To cite this version :

Chady GHNATIOS, Elias CUETO, Antonio FALCO, Jean-Louis DUVAL, Francisco CHINESTA - Spurious-free interpolations for non-intrusive PGD-based parametric solutions: Application to composites forming processes - International Journal of Material Forming - Vol. 14, n°1, p.83-95 - 2020

Spurious-free interpolations for non-intrusive PGD-based parametric solutions: Application to composites forming processes

Chady Ghnatios[1] · Elias Cueto[2] · Antonio Falco[3] · Jean-Louis Duval[4] · Francisco Chinesta[5]

Abstract

Non-intrusive approaches for the construction of computational vademecums face different challenges, especially when a parameter variation affects the physics of the problem considerably. In these situations, classical interpolation becomes inaccurate. Therefore, classical approaches for the construction of an offline computational vademecum, typically by using model reduction techniques, are no longer valid. Such problems are faced in different physical simulations, for example welding path problems, resin transfer molding, or sheet compression molding, among others. In such situations, the interpolation of precomputed solutions at prescribed parameter values (built using either intrusive or non intrusive techniques) generates spurious numerical artifacts. In this work, we propose an alternative interpolation and simulation strategy by using physically-based morphing of spaces. The morphing will transform the uncompatibe physical domains of the problem's solution into a compatible one, where an interpolation free of artifacts can be performed. Later on, an inverse transformation can be used to push-back the solution. Different relevant examples are illustrated in this work to motivate the use of the proposed method.

Keywords Non-intrusive PGD · Smart interpolation · Geometrical mapping · Interpolation · Real-time · Computational vademecum

Introduction

Computational abaci or vademecums can be built either through the use of numerical simulation or data-driven modeling [7, 13, 18]. They consist of a compilation or a handbook of solutions precomputed "offline", to be used on the fly, "online". The possibility of performing real time simulation in an industrial framework at the expense of offline heavy computations is motivating the industrial companies to move in this direction. The development of such computational vademecums became possible thanks to the development of model order reduction techniques [3], especially the Proper Generalized Decomposition, PGD, which is probably a very suitable "a priori" model order reduction technique. However, PGD is by nature the most intrusive model reduction technique due to its "a priori" formulation. Recent efforts have been undertaken to formulate the PGD in a non-intrusive manner as much as possible [4, 5, 27]. Some non-intrusive versions of the PGD rely on external flow processing codes that controls validated open-source finite element (FE) codes or commercial software [16, 20, 26]. The external, home-made, code manages the assigned FE problems to the commercial software. The assigned problems are managed by classical PGD decomposition procedures [1, 6, 11]. Another way of constructing a computational vademecum through the use of non-intrusive PGD relies on the calculation of snapshots to later on construct a manifold of solutions using the locally linear embedding (LLE) for example or other manifold learning techniques [2, 4, 5, 12, 15, 17, 21, 23].

✉ Chady Ghnatios
cghnatios@ndu.edu.lb

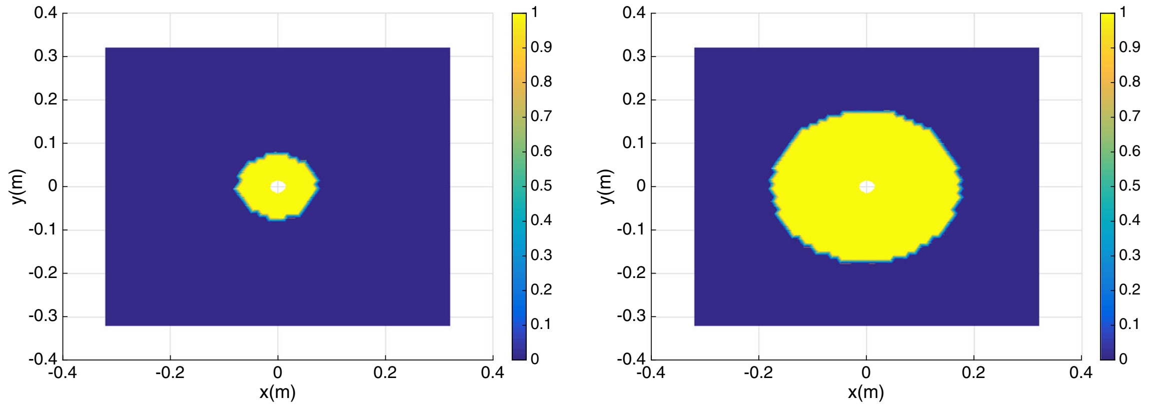
[1] Department of mechanical engineering, Notre Dame University-Louaize, Zouk Mosbeh, PO Box 72, Lebanon

[2] Aragon Institute of Engineering Research, Universidad de Zaragoza, Maria de Luna, 50018 Zaragoza, Spain

[3] ESI Chair @ CEU-UCH, Universidad Cardenal Herrera-CEU, San Bartolome, 46115 Valencia, Spain

[4] ESI Group, Batiment Seville, 3 bis rue Saarinen, F-50468, Rungis, France

[5] PIMM Lab and ESI Chair @ Arts et Metiers Institute of Technology, Boulevard de l'Hôpital, 75013, Paris, France



(a) RTM front propagation for a permeability $k_1 = 5 \times 10^{-12}$

(b) RTM front propagation for a permeability $k_2 = 5 \times 10^{-11}$

Fig. 1 RTM front propagation solution for two different permeabilities k_1 and k_2 (Metric system units) at the same filling time

Despite numerous developments during the recent years towards the development of computational vademecums in both intrusive and non-intrusive manners [11], the calculation of solution handbooks using any of the aforementioned methods faces a major challenge when changing a parameter induces drastic changes in the solution map. The changes at a given point can be very aggressive such that interpolation, a finite element mesh in a parametric domain, for instance, leads to unrealistic numerical artifacts [10]. Thus, in resin transfer molding, the interpolation of the resin flow front between solutions computed at different permeabilities may lead to multiple fronts appearing in the solution. For example, interpolating the solutions shown in Fig. 1 lead to the one illustrated in Fig. 2. In such case, a simulation using standard non-intrusive-PGD fails.

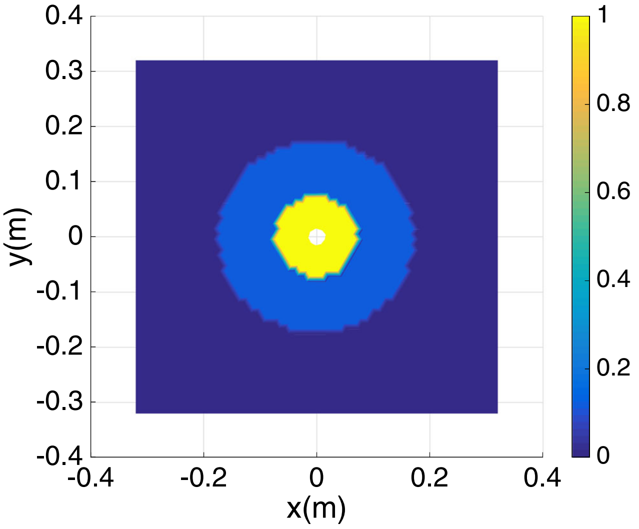


Fig. 2 Classical front interpolation of two RTM fluid front at a permeability $k = 10^{-11}$ from two snapshots at $k_1 = 5 \times 10^{-12}$ and $k_2 = 5 \times 10^{-11}$

We can clearly identify that a change in the permeability of the domain will compromise the solution of the resin front propagation and therefore computing a solution handbook becomes useless. Computational vademecums in resin transfer molding appear to be a cornerstone of interest when identifying anisotropic permeabilities [9] and even more importantly, for parametric simulation of resin injection in complex molds.

Other challenging problems with localized non-interpolable effects are path-dependent problems. Path dependent problems appear in different domains ranging from economy [8] to 3D printing, structural analysis [22, 25], time-dependent vehicle routing problems [14] and welding [24] among many others. Recently, the use of experimental and machine learning techniques to optimize path dependent problems in welding were considered, for example [19, 24]. However, the problem of simulating and optimizing a path-dependent problem involves an infinite number of parameters and is impossible to simulate and optimize using classical techniques.

In this work, we aim to tackle problems with non-interpolable localized effects, through proposing an appropriate interpolation technique. The suggested method starts by using a "physics-based" geometrical mapping, to transform different offline computed snapshots into another geometrically similar domain. In the new domain, which will be named "the reference domain", all localized effects overlap, thus rendering the interpolation possible. Once the interpolation is performed, an inverse transformation is used to transfer the constructed solution back to the real domain. Eventually the transformation and interpolation are both performed online, as fast as possible, to allow real time exploration and optimization of the solution. Several

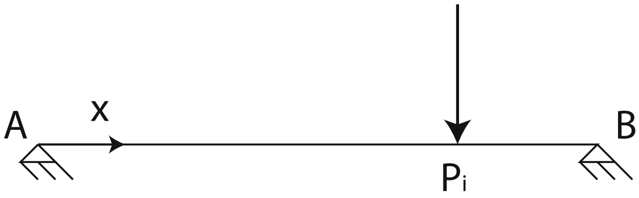


Fig. 3 An elastic rope supporting a load at a position $x = X$

examples are illustrated in this paper using those adopted mappings, in Resin Transfer Molding (RTM), path dependent heat transfer problems and sheet molding compound (SMC) involving flow fronts, starting with an illustrative 1D example.

An illustrative example in 1D

As a first example, we consider an elastic rope fixed at its two ends A and B, with an applied displacement at a point P_i of coordinate $x = X$ as illustrated in Fig. 3. For the sake of simplicity, the vertical displacement is assumed to be equal to 1 unit. The generated deformation of the rope is assumed linear between the two fixed support and the position of application of the load.

Now placing 2 different loads at P_1 and P_2 would give two different deformations as per Fig. 4. If we consider a unit displacement applied at the middle point between P_1 and P_2, the classical interpolation of the two displacement fields of the rope would lead to the solution in red illustrated in Fig. 5. However, we expect to find is the solution in green, also illustrated in Fig. 5.

The problem of the deformation of the rope illustrated in Fig. 4 is a 2D problem defined in the domain (x, X) where x is the coordinate along the length of the rope $x \in [0, L]$, with L the initial length of the rope, and X is the position of the applied displacement such that $X \in [a, b]$, where in the limit a tends to 0 and b tends to L. The analytical solution of the vertical displacement v can be written as a function of x and X such as:

$$\begin{cases} v = -\frac{x}{X} \text{ if } x < X, \\ v = -\frac{L-x}{L-X} \text{ elsewhere.} \end{cases} \tag{1}$$

The solution is illustrated in the 2D domain (x, X) in Fig. 6a for $L = 1$. With the help of a geometrical transformation, we can transform the point of application of the displacements P_i to the same position, for example the center point of the domain. Thus, we can transform the domain $x \in [0, L]$ into another domain $s \in [0, 2]$ such that:

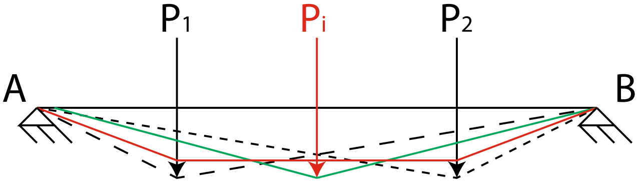


Fig. 5 Interpolation of the displacement field using classical interpolation technique in red, and the reference solution in green

$$\begin{cases} x = sX, \text{ if } s \in [0, 1], \\ x = (s-1)(L-X) + X, \text{ if } s \in [1, 2]. \end{cases} \tag{2}$$

Using the transformation given by Eq. 2, the displacement is always applied at $s = 1$, and therefore the solution in the (s, X) domain results the one illustrated in Fig. 6b for $L = 1$.

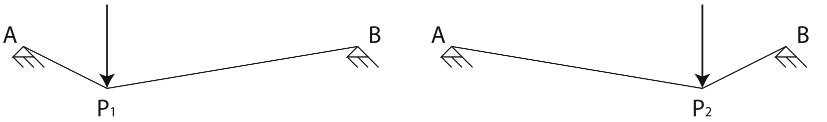
As one may notice, in the s domain, displacements are always applied at the same position, and therefore the solution can be interpolated without major issues. To check the described approach, the solution is computed for four different positions of the applied displacement in the x domain, named 4 snapshots, at $X = 0^+$, $X = L/3$, $X = 2L/3$ and $X = L^-$. Figure 7a shows the solution obtained using a classical interpolation in the x domain at $X = L/2$, while Fig. 7b shows the result of the interpolation when performed in the s domain and then using an inverse transformation.

Geometrical mapping in 2D

In this section we illustrate the geometrical mapping used to transform the domains. For the sake of simplicity, we are first illustrating the process on a 2D example of a localized heat source and homogeneous Dirichlet boundary conditions. Thermal fields are computed in the domain at different location of the heat source. The original domain Ω is depicted in Fig. 8. The governing equation is:

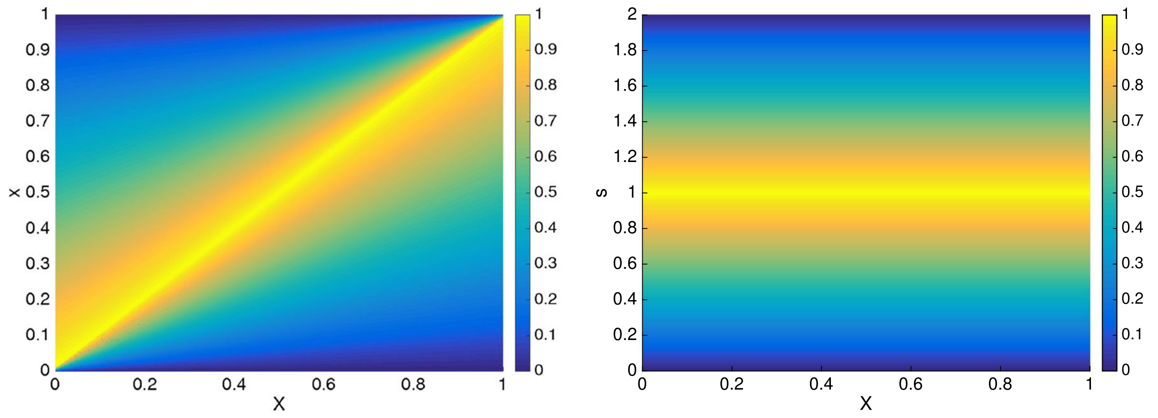
$$-k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) = Q, \tag{3}$$

Fig. 4 Solution for two different applied displacements



(a) Displacement enforced at position P_1

(b) Displacement enforced at position P_2

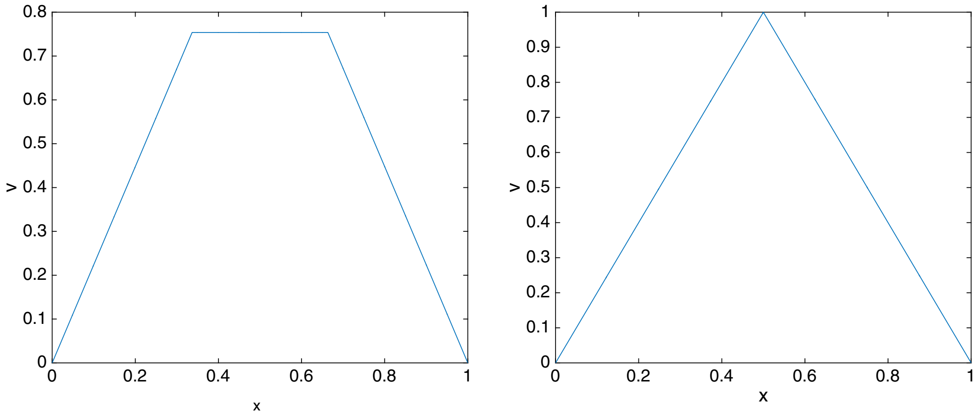


(a) Magnitude of the rope vertical displacement in (x, X) domain

(b) Magnitude of the rope vertical displacement in (s, X) domain

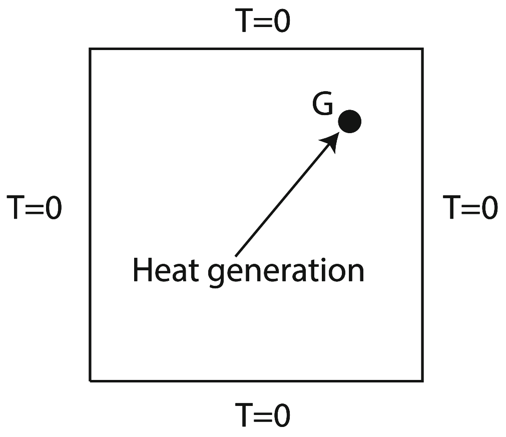
Fig. 6 Solution in the $2D$ real domain (x, X) and transformed domain (s, X)

Fig. 7 Solution at $X = L/2$ using classical interpolation and the proposed mapping technique



(a) Solution at $X = L/2$ using an interpolation in x

(b) Solution at $X = L/2$ using an interpolation in s



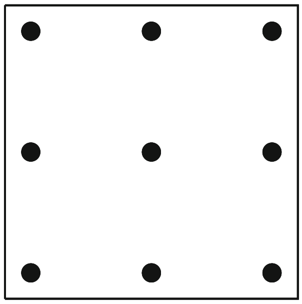
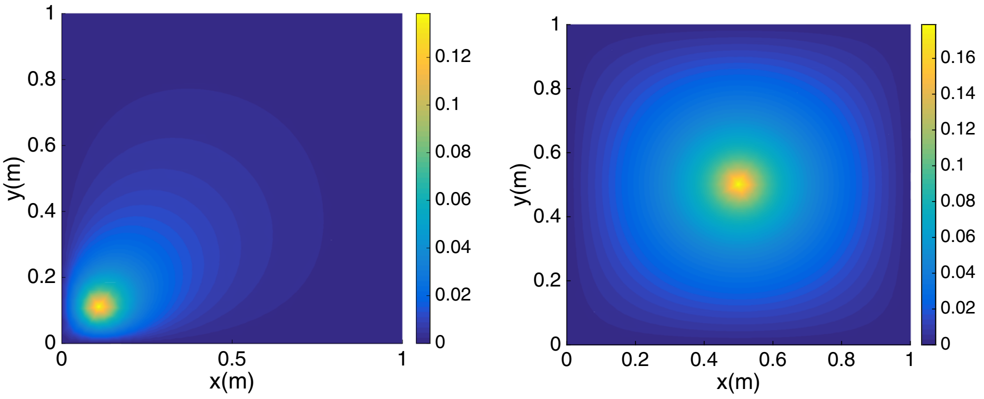


Fig. 8 The problem domain Ω involving localized heat source Q, with homogeneous Dirichlet boundary conditions

Fig. 9 Locations of different thermal sources G_i in the different computed snapshots

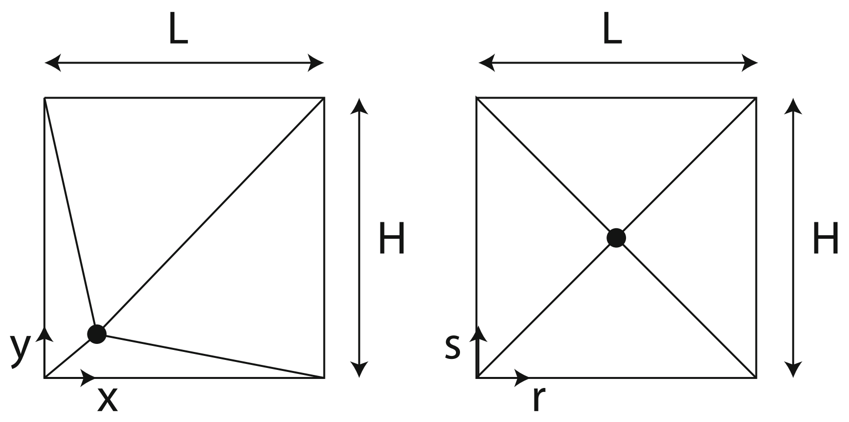
Fig. 10 Temperature fields for two different "snapshot" solutions



(a) Temperature for heat source at location G_1

(b) Temperature for heat source at location G_5

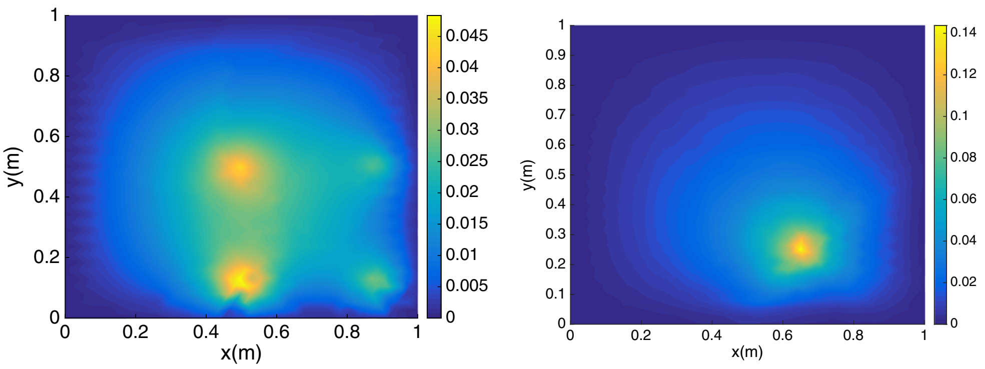
Fig. 11 Real domain (x, y) and reference domain (r, s)



(a) Real domain (x, y)

(b) Reference domain (r, s)

Fig. 12 Solution using classical interpolation and the one based on the proposed mapping



(a) Classical interpolation of a heat source at a different location than the computed ones

(b) Interpolation based on morphing

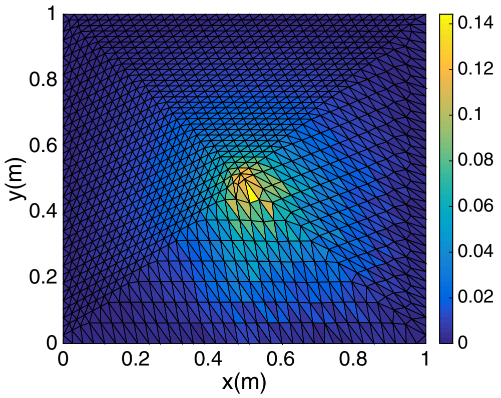


Fig. 13 Interpolated solution in the reference domain

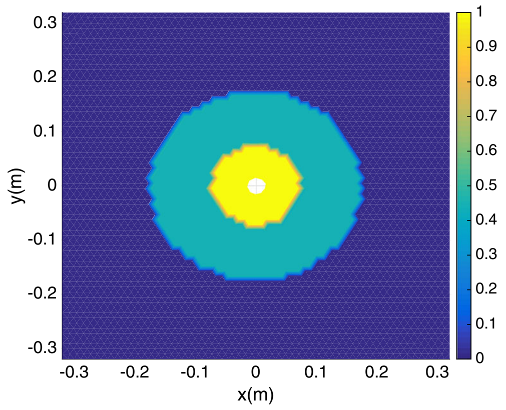


Fig. 15 Solution using classical interpolation for $k_x = k_y = 0.5k$

where k is the thermal conductivity and Q the heat generation term. Q is defined as a localized heat generation inside a domain G such as:

$$Q = \begin{cases} 1 \text{ if } (x, y) \in G, \\ 0 \text{ elsewhere.} \end{cases} \tag{4}$$

Different snapshots are obtained, capturing different temperature fields $T_i(\mathbf{x})$ for different locations of the heat source G_i. The location of the different thermal sources is depicted in Fig. 9. Snapshot solutions (two of them depicted in Fig. 10) can be computed using any direct numerical solver. For instance, in this work we used the finite element method.

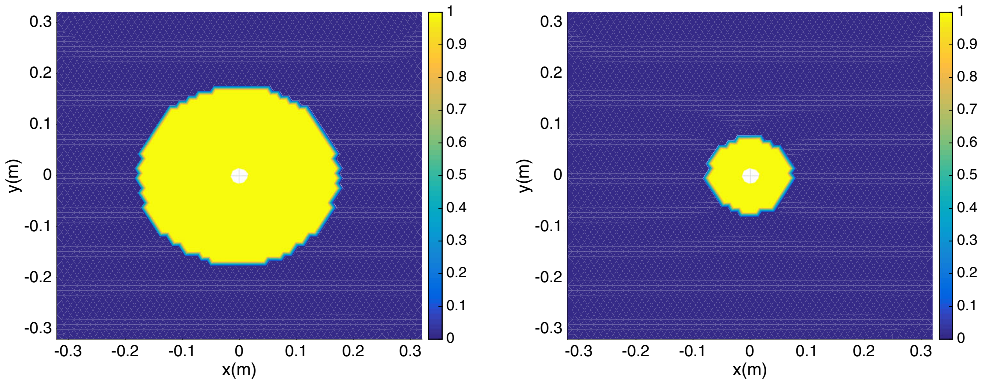
Once the snapshots are available in the physical (x, y) domain, we can transform the domain into a "reference" domain (r, s) where all heat sources apply at the same location, as illustrated in Fig. 11. Transformations between the two coordinate systems $\mathbf{X} = (x, y)$ and $\mathbf{R} = (r, s)$ for each triangle in Fig. 11 read:

$$\begin{cases} \mathbf{X} = \mathbf{X}_1\phi_1(r, s) + \mathbf{X}_2\phi_2(r, s) + \mathbf{X}_3\phi_3(r, s) \\ \mathbf{R} = \mathbf{R}_1\xi_1(x, y) + \mathbf{R}_2\xi_2(x, y) + \mathbf{R}_3\xi_3(x, y) \end{cases} \tag{5}$$

Equation 5 is defined per triangle in the domains illustrated in Fig. 11, ϕ_i being the standard Lagrange interpolation functions defined in the reference (r, s) domain, while ξ_i the Lagrange shape functions defined in the (x, y) real domain whose expression can be found in any finite elements textbook. The element connectivity remains unchanged while mapping between the real and reference elements. The values $\mathbf{X}_i$ and $\mathbf{R}_i$ are the coordinates of the nodes of the corresponding triangles in the (x, y) and (r, s) domains respectively.

In the (r, s) domain, the interpolation can be performed using any classical interpolation technique, for instance the SSL-PGD technique is used [10]. Figure 12 illustrates the interpolated solution associated with a heat source at a location different than the ones related to the pre-computed snapshots using classical interpolation technique in Fig. 12a and using the proposed mapping in Fig. 12b. Figure 13 illustrates the interpolation result in the reference (r, s) domain. Figure 13 clearly illustrates the deformed mesh that is induced by the mapping. As a consequence, the sampling should rich enough for avoiding excessive mesh distortion. Only the closest solutions to the searched one are mapped, and consequently the closer they are, the smaller are the associated mesh distortions induced by the mapping.

Fig. 14 Different snapshots of mold filling at time $t = 120s$



(a) Flow front for $k_x = k_y = k$

(b) Flow front for $k_x = k_y = 0.1k$

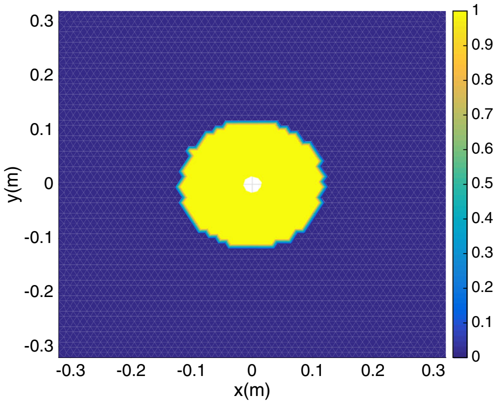


Fig. 16 Solution using flow front radius interpolation in the $(r;\theta)$ coordinates for $k_x = k_y = 0.5k$

Flow fronts interpolation

Parametric domain properties

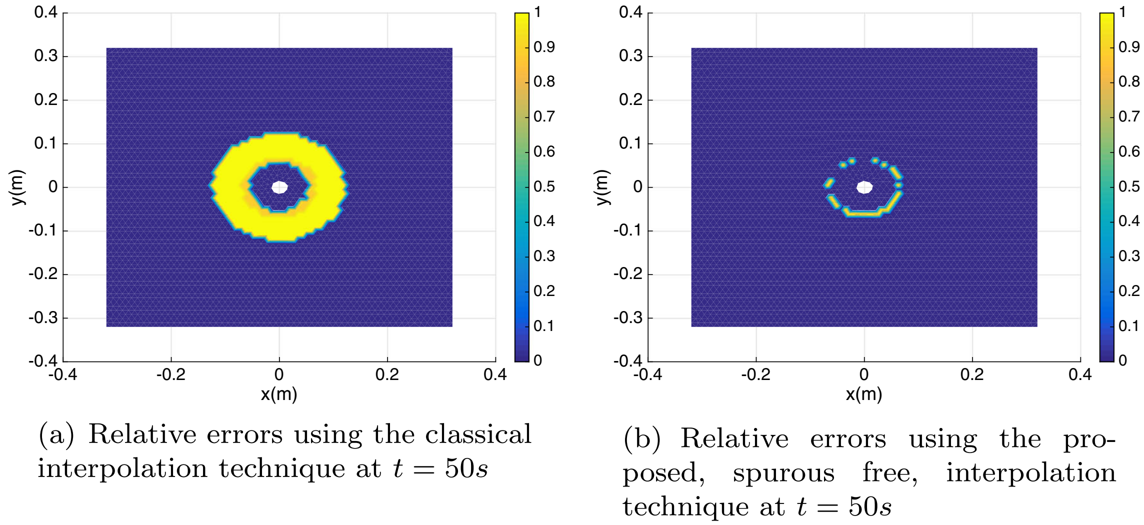
The propagation of a front remains a tricky issue when using again interpolation. In this section we tackle the propagation of a front in resin transfer molding (RTM) processes. The front propagation can be computed with a non intrusive approach using any classical software. We compute first different snapshots using different permeability values. Figure 14a shows a snapshot for $k_x = k_y = k$, while Fig. 14b shows a snapshot for $k_x = k_y = 0.1k$, both for the same filling time. Interpolating the two solutions using classical interpolations at $k_x = k_y = 0.5k$ would yield the result illustrated in Fig. 15, where spurious interpolation artifacts can be noticed.

The technique introduced in Section "Geometrical mapping in 2D" does not apply in the present case, being the injection point located at the same position and remaining unchanged. In that situation a possible solution consists of transforming the 2D domain (x, y) into the 2D polar coordinates domain (r, θ) attached to the injection point. In (r, θ), for the considered problem, the front of snapshot one is defined from $R_1(\theta)$, while the front in snapshot two reads $R_2(\theta)$. We can therefore interpolate the level sets R_1 and R_2 at each value of θ to define R_i related to $k_i = 0.5k$. The resulting solution is depicted in Fig. 16. Figure 17 shows the error maps in the domain with respect to the exact solution, for both cases using classical interpolation and the one here proposed, spurious free, interpolation technique. An excellent accuracy can be noticed. Only very few nodes on the outer interface of the propagation region are not exactly captured, while classical interpolation exhibits completely wrong results.

This idea can be further generalized for different values of k_x and k_y. Figure 18 shows a classical interpolation of 2 solutions for different k_x and k_y, while Fig. 19 shows the fronts $R_1(\theta)$ and $R_2(\theta)$. The interpolation of the front at every θ yields the result illustrated in Fig. 20.

Flow front prediction with the injection point location as parameters

In this section, the techniques discussed in Sections "Geometrical mapping in 2D" and "Flow fronts interpolation", can be combined for simulating the flow front evolution for any permeability and any position of the injection point using only few snapshots. To check the method, we generated a matlab GUI application to modify the permeability values and the injection point coordinates. The solution is therefore obtained in real time. Figure 21 illustrates the front propagation starting from different points and for different permeabilities k_x and k_y, combining the techniques previously introduced in Sections "Geometrical mapping in 2D" and



(a) Relative errors using the classical interpolation technique at $t = 50s$

(b) Relative errors using the proposed, spurous free, interpolation technique at $t = 50s$

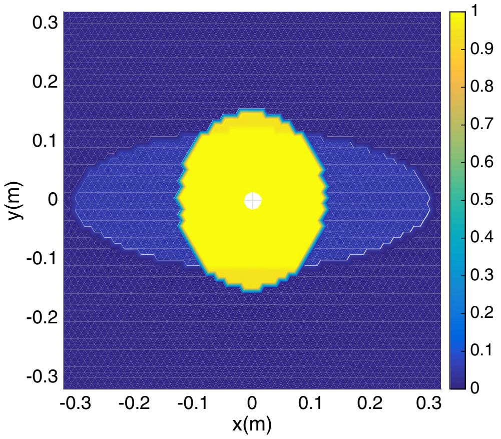


Fig. 18 Solution using classical interpolations for different permeability tensors

"Flow fronts interpolation". Figure 21 also illustrates 4 sliders that are used to change the values of the permeability and injection point coordinates.

Moreover, we can compute the associated pressure fields at different points and at different time steps. However a slight adaptation for interpolating the pressure is needed. In fact, we first extract the pressure pattern $P(r, \theta)$ at every θ_j, with $\theta_j \in [0, 2\pi]$:

$$\begin{cases} P_j^1 = P(r_1, \theta = \theta_j) \text{ for } R = R_1, \\ P_j^2 = P(r_2, \theta = \theta_j) \text{ for } R = R_2. \end{cases} \tag{6}$$

Later on, we scale the pressures P_j^1 and P_j^2 to act in the domain with the flow front located at R_j:

$$\begin{cases} P_{j\,scaled}^1(r_1 \cdot \frac{R_j}{R_1}, \theta_j) = P_j^1(r_1, \theta_j), \\ P_{j\,scaled}^2(r_2 \cdot \frac{R_j}{R_2}, \theta_j) = P_j^2(r_2, \theta_j). \end{cases} \tag{7}$$

Finally we interpolate the scaled pressures to obtain the final pressure, as represented in Fig. 22 for a given angle θ_j. Figure 23 shows the pressure fields for different injection point locations and different permeabilities.

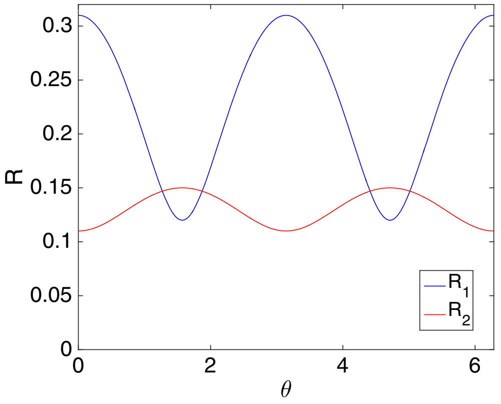


Fig. 19 Flow fronts at the same time for two different permeability tensors

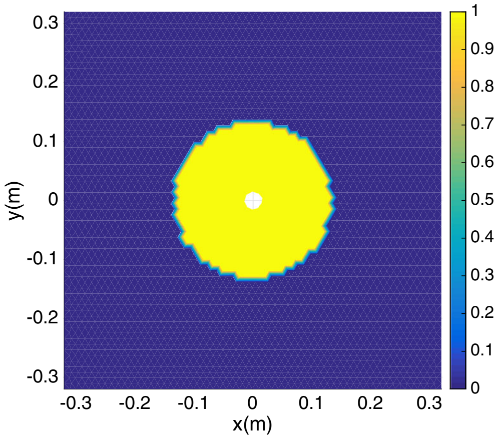


Fig. 20 Flow front prediction using the interpolation in the $(r; \theta)$ coordinates

Figure 24 and 25 depicts similar results for larger times when the flow front is affected by the presence of the mold walls.

For quantifying the error, the pressure field is computed for $k_x = k_y = 5\ 10^{-11}$ Pa.s and the injection point located at $x = y = 0.2m$. The reference solution is depoicted in Fig. 26a, the interpolated solution using the proposed mapping in Fig. 26b and the error between both in Fig. 26c. As it can be noticed, the maximum relative error does not exceed 6% as shown in Fig. 26c.

A more elaborate case: SMC process

This section addresses the flow front evolution in sheet molding compound (SMC) processes. We consider the shell geometry illustrated in Fig. 27. The considered curvilinear coordinate s at any node M can be computed starting from $s = 0$ at $x = 0$

The computed solutions are then mapped into the (s, y) domain, where the considered shell becomes a plate 2D part. This transformation leads to the $2D$ domain illustrated in Fig. 28. The preform flow front induced by the squeezing is illustrated in the (s, y) domain, where the center of the flow front O is considered to be the center of the initially deposited preform. To correctly interpolate the preform front, coordinates (s, y) are mapped into the polar coordinates, with the flow front defined by $R(\theta)$ for every $\theta \in [0, 2\pi]$.

Five transient SMC simulations were performed using commercial ESI software, with different initial position of the squeezed preform that resulted in five different flow fronts $R_i(\theta, t)$, with $i = 1, \ldots, 5$. Figures 29a and b illustrate the preform deformation in the mold at the same time step for two initial preforms at different initial positions.

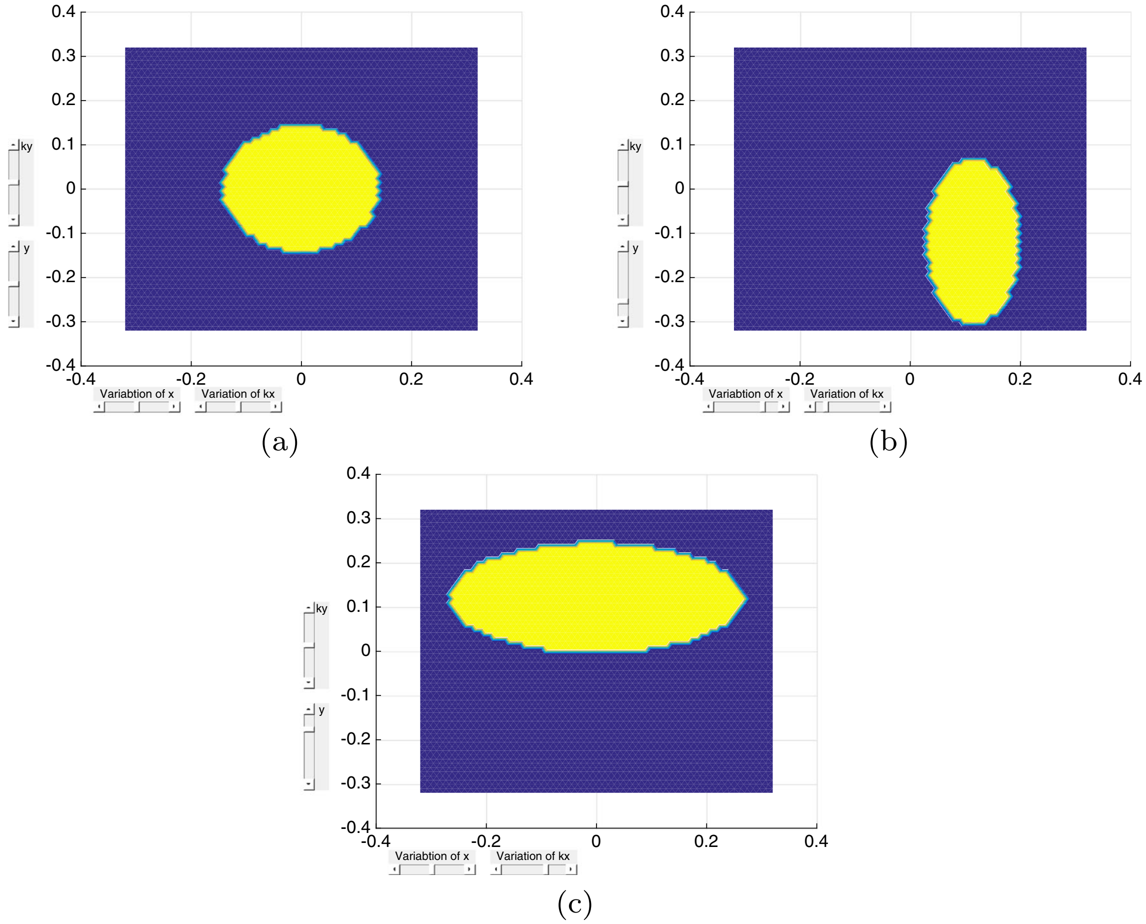


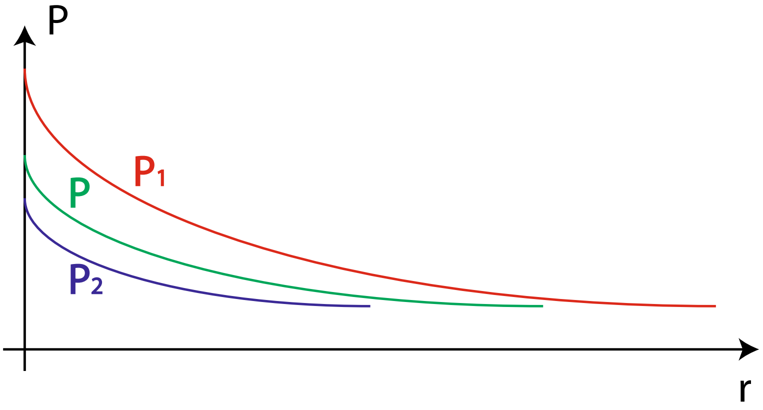
Fig. 21 Different flow front interpolated solutions at the same time step with permeability and injection point location as parameters

s_{O_i} represents the curvilinear coordinate of the center of the simulated preform. The compression of a new preform of center C can be simulated by using the interpolation of $R_i(\theta, t)$ with respect to its initial position of C. Figure 30 illustrates the solution using classical interpolation, from the two illustrated configurations shown in Fig. 29 with $s_C = s_{O_2} \cdot 0.4 + s_{O_3} \cdot 0.6$, while Fig. 31 illustrates the same solution by employing the proposed strategy.

Conclusion

In this work we illustrated the possibility of constructing computational vademecums by adapting the interpolations of snapshots with highly localized effects such as front propagation and localized deformations. This approach relies on the use of physically-based mappings. The approach is later leveraged for different applications allowing

Fig. 22 Pressure interpolation at a given θ_j



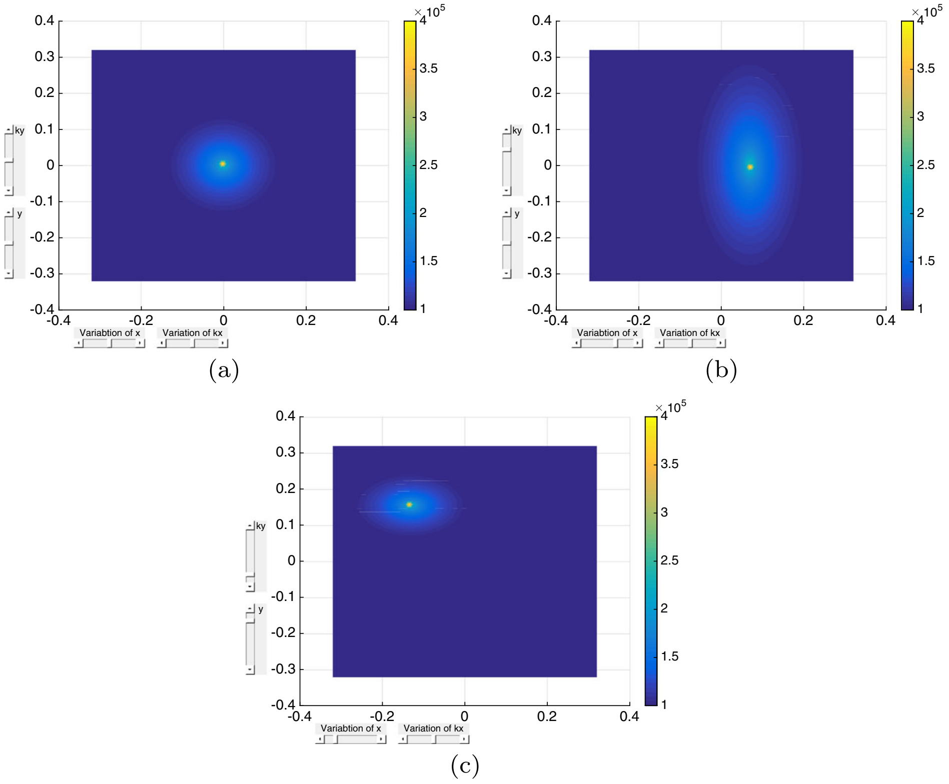
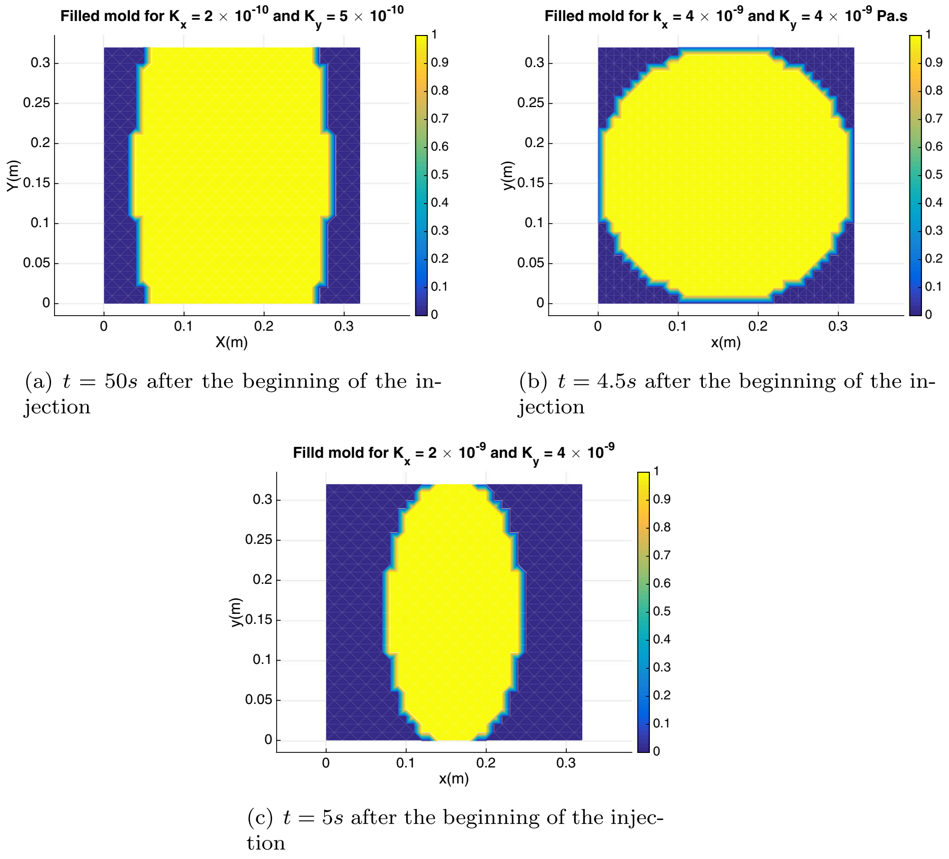


Fig. 23 Different interpolated pressure fields at the same time step


Filled mold for $K_x = 2 \times 10^{-10}$ and $K_y = 5 \times 10^{-10}$

Filled mold for $k_x = 4 \times 10^{-9}$ and $K_y = 4 \times 10^{-9}$ Pa.s

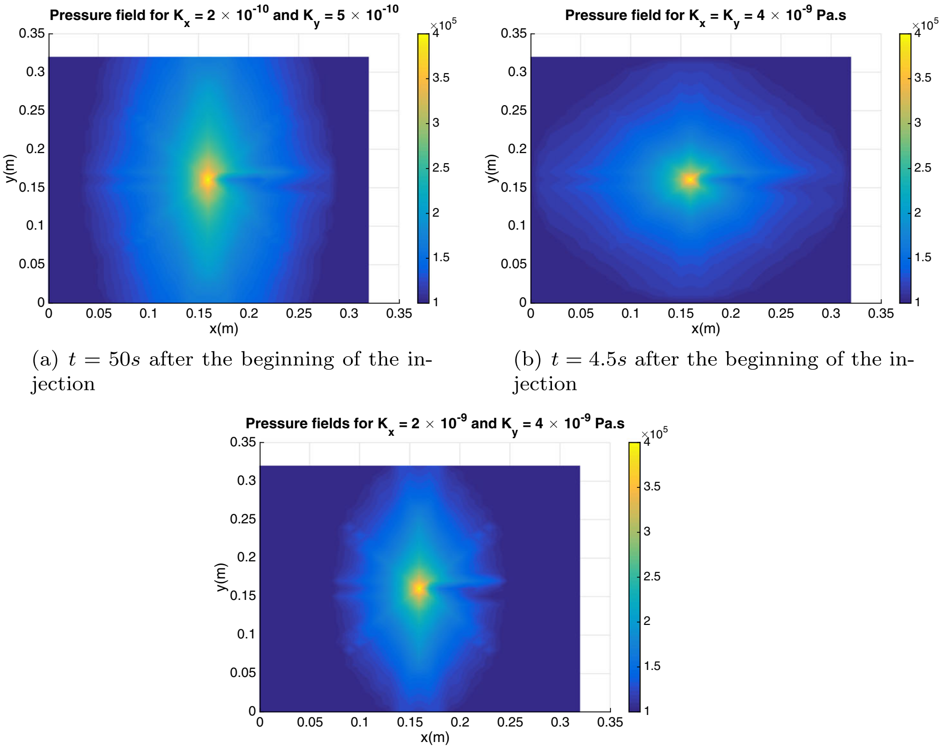
Filld mold for $K_x = 2 \times 10^{-9}$ and $K_y = 4 \times 10^{-9}$

(a) $t = 50s$ after the beginning of the injection

(b) $t = 4.5s$ after the beginning of the injection

(c) $t = 5s$ after the beginning of the injection

Fig. 24 Interpolated flow front positions for different combination of permeabilities

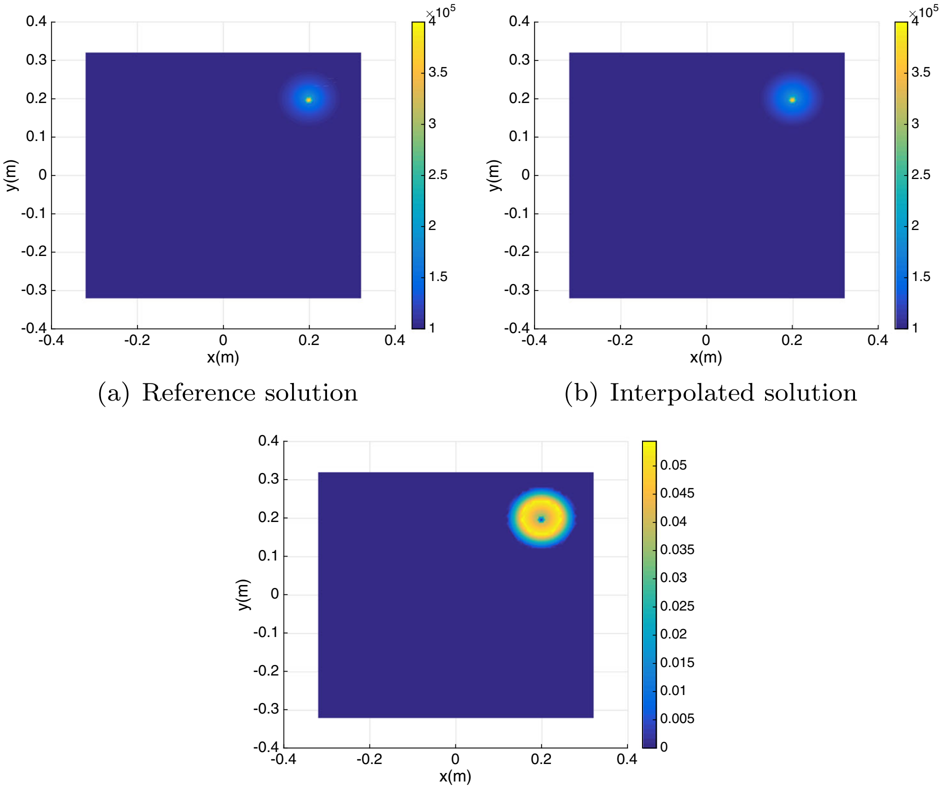


(a) $t = 50s$ after the beginning of the injection

(b) $t = 4.5s$ after the beginning of the injection

(c) $t = 5s$ after the beginning of the injection

Fig. 25 Interpolated pressure fields for different combination of permeabilities



(a) Reference solution

(b) Interpolated solution

(c) Relative error

Fig. 26 Quantifying the error in the pressure calculation at $t = 97.5$s

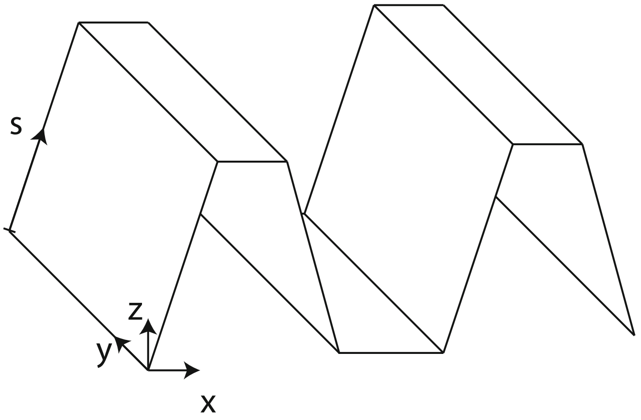


Fig. 27 The shell part considered for the SMC simulation. The coordinate s is a curvilinear coordinate used to map the $3D$ shell part into a $2D$ domain

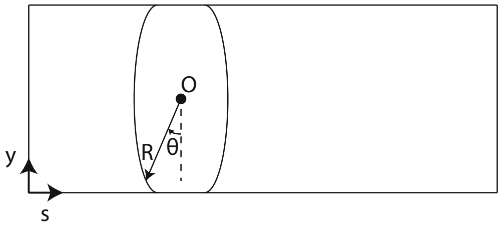
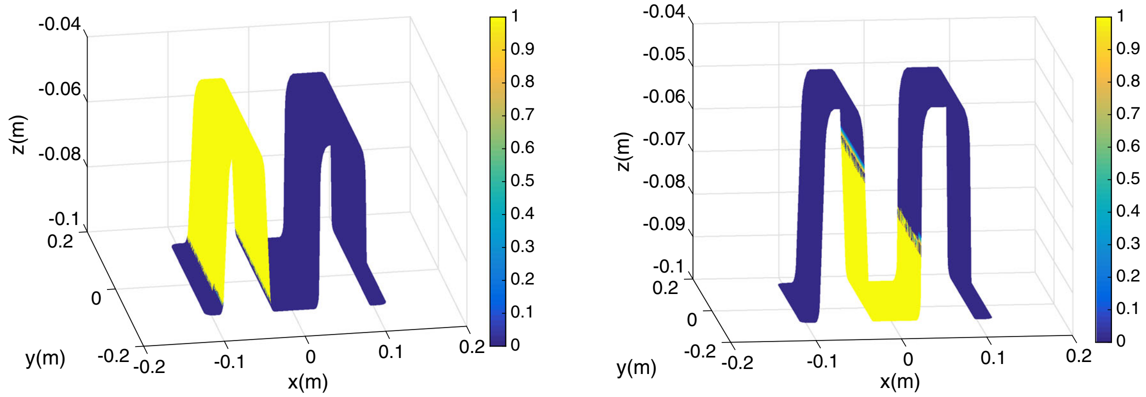


Fig. 28 The shell shape mapped into the (s, y) domain. The preform center O is considered as the initial center of the initially deposited preform



(a) Initial position O_2

(b) Initial position O_3

Fig. 29 Simulation of the deformation of a preform in a mold obtained from ESI® commercial software for two initial preform placement

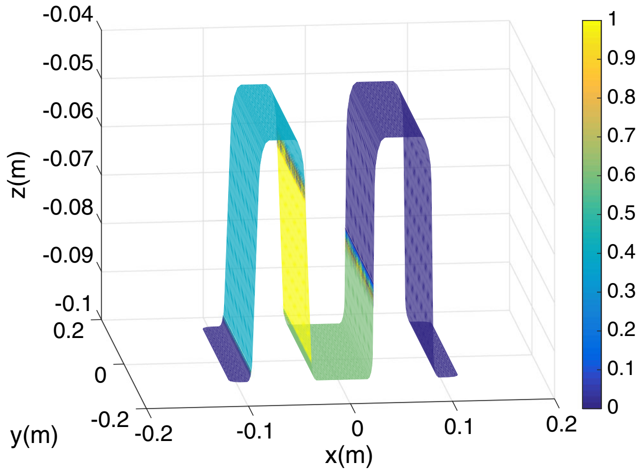


Fig. 30 Classical interpolation for an intermediate preform placement

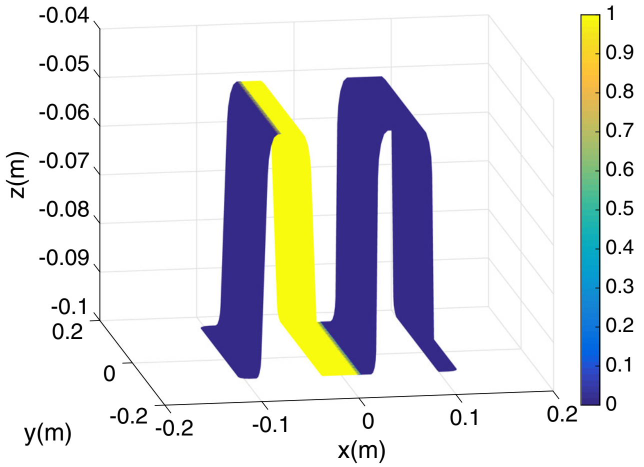


Fig. 31 Interpolation by using the adapted mapping

real-time simulations in previously unreachable parametric simulations. The approach does not require any re-meshing of the domain since all the mathematical procedures apply in the reference domain.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Aguado JV, Borzacchiello D, Ghnatios C, Lebel F, Upadhyay R, Binetruy C, Chinesta F (2017) A simulation app based on reduced order modeling for manufacturing optimization of composite outlet guide vanes. Adv Model Simul Eng Sci 4(1): 1–26
2. Aguado JV, Borzacchiello D, Lopez E, Abisset-Chavanne E, Gonzalez D, Cueto E, Modeling F. Chinesta. (2017) From Microstructure Investigations to Multiscale Bridging the Gap, chapter New trends in computational mechanics : model order reduction, manifold learning and data-driven. Wiley, pp 1–20
3. Bordeu F, Ghnatios C, Boulze D, Carles B, Sireude D, Leygue A, Chinesta F (2015) Parametric 3d elastic solutions of beams involved in frame structures. Adv Aircraft Spacecraft Sci 2(3):233–248
4. Borzacchiello D, Aguado JV, Chinesta F (2019) Non-intrusive sparse subspace learning for parametrized problems. Arch Comput Methods Eng 26:303–326
5. Chinesta F, Cueto E, Abisset E, Duval JL, El Khaldi F (2020) Virtual, digital and hybrid twins. a new paradigm in data-based engineering and engineered data. Arch Comput Methods Eng 27:105–134
6. Chinesta F, Keunings R, Leygue A (2014) The proper generalized decomposition for advanced numerical simulations. Springer, Berlin
7. Cueto E, Ghnatios C, Chinesta F, Monte N, Sanchez F, Falco A (2014) Improving computational efficiency in lcm by using computational geometry and model reduction techniques. Key Eng Mater 611:339–343
8. Dohi T, Osaki S (1993) A note on portfolio optimization with path-dependent utility. Ann Oper Research 45(1):77–90
9. Sol H, Morren G, Bossuyt S (2008) 2d permeability tensor identification of fibrous reinforcements for rtm using an inverse method. Compos Part A 39:1530–1536
10. Ghnatios C, Montes N, Tertrais H, Duval J-L, Abisset-Chavanne E, Falco A, process F. Chinesta. (2019) Towards parametric rtm the interpolative mapping. AIP Conf Proc 100004:2113
11. Ghnatios C, Mathis CH, Simic R, Spencer ND, Chinesta F (2017) Modeling soft permeable matter with the proper generalized decomposition (pgd) approach, and verification by means of nanoindentation. Soft Matter 13:4482–4493
12. Giraldi L, Litvinenko A, Liu D, Matthies HG, Nouy A (2014) To be or not to be intrusive? the solution of parametric and stochastic equations — the “plain vanilla” galerkin case. SIAM J Sci Comput 36(6):A2720–A2744
13. Gonzalez D, Alfaro I, Quesada C, Cueto E, Chinesta F (2015) Computational Biomechanics for Medicine, chapter Vademecums for Real-Time Computational Surgery. Springer, pp 3–12
14. Huang Y, Zhao L, Woensel TV, Gross J-L (2017) Time-dependent vehicle routing problem with path flexibility. Transp Res Part B: Methodol 95:169–195
15. Ibanez R, Abisset-Chavanne E, Ammar A, Gonzalez D, Cueto E, Huerta A, Duval JL, Chinesta F (2018) A multi-dimensional data-driven sparse identification technique: the sparse proper generalized decomposition. Complexity 2018(5608286):11
16. Leon A, Mueller S, de Luca P, Said R, Duval JL, Chinesta F (2019) Non-intrusive proper generalized decomposition involving space and parameters: application to the mechanical modeling of 3d woven fabrics. Adv Model Simul Eng Sci 6(13):1–20
17. Lopez E, Gonzalez D, Aguado JV, Abisset-Chavanne E, Cueto E, Binetruy C, Chinesta F (2018) A manifold learning approach for integrated computational materials engineering. Arch Comput Method Engineering 5(1):59–68
18. Lu Y, Blal N, Gravouil A (2019) Datadriven hopgd based computational vademecum for welding parameter identification. Computational mechanics:1
19. Ojo OO, Taban E, Kaluc E (2018) Loop travel-path of fibre laser welded alclad aa2219-o alloy. J Mater Process Technol 251:118–126
20. Quaranta G, Ziane M, Haug E, Duval JL, Chinesta F (2019) A minimally-intrusive fully 3d separated plate formulation in computational structural mechanics. Adv Model Simul Eng Science 6(11):1–22
21. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
22. Tsay JJ, Arora JS (1990) Nonlinear structural design sensitivity analysis for path dependent problems. part 1 General theory. Comput Methods Appl Mech Eng 81(2):183–208
23. Tsiolakis V, Giacomini M, Sevilla R, Othmer C, Huerta A (2019) Nonintrusive proper generalised decomposition for parametrised incompressible flow problems in openfoam. Open source arXiv:1906.05403, pp 1–35
24. Wang X, Yan Y, Gu X (2019) Spot welding robot path planning using intelligent algorithm. J Manuf Process 42:1–10
25. Zhu Z, Ghong J, Huang Y, Zhan Y, Gong M, Zhang L (2019) Experimental research on transition from scale 3d printing to full-size printing in construction. Construct Build Mater 208:350–360
26. Zou W, Conti M, Diez P, Auricchio F (2017) A nonintrusive proper generalized decomposition scheme with application in biomechanics. Int J Numer Methods Engineering 113:230–251
27. Zou X, Conti M, Diez P, Auricchio F (2017) Pgd for constrained parametric space with a non-intrusive implementation. In: International conference on adaptive modeling and simulation Admos 2017, Verbania, Italy. ECCOMAS