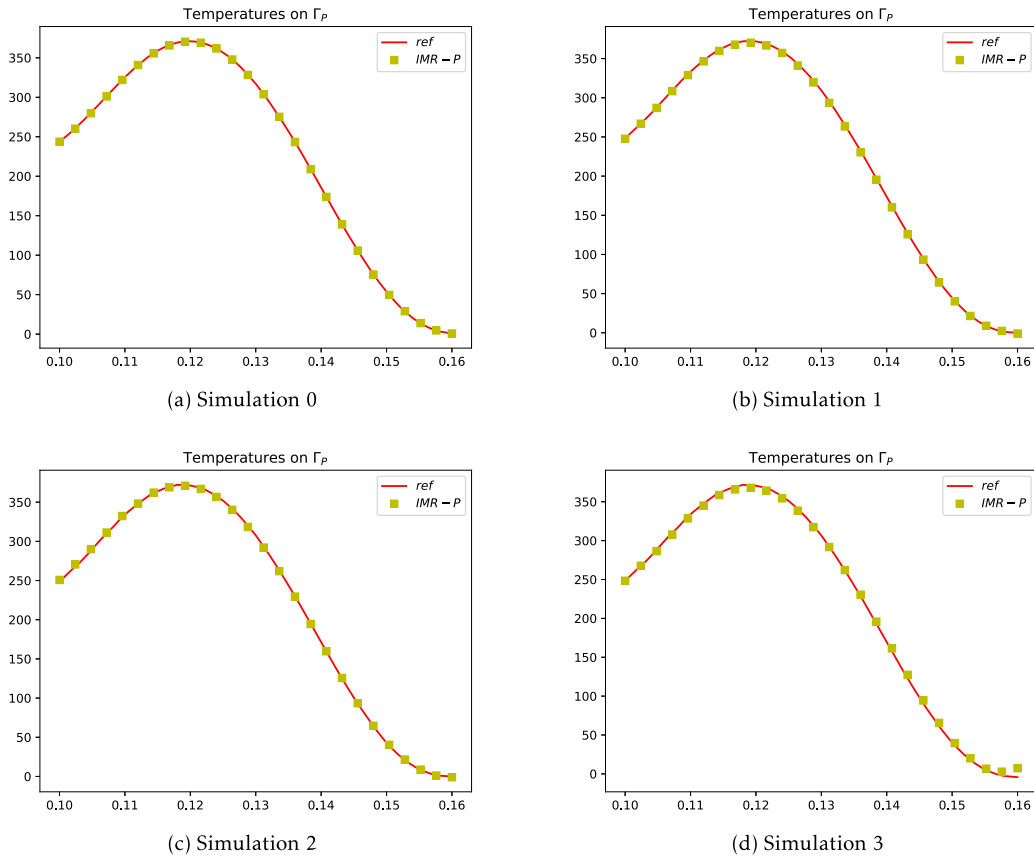
Fig. 11. Temperature on Γ_P with modified IMR.

$$\varepsilon_U = \frac{\left|U_{\max}^{\text{K-FEM}} - U_{\max}\right|}{U_{\max}^{\text{K-FEM}}}; \quad \varepsilon_\Phi = \frac{\left|\Phi_{\max}^{\text{K-FEM}} - \Phi_{\max}\right|}{\Phi_{\max}^{\text{K-FEM}}} \tag{28}$$

As we can see, the PGD algorithm does not provide the relevant solution, at least for the computed flux, when convection increases. Actually, the optimality system leads to the right temperature on boundary Γ_P, but the needed heat flux to reach it is far away from the reference one.

The IMR solutions provide the right temperature on boundary Γ_P (corresponding to the desired state u_d). However, the required fluxes are very far from the reference ones, even at low speed.

6. Unmixing the optimality system

As seen previously, applying the PGD framework on the optimality system (19) can produce weird results, not only because of the value of velocity, but also due to the command cost parameter.

In this section, we propose to solve this system uncoupling its equations. For this purpose, we use Uzawa's technique [28,29].

Given an initial guess $\psi^{(0)}$ for ψ, Uzawa's method consists, in our case, of the following coupled iteration:

$$\begin{cases} \left(\mathbb{K}^u + \mathbb{A}^u\right) u^{(k+1)} = \mathbb{B}_P^{u\psi} \psi^{(k)} \\ \psi^{(k+1)} = \psi^{(k)} + \omega \left(\mathbb{B}_P^{u\psi} u^{(k+1)} - \alpha \left(\mathbb{K}^\psi - \mathbb{A}^\psi - \mathbb{B}_O^\psi\right) p^{(k)} - u_d\right) \end{cases} \tag{29}$$

where $\omega > 0$ is a relaxation parameter.

This coupled iteration is computed within a fixed-point loop, each field u and ψ being solved by PGD.

As in the previous section, we gather in Fig. 9 (respectively Fig. 10) the solutions (resp. the flux) on boundary Γ_P for each considered speed, computed with any of the methods (excepted with the PGD applied on the optimality system).

Table 5 summarises the previous simulations, collecting some key values.

As we can see, the Uzawa's technique drastically improves the results, even if there is no perfect match with the reference solutions at the higher speed. In addition, this method increases the computational cost, since it uses an extra loop, which is quite prohibiting coupled with the IMR formulation. For instance, instead of 50 IMR iterations, we need these 50 steps
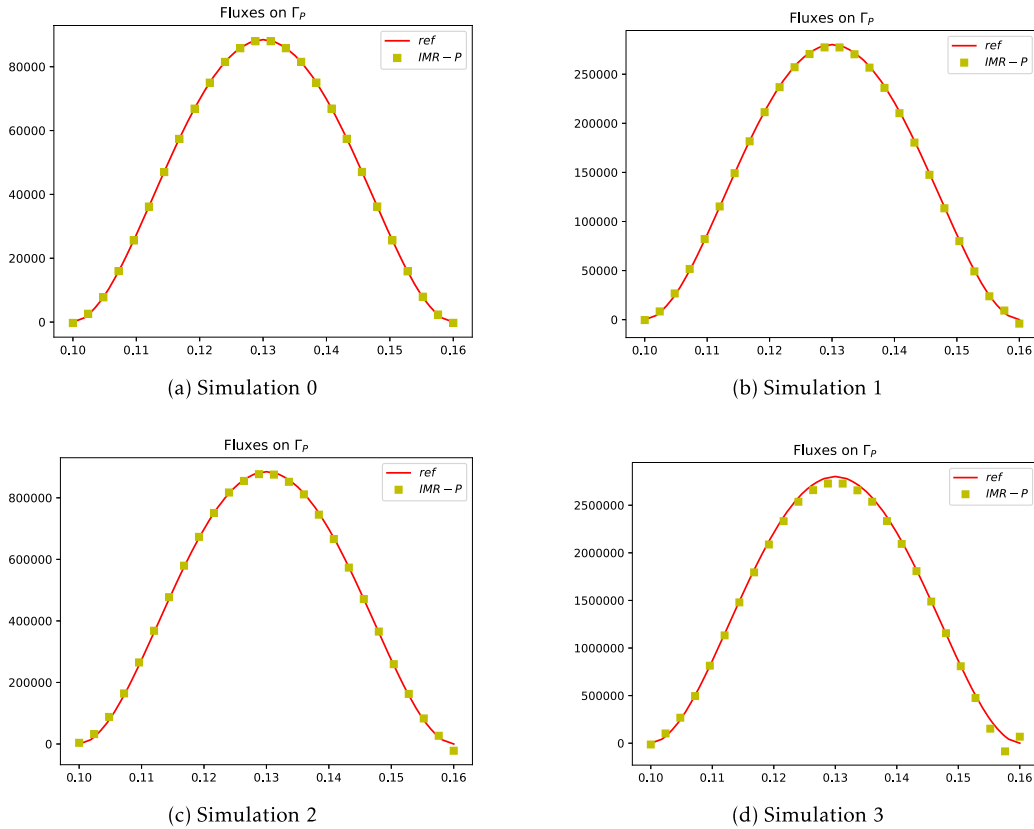
(a) Simulation 0

(b) Simulation 1

(c) Simulation 2

(d) Simulation 3

Fig. 12. Flux on Γ_P with modified IMR.

within every 20 Uzawa's iterations. As we need approximately two minutes to compute about thirty PGD modes, this PGD IMR Uzawa solver takes $50 \times 20 \times 2$ min ≈ 33 h, while the PGD Uzawa solver leads to the solution in less than one hour.

7. Modifying the IMR method

We have seen in the previous results that the IMR algorithm is not usable here, since it requires the computation of AA^T, which is ill-conditioned because of the particularly low value of the command cost parameter α.

We propose here an adaptation of this technique, called *IMR-P* for "preconditioner". Actually, instead of solving

$$\begin{cases} AA^T y = B - A\Psi^k \\ \Psi^{k+1} = \Psi^k + A^T y \end{cases} \tag{30}$$

we now write

$$\begin{cases} \tilde{A}\tilde{A}^T y = B - A\Psi^k \\ \Psi^{k+1} = \Psi^k + \tilde{A}^T y \end{cases} \tag{31}$$

where $\tilde{A}$ is an "approximation" of A. In this paper, we solve Eq. (31) with

$$\tilde{A} = \begin{bmatrix} \mathbb{K}^u + \mathbb{A}^u & 0 \\ 0 & -\beta\mathbb{M}^\psi \end{bmatrix} \tag{32}$$

$\mathbb{M}^\psi$ being the mass matrix on ψ and $\beta = 100$ a relaxation parameter ensuring the convergence of the algorithm.

Figs. 11 and 12 show the temperatures and the fluxes computed on boundary Γ_P for each considered simulation.

Thus, this technique avoids the use of Uzawa's loop and its cost, whilst providing good results compared to the reference. Moreover, with the IMR formulation, there is no link between the number of iterations needed and the velocity of the heat source.

8. Conclusion

This paper shows that it is possible to solve an optimal problem with the so-called Proper Generalised Decomposition, not directly, but through a kind of relaxation method, leading to a computational extra cost. This drawback can be avoided by slightly modifying the operator in the Ideal minimal residual-based proper generalized decomposition.

Future work will be the exploration of other approximations of tensor A in the writing of the IMR-P method. We plan, for instance, to use the nearest Kronecker product, since this technique allows the reduction of the operators in the expression of A, leading to a theoretical speed up in the PGD computation.

Moreover, for now the objective function is simply taken as a temperature profile within the heated zone. It is clear that this function has to take into account the physics about reticulation of the composite, as we aim to deal with the *in situ* consolidation.

References

[1] D.H.-J.A. Lukaszewicz, C. Ward, K.D. Potter, The engineering aspects of automated prepreg layup: history, present and future, Composites, Part B, Eng. 43 (3) (2012) 997–1009.

[2] F.O. Sonmez, H.T. Hahn, M. Akbulut, Analysis of process-induced residual stresses in tape placement, J. Thermoplast. Compos. Mater. 15 (2002) 525–544.

[3] R. Pitchumani, S. Ranganathan, R.C. Don, J.W. Gillespie, M.A. Lamontia, Analysis of transport phenomena governing interfacial bonding and void dynamics during thermoplastic tow-placement, Int. J. Heat Mass Transf. 39 (9) (1996) 1883–1897.

[4] R. Schledjewski, M. Latrille, Processing of unidirectional fiber reinforced tapes—fundamentals on the way to a process simulation tool (ProSimFRT), Compos. Sci. Technol. 63 (14) (2003) 2111–2118.

[5] M.A. Lamontia, M.B. Gruber, J. Tierney, J.W. Gillespie, B.J. Jensen, R.J. Cano, Modeling the accudyne thermoplastic in situ ATP process, in: Jec-Sampe, 2009, pp. 1–8.

[6] F. Chinesta, A. Leygue, B. Bognet, C. Ghnatios, F. Poulhaon, F. Bordeu, A. Barasinski, A. Poitou, S. Chatel, S. Maison-Le-Poec, First steps towards an advanced simulation of composites manufacturing by automated tape placement, Int. J. Mater. Form. 7 (2014) 81–92.

[7] 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-Newton. Fluid Mech. 139 (3) (2006) 153–176.

[8] 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 modelling of complex fluids. Part II: transient simulation using space–time separated representations, J. Non-Newton. Fluid Mech. 144 (2–3) (2007) 98–121.

[9] É. Prulière, F. Chinesta, A. Ammar, On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition, Math. Comput. Simul. 81 (4) (2010) 791–810.

[10] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, A. Poitou, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, Comput. Methods Appl. Mech. Eng. 201–204 (2012) 1–12, accepted.

[11] C. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, A. Poitou, P. Breitkopf, P. Villon, Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion, Composites, Part A, Appl. Sci. Manuf. 42 (9) (2011) 1169–1178.

[12] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, F. Chinesta, Proper Generalized Decomposition based dynamic data-driven control of thermal processes, Comput. Methods Appl. Mech. Eng. 213–216 (2012) 29–41.

[13] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. González, I. Alfaro, A. Ammar, A. Huerta, PGD-based computational vademecum for efficient design, optimization and control, Arch. Comput. Methods Eng. 20 (1) (2013) 31–59.

[14] C. Dedieu, A. Barasinski, F. Chinesta, J.-M. Dupillier, About the origins of residual stresses in in situ consolidated thermoplastic composite rings, Int. J. Mater. Form. 10 (5) (2017) 779–792.

[15] N. Bur, P. Joyot, C. Ghnatios, P. Villon, E. Cueto, F. Chinesta, On the use of model order reduction for simulating automated fibre placement processes, Adv. Model. Simul. Eng. Sci. 3 (1) (2016) 4.

[16] R.E. Lynch, J.R. Rice, D.H. Thomas, Tensor product analysis of partial difference equations, Bull. Amer. Math. Soc. 70 (1964) 378–384.

[17] R.E. Lynch, J.R. Rice, D.H. Thomas, Direct solution of partial difference equations by tensor product methods, Numer. Math. 6 (1) (1964) 185–199.

[18] M. Marcus, Basic Theorems in Matrix Theory, U.S. Government Printing Office, 1960.

[19] P.R. Halmos, Finite-Dimensional Vector Spaces, 2nd edition, Undergrad. Texts Math., Springer New York, New York, NY, 1974.

[20] N. Bur, Développement d'algorithmes de réduction de modèles pour l'optimisation du procédé PFR, PhD thesis, UTC, Compiègne, France, 2015.

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

[22] F. Chinesta, A. Ammar, A. Leygue, R. Keunings, An overview of the proper generalized decomposition with applications in computational rheology, J. Non-Newton. Fluid Mech. 166 (11) (2011) 578–592.

[23] F. Chinesta, P. Ladevèze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395–404.

[24] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, Springer, 2014.

[25] J.-L. Lions, Optimal Control of Systems Governed by Partial Differential Equations, Springer-Verlag, 1971.

[26] M. Billaud-Friess, A. Nouy, O. Zahm, A tensor approximation method based on ideal minimal residual formulations for the solution of high-dimensional problems, ESAIM, Math. Model. Numer. Anal. 48 (6) (2014) 1777–1806.

[27] L. Boucinha, A. Ammar, A. Gravouil, A. Nouy, Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models – application to transient elastodynamics in space–time domain, Comput. Methods Appl. Mech. Eng. 273 (2014) 56–76.

[28] H. Uzawa, Iterative methods in concave programming, in: K.J. Arrow, L. Hurwicz, H. Uzawa (Eds.), Studies in Linear and Nonlinear Programming, Stanford University Press, Stanford, CA, USA, 1958, pp. 154–165.

[29] R. Simon, Multigrid Solvers for Saddle Point Problems in PDE-Constrained Optimization Dissertation, PhD thesis, Johannes Kepler Universität, Linz, Austria, 2008.