Surrogate parametric metamodel based on Optimal Transport

Sergio Torregrosa, Victor Champaney, Amine Ammar, Vincent Herbert, Francisco Chinesta

Surrogate parametric metamodel based on Optimal Transport

Sergio Torregrosa[a,b,*], Victor Champaney[c], Amine Ammar[d], Vincent Herbert[b], Francisco Chinesta[c]

[a] *PIMM, Arts et Métiers Institute of Technology, 151 Boulevard de l'Hopital, F-75013 Paris, France*
[b] *STELLANTIS, Route de Gisy, 78140 Vélizy-Villacoublay, France*
[c] *ESI Chair, PIMM, Arts et Métiers Institute of Technology, 151 Boulevard de l'Hopital, F-75013 Paris, France*
[d] *ESI Chair, LAMPA, Arts et Métiers Institute of Technology, 2 Boulevard du Ronceray BP 93525, 49035 Angers cedex 01, France*

Abstract

The description of a physical problem through a model necessarily involves the introduction of parameters. Hence, one wishes to have a solution of the problem that is a function of all these parameters: a parametric solution. However, the construction of such parametric solutions exhibiting localization in space is only ensured by costly and time-consuming tests, which can be both numerical or experimental. Numerical methodologies used classically imply enormous computational efforts for exploring the design space. Therefore, parametric solutions obtained using advanced nonlinear regressions are an essential tool to address this challenge. However, classical regression techniques, even the most advanced ones, can lead to non physical interpolation in some fields such as fluid dynamics, where the solution localizes in different regions depending on the problem parameters choice. In this context, Optimal Transport (OT) offers a mathematical approach to measure distances and interpolate between general objects in a, sometimes, more physical way than the classical interpolation approach. Thus, OT has become fundamental in some fields such as statistics or computer vision, and it is being increasingly used in fields such as computational mechanics. However, the OT problem is usually computationally costly to solve and not adapted to be accessed in an online manner. Therefore, the aim of this paper is combining advanced nonlinear regressions with Optimal Transport in order to implement a parametric real-time model based on OT. To this purpose, a parametric model is built offline relying on Model Order Reduction and OT, leading to a real-time interpolation tool following Optimal Transport theory. Such a tool is of major interest in design processes, but also within the digital twin rationale.

Keywords: Artificial intelligence; Optimal Transport; Fluid dynamics; sPGD; Model Order Reduction

1. Introduction

When an object, available at a given point, needs to be moved at a target point using the least possible effort, the shortest path gives the optimal solution. The theory of Optimal Transport (OT) [55] generalizes that idea: several objects (or a continuous distribution) from a configuration are moved to another configuration while minimizing

* Corresponding author at: PIMM, Arts et Métiers Institute of Technology, 151 Boulevard de l'Hopital, F-75013 Paris, France.
E-mail addresses: sergio.torregrosa@stellantis.com (S. Torregrosa), victor.champaney@ensam.eu (V. Champaney), amine.ammar@ensam.eu (A. Ammar), vincent.herbert@stellantis.com (V. Herbert), francisco.chinesta@ensam.eu (F. Chinesta).

the cost of the transport [40]. In this context, Optimal Transport offers a general mathematical framework to define distances (also known as Wasserstein or Earth Mover's Distances) and interpolate between functions on general metric spaces, taking into account the geometry of the underlying space.

Initially considered by Monge [37] in the 18th century, who, motivated by military applications, studied how to transport a certain amount of soil from the source to the construction places in the cheapest way. In this original problem, the cost of moving a unit of soil was the distance traveled [55]. Later on, in the 20th century, Kantorovich [23] introduced the linear programming formulation of the problem, presented in detail by C. Villani in [54]. Today, Optimal transport has a wide range of applications, such as, computer vision [15,44], image processing [42] and machine learning [12,51].

Interpolation is a mathematical fundamental operation widely used in numerical simulation. However, despite of its simplicity, usual interpolation does not always yield physical meaningful results. Optimal transport defines a radically different way to interpolate and calculate distances between functions having disjoint supports. Indeed, while the classical Euclidian interpolation leads to a mixture of the two functions (one which progressively disappears and the other which progressively appears), OT-based interpolation leads to a progressive translation and scaling, as it can be seen in Fig. 14. Such a numerical solution is much more realistic in several fields, such as, computer graphics or fluid dynamics [25], justifying its widespread use. In this sense, Optimal Transport quantifies the distance between two functions by considering the cheapest way to move all mass units describing one function to reshape it into the other. Thus, two almost identical functions differing only by a little displacement are considered to be very close by the OT approach while they can be considered as very dissimilar by the usual $L1$ or $L2$ norms.

However, computing optimal transport is usually numerically expensive and not adapted to online approaches since it is usually performed by sampling from the source and target distributions and solving then a discrete linear programming problem which results numerically costly and statistically inefficient [57]. Indeed, classical linear programming approaches to OT [5] scale roughly with cubic complexity. Some approaches have been developed in order to alleviate such an issue but remain very restrictive working only in specific frameworks: e.g., a semi-discrete approach with Laguerre's cells [25,36] or a dynamical formulation [2]. It can also be noted that entropy regularized approaches to solve OT have been developed [12]. In particular, Sinkhorn's algorithm [49] involves only matrix–vector products and allows the regularized distance to be a smooth function of all its parameters. Nevertheless, since the problem depends on a fixed sampling of points, it is not adapted to continuously sampled data or to a real-time tool [35]. Moreover, now that Optimal Transport can be more efficiently approximated, more interesting OT problems can be considered such as the Wassertein Barycenters [1]. Finally, building upon recent advances in the field of Neural Networks, these are nowadays being employed for transforming probability distributions and to estimate the Optimal Transport mapping [8,28,30]. Hence, they could open new routes for Optimal Transport.

Despite of the recent advances to solve OT, the problem remains numerically expensive and not accessible in online applications. Therefore, the aim of this paper is to propose a parametric model based on Optimal Transport and accessible in an online manner that allows to interpolate between several simulations of a parametric problem in real time and following the motion-based interpolation that characterizes OT. Based on displacement interpolation, introduced in [34], the method developed in this paper approaches the interpolation problem as a optimal mass transport problem where each unit of mass describing the source distribution needs to be transported to the target distribution while minimizing the total cost associated to the transport [3]. For two distributions, the problem can thus be written in terms of bipartite graph matching [20]. The paths taken by each unit of mass are parameterized and an interpolated solution can thus be accessed by partially displacing the mass along the paths at given values of the parameters. To this purpose, a parametric model is built offline relying on Smoothed-Particle Hydrodynamics decomposition and Model Order Reduction.

It can be noted that interpolation of 1D functions can be achieved using inverse cumulative distribution functions (CDFs) [33,43]. Although this approach (which is a special case of displacement interpolation) gives the desired result, it cannot be applied in higher dimension problems in which we are interested in. Moreover, mass transport approaches have been developed in order to calculate distances between images [45] or to interpolate between 2D shapes [29]. Finally, in computer graphics, it has been proposed to use Gaussians in order to fit normal distributions and then interpolate the parameters of the Gaussians [16,52].

The approach introduced in this paper proceeds in two stages: an offline non-intrusive training of the parametric optimal transport based model, followed by an online application of the model in the parametric space. Indeed, high-fidelity simulations of the parametric problem are used to train the model without any need of an insight of the problem equations, hence the non-intrusive terminology.

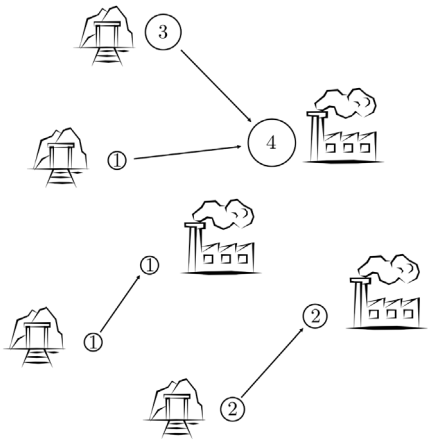

Fig. 1. Discrete Optimal Transport between $N = 4$ mines and $M = 3$ factories. The resource produced by the mines is: $a_1 = 3$, $a_2 = 1$, $a_3 = 1$ and $a_4 = 2$. The resource consumed by the factories is: $b_1 = 4$, $b_2 = 1$ and $b_3 = 2$. The minimized cost is the square of the Euclidian distance.

First, in order to train the model, high-fidelity simulations are used in the offline stage. These simulations belong to a parametric space where our parametric problem is defined. The offline stage proceeds in several steps: first, the high fidelity simulations of the training set are decomposed into the sum of elementary identical Gaussians. Then, each Gaussian from each simulation is paired with one Gaussian of every other simulation in order to respect Optimal Transport behavior. A Genetic Algorithm is employed in order to approximate the k-Dimensional Matching problem obtained. Next, a sparse Proper Generalized Decomposition (sPGD) regression is applied over the coefficients of the Proper Orthogonal Decomposition (POD) of the matrix of snapshots, which have just been decomposed into Gaussians, in order to create the parametric model.

Then, this model can be evaluated in the parametric space of the problem in the online stage leading to a partial advection of all the Gaussians. Summing all those Gaussians allows to reconstruct, in real-time, the high-fidelity simulation interpolated in this point of the parametric space following the Optimal Transport theory.

In this paper, first, the main concepts of optimal transport used to build the offline approach are introduced and illustrated. Then, the underlying idea and the methodology followed to develop the offline stage are presented. Finally, the online application of the parametric model is presented and discussed.

2. Revisiting Optimal Transport

The aim of this section is to introduce the Optimal Transport and its theoretical tools used in the later proposed parametric Optimal Transport. This section should not be considered as an exhaustive presentation of OT but as an introduction to the main concepts. For additional details the reader can refer to [40] and the references therein.

The initial optimal transport problem, introduced by Monge [37], was how to calculate the most optimal way to transport a certain amount of soil from an initial landscape to a desired target landscape. The cost to minimize was the distance traveled. In order to illustrate the Monge's problem from a discrete perspective, let us consider the optimal transport of a resource produced by N mines and transported to M factories that minimizes the square of the Euclidian distance traveled, as it is illustrated in Fig. 1.

Each mine $n \in [\![N]\!]$ (where the notation $[\![N]\!]$ should be read as $\{1, \ldots, n, \ldots, N\}$), located at x_n, produces a quantity a_n of the resource and each factory $m \in [\![M]\!]$, located at y_m, consumes a quantity b_m of this same resource. Then, using the notion of measure, both distributions of the resource produced by the mines, α, and consumed by the factories, β, are defined as

$$\alpha = \sum_{n=1}^{N} a_n \delta_{x_n} \quad \text{and} \quad \beta = \sum_{m=1}^{M} b_m \delta_{y_m} \tag{1}$$

where δ_{x_n} and δ_{y_m} are the Dirac at positions x_n and y_m respectively.

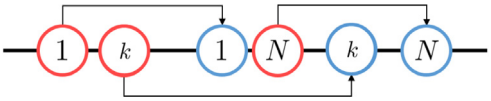

Fig. 2. 1D Monge problem with $N = M$ and $a_n = b_m = 1/N$. Mines are represented by red circles and factories by blue ones. The optimal matching is illustrated by black arrows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The Monge problem consists, thus, in looking for a map T that connects each point x_n with a single point y_m such that it pushes the resource mass of α produced by the mines towards the resource mass of β consumed by the factories. Since no resource can be produced or destroyed during the transport, this map $T : \{x_1, \dots, x_N\} \rightarrow \{y_1, \dots, y_M\}$ must satisfy the mass conservation

$$\forall m \in [\![M]\!], b_m = \sum_{n:T(x_n)=y_m} a_n, \tag{2}$$

which is equivalent to the compact form $T_{\#}\alpha = \beta$. It can be noted that this map is surjective. Moreover, this map must minimize the transportation cost defined as the square of the Euclidian distance between the mine n and its associated factory m:

$$C_{x_n,y_m} = \|x_n - y_m\|_2^2, \tag{3}$$

resulting in the minimization problem:

$$\min_T \sum_{n=1}^{N} C_{x_n,T(x_n)}. \tag{4}$$

Let now simplify this problem by supposing that there are the same number of mines than factories, i.e. $N = M$. Moreover, suppose that each mine produces the same amount of resource and that each factory consumes this same amount of resource, i.e. $a_n = b_m = 1/N$. Therefore, the map T is a bijection and the minimization problem (4) becomes a deterministic assignment problem.

Under these hypotheses, the 1D Monge problem, which is equivalent to the 1D optimal matching problem, represents the simplest Optimal Transport problem. Indeed, in this case the solution simply consists in sorting the mines and factories locations along the 1D axis in ascending order. Then, one can match the first mine with the first factory and this successively, i.e. the kth mine with the kth factory, as it can be seen in Fig. 2 where the mines are represented by red circles and the factories by blue ones.

In 2D, this same problem complexifies since no order relation exists, but remains easy to solve using, for instance, linear programming. Indeed, the problem consists in the optimal assignment problem between two point clouds (where each cloud has the same number of points, each point has the same amount of mass and the cost is the square of the Euclidian distance between points), as it is illustrated in Fig. 3 where each point has a x and y coordinates. Again, the mines are represented by red circles and the factories by blue ones.

The red distribution of the resource produced by the mines, α, and the blue one of the resource consumed by the factories, β, are called source and target distributions respectively. It can be noted that in higher dimensions the problem does not further complexifies unless by the increased computational time for the distances calculation.

Once both point clouds are optimally matched, it is possible to interpolate, in an Optimal Transport way, between the two distributions by partially displacing all the points over the corresponding segments built between each point of the source distribution and its corresponding optimal match on the target distribution. The resulting interpolated distribution (or point cloud) is illustrated in Fig. 3 by violet points.

3. Parametric Optimal Transport based model

In this section, our methodology is presented in detail. First, the methodology is as a whole is introduced. Then, a full description of each different step is presented.

The construction of the parametric Optimal Transport based model represents the offline stage of the approach and follows four steps. First, all the high-fidelity simulations, among which it is intended to interpolate, are decomposed

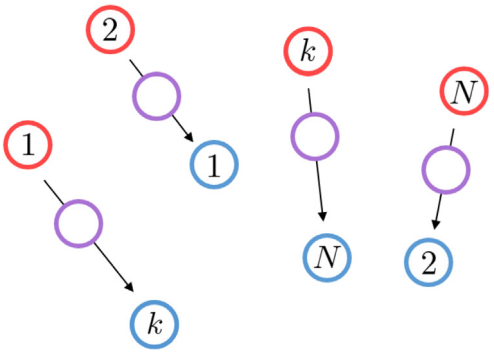

Fig. 3. 2D Monge problem with $N = M$ and $a_n = b_m = 1/N$. Mines are represented by red circles and factories by blue ones. The optimal matching is illustrated by black arrows. The interpolated distribution is illustrated by violet points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

into a sum of Gaussians, of which the standard-deviation is fixed as an hyper-parameter and the only variables are the means. Then, each Gaussian from each simulation is paired with one, and only one, Gaussian from every other simulation using a Genetic Algorithm, in order to minimize the OT based cost. Next, a Proper Orthogonal Decomposition (POD) is applied over the matrix of snapshots, composed by the positions of all the Gaussians of each simulation. Finally, a regression is applied over the retained coefficients of this POD, leading to a parametric Optimal Transport based model which can be accessed in an online manner. Indeed, once the parametric model has been trained using the high-fidelity simulations, it can be used in an online manner to explore the whole parametric space of the problem.

3.1. Parametric OT methodology

As it has been presented before, under some simplifying hypotheses, the 2D minimization problem (4) becomes a deterministic assignment problem between 2 point clouds. The parametric model based on OT developed in this paper relies on this idea. Without loss of generality, consider a distribution $\psi : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}^+$. Note that we suppose here that the image of ψ is strictly positive. First, the distribution is normalized:

$$\rho = \frac{\psi}{\mathcal{I}} \quad \text{where} \quad \mathcal{I} = \int_{\Omega} \psi \mathrm{d}\Omega. \tag{5}$$

Therefore, the idea is to decompose the source and target distributions, ρ_0 and ρ_1, into a sum of N points, or particles, as it can be seen in the first and third columns of Fig. 4. Following a Smoothed-Particle Hydrodynamics (SPH) [26] rationale, each particle is modeled by a Gaussian function of constant standard-deviation σ, which becomes an hyper-parameter of the problem (optimal choices of σ will be addressed later). The only variables, for each distribution, are, thus, the means $\boldsymbol{\mu}$ of each particle, i.e. N vectors of 2 components: μ_x and μ_y (since it is a 2D problem). To illustrate the idea let us take the normalized distribution $\rho : \Omega \in \mathbb{R}^2 \rightarrow \mathbb{R}^+$. The distribution is thus approximated as follows

$$\bar{\rho}(\mathbf{x}) = \sum_{n=1}^{N} G_{\boldsymbol{\mu}_n,\sigma}(\mathbf{x}), \tag{6}$$

where $G_{\boldsymbol{\mu}_n,\sigma}$ is a 2D-Gaussian of standard-deviation σ, mean $\boldsymbol{\mu}_n$ and $1/N$ mass:

$$G_{\boldsymbol{\mu}_n,\sigma}(\mathbf{x}) = \frac{1}{N\sigma^2 2\pi} \exp^{\frac{-(\mathbf{x}-\boldsymbol{\mu}_n)^2}{2\sigma^2}}. \tag{7}$$

Once the source and target distributions are decomposed into N particles, the optimal match between the source and target distributions, represented by their N Gaussians, can be calculated. This can be seen as the optimal assignment problem between two point clouds, where each point is a particle identified by its 2 coordinates μ_x and μ_y. Several algorithms exist to solve this optimal matching problem. The solution is presented in the central column of Fig. 4 where the assignment between clouds is presented by dotted lines.

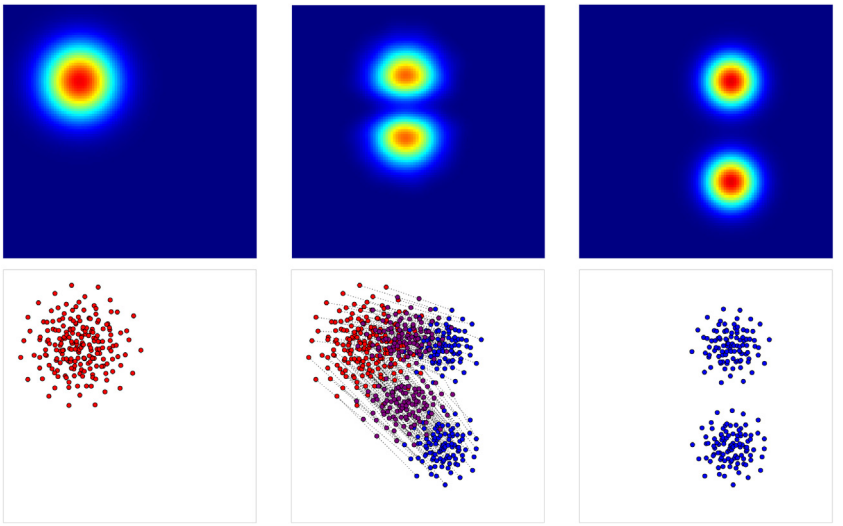

Fig. 4. 1st column: Source distribution ρ_0 and its particle decomposition (red cloud). 2nd column: Interpolated distribution $\hat{\rho}$ and its particle decomposition (violet cloud). The optimal between clouds is illustrated by dotted lines. 3th column: Target distribution ρ_1 and its particle decomposition (blue cloud). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Then, it is possible to interpolate between the two distributions by partially displacing all the particles over the corresponding segments built between each Gaussian particle from the source distribution and its corresponding optimal match on the target distribution, as it is presented in the central column of Fig. 4 by the third violet points cloud. The parametric model is, therefore, built for two distributions. Indeed, one can access to any interpolated solution between the two distributions by adding all the Gaussians functions evaluated at the desired point over their corresponding segments.

However, the problem becomes much more complex when one wants to interpolate between more than two distributions. Indeed, the optimal matching must be done between each distribution and every other one. Hence, each particle of each distribution should be matched with one, and only one, particle from every other distribution such that the cost is minimized. Note that the cost is the sum of the distances between the paired particles from two distributions, and this, summed for each distribution with every other one.

In order to formalize this, let introduce our parametric space $\mathcal{W}(\eta_1, \ldots, \eta_q, \ldots, \eta_Q)$ corresponding to the parametric problem studied where $\eta_q, q \in [\![Q]\!]$ are the parameters. Then, let introduce P distributions (in 2D) in $\mathcal{W}(\eta_1, \ldots, \eta_q, \ldots, \eta_Q)$ corresponding to the high-fidelity simulations of the parametric problem. Each distribution is decomposed into N particles where each particle is described by 2 coordinates, μ_x and μ_y, since the distributions are in 2D. Thus, the nth$\in [\![N]\!]$ particle of the pth$\in [\![P]\!]$ distribution is noted:

$$G_{\boldsymbol{\mu}_n^p,\sigma} \quad \text{where} \quad \boldsymbol{\mu}_n^p = [\mu_{n_x}^p, \mu_{n_y}^p] \in \mathbb{R}^2. \tag{8}$$

Therefore, one can introduce $\boldsymbol{\mu}^p \in \mathbb{R}^{N\times 2}$ as the matrix composed by the coordinates of every particle of the distribution p:

$$\boldsymbol{\mu}^p = \begin{bmatrix} \boldsymbol{\mu}_1^p \\ \vdots \\ \boldsymbol{\mu}_N^p \end{bmatrix} = \begin{bmatrix} [\mu_{1_x}^p, \mu_{1_y}^p] \\ \vdots \\ [\mu_{N_x}^p, \mu_{N_y}^p] \end{bmatrix} \in \mathbb{R}^{N\times 2}. \tag{9}$$

Hence, the cost between 2 distributions p and p' is defined as the sum of the square of the Euclidian distances between the paired particles between the 2 distributions:

$$C_{p,p'}(\phi_p, \phi_{p'}) = \sum_{n=1}^{N} \left\| \boldsymbol{\mu}_{\phi_p(n)}^p - \boldsymbol{\mu}_{\phi_{p'}(n)}^{p'} \right\|_2^2 \tag{10}$$

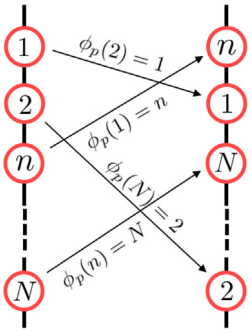

Fig. 5. Illustration of the ordering function ϕ_p. For each particle n of the distribution, the function ϕ_p indicates the new position.

where ϕ_p is a bijection in the set of permutations of N elements. Indeed, $\phi_p : \mathbb{N} \to \mathbb{N}$ associates to each particle n of the distribution p its new position in the sense of order in $\boldsymbol{\mu}^p$. The function ϕ_p represents the ordering of the N particles of the distribution p, as it is illustrated in Fig. 5. One seeks for the optimal function ϕ_p, i.e. the optimal ordering that corresponds to the optimal matching in order to minimize the defined cost.

Then, the cost between the P distributions is defined as the sum of the cost between 2 distributions for all the possible pairs of distributions, i.e. the sum of the cost between all possible pairs:

$$\mathbf{C}_P(\phi_1, \ldots, \phi_p, \ldots, \phi_{P-1}) = \sum_{p=1}^{P-1} \sum_{p'=p+1}^{P} C_{p,p'}(\phi_p, \phi_{p'}). \tag{11}$$

Thus, the P-dimensional optimal matching problem seeks for the $P-1$ orderings ϕ_p, $p \in [\![P]\!]$ of the N particles of each distribution (all except one since when matching two sets it is enough to permute one). This P-dimensional optimal matching problem (12) reads

$$\min_{\phi_1, \ldots, \phi_{P-1}} \mathbf{C}_P(\phi_1, \ldots, \phi_{P-1}). \tag{12}$$

This minimization problem can be seen as seeking for the N distinct P-tuples that minimize the cost $\mathbf{C}_P$, where each P-tuple contains one particle of each distribution. This is illustrated in Fig. 6(b), which has been inspired by the notion of hyper-graph [31]. Indeed, the cost to minimize can be seen from an hyper-graph point of view, where the hyper-graph is composed by P sets of N nodes each and a cost $\mathcal{C}_n$ (sum of the distances of all the possible particle pairs of the hyper-edge) associated to the hyper-edge n joining exactly one node from each set.

One seeks for the optimal matching which is a set of N disjoint hyper-edges such that each node is associated to one and only one hyper-edge of the matching and such that the total cost (sum of the cost of all the hyper-edges) is minimized.

This P-dimensional matching problem corresponds to the generalization of a well known Karp NP-complete problem. Therefore, the aim of this approach is not to find the perfect matching but an optimal reachable one. Once the P-matching problem is solved, each particle of each distribution is linked with exactly one particle from every other distribution, as it is illustrated in Fig. 6(a). Thus, one can "follow" each particle over the P distributions.

Then, the regression over the positions of particles (i.e. the means of the Gaussian functions which are equivalent to the x and y coordinates of the particles, since we are in 2D) leads to a function from the parametric space $\mathcal{W}(\eta_1, \ldots, \eta_Q)$ to $\Omega \in \mathbb{R}^2$. Indeed, for a set of parameters, the regression returns the x and y positions in $\mathbb{R}^2$ of the N particles. Finally, summing these particles recovers the interpolated simulation. The parametric real-time model is thus built and it can be applied in an online manner as schematized in Fig. 7.

Finally, the total mass $\hat{\mathcal{I}}$ is interpolated in order to recover $\hat{\psi}$. For instance, if there are only 2 distributions and a parameter η:

$$\hat{\mathcal{I}} = (1 - \eta)\mathcal{I}_1 + \eta\mathcal{I}_2. \tag{13}$$

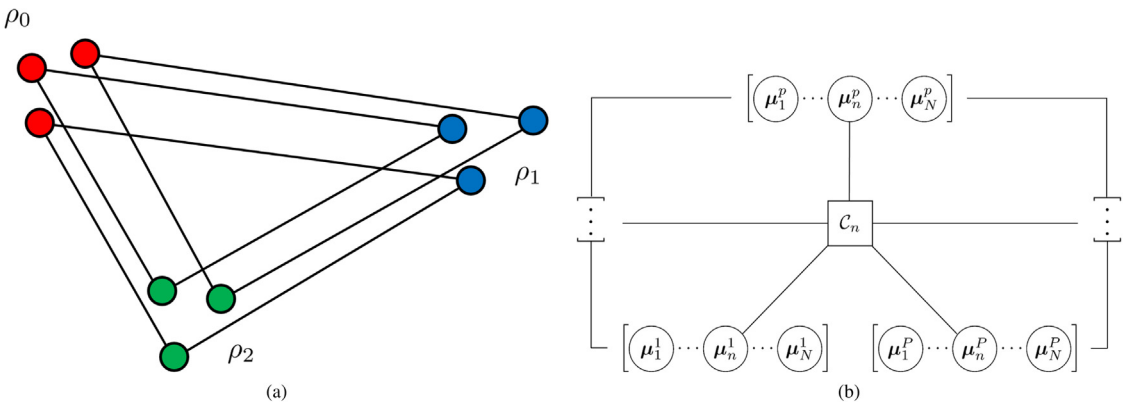

Fig. 6. (a) 3-dimensional matching between the distributions ρ_0, ρ_1 and ρ_2. (b) Hyper-graph point of view of the problem. Three of the P sets are represented (1, p and P) containing each N nodes. The hyper-edge n and its associated cost $\mathcal{C}_n$ are presented.

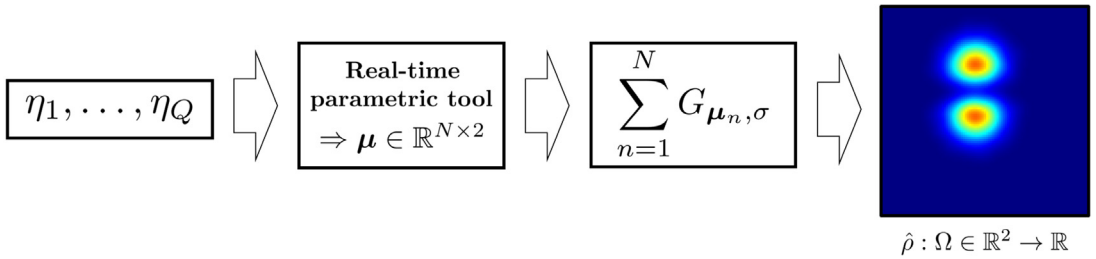

Fig. 7. Interpolated distribution procedure from a not simulated set of parameters using the real-time model.

3.2. Particle Decomposition Methodology

Let introduce now the methodology followed to decomposed the pth distribution $\rho^p : \Omega \in \mathbb{R}^2 \to \mathbb{R}^+$ in Gaussian functions. Its decomposition into N particles (or Gaussians) consists in the optimization problem (14) where the variable is $\boldsymbol{\mu}^p \in \mathbb{R}^{N\times 2}$ and $\mathbf{x}_i, i \in [\![D]\!]$ are the discretisation points of Ω following a uniform meshing. Note that the standard-deviation σ of the Gaussian functions has been fixed for the whole problem as an hyper-parameter. Thus, the optimization problem reads

$$\min_{\boldsymbol{\mu}^p} \frac{1}{2} \left\| \rho^p - \bar{\rho}^p \right\|_2^2 = \min_{\boldsymbol{\mu}^p} \frac{1}{2} \left[\sum_{i=1}^{D} \left(\rho^p(\mathbf{x}_i) - \sum_{n=1}^{N} G_{\boldsymbol{\mu}_n^p,\sigma}(\mathbf{x}_i) \right)^2 \right], \tag{14}$$

where D is the number of points where the distribution ρ^p has been simulated. This optimization problem is solved by using a Gradient Descent approach. First, the optimal choice of the hyper-parameter σ is presented.

3.2.1. Optimal choice of σ

The standard-deviation σ of the Gaussian particles is fixed in advance for all the particles and distributions. Thus, its choice is important. On the one hand, the particles need to be able to catch every feature of the distributions: if σ is too large some features will be filtered. On the other hand, one wishes to minimize the required number of particles in order to reduce the computational cost: if σ is too small the number of particles needed will be computationally too expensive.

In order to chose the best σ, the eigenvalues of the Hessian matrix of the distribution are calculated at each point $\mathbf{x}_i$ of the mesh. At each point, the largest eigenvalue indicates the greatest curvature of the distribution along the corresponding eigenvector. Then, for all the D points, the largest eigenvalue indicates the greatest curvature of the distribution.

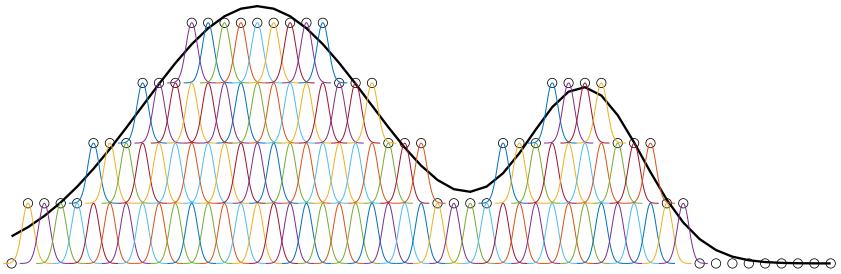

Fig. 8. Example of initialization of the problem (14) in a 1D case (before application of small random displacements) by uniform quantization.

Since the goal of the σ choice is to catch every feature of the distribution, the standard-deviation of the particle that would model the fastest deformation of the distribution should be equal to the standard-deviation of the distribution at this point, or more generally to the greatest eigenvalue of the Hessian matrix at this point. Therefore,

$$\sigma = \sqrt{\left| \frac{\rho_{\lambda_{max}}}{\lambda_{max}} \right|}, \tag{15}$$

where λ_{max} is the largest eigenvalue for all the distribution points and where $\rho_{\lambda_{max}}$ is the distribution value corresponding to the point where λ_{max} is reached. Note that this calculation is done for all the distributions and the smallest σ is retained.

3.2.2. Gradient Descent initialization

In order to initialize the Gradient Descent, a uniform quantization of the continuous distribution ρ^p is realized. Let $\mathbf{s}_j$, $j = 1, \ldots, S$ be the points on which the problem initialization is done. A valuable initialization of $\boldsymbol{\mu}^p$ for a given ρ^p is to place N_j particles on each $\mathbf{s}_j$ point such that

$$N_j = E\left(\frac{\rho^p(\mathbf{s}_j)}{\sum_{j'=1}^{S} \rho^p(\mathbf{s}_{j'})} N \right), \tag{16}$$

where $E(.)$ is the integer part function. It can be noted that this initialization corresponds to first calculate the partition of the total mass present at $\mathbf{s}_j$ and then multiply it by the total number of particles N in order to get the partition of all the particles that should be initialized at that point.

At the end of this partition, $N - \sum_{j=1}^{S} N_j$ particles remain not placed. Those particles are distributed according to the rule of the strongest rest by positioning the unplaced particles on the s_j positions with the highest values of

$$\frac{\rho^p(\mathbf{s}_j)}{\sum_{j'=1}^{S} \rho^p(\mathbf{s}_{j'})} - \frac{N_j}{N}. \tag{17}$$

Fig. 8 illustrates an example of initialization in a 1D case. Following this initialization, all positions are perturbed with a small random displacement in order to avoid gradient problems since several particles would be initialized at exactly the same position. Indeed, this distribution is very localized and not natural at all.

Finally, the optimization problem (14) is solved to optimize the positions $\boldsymbol{\mu}^p$ to best represent ρ^p.

3.2.3. Gradient Descent

In order to solve the problem (14), the objective function to minimize is

$$J(\boldsymbol{\mu}^p) = \frac{1}{2} \left\| \rho^p - \sum_{n=1}^{N} G_{\boldsymbol{\mu}_n^p, \sigma} \right\|_2^2. \tag{18}$$

A gradient method with optimal step is used to find the optimal parameters $\boldsymbol{\mu}^p$. The advantages of this method are:

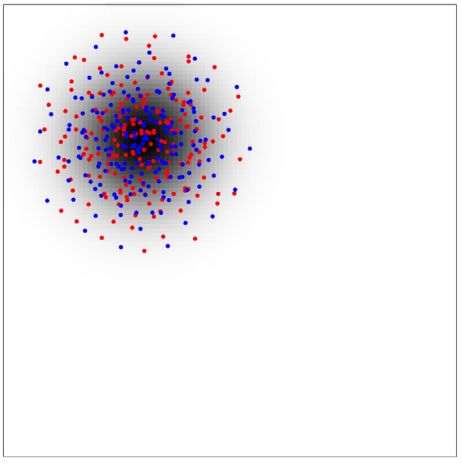

Fig. 9. Example of the Particle Decomposition of a 2D distribution presented in gray. The initialization of the particles by uniform quantization is illustrated in red and the converged solution in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- The objective function is expressed with continuous functions, thus, the gradient can be calculated with an analytical expression.
- The degree of complexity with respect to the number of particles N remains linear.
- With the optimal step scheme the computation of the Hessian matrix of the objective function is avoided and, thus, quadratic complexity.

First, the gradient is written in an analytical form for the nth particle of the pth distribution:

$$\frac{\partial J(\boldsymbol{\mu}^p)}{\partial \boldsymbol{\mu}_n^p} = -\sum_{i=1}^{D} \frac{\left(\boldsymbol{\mu}_n^p - \mathbf{x}_i\right)}{\sigma^2} G_{\boldsymbol{\mu}_n^p} \sigma^2 \left(\rho^p(\mathbf{x}_i) - \sum_{n=1}^{N} G_{\boldsymbol{\mu}_n^p,\sigma}(\mathbf{x}_i) \right). \tag{19}$$

Note that there are two components (along the two axes of the 2D space) since $\boldsymbol{\mu}_n^p \in \mathbb{R}^2$. Then, once the gradient is calculated, the optimal step ξ^* results from a single variable minimization problem:

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \left(J\left(\boldsymbol{\mu}_n^p - \xi \frac{\partial J(\boldsymbol{\mu}^p)}{\partial \boldsymbol{\mu}_n^p} \right) \right). \tag{20}$$

Next, the position of the N particles $\boldsymbol{\mu}^p$ is updated following the gradient descent scheme

$$\boldsymbol{\mu}_n^p \leftarrow \boldsymbol{\mu}_n^p - \xi^* \frac{\partial J(\boldsymbol{\mu}^p)}{\partial \boldsymbol{\mu}_n^p} \quad \forall n \in [\![N]\!]. \tag{21}$$

Finally, the gradient is recalculated and the process is repeated until the value $J(\boldsymbol{\mu}^p)$ decreases below a certain tolerance. This methodology is applied to each distribution leading to P particle decompositions $\boldsymbol{\mu}^p \in \mathbb{R}^{N\times 2}$, $p \in [\![P]\!]$. To illustrate this methodology, Fig. 9 shows in 2D both the initialization (red particles) and the converged solution (blue particles) for the gray continuous distribution.

3.3. P-dimensional matching

As it has been introduced previously, matching two discrete distributions (two N point clouds) remains a quite simple problem that can be solved using, for instance, linear programming. However, the problems becomes much more harder when one wants to match P points clouds of N particles each one. Indeed, it becomes a NP-complete minimization problem.

To this purpose, a Genetic Algorithm (GA) is implemented. GAs are inspired from the theory of evolution by natural selection: successive generations can differ from previous ones; children inherit characteristics of their parents which are combined and which can mutate leading to genetic variation across the generations; only the best individuals survive with time. Therefore, successive generations tend to perform better than previous ones [48].

GAs implement such a process. A population of individuals (i.e. possible solutions of the problem) are iteratively updated. Individuals are evaluated using a cost function. A new generation of population is created by selecting optimal individuals from the current generation. Some of them are admitted to the new generation unchanged. Others are subjected to genetic operators such as crossover or mutations. As individuals evolve across generations, they tend to be more optimal, reducing the cost function. At the end of the G iterations, the best individual of the last population is selected as the optimal solution of the problem.

In order to present this algorithm, let introduce M the number of individuals of the population Φ. The individual $\boldsymbol{\varphi}^m, m \in [\![M]\!]$ is composed by the P distributions of the problem where each distribution is modeled by the particles $\boldsymbol{\mu}^p \in \mathbb{R}^{N\times 2}$:

$$\boldsymbol{\varphi}^m = \left[\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^p, \ldots, \boldsymbol{\mu}^P\right] \in \mathbb{R}^{N\times(2\times P)}. \tag{22}$$

The difference from one individual to another is the order of the N particles of each of the P distributions. Therefore, each individual is characterized by a different cost $\mathbf{C}_P^m$.

The three process that are involved in one iteration are now presented. Note that γ is the fraction of the population which is replaced by crossover and ω the mutation rate in each iteration.

- First, the copy process is introduced. Since γ is the fraction of the new generation created by crossover, then $1-\gamma$ is the fraction that is copy from the old generation to the new one without modification. The $(1-\gamma)\times M$ individuals that are copied are selected by a tournament selection. Suppose that one wants to pick 20 individuals from 100. A small group of individuals (e.g. fewer than 10) is randomly selected (with uniform probability) from the initial 100 (with replacement). Then, the best one of the 10 individual is kept. This is repeated until 20 individuals are selected. It can be note that tournament selection is slower and more complicated than other selection techniques. However, it creates more diverse populations.
- Then, the crossover process is presented. During crossover, two individuals from the old generation, the parents, are combined to create an individual of the new generation, the child. Each parent is selected by a tournament selection. The child is created by merging a random number of distributions from the father and the complementary distributions (the ones that have not been selected from the father) from the mother. Note that $\gamma \times M$ individuals are created this way.
- Finally, the mutation process is explained. At this stage, the new generation is already composed by M individuals: $(1-\gamma)\times M$ from the copy process and $\gamma \times M$ from the crossover process. A fraction ω of this new generation, i.e. $\omega \times M$ individuals, is mutated. Those individuals are randomly selected with uniform probability. For each chosen individual, two of its distributions, p and p', are randomly selected are perfectly matched using linear programming in order to minimize $C_{p,p'}$. Mutation is a more random process than crossover and is implemented with $\omega < \gamma$. This random process might help to create essential features missing in the current generation.

3.4. Parametric model construction

Finally, once the N particles of each of the P distributions are "reasonably" matched, the parametric model can be built. This model relies on a Model Order Reduction approach, based on POD and sPGD, and yields to the interpolation of each particle between all the distributions. The construction of this real-time model follows the next two steps.

First, a POD [11] is conducted over the matrix $\boldsymbol{\mu}$ composed by the components of the N particles reshaped as a column (which reads along the rows) and this for the P distributions(which reads along the columns). The POD methodology is presented in Appendix A. Indeed,

$$\boldsymbol{\mu} = \left[\boldsymbol{\mu}^1(:)\cdots\boldsymbol{\mu}^p(:)\cdots\boldsymbol{\mu}^P(:)\right] \in \mathbb{R}^{2N\times P}, \tag{23}$$

Algorithm 1 Genetic Algorithm (M,G,γ,ω,$\mathbf{C}_p^m$)

Initialize generation 0:
  $g = 0$;
  Φ_g = a population of M randomly-generated individuals;
Evaluate Φ_g **:** Compute $\mathbf{C}_P^m \ \ \forall m \in [\![M]\!]$;
for $g \in [\![G]\!]$ **do**
  Create generation $g+1$ **:**
    1. Copy:
    Select $(1-\gamma) \times M$ individuals from Φ_g and insert hem into Φ_{g+1};
    2. Crossover:
    Select $\gamma \times M$ pairs from Φ_g, merge them and insert the merge into Φ_{g+1};
    3. Mutation:
    Select μ individuals from Φ_{g+1} and mutate them;
  Evaluate Φ_{g+1} **:** Compute $\mathbf{C}_P^m \ \ \forall m \in [\![M]\!]$;
end for
Return the best (minimum cost $\mathbf{C}_P^m$) individual φ^m of Φ_G;

where $\boldsymbol{\mu}^p(:) \in \mathbb{R}^{2N}$ is defined as $\boldsymbol{\mu}^p \in \mathbb{R}^{N\times 2}$ reshaped as a column. The POD decomposition of $\boldsymbol{\mu}$, where R modes have been retained, leads to

$$\underset{}{\boldsymbol{\mu}} \quad \simeq \quad \underset{}{\boldsymbol{U}_R} \quad \underset{}{\boldsymbol{\Sigma}_R \boldsymbol{V}_R^T}$$

$$\begin{bmatrix} | & & | & & | \\ \boldsymbol{\mu}^1(:) & \cdots & \boldsymbol{\mu}^p(:) & \cdots & \boldsymbol{\mu}^P(:) \\ | & & | & & | \end{bmatrix} \simeq \begin{bmatrix} | & & | & & | \\ \Upsilon^1 & \cdots & \Upsilon^r & \cdots & \Upsilon^R \\ | & & | & & | \end{bmatrix} \begin{bmatrix} — & \boldsymbol{\alpha}^1 & — \\ & \vdots & \\ — & \boldsymbol{\alpha}^r & — \\ & \vdots & \\ — & \boldsymbol{\alpha}^R & — \end{bmatrix}. \tag{24}$$

$$2N \times P \qquad 2N \times R \qquad R \times P$$

Then, after selecting the R POD modes containing enough information (based on a relative energy of the retained snapshots criterion), a sPGD regression [41,47] is conducted over the coefficients of the $\boldsymbol{\Sigma}_R \boldsymbol{V}_R^T$ matrix for each retained mode $r \in [\![R]\!]$ of the POD (the sPGD methodology is presented in Appendix B). Therefore, the separated variables regression technique leads to R rank-$\mathcal{M}$ separated approximations

$$\hat{\boldsymbol{\alpha}}^r = \sum_{m=1}^{\mathcal{M}} \prod_{q=1}^{Q} \zeta_m^q(\eta_q), \quad \forall r \in [\![R]\!], \tag{25}$$

where $\mathcal{M}$ denotes the number of PGD [9] modes and $\zeta_m^q : \mathcal{W}_q \to \mathbb{R}$ is the univariate function related to the qth dimension (or parameter) and mth mode. The parametric domain $\mathcal{W}$ (where our parametric problem is solved), expressed as $\mathcal{W} = \mathcal{W}_1 \times \ldots \times \mathcal{W}_Q$, is assumed to be separable and an equal number L of degrees of freedom along each dimension q is fixed. Therefore, the univariate function ζ_m^q can be linearly decomposed in a chosen base $N_l^q : \mathcal{W}_q \to \mathbb{R}$ with the unknown coefficients $a_{m,l}^q \in \mathbb{R}, \forall l \in [\![L]\!]$:

$$\zeta_m^q(\eta_q) = \sum_{l=1}^{L} N_l^q(\eta_q) a_{m,l}^q. \tag{26}$$

Finally, the R minimization problems for the unknown coefficients read

$$\min_{a_{m,l}^q} \left\| \boldsymbol{\alpha}^r - \hat{\boldsymbol{\alpha}}^r \right\|_2^2, \quad \forall r \in [\![R]\!]. \tag{27}$$

In order to solve those minimization problems, the sPGD approach is adopted relying on a greedy algorithm and an Alternating Direction Strategy, as explained in Appendix B. Once the R minimization problems are solved, the parametric model is built, leading to the end of the offline stage.

Then, one can explore the parametric space and get the interpolated simulation for any new set of parameters. Indeed, the parametric model outputs the location of the N particles $\hat{\boldsymbol{\mu}}$ for a set of parameters $\eta_q, q \in [\![Q]\!]$ in an online manner:

$$\hat{\boldsymbol{\mu}} = \sum_{r=1}^{R} \hat{\boldsymbol{\alpha}}^r(\eta_1, \ldots, \eta_Q)\, \Upsilon^r. \tag{28}$$

Then, the interpolated distribution $\hat{\rho}$ can be reconstructed thanks to the position $\hat{\boldsymbol{\mu}}$ of the N particles,

$$\hat{\rho} = \sum_{n=1}^{N} G_{\hat{\boldsymbol{\mu}}_n, \sigma}. \tag{29}$$

Finally, the total mass $\hat{\mathcal{I}}$ is also interpolated in order to recover $\hat{\psi}$.

3.5. Model overview

In this subsection, the methodology presented so far is summarized, as it is illustrated in Fig. 10. Given $P = 4$ high-fidelity simulations of a problem defined in the parametric space $\Omega\left(\eta_1, \ldots, \eta_Q\right)$ relying on $Q = 2$ parameters as an input, the aim is to build a parametric model combining Optimal Transport and sPGD regressions able to predict the solution for any set of parameters in the Ω.

The methodology proceeds in two stages. First, the offline training of the model using the high-fidelity simulations follows the next steps (colored in blue in Fig. 10):

1. **Pre-processing:** The high-fidelity simulations from the parametric space are normalized in order to have unitary integral. This simulations are used as training data points for the parametric model.
2. **SPH decomposition:** Each simulation is decomposed into a sum of N Gaussian particles of standard deviation σ and mass $1/N$. Both, the number of particles and the standard deviation of each particle, are fixed from now on for all the simulations. They are, indeed, hyperparameters of our methodology. Thus, P optimization problems (one for each simulation) need to be solved to locate the N Gaussian functions in order to minimize the error with respect to the original data. A gradient descent approach is chosen to this purpose.
3. **P-dimensional matching:** We have then the positions of the N particles for the P simulations. In order to emulate the Optimal Transport behavior, each Gaussian from each simulation is paired with one Gaussian of every other simulation. A Genetic Algorithm is implemented to solved the P-Dimensional Matching problem obtained leading to the positions of the N particles for the P simulations relatively ordered in such a way as to minimize the OT cost.
4. **Parametric model training:** Finally, in order to train the parametric model, a Model Order Reduction approach is conducted. A matrix of snapshots is built with the matched particles. Then, sPGD regressions are trained over the coefficients of the POD that has been applied to the matrix of snapshots. Note that as many regressions as retained modes of the POD are trained. The parametric model is thus built.

Then, the model can be used in real-time over the parametric space which represents the online stage (colored in red in Fig. 10):

1. For an unknown set of parameters (i.e. that do not belong to the training set), the trained regressions return the positions of the N Gaussian functions.
2. Then, these functions are summed in order to reconstruct the interpolated high-fidelity simulation following the Optimal Transport behavior. The parametric space can thus be explored.

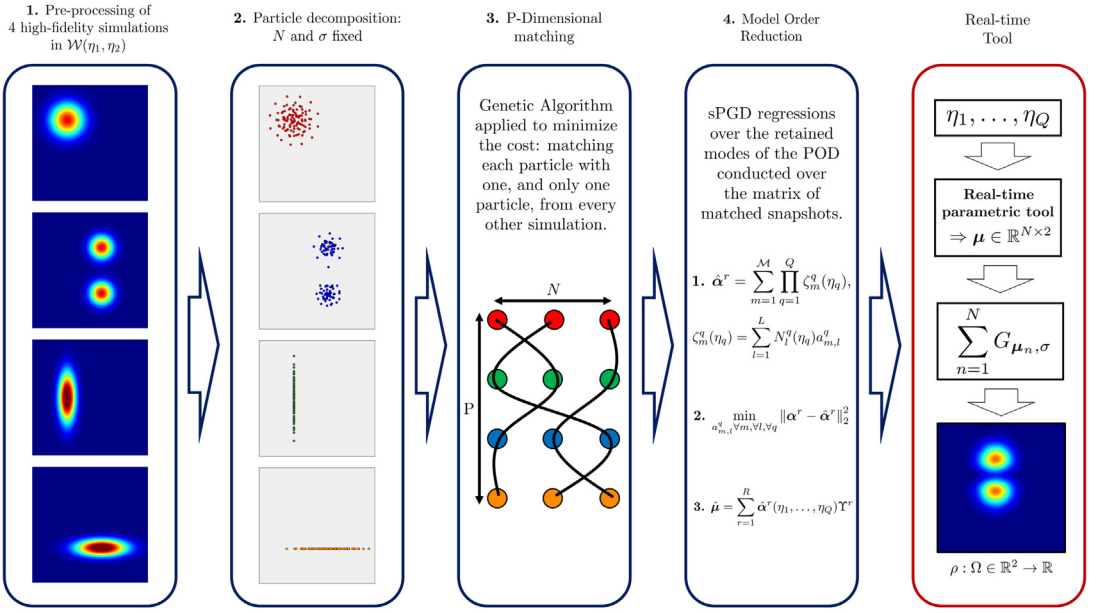

Fig. 10. Methodology summary diagram: the offline stage is colored in blue and the online stage in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Model improvements

4.1. Mass centering

On the one hand, in order to improve the quality of the regression, the mass of every distribution is centered in a pre-processing step (see Fig. 11(a)). Indeed, for a given distribution p, the means of the x and y positions of the particles are subtracted to the x and y coordinates of every particle. Thus, for the pth distribution:

$$\begin{aligned} \forall n \in [\![N]\!], \quad \mu_{n_x}^p &\leftarrow \mu_{n_x}^p - \mathbb{E}_n(\mu_{n_x}^p) \\ \mu_{n_y}^p &\leftarrow \mu_{n_y}^p - \mathbb{E}_n(\mu_{n_y}^p), \end{aligned} \tag{30}$$

where $\mathbb{E}_n(\mu_{n_x}^p)$ and $\mathbb{E}_n(\mu_{n_y}^p)$ are the means of the x and y positions of the particles of the distribution p respectively, that is,

$$\mathbb{E}_n\left(\mu_{n_d}^p\right) = \frac{1}{N}\sum_{n=1}^{N} \mu_{n_d}^p, \qquad d \in \{x, y\}. \tag{31}$$

Then, two supplementary sPGD regressions are implemented for these x and y means in the separable parametric space $\mathcal{W} = \mathcal{W}_1 \times \ldots \times \mathcal{W}_Q$. The separated variables regression technique leads to two rank-$\mathcal{M}$ approximations

$$\hat{\mathbb{E}}_n\left(\mu_{n_d}^p\right) = \sum_{m=1}^{\mathcal{M}} \prod_{q=1}^{Q} \epsilon_m^q(\eta_q), \qquad d \in \{x, y\}, \tag{32}$$

where $\mathcal{M}$ denotes the number of PGD modes and $\epsilon_m^q : \mathcal{W}_q \to \mathbb{R}$ is the univariate function related to the qth dimension (or parameter) and mth mode.

The POD in 3.4 is thus applied to a centered matrix of snapshots $\boldsymbol{\mu}$. Then, in order to reconstruct the solution for a set of parameters $\eta_q, q \in [\![Q]\!]$, first the sPGD regression developed in (28) is applied and then regressions of the means are calculated and added to the result.

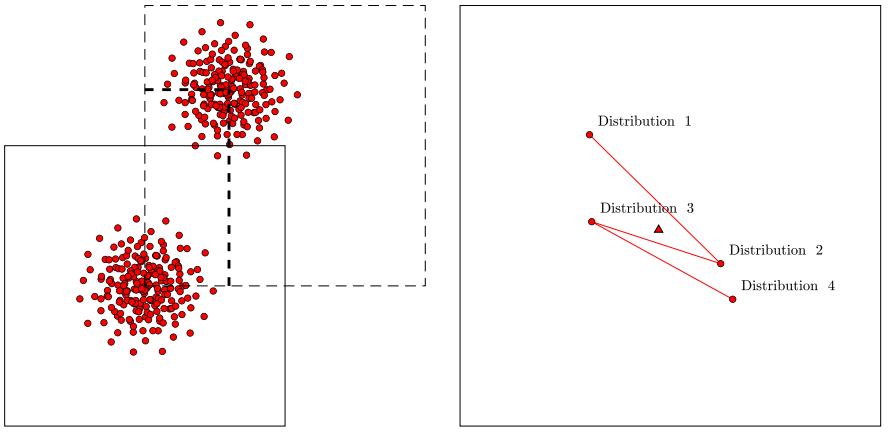

Fig. 11. (a) Example of mass centering for a given distribution. The dashed frame represents the initial distribution with the mean position illustrated with bold dashed black lines. The solid frame represents the mass centered distribution. (b) Example of particle centering for a given distribution. The red points represent the positions of a given particle in the four distributions linked by the red line which represents the trajectory. The red triangle represents the trajectory mean.

4.2. Particle centering

The quality of the parametric model is further improved by subtracting to each particle the mean of their x and y coordinates along its displacement over the distributions. Since the particles are matched between distributions, one can follow the trajectory of each particle along the different P distributions (see Fig. 11(b)). Indeed, the mean of this trajectory is subtracted to every particle respectively. Thus, for the pth distribution:

$$\begin{aligned} \forall n \in [\![N]\!], \qquad & \mu^p_{n_x} \leftarrow \mu^p_{n_x} - \mathbb{E}_p(\mu^p_{n_x}) \\ & \mu^p_{n_y} \leftarrow \mu^p_{n_y} - \mathbb{E}_p(\mu^p_{n_y}), \end{aligned} \tag{33}$$

where $\mathbb{E}_p(\mu^p_{n_x})$ and $\mathbb{E}_p(\mu^p_{n_y})$ are the means of the x and y positions of the particle n respectively while moving in the parametric space, that is,

$$\mathbb{E}_p\left(\mu^p_{n_d}\right) = \frac{1}{P}\sum_{p=1}^{P} \mu^p_{n_d}, \qquad d \in \{x, y\}. \tag{34}$$

It can be noted that this second mean is calculated over the already centered distributions. Therefore, in order to reconstruct the solution for a set of parameters $\eta_q, q \in [\![Q]\!]$, first the sPGD regression developed in (28) is applied, then the regressions of the means are calculated and added to the result and finally the mean of each particle is added.

4.3. Clustering

Finally, a last improvement is implemented. It can be observed that when particles are transported from one distribution to another, some particles follow a very similar displacement. Therefore, instead of applying a POD over the whole set of particles, the particles are clustered following similar behaviors and, then, an independent POD is applied to every cluster.

To this purpose, a `k-means` algorithm [27] is applied to the trajectories of the particles in order to cluster them by similar patterns. In order to determine the optimal number of clusters, the Calinski–Harabasz [6] criterion is used.

It can be noted that the resulting clustering presents a geometrical sense, as it can be seen in Fig. 12. Indeed, the cloud of points is partitioned by angular sectors since particles which are close tend to move following a similar pattern. In the example presented in Fig. 12, clustering the distributions into two groups seems the correct optimum since the source distributions divide into two smaller distributions.

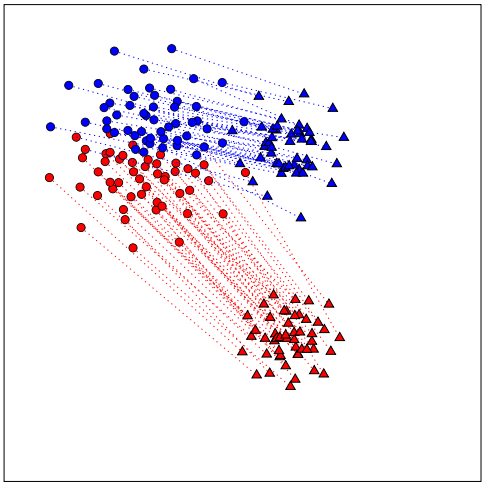

Fig. 12. Clustering of particles for two distributions. The first distribution is represented by circles and the second one by triangles. The first cluster is colored in red and the second one in blue. The matching between particles is represented by dashed lines and colored with respect to the clustering. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Results

The results obtained with the presented methodology are now presented and compared with other Optimal Transport and regression techniques. Four cases are here studied. First, two synthetic data cases are addressed, a 1D problem and a 2D one. Then, two physical cases are addressed concerning heat and mass transfers. The values of the different methodology hyper-parameters that have been chosen are gathered in Appendix C. It can be noted that an error evaluation methodology has been defined based on the L^2-Wasserstein metric [23].

5.1. Error evaluation

The error evaluation methodology is first of all presented. Given a subset of the parametric space $\mathcal{W}$ defining the testing set, the P_{test} high-fidelity solutions of this subset are compared with the real-time solutions obtained with the developed OT based approach. To this purpose, the L^2-Wasserstein metric $W(\rho_p^{test}, \hat{\rho}_p^{test})_2^2$ is calculated between the reference ρ_p^{test} and the interpolated $\hat{\rho}_p^{test}$ solutions where $p \in [\![P_{test}]\!]$. In order to calculate $W\left(\rho_p^{test}, \rho_p^{test}\right)$, a Linear Programming approach is adopted.

The Kantorovich Optimal Transport problem [23] reads as an infinite dimensional Linear Program. For the two distributions ρ_p^{test} and $\hat{\rho}_p^{test}$, with supports X and Y, the problem writes

$$W\left(\rho_p^{test}, \hat{\rho}_p^{test}\right)_2^2 = \min_{\pi \in \Pi(\rho_p^{test}, \hat{\rho}_p^{test})} \int_{X \times Y} c(x, y) \mathrm{d}\pi(x, y), \tag{35}$$

where Π is the set of transference plans and $c(x, y) : X \times Y \to \mathbb{R}$ the cost function. It has been proved [39] that the discretization of this problem into a finite dimensional Linear Program converge weakly to the solution of the infinite dimensional formulation as the mesh granularity tends to zero. Let define the discretized measures ρ_p^{test} and $\hat{\rho}_p^{test}$ as a weighted sum of Dirac masses whose weights correspond to value of the continuous measures at the corresponding mesh nodes $\mathbf{x}_i$ and $\mathbf{y}_i$, that is,

$$\rho_p^{test} = \sum_{i=1}^{D} \left(\rho_p^{test}\right)_i \delta_{\mathbf{x}_i} \quad \text{and} \quad \hat{\rho}_p^{test} = \sum_{i=1}^{D} \left(\hat{\rho}_p^{test}\right)_i \delta_{\mathbf{y}_i}. \tag{36}$$

The discrete cost function is given by

$$C_{i,j} = c(\mathbf{x}_i, \mathbf{y}_j) = \left\| \mathbf{x}_i - \mathbf{y}_j \right\|_2^2. \tag{37}$$

Thus, the discrete Kantorovich OT problem writes

$$W\left(\rho_p^{test}, \hat{\rho}_p^{test}\right)_2^2 = \min_{\pi \in \Pi(\rho_p^{test}, \hat{\rho}_p^{test})} \sum_{i,j} C_{i,j}\pi_{i,j}, \tag{38}$$

where $\pi_{i,j}$ describes the amount of mass transported from $\mathbf{x}_i$ towards $\mathbf{y}_j$ and the set of transference plans becomes

$$\Pi\left(\rho_p^{test}, \hat{\rho}_p^{test}\right) = \left\{ \pi = (\pi_{ij}) \left| \sum_j \pi_{ij} = \left(\rho_p^{test}\right)_i, \sum_i \pi_{ij} = \left(\hat{\rho}_p^{test}\right)_j \right. \right\} \tag{39}$$

In order to have a meaningful error estimator (given as a percentage), the L^2-Wasserstein metric $W(\rho_p^{test}, \hat{\rho}_p^{test})_2^2$ is divided by the greatest L^2-Wasserstein distance between the P high-fidelity training solutions of the parametric space $\mathcal{W}$.

5.2. Synthetic 1D problem

First, as it is illustrated in Fig. 13(a), a problem with two 1D-distributions is studied. The source distribution ρ_1 (in red) is a Gaussian function and the target distribution ρ_2 (in blue) is a combination of two Gaussian functions. The mathematical expressions of these two distributions are:

$$\begin{aligned} \rho_1(x) =& G_{0.2,0.05}(x) \\ \rho_2(x) =& \frac{1}{2}\left(G_{0.65,0.04}(x) + G_{0.8,0.05}(x)\right), \end{aligned} \tag{40}$$

where $G_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{-(x-\mu)^2}{2\sigma^2} \right\}$ and $x \in [0, 1]$. Note that both distributions have unit mass.

First, the two distributions are decomposed into N particles following the SPH rationale. The results of such a decomposition are presented in Fig. 13(b) where the original (in red/blue solid lines) and the reconstructed curves, i.e., $\bar{\rho}_1$ and $\bar{\rho}_2$, (in black crosses) are superimposed.

Then, Optimal Transport applies between the two distributions leading to $\hat{\rho}_t$. Since it is a 1D case, the optimal assignment is calculated by sorting the particles over the 1D axis and matching the particles progressively along the axis. The distribution can thus be translated by moving every particle from the source to the target positions. As it can be seen in Fig. 14(a), the source distribution is translated along the axis, its support is increased and its amplitude modified to adapt to the target distribution.

The optimal transport based solution is compared with the classic interpolation solution, presented in Fig. 14(b): $\hat{\rho}_t = (1-t)\rho_1 + t\rho_2$. Here, it can be observed how the source distribution does not move along the axis. Its amplitude decreases while a new distribution appears and increases at the target position.

5.3. Synthetic 2D problem

Next, a problem with four 2D-distributions is studied. Here, two parameters, η_1 and η_2, are introduced leading to the parametric space $\mathcal{W}(\eta_1, \eta_2)$. In this parametric space four distributions are created combining Gaussian functions. The mathematical expressions of the four distributions are:

$$\begin{aligned} \rho_1(x, y) =& G_{[0.3,0.7],[0.1,0.1]}(x, y) \\ \rho_2(x, y) =& \frac{1}{2}\left(G_{[0.6,0.3],[0.07,0.07]}(x, y) + G_{[0.6,0.7],[0.07,0.07]}(x, y)\right) \\ \rho_3(x, y) =& G_{[0.3,0.5],[0.05,0.15]}(x, y) \\ \rho_4(x, y) =& G_{[0.6,0.3],[0.15,0.05]}(x, y), \end{aligned} \tag{41}$$

where $G_{[\mu_1,\mu_2],[\sigma_1,\sigma_2]}(x, y) = \frac{1}{\sigma_1\sigma_2 \times 2\pi} \exp\left\{ -\frac{1}{2}\left(\frac{(x-\mu_1)^2}{\sigma_1^2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right) \right\}$ and $(x, y) \in [0, 1]^2$.

Note that all distributions have unit mass. The four unit mass distributions ρ are presented in Fig. 15(a). Then, each distribution is decomposed following our Particle Decomposition Methodology. Fig. 15(b) shows the four recomposed distributions $\bar{\rho}$. It can be noted that the result is very close to the original. Indeed, the SPH approach is able to catch all the features of the distributions.

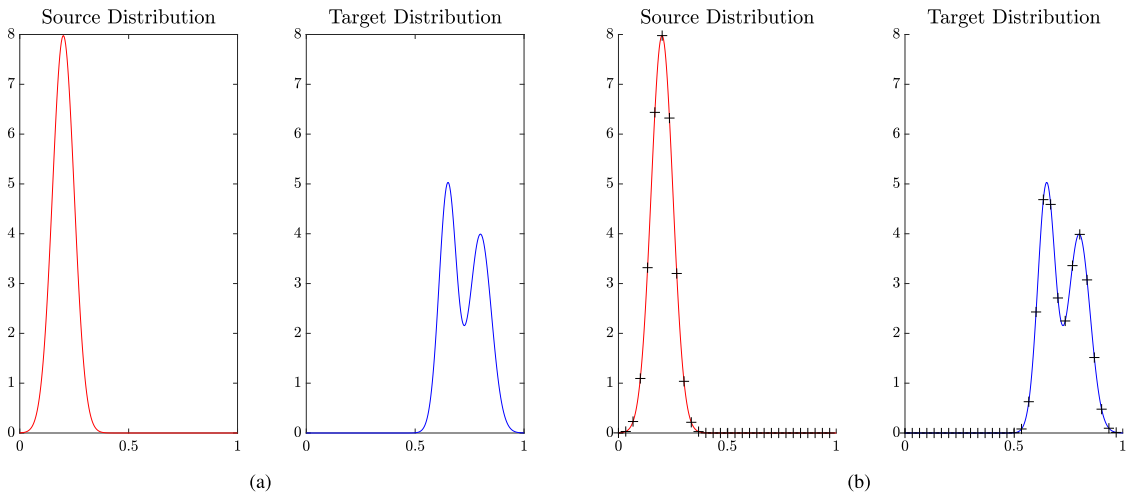

Fig. 13. (a) Source ρ_1 (in red) and target ρ_2 (in blue) distributions. (b) Original (in red/blue solid lines) and reconstructed after SPH decomposition (in black crosses) distributions superimposed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

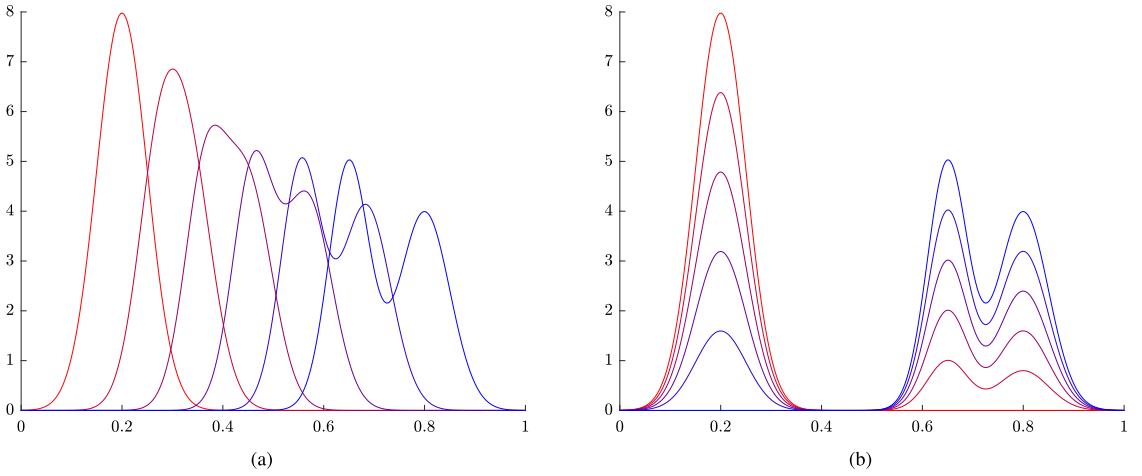

Fig. 14. Interpolated distribution $\hat{\rho}_t$ for different t values based on the Optimal Transport approach (a) and following the classic interpolation method $\hat{\rho}_t = (1-t)\rho_1 + t\rho_2$ (b).

First, a two 2D-distributions problem is presented. Here, the transport applies from $\bar{\rho}_1$ to $\bar{\rho}_2$. Since there are only two images, the linear assignment problem can be solved using the MATLAB function `matchpairs` [13]. Each particle of $\bar{\rho}_1$ is thus matched with a particle of $\bar{\rho}_2$ in such a way that the cost $C_{\bar{\rho}_1,\bar{\rho}_2}$ is minimized. Then, the Optimal Transport between distributions can be achieved by fractionally displacing each particle of $\bar{\rho}_1$ along the segment that connects it with its perfect match in $\bar{\rho}_2$. The solution is presented in Fig. 16. It can be observed how the single geometry at $t = 0$ divides progressively into two distinct geometries at $t = 1$. The mass from $\bar{\rho}_1$ is divided in two smaller masses and transported to the other side of Ω to represent $\bar{\rho}_2$.

Therefore, this solution represents the Optimal Transport from $\bar{\rho}_1$ to $\bar{\rho}_2$ and is very different from the solution obtained by a classical linear interpolation between both distributions such that $\hat{\rho}_t = (1-t)\rho_1 + t\rho_2$. Indeed, as it can be seen in Fig. 17 the distribution ρ_1 progressively disappears while the distribution ρ_2 appears in a analogous way to the interpolation presented in Fig. 14(b). This interpolation leads to a non-physical result.

Finally, the Optimal Transport based interpolation obtained by applying our proposal is compared with the solution calculated using a Barycenter approach following the methodology described in [50] and presented in

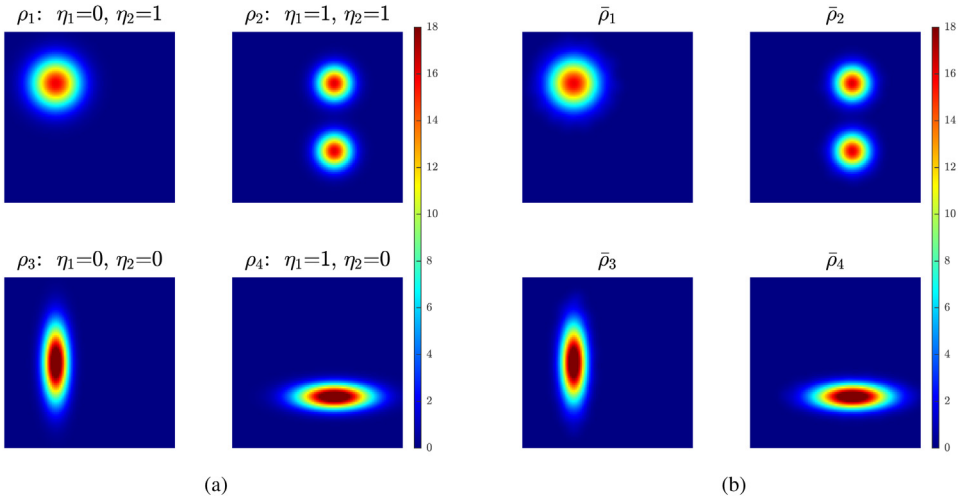

Fig. 15. (a) Four distributions ρ_1, ρ_2, ρ_3 and ρ_4 of the parametric space $\mathcal{W}(\eta_1, \eta_2)$. (b) Four recomposed distributions $\bar{\rho}$ after Particle Decomposition.

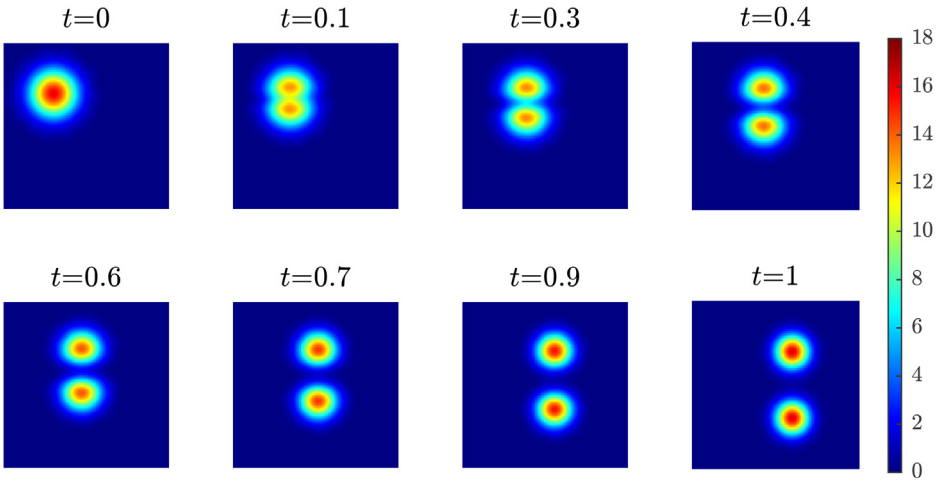

Fig. 16. Interpolated distribution $\hat{\rho}_t$ from ρ_1 to ρ_2 for different t values based on the Optimal Transport approach.

Fig. 18. It can be observed that the Barycenter Optimal Transport solution between the two distributions follows the same behavior that our methodology solution. This validates and gives confidence on the approach here proposed.

Then, the four 2D-distributions problem is studied. The transport applies simultaneously between $\bar{\rho}_1$, $\bar{\rho}_2$, $\bar{\rho}_3$ and $\bar{\rho}_4$ where each distribution has an associated set of parameters η_1 and η_2. Once the particle decomposition is done, the optimal matching is conducted using the Genetic Algorithm and the Parametric Model is built. Then, the distribution $\hat{\rho}(\eta_1, \eta_2)$ can be interpolated for a non-simulated set of parameters η_1 and η_2.

The real-time particularization of the solution is presented in Fig. 19. It can be observed that the solutions obtained correspond to the Optimal Transport between the four training distributions. Indeed, one can evaluate the solution of the problem for any unknown set of parameters of the parametric space $\mathcal{W}(\eta_1, \eta_2)$ in a OT and online manner.

Again, the Optimal Transport based solutions obtained with our approach are compared with the solution calculated using a Barycenter approach [50], presented in Fig. 20. It can be noticed that the OT based on barycenters agrees with the one based on our methodology.

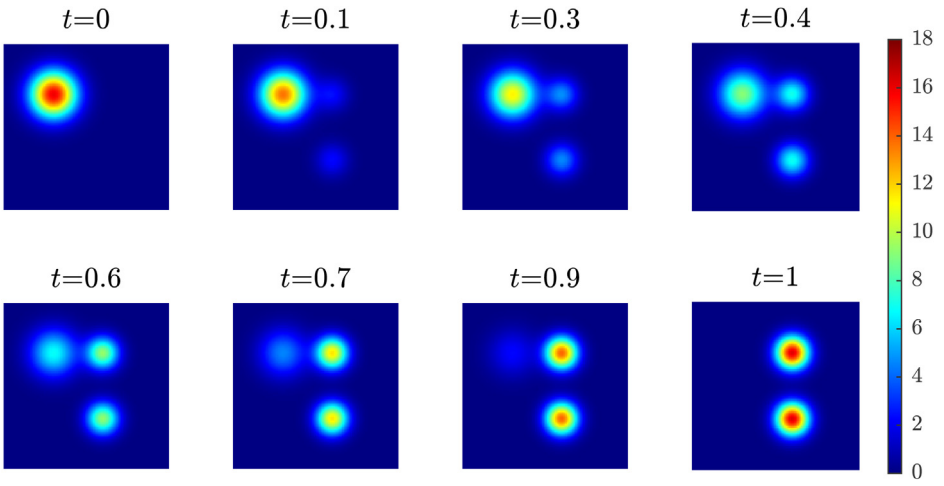

Fig. 17. Interpolated distribution $\hat{\rho}_t$ from ρ_1 to ρ_2 for different t values following the classic interpolation method $\hat{\rho}_t = (1-t)\rho_1 + t\rho_2$ with the associated spurious effects.

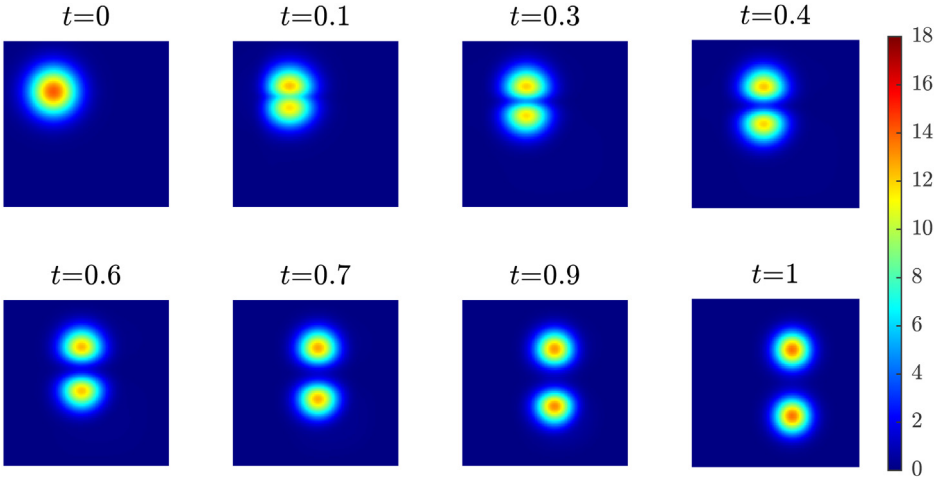

Fig. 18. Interpolated distribution $\hat{\rho}_t$ from ρ_1 to ρ_2 for different t values following the Barycenter approach described in [50].

5.4. Heat transfer problem

A physical case study is now addressed. The heat equation is solved (using a finite elements methodology) in a 2D domain Ω with an-isotropic thermal conductivity where the initial condition is a Gaussian heat source that diffuses in time. A homogeneous Newman condition is imposed on the domain boundary. Thus, the problem writes

$$\begin{cases} k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} = \rho C_P \frac{\partial T}{\partial t} & \text{in } \Omega \times [0, T_f), \\ \nabla T \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \\ T(x, y, t = 0) = \frac{1}{\sigma^2 2\pi} \exp^{-\frac{(x-s_x)^2 + (y-s_y)^2}{2\sigma^2}} & \text{in } \Omega, \end{cases} \tag{42}$$

where C_P is the specific heat, ρ is the density, k_x and k_y the thermal conductivity along the x and y directions respectively, T_f the final time and $\mathbf{n}$ the outward normal from Ω. Therefore, the parametric space is defined by the parameters s_x, s_y, t, k_x and k_y: $\mathcal{W}(s_x, s_y, t, k_x, k_y) \in \mathbb{R}^5$.

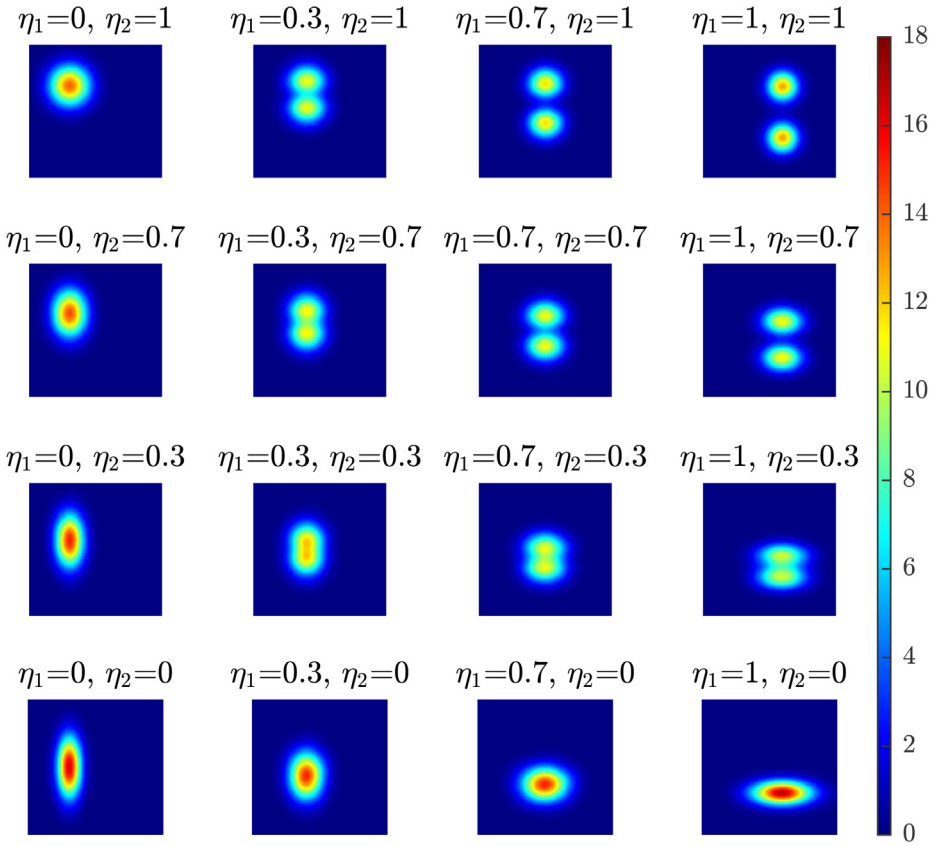

Fig. 19. Interpolated distribution $\hat{\rho}(\eta_1, \eta_2)$ in the parametric space $\mathcal{W}(\eta_1, \eta_2)$ computed from the four solutions in the four corners: top-left, top-right, bottom-left and bottom-right.

Table 1
Error estimator of the 4 testing set simulations for the heat transfer problem.

Solution	ρ_1^{test}	ρ_2^{test}	ρ_3^{test}	ρ_4^{test}
Error estimator (%)	3.67	5.52	5.66	5.01

Once the parametric model is trained, the real-time solution is compared with the high-fidelity solution for four points in the parametric space $\mathcal{W}$ defining the testing set. The high-fidelity simulations are presented in Fig. 21(a) and the interpolated solutions in Fig. 21(b). The error estimator values are presented in Table 1. It can be noticed the excellent agreement between the OT based interpolation and the reference solutions.

5.5. Mass transfer problem

Finally, a fluid dynamics problem is addressed. The mass transfer equation is solved (using a finite elements methodology) in a 2D isotropic domain Ω where the initial condition is a Gaussian concentration source that diffuses in time while convected by a flow. An homogeneous Newman condition is imposed on the domain boundary. Thus, the problem writes

$$\begin{cases} \frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{v}C) = D \quad \Delta C & \text{in } \Omega \times [0, T), \\ \nabla C \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \\ C(x, y, t = 0) = \frac{1}{\sigma^2 2\pi} \exp^{-\frac{(x-s_x)^2 + \left(y-s_y\right)^2}{2\sigma^2}} & \text{in } \Omega, \end{cases} \tag{43}$$

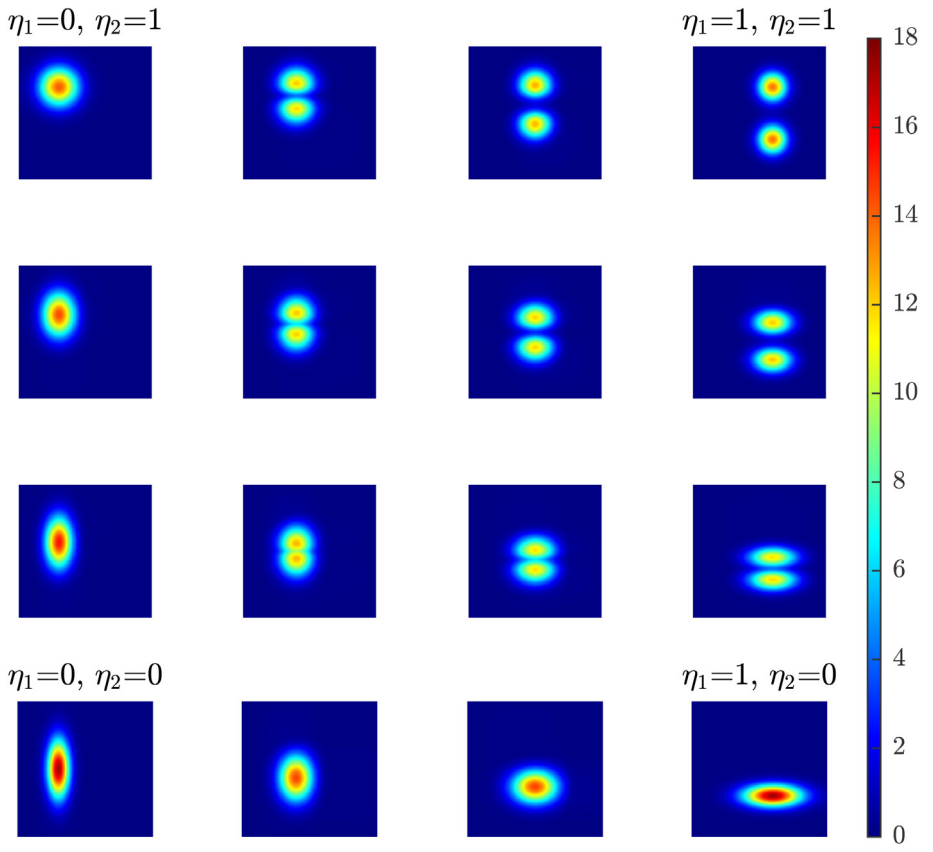

Fig. 20. Interpolated distribution $\hat{\rho}(\eta_1, \eta_2)$ in the parametric space $\mathcal{W}(\eta_1, \eta_2)$ based on the Barycenter approach [50].

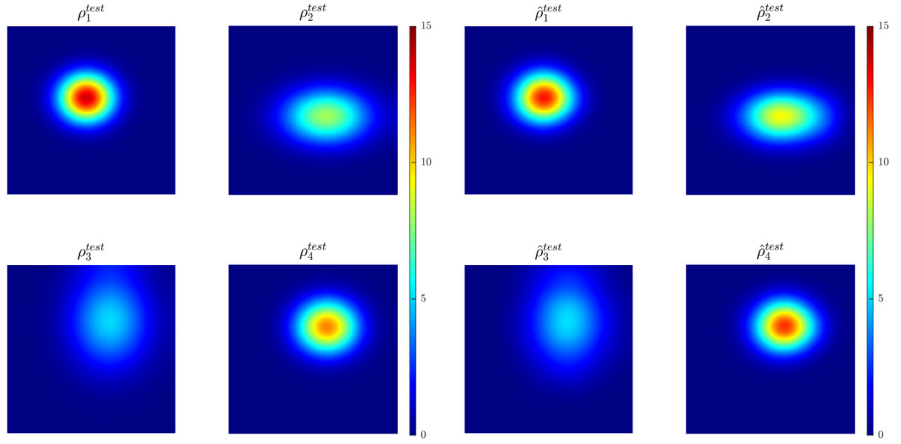

Fig. 21. (a) Four high-fidelity solutions in the parametric space $\mathcal{W}$ corresponding to the testing set considered as reference solutions. (b) Solutions interpolated at the testing set points.

where C is the concentration of the species of interest, D is the diffusivity, T the final time, s_y the y coordinate of the initial concentration (fixed for all the simulations at $s_y = 0.2$), $\mathbf{n}$ the outward normal from Ω and $\mathbf{v}$ the velocity field defined as $\mathbf{v}(x, y) = (0, 2(-x^2 + x) + 0.5y)$, as it is presented in Fig. 22. Therefore, the parametric space is defined by the parameters s_x, t, D and σ: $\mathcal{W}(s_x, t, D, \sigma) \in \mathbb{R}^4$.

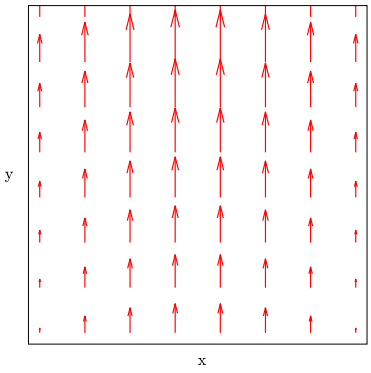

Fig. 22. Velocity field of the mass transfer problem: $\mathbf{v}(x, y) = (0, 2(-x^2 + x) + 0.5y)$.

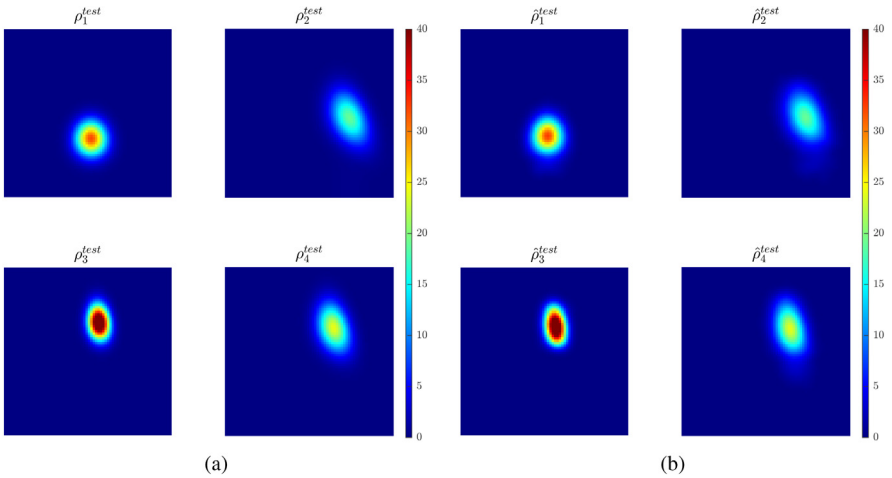

Fig. 23. (a) Four high-fidelity solutions in the parametric space $\mathcal{W}$ corresponding to the testing set considered as reference solutions. (b) Solutions interpolated at the testing set points.

Table 2
Error estimator of the 4 testing set simulations for the mass transfer problem.

Solution	ρ_1^{test}	ρ_2^{test}	ρ_3^{test}	ρ_4^{test}
Error estimator (%)	5.04	4.00	6.09	5.07

Again, the real-time solution is compared with the high-fidelity solution for four points in the parametric space $\mathcal{W}$ corresponding to the testing set. The results are presented in Fig. 23 while Table 2 shows error estimator values.

6. Conclusion

Although techniques using decomposition into particles and mass transport have already been implemented to interpolate between two distributions, they do not allow to interpolate between several distributions from a parametric space in an online manner and under the stringent real-time constraint. Combining the optimal deterministic assignment simplification of the Optimal Transport Monge problem with the sPGD Model Order Reduction technique, our approach leads to a real-time parametric model based on OT interpolation point of view. The key step of our approach is the SPH decomposition of the interpolated simulations which allows displacement interpolation in the parametric space. The main drawback of our approach is the matching of the particles of the different distributions which consists on the resolution of a P-dimensional deterministic assignment problem (a

known NP-complete problem). The work in progress aims at extending our approach in a given geometric structure, such as a discrete surface, and between distributions defined on different domains.

Appendix A. Pod: Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition is a numerical method belonging to Model Order Reduction algorithms and associated with the field of machine learning. The POD enables a reduction in the complexity of data by obtaining low-dimensional approximate descriptions of high-dimensional processes [7]. In data analysis, the POD is often applied to extract 'mode shapes' or basis functions, from experimental data or high-fidelity simulations of high-dimensional systems. The POD is assimilated with the Principal Component Analysis from Pearson in the field of statistics, or the Singular Value Decomposition in linear algebra because it refers to eigenvalues and eigenvectors of a physical field.

The idea behind the POD is to approximate a function $z(x,t)$ over a domain of interest as a finite sum in the variables-separated form:

$$z(x,t) \approx \sum_{k=1}^{M} a_k(t)\phi_k(x), \tag{44}$$

where the functions $\phi_k(x)$ are orthonormal ordered functions.

In a finite-dimensional case, in which we are interested here, let us first consider a system where m state variables are measured at N instants of time. This measurements are arranged in a $N \times m$ matrix A such that the element A_{ij} of the matrix is the measurement in the jth location at the ith time instant.

Then, a Singular Value Decomposition (SVD) of the matrix A is computed. The goal of the SVD [14] is to decompose the rectangular matrix A as the product of three matrices: U, Σ and V^T

$$A = U\Sigma V^T, \tag{45}$$

where U is an $N \times N$ orthogonal matrix, V an $m \times m$ orthogonal matrix and Σ a matrix such that

$$\Sigma = \begin{pmatrix} \Sigma' & 0 \\ 0 & 0 \end{pmatrix}, \tag{46}$$

where Σ' is a diagonal $s \times s$ matrix, where $s = min(N,m)$.

Columns of U are the space principal components of A, also known as modes. Likewise, columns of V are the time principal components. The diagonal elements of Σ' are $s = min(N,m)$ non-negative numbers σ_i arranged in decreasing order, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$, which are called singular values of A.

Finally, let $Q = U\Sigma$ a matrix of dimensions $N \times m$. Then,

$$A = QV^T = \sum_{k}^{m} q_k v_k^T \tag{47}$$

where q_k and v_k are the kth column of Q and V respectively. Therefore, (47) is the discrete form of (44). The function $a_k(t)$ is represented by the column matrix q_k and the function $\phi_k(x)$ by the row matrix v_k^T.

With this decomposition, one can easily approximate the ith column of A, a_i, by projecting it on the principal components in order to compute its r reduced coordinates $(\alpha_{i1}, \ldots, \alpha_{ir})$

$$[\alpha_{i1}, \ldots, \alpha_{ir}] = a_i^T \tilde{U}, \tag{48}$$

where $\tilde{U}$ contains the first r columns of U. Then, the columns of A can be approximated in the reduced basis composed by the columns $\tilde{u}_j$ of the reduced matrix $\tilde{U}$:

$$a_i \approx \sum_{j=1}^{r} \alpha_{ij}\tilde{u}_j. \tag{49}$$

Therefore, one can approximate any rectangular matrix A by projecting its columns in a reduced basis composed by r dimensions instead of N, where $N \gg r$. Then, the question arises of how many r modes to take. To this purpose, the "kept" information is observed. An indication of this information is given by

$$I_{POD}(r) = \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{s} \sigma_i^2}, \tag{50}$$

which represents the relative energy of the snapshots captured by the r first POD basis vectors. The number of modes r is thus chosen as the minimum integer for which $I_{POD}(r)$ is superior to a certain criterion. For additional details about the POD, the reader can refer to [17,32,56].

Appendix B. sPGD: sparse Proper Generalized Decomposition

Model Order Reduction (MOR) techniques express the solution of a given problem into a reduced basis with strong physical or mathematical content. These bases can be extracted from solutions of the problem obtained offline. This can be done, for instance, by applying the Proper Orthogonal Decomposition. Despite of the loss of generality, the use of a reduced basis enables impressive computing time savings with an considerable accuracy. To improve generality while ensuring accuracy, one can construct the reduced basis and solve the problem simultaneously, following the Proper Generalized Decomposition (PGD) [11]. However, this option is very intrusive.

Therefore, in order to keep non-intrusiveness, parametric models are built on high-fidelity solutions performed offline. Despite of its non-intrusiveness, these procedures are very expensive from the computing time viewpoint. Within the PGD rationale, the sparse PGD (sPGD) reduces the amount of required data by considering a sparse sampling [41]. The sPGD can be viewed as a nonlinear regression that makes use of the separation of variables. Thus, it can be used in multi-parametric settings.

Regressions are a key ingredient in order to build models. Thus, they are widely used in artificial intelligence, in particular in supervised machine learning [4,19,53], and in the development of hybrid twins [10,38,46]. However, the high-dimensional low-data limit context of the sPGD leads to nonlinear behaviors issues. Indeed, the number of data points needed for an accurate model increase exponentially with the number of degrees of freedom [24]. Nevertheless, it is important to keep models as simple as possible [4,53] and, therefore, impose sparsity in the regression [4,18,21,22]. In order to construct parsimonious models, able to deal with sparse data, regularization is performed.

Let us now consider an unknown function that we aim to approximate:

$$f\left(\eta_1, \ldots, \eta_Q\right) : \Omega \subset \mathbb{R}^Q \rightarrow \mathbb{R}, \tag{51}$$

which depends on Q different variables $\eta_q, q = 1, \ldots, Q$, considered as dimensions of the parametric space.

The sPGD tries, as standard PGD procedures, to approximate the objective function by a sum of products of one-dimensional functions. Each one of these function represents one dimension and each sum is known as a mode. This separated approximate expression reads:

$$f\left(\eta_1, \ldots, \eta_Q\right) \approx \hat{f}\left(\eta_1, \ldots, \eta_Q\right) = \sum_{m=1}^{\mathcal{M}} \prod_{q=1}^{Q} \zeta_m^q(\eta_q), \tag{52}$$

where $\hat{f}$ is the approximation, $\mathcal{M}$ denotes the number of PGD modes and ζ_m^q are the one-dimensional functions of the mode m and the dimension q.

In the sPGD context, the $\zeta_m^q, m = 1, \ldots, M$ and $q = 1, \ldots, Q$ functions are expressed from standard approximation functions:

$$\zeta_m^q(\eta_q) = \sum_{l=1}^{L} N_{m,l}^q(\eta_q) a_{m,l}^q = \left(\vec{N}_m^q\right)^T \vec{a}_m^q, \tag{53}$$

where L is the number of degrees of freedom of the chosen approximation. Moreover, $\vec{N}_m^q$ is a column vector composed by the basis functions (chosen by the user, in our case we have chosen Chebyshev polynomials) and $\vec{a}_m^q$ is a column vector that contains the coefficients for the qth dimension and the mth mode. The choice of the set of basis functions is here important and needs to suit the problem studied.

Finally, as for any other regression, the aim is to minimize the distance (here related to the L2-norm) to the measured function finding the best $\hat{f}$ approximation. This leads to the following minimization problem:

$$\hat{f} = \underset{f^*}{\operatorname{argmin}} \sum_{i=1}^{n_t} \left\| f\left(\vec{\eta}_i\right) - f^*\left(\vec{\eta}_i\right) \right\|_2^2, \tag{54}$$

where f^* is expressed following the separated form (52), n_t is the number of training points and $\vec{\eta}_i$ are the vectors containing the parameters of the corresponding training point.

A greedy algorithm is employed in order to determine the coefficients of each one-dimensional function for each mode, such that once the approximation up to order $\mathcal{M}-1$ is known, the $\mathcal{M}$th order is solved:

$$\hat{f}_{\mathcal{M}}\left(\eta_1,\ldots,\eta_Q\right)=\sum_{m=1}^{\mathcal{M}-1}\prod_{q=1}^{Q}\zeta_m^q(\eta_q)+\prod_{q=1}^{Q}\zeta_{\mathcal{M}}^q(\eta_q), \tag{55}$$

where the subscript $\mathcal{M}$ highlights the rank of the sought function. To solve the resulting non-linear problem of the $\mathcal{M}$th order, an iterative scheme based on an Alternating Direction Strategy is used.

For the ease of the explanation and without loss of generality, let us continue by supposing that the unknown function depends on $Q=2$ different variables: x and y. Therefore, the objective function is

$$f(x,y):\Omega\subset\mathbb{R}^2\rightarrow\mathbb{R}, \tag{56}$$

which can be written, in the separated form, as

$$\hat{f}_{\mathcal{M}}(x_i,y_i)=\sum_{m=1}^{\mathcal{M}}\left(\left(\vec{N}_m^x(x_i)\right)^T\vec{a}_m^x\cdot\left(\vec{N}_m^y(y_i)\right)^T\vec{a}_m^y\right), \tag{57}$$

where $\vec{N}_m^x(x_i)$ and $\vec{N}_m^y(y_i)$ are the vectors containing the evaluation of the interpolation basis functions of the mth mode at x_i and y_i respectively. Therefore, the optimization problem writes:

$$\hat{f}=\underset{f^*}{\operatorname{argmin}}\sum_{i=1}^{n_t}\left\|\hat{f}_{\mathcal{M}}(x_i,y_i)-f^*(x_i,y_i)\right\|_2^2. \tag{58}$$

Then, the Alternating Direction Strategy computes $\vec{a}_{\mathcal{M}}^{x,k}$ from $\vec{a}_{\mathcal{M}}^{y,k-1}$ and $\vec{a}_{\mathcal{M}}^{y,k}$ from $\vec{a}_{\mathcal{M}}^{x,k}$ where $\vec{a}_{\mathcal{M}}^{x,k}$ indicates the values of $\vec{a}_{\mathcal{M}}^{x}$ at the kth iteration of the nonlinear iteration algorithm. A tolerance is specified by the user such that the iterations proceed until it is not reached.

Finally, the system to solve can be written as:

$$\begin{aligned}\mathbf{M}_x\cdot\vec{a}_{\mathcal{M}}^x&=\vec{r},\\ \mathbf{M}_y\cdot\vec{a}_{\mathcal{M}}^y&=\vec{r},\end{aligned} \tag{59}$$

where:

$$\begin{aligned}\vec{r}&=\begin{pmatrix}f(x_1,y_1)-\hat{f}_{\mathcal{M}-1}(x_1,y1)\\ \vdots\\ f(x_{n_t},y_{n_t})-\hat{f}_{\mathcal{M}-1}(x_{n_t},yn_t)\end{pmatrix},\\ \mathbf{M}_x&=\begin{pmatrix}\left(\vec{N}_{\mathcal{M}}^y(y_1)\right)^T\vec{a}_{\mathcal{M}}^y\cdot\left(\vec{N}_{\mathcal{M}}^x(x_1)\right)^T\\ \vdots\\ \left(\vec{N}_{\mathcal{M}}^y(y_{n_t})\right)^T\vec{a}_{\mathcal{M}}^y\cdot\left(\vec{N}_{\mathcal{M}}^x(x_{n_t})\right)^T\end{pmatrix},\\ \mathbf{M}_y&=\begin{pmatrix}\left(\vec{N}_{\mathcal{M}}^x(x_1)\right)^T\vec{a}_{\mathcal{M}}^x\cdot\left(\vec{N}_{\mathcal{M}}^y(y_1)\right)^T\\ \vdots\\ \left(\vec{N}_{\mathcal{M}}^x(x_{n_t})\right)^T\vec{a}_{\mathcal{M}}^x\cdot\left(\vec{N}_{\mathcal{M}}^y(y_{n_t})\right)^T\end{pmatrix}.\end{aligned} \tag{60}$$

To conclude, the sPGD regression faces classical machine learning problems of regressions: the approximation needs to fit not only the training set but also the test one. This second goal is particularly difficult when confronted with sparse date of a high-dimensional problem. In this low-data limit, the risk of overfitting increases. To solve this problem, improved sPGD regressions are proposed implementing L1 and L2 regularization [47].

Appendix C. Hyper-parameters values for the numerical results

See Table 3.

Table 3
Hyper-parameters values used in the problems presented for the numerical results.

Problem	Parameter						
	N	σ	Clusters	sPGD for POD		sPGD for mass centering	
				L	$\mathcal{M}$	L	$\mathcal{M}$
Synthetic 1D problem	15	0.0412	Not applied	Not applied	Not applied	Not applied	Not applied
Synthetic 2D problem	100	0.0503	2	2	1	2	2
Heat transfer problem	100	0.0797	3	2	1	2	4
Mass transfer problem	300	0.0318	2	2	1	2	4

References

[1] M. Agueh, G. Carlier, Barycenters in the Wasserstein space, SIAM J. Math. Anal. 43 (2) (2011).
[2] J.D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem, Numer. Math. 84 (3) (2000) 375–393.
[3] N. Bonneel, M. van de Panne, S. Paris, W. Heidrich, Displacement interpolation using Lagrangian mass transport, ACM Trans. Graph. 30 (6) (2011) 1–12.
[4] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937.
[5] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, Society for Industrial and App. Math, 2009.
[6] T. Calinski, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. 3 (1) (1974) 1–27.
[7] A. Chatterjee, An introduction to the proper orthogonal decomposition, Curr. Sci. 78 (7) (2000) 808–817.
[8] B. Chen, G. Becigneul, O.E. Ganea, R. Barzilay, T. Jaakkola, Optimal transport graph neural networks, 2020, arXiv preprint arXiv:2006.04804.
[9] 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.: State-of-the-Art Rev. 17 (4) (2010) 327–350.
[10] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J.L. Duval, F. El Khaldi, Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data, Arch. Comput. Methods Eng. 27 (1) (2020) 105–134.
[11] F. Chinesta, A. Huerta, G. Rozza, K. Willcox, Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd, 2015.
[12] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in: Adv. in Neural Information Proc. Sys., 2013.
[13] I.S. Duff, J. Koster, On algorithms for permuting large entries to the diagonal of a sparse matrix, SIAM J. Matrix Anal. Appl. 22 (4) (2000) 973–996.
[14] G.H. Golub, C. Reinsch, Singular value decomposition and least squares solutions, Linear Algebra 656 (1971) 134–151.
[15] S. Haker, L. Zhu, A. Tannenbaum, S. Angenent, Optimal mass transport for registration and warping, Int. J. Comput. Vis. 60 (3).
[16] C. Han, B. Sun, R. Ramamoorthi, E. Grinspun, Frequency domain normal map filtering, ACM Trans. Graph. (SIGGRAPH'07) 26 (3) (2007) 1–12.
[17] E.R. Henry, J. Hofrichter, Singular value decomposition: Application to analysis of experimental data, Methods Enzymol. 210 (1992) 129–192.
[18] Q. Hernandez, A. Badias, D. Gonzalez, F. Chinesta, E. Cueto, Deep learning of thermodynamics-aware reduced-order models from data, 2020, arXiv preprint arXiv:2007.03758.
[19] Q. Hernandez, A.D. Gonzalez, F. Chinesta, E. Cueto, Structure-preserving neural networks, J. Comput. Phys. 426 (109950) (2021).
[20] F.S. Hillier, G.J. Lieberman, Introduction to Operations Research, McGraw-Hill, 1990.
[21] R. Ibanez, E. Abisset-Chavanne, A. Ammar, D. Gonzalez, E. Cueto, A. Huerta, J.L. Duval, F. Chinesta, A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition, Complexity (2018).
[22] R. Ibanez, E. Abisset-Chavanne, E. Cueto, A. Ammar, J.L. Duval, F. Chinesta, Some applications of compressed sensing in computational mechanics: model order reduction, manifold learning, data-driven applications and nonlinear dimensionality reduction, Comput. Mech. 64 (5) (2019) 1259–1271.
[23] L. Kantorovich, On the transfer of masses (in russian), Dokl. Akad. Nauk 37 (2) (1942) 227–229.
[24] R.B. Laughlin, D. Pines, The theory of everything, Proc. Natl. Acad. Sci. USA 97 (1) (2000) 28.
[25] B. Lévy, E.L. Schwindt, Notions of optimal transport theory and how to implement them on a computer, Comput. Graph. 72 (2018) 135–148.
[26] S. Lind, B. Rogers, P. Stansby, Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling, Proc. R. Soc. A 476 (2020).
[27] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (2) (1982) 129–137.
[28] Y. Lu, J. Lu, A universal approximation theorem of deep neural networks for expressing probability distributions, 2020, arXiv preprint arXiv:2004.08867.
[29] Y. Makihara, Y. Yagi, Earth mover's morphing: topology-free shape morphing using cluster-based emd flows, in: Proc. 10th Asian Conference on Computer Vision, 2011, pp. 202–215.
[30] A.V. Makkuva, A. Taghvaei, J.D. Lee, S. Oh, Optimal transport mapping via input convex neural networks, 2019, arXiv preprint arXiv:1908.10962.

[31] O.C. Martin, M. Mézard, O. Rivoire, Random multi-index matching problems, J. Stat. Mech. Theory Exp. 2005 (9) (2005).
[32] C.D. Martin, M.A. Porter, The extraordinary SVD, Linear Algebra 119 (2012) 838–851.
[33] W. Matusik, M. Zwicker, F. Durand, Texture design using a simplicial complex of morphable texturess, ACM Trans. Graph. (SIGGRAPH'05) 24 (3) (2005) 787–794.
[34] R.J. McCann, A convexity principle for interacting gases, Adv. Math. 128 (1997) 153–179.
[35] A. Mensch, G. Peyré, Online Sinkhorn: Optimal Transport Distances from Sample Streams, Vol. 33, Curran Associates, Inc., 2020, pp. 1657–1667.
[36] Q. Mérigot, A multiscale approach to optimal transport, Comput. Graph. Forum 30 (5) (2011).
[37] G. Monge, Mémoire sur la théorie des déblais et des remblais, Hist. Acad. R. Sci. Paris (1781) 666–704.
[38] B. Moya, A. Badıas, I. Alfaro, F. Chinesta, E. Cueto, Digital twins that learn and correct themselves, Internat. J. Numer. Methods Engrg. (2020).
[39] M. Oberman, Y. Ruan, An efficient linear programming method for optimal transportation, Numer. Anal. (2015) arXiv:.
[40] G. Peyré, M. Cuturi, Computational optimal transport, Found. Trends Mach. Learn. 11 (5–6) (2019) 355–607.
[41] R.I. Pinillo, E. Abisset-Chavanne, A. Ammar, D. Gonzalez, E. Cueto, A. Huerta, J.L. Duval, F. Chinesta, A multidimensional data-driven sparse identification technique: The sparse proper generalized decomposition, Complexity 11 (2018) 1–11.
[42] J. Rabin, N. Papadakis, Convex color image segmentation with optimal transport distances, in: Proc. SSVM'15, 2015.
[43] A.L. Read, Linear interpolation of histograms, Nucl. Instrum. Methods Phys. Res. 425 (1999) 357–360.
[44] Y. Rubner, C. Tomasi, M.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2000) (2000).
[45] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2) (2000) 99–121.
[46] A. Sancarlos, M. Cameron, A. Abel, E. Cueto, J.L. Duval, F. Chinesta, From rom of electrochemistry to ai-based battery digital and hybrid twin, Arch. Comput. Methods Eng. (2020) 1–37.
[47] A. Sancarlos, V. Champaney, J.L. Duval, E. Cueto, F. Chinesta, Pgd-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit, arXiv:2103.05358 (2021).
[48] K. Sastry, D. Goldberg, G. Kendall, Genetic Algorithms, Springer US, Boston, MA, 2005, pp. 97–125.
[49] R. Sinkhorn, A relationship between arbitrary positive matrices and doubly stochastic matrices, Ann. Math. Stat. 35 (1964).
[50] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, Convolutional wasserstein distances: Efficient optimal transportation on geometric domains, ACM Trans. Graph. 34 (4) (2015) 1–11.
[51] J. Solomon, R. Rustamov, L. Guibas, A. Butscher, Wasserstein propagation for semi-supervised learning, in: Int. Conf. on Machine Learning, 2014.
[52] P. Tan, S. Lin, B. Guo L. Quan, H.Y. Shum, Multiresolution reflectance filtering, Render. Tech. (2005) 111–116.
[53] S.M. Udrescu, A. Tan, J. Feng, O. Neto, T. Wu, M. Tegmark, Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity, 2006, arXiv preprint arXiv:2006.10782.
[54] C. Villani, Topics in Optimal Transportation, Vol. 58, American Mathematical Soc, 2003.
[55] C. Villani, Optimal Transport, Old and New, Springer, 2006.
[56] M.E. Wall, A. Rechtsteiner, L.M Rocha, Singular Value Decomposition and Principal Component Analysis, Springer, Berlin/Heidelberg, Germany, 2003, pp. 91–109.
[57] J. Weed, F. Bach, Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance, Bernouilli 25 (4A) (2019) 2620–2648.