



**HAL**  
open science

# Efficient Mold Cooling Optimization by Using Model Reduction

Fabrice Schmidt, Nicolas Pirc, Marcel Mongeau, Francisco Chinesta

► **To cite this version:**

Fabrice Schmidt, Nicolas Pirc, Marcel Mongeau, Francisco Chinesta. Efficient Mold Cooling Optimization by Using Model Reduction. International Journal of Material Forming, 2011, 4 (1), pp.73-82. 10.1007/s12289-010-0988-5 . hal-01006784

**HAL Id: hal-01006784**

**<https://hal.science/hal-01006784>**

Submitted on 6 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Efficient mold cooling optimization by using model reduction

Fabrice Schmidt · Nicolas Pirc · Marcel Mongeau ·  
Francisco Chinesta

**Abstract** Optimization and inverse identification are two procedures usually encountered in many industrial processes reputed gourmand for the computing time view point. In fact, optimization implies to propose a trial solution whose accuracy is then evaluated, and if needed it must be updated in order to minimize a certain cost function. In the case of mold cooling optimization the evaluation of the solution quality needs the solution of a thermal model, in the whole domain and during the thermal history. Thus, the optimization process needs several iterations and then the computational cost can become enormous. In this work we propose the use of model reduction for accomplishing this kind of simulations. Thus, only one thermal model is solved using the standard discretization technique. After that, the most important modes defining the temperature evolution are extracted by invoking the proper orthogonal decomposition, and all the other thermal model solutions are performed by using the reduced order approximation basis just extracted. The CPU time savings can be impressive.

**Keywords** BEM · Optimisation · Model reduction · Injection moulding

## Introduction

Process description—injection molding

About 30% of the annual polymer production is transformed by injection molding. Injection molding is a cyclic process of forming a plastic into a desired shape by forcing the molten polymer under pressure into a hollow cavity [1, 25]. For thermoplastic polymers, the solidification is achieved by cooling. Typical cycle times range from 1 to 100 s and depend mainly on the cooling time. The complexity of molded parts is virtually unlimited, sizes may range from very small (<1 mm) to very large (>1 m), with an excellent control of tolerances.

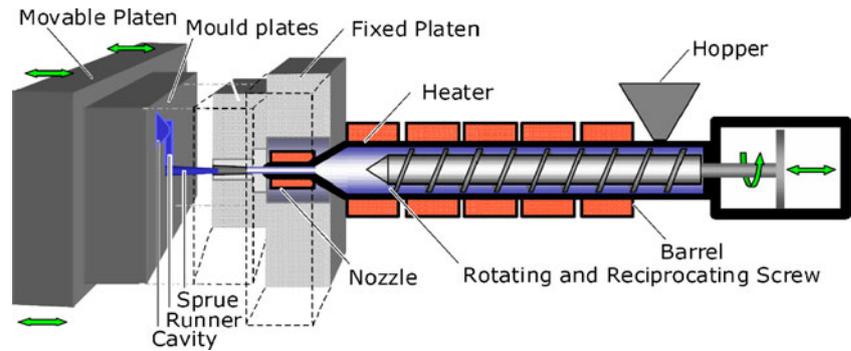
The injection molding equipment

The reciprocation screw injection molding machine is the most common injection unit used (Fig. 1). These machines consist of two basic parts, an injection unit and a clamping unit. The injection unit melts the polymer resin and injects the polymer melt into the mold. Its screw rotates and axially reciprocates to melt, mix, and pump the polymer. A hydraulic system controls the axial reciprocating of the screw, allowing it to act like a plunger, moving the melt forward for injection. The clamping unit holds the mold together, opens and closes it automatically, and ejects the finished part.

Importance of cooling step for manufacturing injected parts

Part cooling during injection molding is the critical step as it is the most time consuming. An inefficient mold cooling may have dramatic consequences on cycle time and part

**Fig. 1** Injection molding machine with reciprocating screw [39]



quality and may require expensive mold rectification. Depending on the wall thickness of the molded parts, it usually takes the major portion of the cycle time to evacuate the heat since polymers are bad conductors as thermal conductivity ranges from 0.1 W/mK to 1.8 W/mK [25]. The cooling cycle can represent more than 70% of the injection cycle [18, 23]. The cooling rate is an important factor for productivity, and important benefits can be achieved by decreasing the cooling time of parts with hot zones badly cooled. A bad design of the cooling channels may generate zones with higher temperatures in the mold, increasing the cooling time.

In addition, different types of injection defects due to a bad thermal regulation of the mold can appear: dimensional defects, structural defects and aspect defects [21, 40].

In order to reduce mold and production costs, an automatic optimization of cooling device geometry and processing parameters (temperature, flow rate...) may be developed. The optimization procedure necessitates to compute numerous transient heat balance problem (eventually non linear). For solving the thermal problem, we need an efficient meshing technique. Boundary elements method (BEM) is well adapted for such a problem because it only requires a surfacing mesh. The displacement of cooling channels after each optimization iteration is then facilitated (no remeshing). In addition, reduced modeling is useful in order to reduce CPU time of the direct

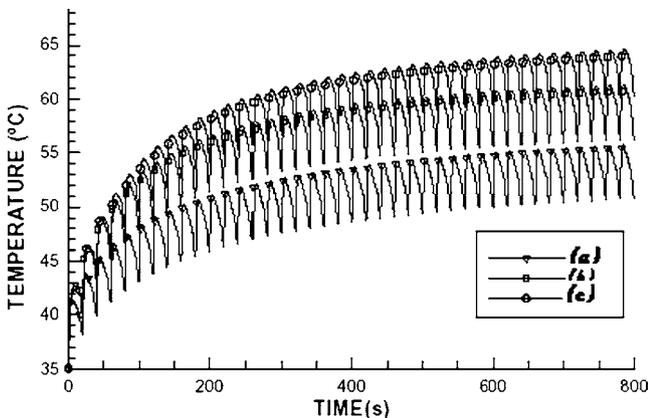
computations, particularly for 3D computations. Injection molding process is a cycling one; which imply computation of numerous cycles. On Fig. 2, an example of temperature history of 40 cycles is plotted versus time. Curves (a) and (b) give respectively the maximum of the temperature in the cavity before and after optimization. Curve (c) represents the average temperature at the cavity surface after optimization.

## Mold cooling optimization

### Introduction

Several CAD and simulation tools are available to help designing the cooling system of an injection mold. Simulation of heat transfer during injection can be used to check a mold design or study the effect of a parameter (geometry, materials...) on the cooling performance of the mold. Several numerical methods such as Finite Elements Method (FEM) [6] or Boundary Element Method (BEM) [4, 7, 8] can be used. Bikas et al. [5] used C-Mold® simulations and design of experiments to find expressions of mean temperature and temperature variation as functions of geometry parameters of the mold.

Numerical simulation can also be used to perform an automatic optimization of the mold cooling. Numerical simulation is used to solve the thermal equations and evaluate a cost function related to productivity or part quality. An optimization method is used to modify the parameters and improve the thermal performance of the mold. Tang et al. [37] used 2D transient FEM simulations coupled with Powell's optimization method [10] to optimize the cooling channel geometry to get uniform temperature in the polymer part. Huang et al. [13] used 2D transient FEM simulations to optimize the use of mold materials according to part temperature uniformity or cycle time. Park et al. [27] developed 2D and 3D stationary BEM simulations in the injection molds coupled with 1D transient analytical computation in the polymer part (throughout the thickness). The heat transfer integral equation is differentiated to get sensitivities of a cost function to the parameters. The



**Fig. 2** Temperature history of the first 40 cycles

calculated sensitivities are then used to optimize the position of linear cooling channels for simple shapes (sheet, box).

In the next section, we present the use of Boundary Element Method (BEM) and DRM applied to transient heat transfer of injection molds. The BEM software, [20, 30, 31], was combined with an adaptive reduced modeling (widely described later). This procedure will be fully described in ‘‘Reduced modeling’’, it allows reducing considerably the computing time during the linear system solution in transient problem. Then, we present a practical methodology to optimize both the position and the shape of the cooling channels in injection molding processes (‘‘Application of BEM reduced model to mold cooling optimization’’). We couple the direct computation with an optimization algorithm: Sequential Quadratic Programming.

### BEM for transient heat balance equation

Using BEM, only the boundary of the domain has to be meshed and internal points are explicitly excluded from the solution procedure. An interesting side effect is the considerable reduction in size of the linear system to be solved [30]. The transient heat conduction in a homogeneous isotropic body  $\Omega$  is described by the diffusion equation where  $\alpha$  is the material thermal diffusivity assumed constant:

$$\Delta T(\mathbf{x}, t) = \frac{1}{\alpha} \frac{\partial T(\mathbf{x}, t)}{\partial t} \quad (1)$$

We define the initial conditions and the boundary conditions (Fig. 3) as:

$$\begin{cases} T(\mathbf{x}, t=0) = T^o(\mathbf{x}) \\ -\lambda \nabla T \cdot \mathbf{n} = q_p \quad \forall \mathbf{x} \in \Gamma_p \\ -\lambda \nabla T \cdot \mathbf{n} = h_c(T - T_c) \quad \forall \mathbf{x} \in \Gamma_c \\ -\lambda \nabla T \cdot \mathbf{n} = h_a(T - T_a) \quad \forall \mathbf{x} \in \Gamma_M \end{cases} \quad (2)$$

Where  $\lambda$  is the thermal conductivity (the medium is assumed homogeneous and isotropic),  $\Gamma_p$  is the boundary of the cavity surface (plastic part),  $\Gamma_c$  the boundary of the cooling channels and  $\Gamma_M$  the mold exterior surface. The temperature of the coolant is  $T_c$  and  $h_c$  and  $h_a$  represent the heat transfer coefficient between the mold and the coolant and the mold and the ambient air at temperature  $T_a$

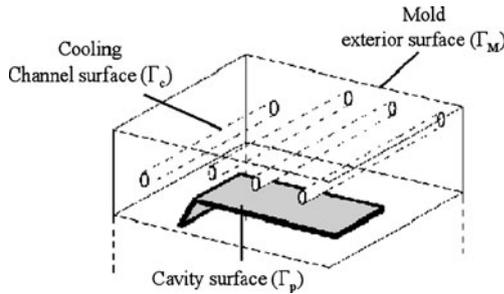


Fig. 3 Boundary conditions applied to the mold

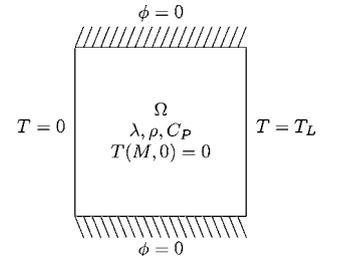
respectively. In order to avoid multi-domains calculation and save computation time, the plastic part is taken into account via a heat flux  $q_p$  imposed on the mold cavity surface. The flux density  $q_p$  is calculated from the cycle time and polymer properties [20, 29].

Different strategies are possible to solve such problems using BEM. Pasquetti et al. [28] propose to use space and time Green’s function. Another solution should be to apply Laplace [36] or Fourier [11] transforms on time variable before spatial integration. In this section, we will use only space Green’s function inspired from stationary heat transfer problem (i.e. Laplace’s equation). To express the domain integral resulting from time derivative in terms of equivalent boundary integrals, we introduce the DRM approximation. The DRM consists in seeking the solution as series of particular solutions and interpolated on  $N$  points inside and on the boundary of the domain. The interpolation function used in this paper is the linear radial function. In what follows we are not considering the detail of the Boundary Element discretization technique that has been addressed in many papers and books. Anyways, DRM procedure used in this paper is detailed in the paper of Mathey et al. [20]. As we knew from the literature that instabilities may occur using this method, and particularly in case of thermal shock, a parametric study has been performed in a 2D configuration [19]. A thermal shock was simulated by applying on one side of a square initially at zero temperature a temperature  $T_L$ . The opposite side is kept at zero temperature and the upper and lower sides are assumed thermally insulated (Fig. 4): where the value of different parameters are:  $\alpha = 10^{-5} \text{m}^2/\text{s}$ ,  $L=0.1 \text{ m}$  and  $T_L=200^\circ\text{C}$ . The analytical solution of such problem could be found in Özisik [24]. Different parameters have been tested such as time step, boundary mesh size, radial function, time integration scheme. From this study, we have highlighted that the influence of the parameters is only significant at the beginning of heating as it is illustrated on Fig. 5 ( $x=0.05 \text{ m}$ ).

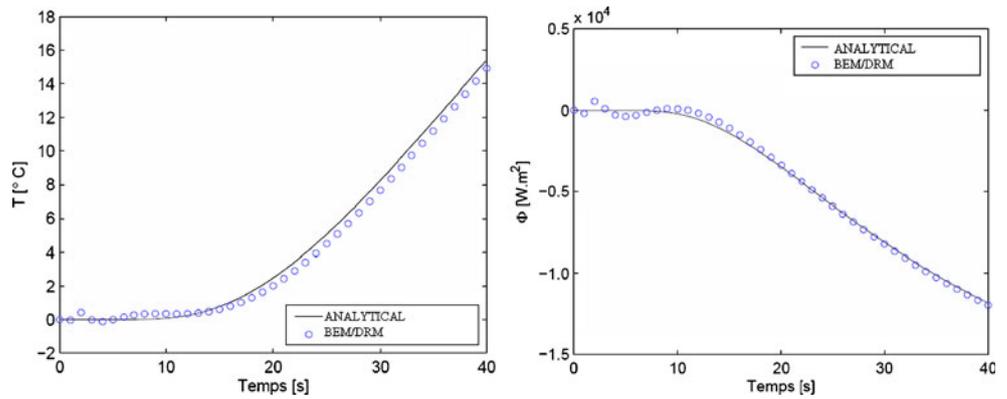
If we want to improve the accuracy of simulations, we need to decrease the time step, to use a finer mesh, and an implicit time integration scheme. Using the results of this study, we are able to optimise the DRM method.

In any case, using the BEM (as was also the case when using the finite element method) one should solve at each time step a linear system of equations  $\mathbf{A}\mathbf{u}^p = \mathbf{f}^{p-1}$ , where in the case

Fig. 4 Thermal shock configuration



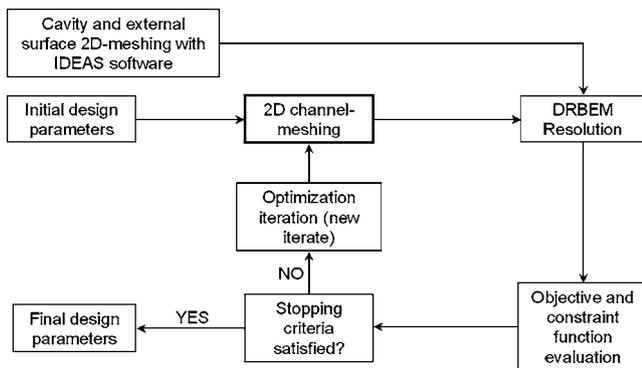
**Fig. 5** Temperature versus time (left)—Heat flux versus time (right)



of the BEM, vector  $\mathbf{u}^p$  contains all the discrete model unknowns (nodal temperatures or nodal fluxes on the domain boundary) at the current time step  $p$  whereas  $\mathbf{f}^{p-1}$  contains all the known information given at the previous time step  $p-1$ . The matrix of coefficients is fully populated and in some cases it changes as time evolves. For these reasons, even if the number of degrees of freedom involved in the BEM discrete model is much lower than that involved in standard finite element discretizations, its solution can be cumbersome. This is especially true if the thermal model must be solved many times, as is the case in mold optimization that implies the solution of a transient thermal model for each trial geometrical configuration.

### Coupling BEM with an optimization method

We couple heat transfer computation with an optimization method to modify automatically the parameters at each optimization iteration as shown in Fig. 6. Given initial parameters, the BEM simulation is performed and the cost function is calculated. The optimization method allows updating parameters according to constraints until a minimum of the cost function is found. SQP (Sequential Quadratic Programming) [19] is used for the optimization of continuous non linear functions with continuous non linear constraints.



**Fig. 6** Heat transfer simulation/optimization coupling

In order to reduce the computing time during the linear system solution (especially in 3D), we propose the use of model reduction within the BEM solver. Before presenting results of optimization, the reduced modeling approach is summarized in detail.

### Reduced modeling

#### Introduction

Most engineering systems can be represented by a continuous model usually expressed by a system of linear or non-linear coupled partial differential equations describing the different conservation balances (momentum, energy, mass and chemically reacting substances). From a practical point of view, the determination of its exact solution, that is, the exact knowledge of the different fields characterizing the physical system at any point and time instant (velocity, pressure, temperature, chemical concentrations, ...) is not possible in real systems due to the complexity of models, geometries and/or boundary conditions. For this reason the solution is searched only at some points and at some times, from which it could be interpolated to any other point and time. Numerical strategies allowing this kind of representation are known as discretization techniques. There exist numerous discretization techniques, e.g. finite elements, finite volumes, boundary elements, finite differences, meshless techniques, among many others. The optimal technique to be applied depends on the model and on the domain geometry. Progresses in numerical analysis and in computation performances make possible today the solution of complex systems involving millions of unknowns related to the discrete model. However, the complexity of the models is also increasing exponentially, and today engineers are not only interested in solving a model, but in solving these models many times (e.g. when they address optimization or inverse identification). For this purpose, strategies able to speed-up the numerical solution, preserving the solution accuracy, are in the focus.

In the context of control, optimization or inverse analysis, numerous problems must be solved, and for this reason the question related to the computation time becomes crucial. The question is very simple: is it possible to perform very fast and accurate simulations? Different answers have been given to this question depending on the scientific community to which this question is addressed. For specialists in computational science the answer to this question concerns the improvement of computational resources, high performance computing and the use of parallel computing platforms. For some specialists in numerical analysis the challenge is in the fast resolution of linear systems via the use of preconditioners or multigrid techniques among many others. For others the idea is to adapt the cloud of nodes (points where the solution is computed) in order to avoid an excessive number of unknowns. Many other answers have been given, however at present all these approaches allow to alleviate slightly the computation efforts but the fast and accurate computation remains a real challenge.

This section describes another different approach based on model reduction allowing fast and accurate computations. The idea is very simple. Consider a domain where a certain model is defined as well as the associated cloud of nodes able to represent by interpolation the solution everywhere. In general the number of unknowns scales with the number of nodes, and for this reason even if the solution is evolving smoothly in time all the nodes are used for describing it at each time step. In the reduced modeling that we describe later the numerical algorithm is able to extract the optimal information describing the evolution of the solution in the whole time interval. Thus, the evolution of the solution can be expressed as a linear combination of a reduced number of functions (defining the reduced approximation basis), and then the size of the resulting linear problems is very small, and consequently the CPU time savings can attain several orders magnitude (in the order of millions sometimes).

The extraction of this relevant information is a well known topic based on the application of the proper orthogonal decomposition, also known as Karhunen-Loève decomposition [14, 16] that is summarized in the next section. This kind of approach has been widely used for weather forecast purposes [17], turbulence [12, 35], solid mechanics [15] but also in the context of chemical engineering for control purposes [26].

Usual reduced modeling performs the simulation of some similar problem or the desired one in a short time interval. From these solutions the Karhunen-Loève decomposition applies, allowing the extraction of the most relevant functions describing the solution evolution. Now, it is assumed that the solution of a “similar” problem can be expressed using this reduced approximation basis, allowing a significant reduction on the discrete problem size and then to significant CPU time savings. However, in general the

question related to the accuracy of the computed solutions is usually ignored. An original approach combining the model reduction and the control of the solution accuracy was proposed by Ryckelynck [32], and applied later in a large catalogue of applications [2, 9, 22, 33, 34, 38]. This model reduction strategy can be coupled with usual finite element or boundary element discretizations [3].

We summarize in this section the main ideas of this reduction strategy for non specialists in numerical analysis, in order to show its potentiality in many domains of engineering and in particular in the context of optimization.

### Revisiting the Karhunen-Loève decomposition

We assume that the evolution of a certain field that depends on the physical space  $\mathbf{x}$  and on time  $t$ ,  $u(\mathbf{x}, t)$  is known. In practical applications, this field is expressed in a discrete form, that is, it is known at the nodes of a spatial mesh and at some times, i.e.  $u(\mathbf{x}_i, t^p) \equiv u_i^p$ . We can also write, introducing a spatial interpolation:  $u^p(\mathbf{x}) \equiv u(\mathbf{x}, t = p\Delta t)$ ;  $\forall p \in [1, \dots, P]$ . The main idea of the Karhunen-Loève (KL) decomposition is how to obtain the most typical or characteristic structure  $\varphi(\mathbf{x})$  among these  $u^p(\mathbf{x})$ ,  $\forall p$ . This is equivalent to obtaining functions  $\varphi(\mathbf{x})$  maximizing  $\alpha$

$$\alpha = \frac{\sum_{p=1}^{p=P} \left[ \sum_{i=1}^{i=N} \varphi(\mathbf{x}_i) u^p(\mathbf{x}_i) \right]^2}{\sum_{i=1}^{i=N} (\varphi(\mathbf{x}_i))^2} \quad (14)$$

which leads to:

$$\sum_{p=1}^{p=P} \left[ \left[ \sum_{i=1}^{i=N} \tilde{\varphi}(\mathbf{x}_i) u^p(\mathbf{x}_i) \right] \left[ \sum_{j=1}^{j=N} \varphi(\mathbf{x}_j) u^p(\mathbf{x}_j) \right] \right] = \alpha \sum_{i=1}^{i=N} \tilde{\varphi}(\mathbf{x}_i) \varphi(\mathbf{x}_i) \quad (15)$$

$\forall \tilde{\varphi}$ , which can be rewritten in the form

$$\sum_{i=1}^{i=N} \left[ \sum_{j=1}^{j=N} \left\{ \sum_{p=1}^{p=P} u^p(\mathbf{x}_i) u^p(\mathbf{x}_j) \varphi(\mathbf{x}_j) \right\} \tilde{\varphi}(\mathbf{x}_i) \right] = \alpha \sum_{i=1}^{i=N} \tilde{\varphi}(\mathbf{x}_i) \varphi(\mathbf{x}_i) \quad (16)$$

Defining vector  $\boldsymbol{\varphi}$  such that its  $i$ -component is  $\varphi(\mathbf{x}_i)$ , Eq. 16 takes the following matrix form

$$\tilde{\boldsymbol{\varphi}}^T \mathbf{k} \boldsymbol{\varphi} = \alpha \tilde{\boldsymbol{\varphi}}^T \boldsymbol{\varphi}; \quad \forall \tilde{\boldsymbol{\varphi}} \Rightarrow \mathbf{k} \boldsymbol{\varphi} = \alpha \boldsymbol{\varphi} \quad (17)$$

where the two points correlation matrix is given by

$$\mathbf{k}_{ij} = \sum_{p=1}^{p=P} u^p(\mathbf{x}_i) u^p(\mathbf{x}_j) \Leftrightarrow \mathbf{k} = \sum_{p=1}^{p=P} \mathbf{u}^p (\mathbf{u}^p)^T \quad (18)$$

which is symmetric and positive definite. If we define the matrix  $\mathbf{Q}$  containing the discrete field history:

$$\mathbf{Q} = \begin{pmatrix} u_1^1 & u_1^2 & \cdots & u_1^P \\ u_2^1 & u_2^2 & \cdots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_N^1 & u_N^2 & \cdots & u_N^P \end{pmatrix} \quad (19)$$

it is easy to verify that the matrix  $\mathbf{k}$  in Eq. 18 results

$$\mathbf{k} = \mathbf{Q} \mathbf{Q}^T \quad (20)$$

### Reduced modeling

If the evolution of a certain field is known

$$u(\mathbf{x}_i, t^p) \equiv u_i^p, \forall i \in [1, \dots, N], \forall p \in [1, \dots, P] \quad (21)$$

from some direct simulations or from experimental measures, then matrices  $\mathbf{Q}$  and  $\mathbf{k}$  can be computed and the eigenvalue problem given by Eq. 17 can be solved. The solution of Eq. 17 results in  $N$  eigenvalue-eigenvector couples. However, in a large number of models involving regular time evolutions of the solution, the magnitude of the eigenvalues decreases very fast, evidencing that the solution evolution can be represented as a linear combination of a reduced number of functions (the eigenvectors related to the largest eigenvalues).

In our numerical applications we consider the eigenvalues ordered  $\alpha_1 > \alpha_2 > \dots > \alpha_N$ . The  $n$  eigenvalues belonging to the interval  $\alpha_1 > \dots > \alpha_n$  with  $\alpha_n > \alpha_1 \cdot 10^{-8}$  and  $\alpha_{n+1} < \alpha_1 \cdot 10^{-8}$  are selected, because their associated eigenvectors are expected to be sufficient to represent accurately the entire solution evolution. In a large variety of models  $n \ll N$  and moreover  $n$  only depends on the regularity of the solution evolution, but neither on the dimension of the physical space (1D, 2D or 3D) nor on the size of the model ( $N$ ).

The reduced approximation basis consists on the  $n$  eigenvectors  $\varphi_1, \dots, \varphi_n$ , allowing to define the basis transformation matrix  $\mathbf{B}$ :

$$\mathbf{B} = (\varphi_1, \varphi_2, \dots, \varphi_n) \quad (22)$$

whose size is  $N \times n$ . Thus, the vector containing the field nodal values  $\mathbf{u}$  can be expressed by:

$$\mathbf{u} = \sum_{i=1}^n \varphi_i \cdot \xi_i(t) = \mathbf{B} \cdot \boldsymbol{\xi}(t) \quad (23)$$

Now, if we consider the linear system of equations resulting from the discretization of a partial differential equation (PDE) in the form

$$\mathbf{A} \mathbf{u}^p = \mathbf{f}^{p-1} \quad (24)$$

where  $\mathbf{f}^{p-1}$  accounts for the solution at the previous time step, taking into account Eq. 23 it reduces to:

$$\mathbf{A} \mathbf{u}^p = \mathbf{f}^{p-1} \Rightarrow \mathbf{A} \mathbf{B} \boldsymbol{\xi}^p = \mathbf{f}^{p-1} \quad (25)$$

and multiplying both terms by  $\mathbf{B}^T$  it results

$$\mathbf{B}^T \mathbf{A} \mathbf{B} \boldsymbol{\xi}^p = \mathbf{B}^T \mathbf{f}^{p-1} \quad (26)$$

which proves that the final system of equations is of low order, i.e. the dimension of  $\mathbf{B}^T \mathbf{A} \mathbf{B}$  are  $n \times n$ , with  $n \ll N$ , and the dimension of both  $\boldsymbol{\xi}$  and  $\mathbf{B}^T \mathbf{f}^{p-1}$  are  $n \times 1$ .

### Reduced basis adaptivity

The just described strategy allows for very fast computation of large size models. For example one could solve the full model using some standard discretization technique (finite differences, finite elements, boundary elements, ...) for a small time interval and then define matrices  $\mathbf{Q}$  and  $\mathbf{k}$  allowing to compute the reduced approximation basis transformation  $\mathbf{B}$  that leads to the reduced solution procedure illustrated by Eq. 26. Another possibility consists in solving a model in the whole time interval and then extracting the most representative functions that could be used for solving some ‘‘similar’’ models. We come back to this discussion later.

However, in any case, it is not guaranteed that this reduced basis that was built in the first scenario from the solution known within a short time interval, and in the second one for a particular model different to the present one, remains accurate for describing the solution in the entire simulation interval or for any other ‘‘similar’’ model respectively. In the first case it is obvious that during the simulation, material properties, boundary conditions, etc. could change, compromising the validity of the reduced basis. In the second case, the model being different to the one that served to extract the reduced basis, nothing guarantees the validity of that reduced approximation basis.

In this manner, if one would compute reduced model solutions and keep the confidence on the related solution, a check of the solution accuracy must be performed and an enrichment strategy must be defined in order to adapt the reduced approximation basis in order to capture the new events present in the solution evolutions which cannot be described accurately from the original reduced approximation basis.

For this purpose, Ryckelynck proposed [32] to start with a low order approximation basis, using some simple functions (e.g. the initial condition in transient problems) or using the eigenvectors of a ‘‘similar’’ problem previously solved or the ones coming from a full simulation in a short time interval. Now, we compute  $S$  iterations of the evolution problem using the reduced model (26) without

changing the approximation basis. After these  $S$  iterations, the complete discrete system (25) is constructed, and the residual  $\mathbf{R}$  is evaluated:

$$\mathbf{R} = \mathbf{A}\mathbf{u}^S - \mathbf{f}^{S-1} = \mathbf{A}\mathbf{B}\boldsymbol{\xi}^S - \mathbf{f}^{S-1} \quad (27)$$

If the norm of the residual is small enough,  $\|\mathbf{R}\| \leq \varepsilon$ , with  $\varepsilon$  a small enough threshold value, we can continue for  $S$  more iterations using the same approximation basis. On the contrary, if the residual norm is too large,  $\|\mathbf{R}\| \geq \varepsilon$ , we need to enrich the approximation basis and compute again the last  $S$  iterations. This enrichment is built using some Krylov's subspaces, in our case the first three subspaces:  $\mathbf{B} \leftarrow (\mathbf{B}, \mathbf{R}, \mathbf{A}\mathbf{R}, \mathbf{A}^2\mathbf{R})$ .

One could expect the enrichment process to increase continuously the size of the reduced approximation basis, but in fact, after reaching convergence, a Karhunen-Loève decomposition is performed on the whole past time interval in order to extract the significant information as well as to define an orthogonal reduced approximation basis. The interested reader can refer to Ryckelynck et al. [34] and the references therein for a more detailed and valuable description of the computational algorithm.

### Application of BEM reduced model to mold cooling optimization

#### Reduced model coupled with BEM

We solve the eigenvalue problem defined in "Reduced modeling" selecting the eigenfunctions associated with the  $n$  largest eigenvalues as described previously. In practice,  $n$  is much lower than  $N$ . The  $\mathbf{B}$  matrix is then assembled and used to approximate the temperature. In fact, for the first trial geometry, we solve the first injection cycle by applying the standard BEM. During this solution procedure some solution snapshots are stored and the eigenvalue problem related to the proper orthogonal decomposition is solved at

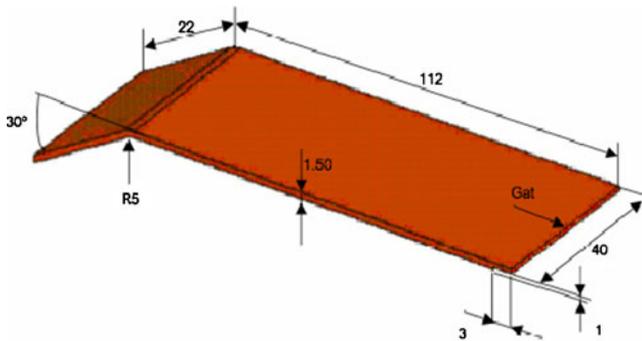


Fig. 7 Plastic part dimensions

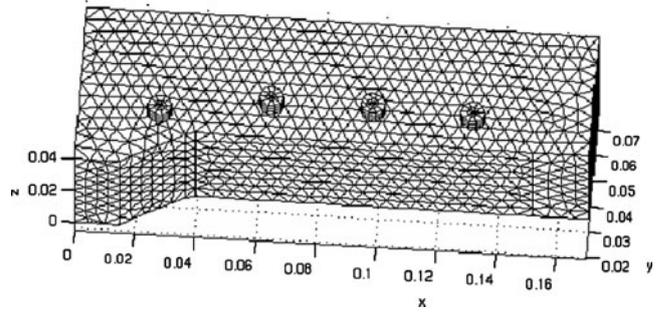


Fig. 8 Upper-part of the mold mesh

the end of the first injection cycle to extract the main representative modes defining the evolution of the thermal fields. These modes are used for defining the matrix  $\mathbf{B}$  that allows moving from the global approximation basis (the one related to the BEM discretization) to the reduced one containing only the most representative modes. Then, the subsequent injection cycles are solved by using the reduced approximation basis contained in matrix  $\mathbf{B}$ . The first column of that matrix at each injection cycle contains the nodal unknowns at the previous injection cycle in order to represent accurately the initial temperature at each injection cycle. This procedure can be repeated when the trial mold geometry is modified during the optimization procedure.

#### Overall optimization methodology

We present in this section how we formulate the problem under a mathematical programming form. In the sequel,  $x$  will denote the vector of optimization variables (position and shape parameters for the cooling channels). Since the output of the heat-transfer problem is a function of  $x$ , we shall make explicit the dependence of the temperature measurements upon the position and shape parameters. Most practical optimization problems involve several (often contradictory) objective functions. The simplest way to proceed in such a multi-criterion context is to consider as objective function a weighted sum of the various criteria. This involves choosing appropriate weighting parameter values. An obvious alternative is to use one criterion as objective function while requiring, in the constraints, maximal threshold levels for the remaining criteria. We choose here the latter approach because we do know a threshold level value for the maximal temperature variation

Table 1 Thermo-physical properties

	Polymer (PP)	Mold (Steel)
$\lambda$ [W.m <sup>-1</sup> .K <sup>-1</sup> ]	0.63	34
$\rho$ [kg.m <sup>-3</sup> ]	891	7,800
$C_p$ [J.kg <sup>-1</sup> .K <sup>-1</sup> ]	2,740	460

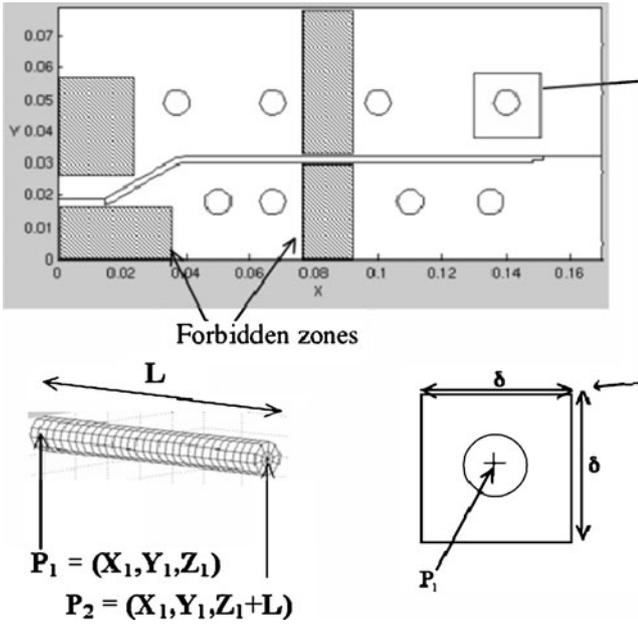


Fig. 9 Constraints and optimization variables

under which any variation is equally acceptable. More precisely, we formulate our problem under the form:

$$\min_x \|T(x)\| \quad \text{subject to } f(T(x)) \text{ and } g(x) \leq 0 \quad (34)$$

where  $f$  is a real-valued function used to stipulate the uniformity-temperature constraint, and  $g(x)$  is a general vector-valued non-linear function. The complete methodology to couple the thermal solver and the optimization algorithm procedure is presented in [30].

The general constraints  $g(x) \leq 0$  represent any geometry related or other industrial constraints, such as:

- upper/lower-bound constraints on the  $x_i$ 's,
- keeping the cooling channels within the mold,
- technically-forbidden zones where we cannot position the cooling channels (for instance due to the presence of ejectors),
- constraints stipulating a minimal distance between every pair of cooling channels to avoid inter-channels collision.

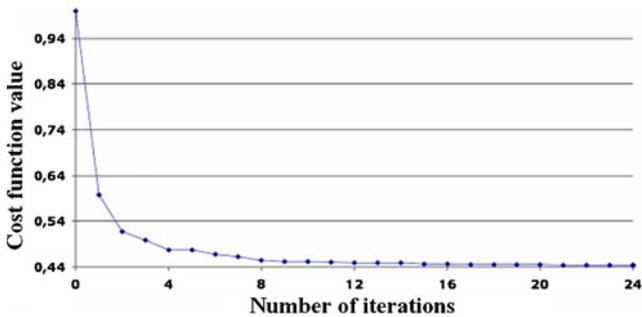


Fig. 10 Convergence analysis

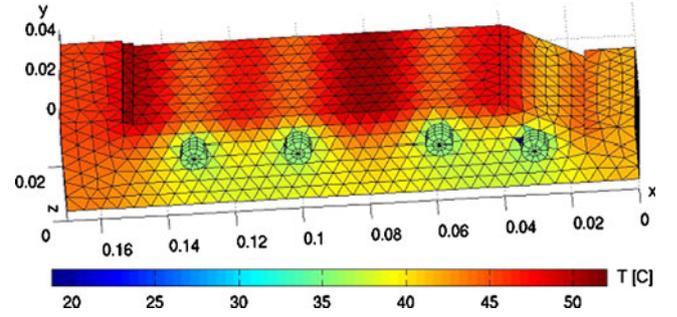


Fig. 11 Surface temperature distribution of the mold

### Application

In this section, we report numerical simulations on a 3D plastic part whose features are displayed on Fig. 7 (units in mm). It is a semi-industrial injection mold design for the European project: Eurotooling 21.

The mold is meshed using 5,592 linear triangles and each cooling channel using 340 quadrangles (Fig. 8).

Thermo-physical properties of the polymer as well the mold material is referenced in Table 1. Boundary conditions are the same as defined in “BEM for transient heat balance equation”.

The history matrix, corresponding to the first injection cycle time, is computed using transient DRBEM code. The mold temperature, for the next injection cycle time, is computed using the reduced model. The optimization objective consists in minimizing maximal temperature while minimizing temperature variations:

$$\text{minimize}_{i \in N} \max(T_i) \quad \text{subject to} \quad \sum_{i \in N} |T_i - T_{av}| \leq \sigma \quad (35)$$

where  $N$  is the number of elements and  $T_{av}$  the average surface temperature. For illustration purposes, we consider here eight cooling channels and the constraints and optimization variables are sketched on Fig. 9.

The geometrical optimization parameters are here the coordinates of the end points,  $P_1$  and  $P_2$  of each cooling channel (Fig. 7). Since  $P_2$  can be expressed in terms of the other coordinates and since the channel length ( $L$ ) is constant,

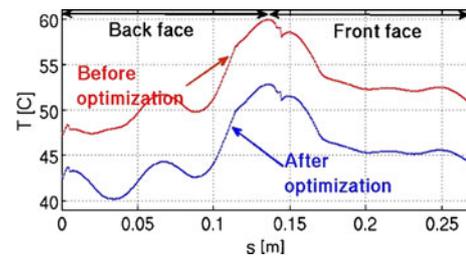


Fig. 12 Temperature profile at the surface of mold cavity before and after optimization ( $z=0.02$  m)

the optimization parameters for locating the  $i$ -th cooling channel are completely determined by  $P_1^i = (X_i, Y_i, Z_i)_{i=1, \dots, 8}$ . For this application,  $Z_i$  is fixed and therefore the problem reduces to 16 optimization variables.

We use as starting point, a heuristic solution provided by an experienced engineer. In average, one objective function evaluation requires 14 min of CPU time (one direct computation). Since, we compute gradients using finite difference approximation (associated with SQP method), one optimization iteration involves 4 h of CPU time on *Macintosh 1.83 GHz Intel Core 2 Duo*. 24 optimization iterations were necessary in order to achieve convergence for one injection cycle (Fig. 10).

In addition, the surface temperature distribution of the mold is presented on Fig. 11 and the temperature profile at the mold surface before and after optimization are displayed in Fig. 12. We observe on Fig. 12 that both temperature variance and temperature average decrease significantly.

We need 96 h to perform complete optimization without reduction model. If we use now reduction model and DRBEM, we reduce the CPU time to 7 h40 (one direct computation of 14 min and 24 optimization iterations of 18 min approximately). CPU time is divided approximately by 13.

## Discussion

We introduced a methodology based on the use of BEM to solve the 3D heat transfer equation during the cooling step of the injection molding process. The preliminary computation tests on a semi-industrial plastic part showed that the approach is viable for optimizing the design of cooling channels for injection molding. The numerical modeling and optimization methodology can easily take into account a large range of industrial constraints. Various optimization criteria can be provided by the user (either directly as a cost function or within constraints).

Another interesting aspect consists in using this technique in order to compute multi-cycles injection mold cooling. The reduction model allows to extract from the first cycle the relevant eigenfunctions associated with the eigenvalues and consequently to calculate very rapidly all the other cycles.

## References

1. Agassant J-F, Avenas P, Sergent J-Ph, Carreau PJ (1991) Polymer processing—principle and modelling. Hanser, New York
2. Ammar A, Ryckelynck D, Chinesta F, Keunings R (2006) On the reduction of kinetic theory models related to finitely extensible dumbbells. *J Non-Newton Fluid Mech* 134:136–147
3. Ammar A, Pruliere E, Férec J, Chinesta F, Cueto E (2009) Coupling finite elements and reduced approximation bases. *Eur J Contracept Reprod Health Care* 18/5–6:445–463
4. Bialecki A, Jurgas P, Kuhn G (2002) Dual reciprocity BEM without matrix inversion for transient heat conduction. *Eng Anal Bound Elem* 26:227–236
5. Bikas A, Kanarachos A (1999) The dependence of cooling channels system geometry parameters on product quality as a result of uniform mold cooling. Proceedings of the 57th Annual Technical Conference ANTEC 99, 578–583
6. Boillat E, Glardon R, Paraschivescu D (2002) *J Phys IV* 102:27–38
7. Brebbia CA, Dominguez J (1992) Boundary elements, an introductory course. WIT Press/Computational Mechanics Publications
8. Brebbia CA, Chen CS, Power H (1999) Dual reciprocity method using compactly supported radial basis functions. *Commun Numer Methods Eng* 15/2:137–150
9. Chinesta F, Ammar A, Lemarchand F, Beauchene P, Boust F (2008) Alleviating mesh constraints: model reduction, parallel time integration and high resolution homogenization. *Comput Methods Appl Mech Eng* 197/5:400–413
10. Fletcher R (1987) Practical methods of optimization. John Wiley & Sons, New York
11. Godinho L, Tadeu A, Simoes N (2004) Study of transient heat conduction in 2.5D domains using the boundary element method. *Eng Anal Bound Elem* 28/6:593–606
12. Holmes PJ, Lumley JL, Berkooz GD, Mattingly JC, Wittenberg RW (1997) Low-dimensional models of coherent structures in turbulence. *Phys Rep* 287:337–384
13. Huang J, Fadel G (2001) *J Mech Des* 123:226–239
14. Karhunen K (1946) Über lineare methoden in der wahrscheinlichkeitsrechnung. *Ann Acad Sci Fennicae ser A1 Math Phys* 37:3–79
15. Krysl P, Lall S, Marsden JE (2001) Dimensional model reduction in non-linear finite element dynamics of solids and structures. *Int J Numer Methods Eng* 51:479–504
16. Loève MM (1963) Probability theory. The University Series in higher mathematics, 3rd edn. Van Nostrand, Princeton
17. Lorenz EN (1956) Empirical orthogonal functions and statistical weather prediction. MIT, Departement of Meteorology, Scientific Report N1, Statistical Forecasting Project
18. Lu Y, Li D, Xiao J (1996) Simulation of cooling process of injection molding. *Prog Nat Sci* 6/2:227–234
19. Mathey E (2004) PhD thesis, Toulouse University (in French)
20. Mathey E, Penazzi L, Schmidt FM, Ronde-Oustau F (2004) Automatic optimization of the cooling of injection mold based on the boundary element method *Materials Processing and Design: Modeling, Simulation and Applications*. Proceedings NUMIFORM 04, 222–227
21. Moller J, Carlson M, Alterovitz R, Swartz J (1998) Post-ejection cooling behavior of injection molded parts. Proceeding of the 56th Annual Technical Conference ANTEC 98, 525–527
22. Niromandi S, Alfaro I, Cueto E, Chinesta F (2008) Real-time deformable models of non-linear tissues by model reduction techniques. *Comput Meth Programs Biomed* 91:223–231
23. Opolski SW, Kwon TW (1987) Injection molding cooling system design. Annual Technical Conference and Exhibition, Society of Plastics Engineers, 264–268
24. Özisik MN (1985) Heat transfer, a basic approach. McGraw Hill
25. Osswald TA (1998) Polymer processing fundamentals. Carl HANSER Verlag, Munich
26. Park HM, Cho DH (1996) The use of the Karhunen-Loève decomposition for the modelling of distributed parameter systems. *Chem Eng Sci* 51:81–98
27. Park S, Kwon T (1998) *Polym Eng Sci* 38:1450–1462
28. Pasquetti R, Petit D (1995) Inverse diffusion by boundary elements. *Eng Anal Bound Elem* 15:197–205
29. Pirc N, Bugarin F, Schmidt FM, Mongeau M (2008) 3D BEM-based cooling-channel shape optimization for injection molding processes.

- International Journal for Simulation and Multidisciplinary Design Optimization 2/3:245–252
30. Pirc N, Schmidt FM, Mongeau M, Bugarin F, Chinesta F (2009) Optimization of 3D cooling channels in injection molding using DRBEM and model reduction. *International Journal of Material Forming* 2(01):271–274
  31. Rao NS, Schumacher G, Schott NR, O'Brien KT (2002) Optimization of cooling systems in injection molds by an easily applicable analytical model. *J Reinf Plast Compos* 21/5:451–459
  32. Ryckelynck D (2005) A priori hyperreduction method: an adaptive approach. *J Comput Phys* 202:346–366
  33. Ryckelynck D, Hermanns L, Chinesta F, Alarcón E (2005) An efficient “a priori” model reduction for boundary element models. *Eng Anal Bound Elem* 29:796–801
  34. Ryckelynck D, Chinesta F, Cueto E, Ammar A (2006) On the “a priori” model reduction: overview and recent developments. *Arch Comput Methods Engng, State Art Rev* 13/1:91–128
  35. Sirovich L (1987) Turbulence and the dynamics of coherent structures part I: coherent structures. *Quart Appl Math* XLV:561–57
  36. Sutradhar A, Paulino GH, Gray LJ (2002) Transient heat conduction in homogeneous and nonhomogeneous materials by the Laplace transform Galerkin boundary element method. *Eng Anal Bound Elem* 26/2:119–132
  37. Tang L, Pochiraju K, Chassapis C, Manoochehri S (1998) *J Mech Des* 120:165–174
  38. Verdon N, Allery C, Béghein C, Hamdouni A, Ryckelynck D (2009) Reduced-order modelling for solving linear and non-linear equations. *Int J Numer Methods Eng*. doi:10.1002/cnm.1286
  39. MH Wesselmann (1998) Impact of moulding conditions on the properties of short fibre reinforced high performance thermoplastic parts. PhD Thesis, Ecole des Mines Albi
  40. Yang SY, Lien L (1996) Effects of cooling time and mold temperature on quality of moldings with precision contour. *Adv Polym Technol* 15(4):289–295