Hybrid twins 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

ARTICLE INFO

Keywords:
Hybrid twin
Artificial intelligence
Optimal transport
Model order reduction
Fluid dynamics

ABSTRACT

Nowadays data is acquiring an indisputable importance in every field including engineering. In the past, experimental data was used to calibrate state-of-the art models. Once the model was optimally calibrated, numerical simulations were run. However, data can offer much more, playing a more important role than calibration or statistical analysis in the modeling/simulation process. Indeed, today data is gathered and used to train models able to replace complex engineering systems. The more and better the training data, the more accurate the model is. However, in engineering experimental data use to be the best data but also the most expensive in time and computing effort. Therefore, numerical simulations, cheaper and faster, are used instead but, even if they are closed to reality, they always present an error related to the ignorance of the engineer over the complex real system. It seems thus coherent to take advantage of each approach. This leads to the "hybrid twin" rationale. On the one hand, numerical simulations are computed as primary data source, assuming their inherent error. On the other hand, some experimental data is gathered to train a machine learning correction model which fills the prediction-measurement gap. However, learning this ignorance gap becomes difficult in some fields such as fluids dynamics, where a regression over the localized solutions can lead to non physical interpolated solutions. Therefore, the "hybrid twin" methodology proposed in this article relies on Optimal Transport theory, which provides a mathematical framework to measure distances between general objects and a completely different interpolation approach between functions.

1. Introduction

Physical problems frequently introduce parameters into their mathematical modeling. This is especially true when the problem is complex. In engineering, the complexity of systems is such that many parameters are involved in their description. Therefore, one wishes to have a parametric model, i.e., for a given system, a solution that is a function of all its parameters. However, the construction and training of such a model relies on experimental and numerical data, which are costly and time consuming. Thus, advanced mathematical regressions are proving to be indispensable tools for the construction of such parametric solutions.

Such mathematical models need to be trained, in an offline stage, with numerical or experimental data, solution of the engineering problem on the training points. The success of the regression methods relies on the quality and quantity of this training data since the models can only be as accurate and precise as the information and data used to obtain them [6]. Therefore, the question arises of which training data to use. Indeed, on the one hand, experimental data is usually considered as the "correct" solution of the engineering system but turns out to be extremely expensive and time-consuming. On the other hand, numerical simulation appeared to be a promising: physically-based, cheaper and faster.

Indeed, as engineering models become more and more complex, their analytical solution is compromised. Moreover, the development and democratization of more and more powerful computers, since the mid-20th century [17], led to the emergence of the so called third paradigm of science: the "virtual twins" or physic-based numerical simulation, reproducing a physical system using mathematical models to emulate its complex behavior [9]. Nowadays, numerical simulation has become an essential tool for scientific investigation and analysis of

* 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).

complex systems in engineering, drastically reducing the number of experimental tests [9,15,7].

However, "virtual twins" are limited by the size and complexity of the problem studied, which can not be overcome despite increasingly computation power [9]. Indeed, computational resources, and its linked cost, are proportional to the intricacy of the engineering system. Numerical simulations are, thus, do not adapted to real-time constraints. Moreover, Albert Einstein stated another major issue of numerical simulation: "As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality". Indeed, physic-based simulations present some significant deviations when compared to measurement data. These deviations are expected to be biased since they represent the ignorance of the modeler on the subjacent physics. Indeed, they are related to the inaccuracy in the employed models, in its numerical time-space discretization or in its parameters [9,15,28].

Although science was mainly experimental at the very beginning, mathematical language has made it theoretic several centuries ago, expressing physics through universal laws. At the mid-20th century, as presented before, science became also computational thanks to affordable computer facilities [10]. Finally, in the late 20th century, science became also data-based. Indeed, data has massively proliferated in the majority of scientific fields and, thanks to widely developed artificial intelligence and machine learning techniques, data-based models (also called "digital twins") have partially or totally substituted physic-based ones thanks to their lower computational complexity. Such models represent a real-world system, with all its complexity, guaranteeing at the same time real-time constraints and without needing an insight into the actual physics of the asset. However, they do need a lot of training data to be built, lead to feelings of waste of acquired knowledge and to some aversion since they usually work as black boxes [9].

This present data-based revolution lead the fourth paradigm, also called "data-intensive science", which provides a new framework where data, theory and simulation can interact and reinforce each other [17]. Indeed, the most appealing solution seems to be combining both the "digital" and "virtual" twins [9]. On the one hand we need the most "correct" solutions to train our parametric solution but experimental data are too expensive and time-consuming. On the other hand, we have access to computational facilities to run "as many as needed" numerical simulations but these present a non negligible deviation with the actual (i.e. experimental) evolution of the system. It seems thus coherent to use both technologies together, using the advantages of each approach [31,7].

The "hybrid twin" (HT) is thus born, composed of a imperfect physic-based simulation of the system (the "virtual twin") and of a data-based model emulating the ignorance gap between measurement and prediction (the "digital twin"). It can be noted that the data-based model acts as a black box. However, this becomes less inconvenient within the hybrid twin rationale, since artificial intelligence is only applied to model the physical part that is beyond our knowledge [9].

The question that arises now is, how do we model the ignorance, i.e. the prediction-measurement gap? One can think about learning this gap as a difference. However, with this approach, the results of the regressions could be non-physical in, for instance, fields where a given choice of the problem parameters leads to a localized solution in different regions (e.g. fluid mechanics). A more physical solution would be to use Optimal Transport (OT) theory [35]. This solution has already been applied by the authors in previous work [33]. Indeed, Optimal Transport provides a mathematical framework to calculate distances between general objects, which can be considered more physical in many fields. Thus, the goal of this article is to present an Hybrid Twin where the data-based model follows the Optimal Transport theory: the ignorance is learnt as a transport of information. Hence, the prediction data is OT-based corrected by being optimally transported to the measurement data.

The Optimal Transport theory [35] generalizes the idea of the optimal solution when transporting an object from an initial to a target point employing the minimum work, i.e. the shortest path. Indeed, this theory minimizes the transport cost when moving a continuous distribution from an initial to a final configuration [29]. Thus, OT defines a new mathematical framework to understand and measure distances where the geometry of the underlying space is taken into account [34].

OT introduces a completely new interpolation approach between functions, even if they are defined over disjoint supports. Hence, in contrast to the usual Euclidean interpolation, which results in the mixture of two functions, interpolation based on OT provides a continuous scaling and translation. Indeed, OT theory quantifies the distance between two functions by determining the cheapest manner to transport and reshape the whole information provided by one function into another. In this sense, OT considers as identical two functions differing exclusively by a small horizontal displacement while classical $L1$ or $L2$ norms consider them as very dissimilar. Therefore, such a interpolation point of view is more accurate to reality in many domains, such as, fluid mechanics or computer graphics [22], hence its increasing popularity.

However, the resolution of the Optimal transport problem remains not accessible in an online approach and numerically expensive [36,5,25], notwithstanding the substantial recent progress [27,22,3,11,30,2,21,23,8]. Among all this advances, the Lagrangian formulation of the problem, introduced in [24], has been used to approach the Optimal Transport problem as a mass transport problem: the source distribution is described by a set of mass units that need to be moved to the target distribution, while minimizing the transport cost [4]. When taken into account the OT between two distributions, this approach can be seen as a bipartite graph matching [18]. The method developed in this paper is based on this interpretation of the problem. Indeed, the OT-based gap is calculated from a Lagrangian approach and a "digital twin", transporting the prediction to the measurement, is trained.

In this paper, a two stages approach is introduced: first, the OT-based "digital twin" is trained offline, then, this data-based correction is applied over the "virtual twin" output in an online manner. Indeed, physic-based simulations and their experimental counterparts are used to train the OT-based correction model. Then, trained "digital twin" can be used in an online manner to correct further simulations from the "virtual twin" (from which we do not have access to the experimental data).

First, high-fidelity simulations (the "virtual twin") and their experimental counterparts are used in the offline stage to train the model. Note that these training solutions are computed in the parametric space of our problem. The steps of the offline stage are as follows: first, based on a Smoothed-Particle Hydrodynamics (SPH) decomposition [20], the training experimental measurements and numerical high-fidelity simulations are decomposed into the sum of identical Gaussian functions (referred as particles). Then, each Gaussian from the "virtual twin" is matched with a Gaussian from its corresponding experimental data. And this for all the prediction-measurements couples of the training set. The OT-based differences between prediction and measurement can therefore be calculated. Next, a Partial Least-square (PLS) regression [1,32,12,14] is applied over the OT-based gaps in order to build the OT-based "digital twin".

Then, the just trained data-based parametric model of the ignorance can be applied over a new numerical simulation, from which we do not have access to the experimental data, in an online manner in order to correct it. To this purpose, the "virtual twin" output is decomposed in Gaussian particles and the OT-based gap, interpolated by the "digital twin", is added to the simulation particles. The numerical simulation is thus corrected following the Optimal Transport theory and the so expected experimental corresponding data is reconstructed by summing all the Gaussians.

In this article, the principal ideas of OT theory, on which the OT-based gap model is based, are first presented and illustrated. Then, the "digital twin" is presented and the offline stage of the methodology is

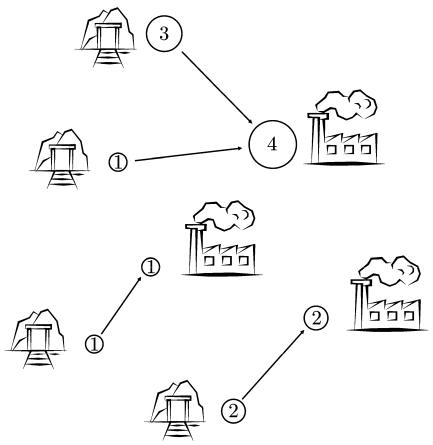


Fig. 1. Discrete OT formulation for $N = 4$ mines and $M = 3$ factories. The resource produced by the mines is: $a_1 = 3$, $a_2 = 1$, $a_3 = 1$ and $a_4 = 2$. The resource consumed by the factories is: $b_1 = 4$, $b_2 = 1$ and $b_3 = 2$. The Euclidean distance traveled by the resource is the cost to minimize.

detailed. Next, the OT-based "digital twin"[2] online application over the "virtual twin" is introduced. Finally, the accuracy of the OT-based correction is evaluated through several examples. It must be noted that the main goal of this paper is to introduce the novel methodology. Thus, the ignorance gap between measurement and prediction for the assessment examples is here simulated, as presented in the results section. However, further work is planned and in progress to apply this approach to a real industrial case.

2. Revisiting optimal transport

In this section, the OT framework is presented and the tools on which the thereafter proposed OT-based parametric model of the ignorance is based introduced. Note that this section is an non exhaustive introduction of the principal ideas of OT. For further documentation on this topic, [29] and the references therein can be consulted.

Introduced by Monge [26], the initial Optimal Transport problem consisted in calculating the most optimal path to move a given quantity of soil from an initial to a target location where the cost function was defined as the distance traveled by the soil. Note that in this article we are interested in the discrete formulation of the problem. In order to introduce the discrete perspective let us consider N mines producing a certain resource that needs to be transported to M factories. We seek for the Optimal Transport solution that minimizes a cost function defined as the square of the total Euclidean distance traveled by the resource, as it is showed in the Fig. 1.

On the one hand, each mine $n \in [\![N]\!]$ (note that the notation $[\![N]\!]$ corresponds to $\{1, \dots, n, \dots, N\}$) produces an amount a_n of the resource and it is located at x_n. On the other hand, each factory $m \in [\![M]\!]$ consumes an amount b_m of this same resource and it is located at y_m. Hence, two distributions, α and β, corresponding to the resource produced and consumed respectively can be defined following the notion of measure:

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

where δ_{x_n} and δ_{y_m} correspond to the Dirac at locations x_n and y_m respectively.

Therefore, solving the Monge problem consists in finding the map T connecting each point x_n with a single target point y_m such that the produced resource distribution, α, is pushed toward the consumed resource distribution, β. It is important to note that the resource cannot be destroyed or produced during its transport. Indeed, the transport map $T : \{x_1, \dots, x_N\} \to \{y_1, \dots, y_M\}$ satisfies the mass conservation

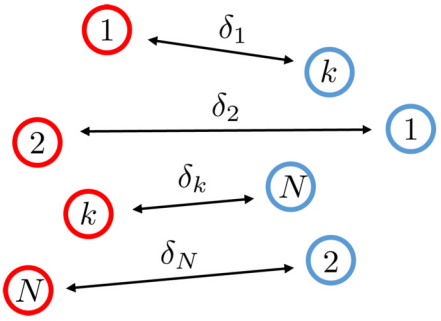


Fig. 2. 2D Monge problem with $N = M$ and $a_n = b_m = 1/N$. Mines and factories are represented by red and blue circles respectively. Black arrows illustrate the optimal matching.

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

which can also be written in a compact form $T_{\#}\alpha = \beta$. Note that the map T is a surjective function. Finally, this transport map is determined such that it minimizes the transport cost. Here, we consider the square of the $L2$ distance between the mine n and its corresponding factory m:

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

Hence, we obtain the following minimization problem:

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

The discrete Monge problem has been presented. However, in this paper we are interested in a simplified version of the problem. Indeed, it is first supposed that the number of mines and factories is the same, i.e. $N = M$. In addition, it is also supposed that the quantity of resource produced by each mine is the same and that each factory also consumes this same quantity of resource, i.e. $a_n = b_m = 1/N$. Thus, the transport map becomes a bijective function and the optimization problem (4) is now a deterministic matching problem.

Under these assumptions, linear programming can be used to easily solve the just simplified Monge problem. Indeed, the problem becomes an optimal matching problem between two particle clouds. As it is illustrated in the Fig. 2 in 2D, each cloud is composed by the same number of particles, every particle has the same amount of mass and the cost is defined as the square of the $L2$ distance between two particles. In higher dimensions than 2D, the computational time needed to calculate the distances increases but the problem does not further complexify.

Once the two particle clouds are optimally paired, the OT-based difference between them can be determined by calculating the Euclidean distances between matched particles. Indeed, the OT-based difference is the total sum of the $L2$ distances δ_k for all the pairs of particles.

3. Hybrid twin based on optimal transport

In this section, the two stages of the "hybrid twin" methodology are presented in detail. First, the offline stage, where the OT-based "digital twin" is trained, is introduced. Then, the online stage, where the "virtual twin" is corrected, is presented. Finally, the whole methodology is summarized and schematized.

The offline stage of the approach consists in the construction of the OT-based "digital twin" and follows the next four steps. First, the numerical high fidelity simulations, from the "virtual twin", and their experimental counterparts are decompounded into a sum of Gaussian functions (referred as particles) following an SPH approach. It can be noted that the number of functions and their standard-deviation are fixed as hyperparameters. The only remaining variables are the means of every Gaussian function (which can be seen as x and y coordinates in $\mathbb{R}^2$). Then, for each prediction-measurement couple, each Gaussian

from the numerical simulation is optimally matched with one Gaussian from the experimental data. Once this matching is done, the difference between the coordinates for the matched particles is calculated. Indeed, this difference constitutes the OT-based gap between prediction and measurement. Lastly, a PLS regression is applied over the OT-based difference, leading to a OT-based "digital twin" accessible in an online manner.

In the online stage, a new numerical simulation from the "virtual twin", from which we do not have access to the experimental data, can be corrected by the "digital twin" leading to the so called "hybrid twin" approach. To this purpose, the "virtual twin" output is decomposed in Gaussian particles and the OT-based gap, interpolated by the "digital twin", is added to the simulation particles.

3.1. Offline stage: the OT-based "digital twin" training

The four steps of the offline stage are here presented. Let suppose that our parametric problem is defined in a parametric space $\mathcal{W}(\eta_1,\dots,\eta_q,\dots,\eta_Q)$ where $\eta_q, q \in [\![Q]\!]$ are the parameters. Next, let consider P prediction-measurement couples in the space $\mathcal{W}(\eta_1,\dots,\eta_q,\dots,\eta_Q)$ corresponding to the 2D high-fidelity simulations and their experimental counterparts of the parametric problem. Hence, the OT-based "digital twin" relies on the simplified 2 dimensional minimization problem (4), i.e. on a deterministic matching problem between 2 particle clouds: the numerical data and its measurement counterpart.

3.1.1. SPH decomposition

Without loss of generality, let suppose a 2D problem. Each numerical and experimental data sample is formally represented by a distribution $\psi : \Omega \in \mathbb{R}^2 \to \mathbb{R}^+$. It can be noted that the image of ψ is supposed strictly positive. Moreover, distributions are normalized:

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

The idea is, for each prediction-measurement couple, to decompose the high-fidelity simulation and it experimental counterpart, i.e. ρ_υ and ρ_e, into a sum of N Gaussian functions, also referred as particles, following a Smoothed-Particle Hydrodynamics [20] rationale. Note that the number of particles N is an hyperparameter of the methodology, i.e. it is fixed for both the numerical and experimental data, and for all the prediction-measurement couples. Moreover, every Gaussian function has the same standard-deviation σ, which becomes also an hyperparameter of the methodology. Hence, the only variables are, for a given distribution, the means $\boldsymbol{\mu}$ of each Gaussian function, i.e. N vectors of 2 components: μ_x and μ_y (because we are in 2 dimensions). Thus, a normalized distribution $\rho : \Omega \in \mathbb{R}^2 \to \mathbb{R}^+$ is 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 function of standard-deviation σ, $1/N$ mass and mean $\boldsymbol{\mu}_n$:

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

Therefore, each 2D distribution, of each prediction-measurement couple, is decomposed into the sum of N particles each of which is described by 2 coordinates, μ_x and μ_y, as it is illustrated in Fig. 3, where the pth prediction-measurement couple is represented. The nth $\in [\![N]\!]$ particle of the "virtual twin" distribution of the pth $\in [\![P]\!]$ couple is noted:

$$G_{\boldsymbol{\mu}^p_{\upsilon_n},\sigma} \quad \text{where} \quad \boldsymbol{\mu}^p_{\upsilon_n} = \left[\mu^p_{\upsilon_{n,x}}, \mu^p_{\upsilon_{n,y}}\right] \in \mathbb{R}^2. \tag{8}$$

Hence, one can introduce the matrix $\boldsymbol{\mu}^p_\upsilon \in \mathbb{R}^{N\times 2}$, composed by the coordinates x and y of all the particles of the "virtual twin" distribution of the pth couple:

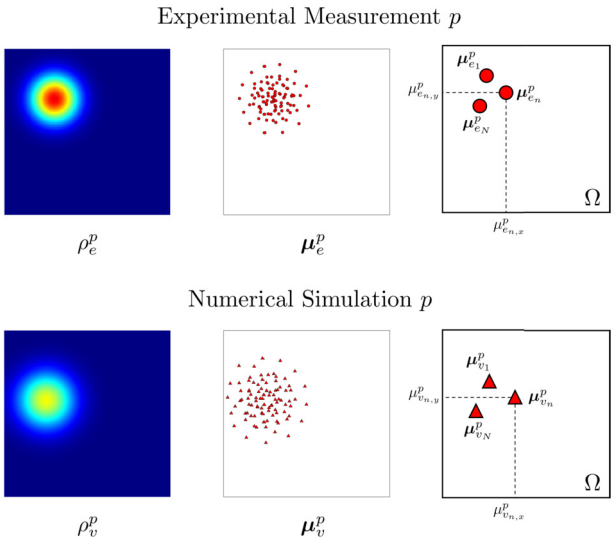


Fig. 3. pth prediction-measurement couple: the first row shows the experimental data and the second row the numerical one. The first column shows the experimental measurement ρ^p_e and it high-fidelity simulation counterpart ρ^p_υ. In the second column the decomposition into N Gaussian particles is represented showing the N-particles clouds $\boldsymbol{\mu}^p_e$ and $\boldsymbol{\mu}^p_\upsilon$, each particle represents a Gaussian function. The third column schematizes the particles decomposition in order to illustrate the defined notation.

$$\boldsymbol{\mu}^p_\upsilon = \begin{bmatrix} \boldsymbol{\mu}^p_{\upsilon_1} \\ \vdots \\ \boldsymbol{\mu}^p_{\upsilon_n} \\ \vdots \\ \boldsymbol{\mu}^p_{\upsilon_N} \end{bmatrix} = \begin{bmatrix} \left[\mu^p_{\upsilon_{1,x}}, \mu^p_{\upsilon_{1,y}}\right] \\ \vdots \\ \left[\mu^p_{\upsilon_{n,x}}, \mu^p_{\upsilon_{n,y}}\right] \\ \vdots \\ \left[\mu^p_{\upsilon_{N,x}}, \mu^p_{\upsilon_{N,y}}\right] \end{bmatrix} \in \mathbb{R}^{N\times 2}. \tag{9}$$

It is important to note that the order of the particles in this matrix $\boldsymbol{\mu}^p_\upsilon$ is not arbitrary but will be used to represent the assignment between point clouds: the nth particle of one cloud being matched with the nth particle of another cloud when needed, as explained in the next section. Finally, the same notation is introduced for the experimental data distribution of the pth $\in [\![P]\!]$ couple replacing subscript υ by e.

The decomposition into N particles of the pth prediction-measurement couple, i.e. ρ^p_υ and ρ^p_e, consists in 2 identical optimization problems. Without loss of generality, we introduce here the one related to ρ^p_υ. In this optimization problem (10), the variable is $\boldsymbol{\mu}^p_\upsilon \in \mathbb{R}^{N\times 2}$. The discretization points of Ω, $\mathbf{x}_i, i \in [\![D]\!]$, follow a uniform meshing. Since the number of functions N and their standard-deviation σ are fixed, the optimization problem writes

$$\min_{\boldsymbol{\mu}^p_\upsilon} \frac{1}{2} \left\| \rho^p_\upsilon - \bar{\rho}^p_\upsilon \right\|_2^2 = \min_{\boldsymbol{\mu}^p_\upsilon} \frac{1}{2} \left[\sum_{i=1}^{D} \left(\rho^p_\upsilon(\mathbf{x}_i) - \sum_{n=1}^{N} G_{\boldsymbol{\mu}^p_{\upsilon_n},\sigma}(\mathbf{x}_i) \right)^2 \right], \tag{10}$$

where D is the number of points of the mesh where the distribution ρ^p_υ is calculated. It can be noted that in each of these optimization problems we seek for the N Gaussian functions that decompose each of the data fields. In (10) the decomposition of the pth numerical simulation has been written as an example. Indeed, $\boldsymbol{\mu}^p_{\upsilon_n}$ represents the nth particle of $\boldsymbol{\mu}^p_\upsilon$ as it has been explained in (9).

A Gradient Descent approach is chosen in order to solve the optimization problem. It can be noted that the offline stage contains $P \times 2$ such optimization problems. The resolution methodology, as well as the optimal choice for the hyperparameter σ, have been presented in detail by the authors in [33].

3.1.2. Prediction-measurement particles matching

Once the high fidelity simulations and their experimental counterparts are decomposed into N particles, the optimal assignment between

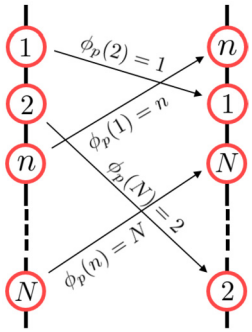


Fig. 4. Illustration of the ordering function ϕ_p. For each particle n of the distribution ρ_v^p, the function ϕ_p indicates the new position.

both distributions for each prediction-measurement couple, can be determined. For a prediction-measurement couple, this can be understood as the optimal matching problem between two N-particles clouds presented in Fig. 2, where each particle is a 2D-Gaussian function represented by its μ_x and μ_y coordinates. This linear assignment problem, where the cost function is the sum of the squared $L2$ distances between matched particles, can be resolved applying several algorithms. Here, the algorithm `matchpairs` from MATLAB is used to solve the problem [13].

Hence, for the pth prediction-measurement couple, the cost between the decomposed numerical simulation and the experimental measurement, ρ_v^p and ρ_e^p, writes:

$$C_{v,e}^p(\phi_p) = \sum_{n=1}^{N} \left\| \boldsymbol{\mu}_{v_{\phi_p(n)}}^p - \boldsymbol{\mu}_{e_n}^p \right\|_2^2 \tag{11}$$

where ϕ_p is a bijective function in the set of permutations of N elements. To each particle n of the distribution ρ_v^p, $\phi_p : \mathbb{N} \to \mathbb{N}$ associates its new position in the sense of order in $\boldsymbol{\mu}_v^p$, as it can be seen in Fig. 4. It is important to remember that the order of the particles in $\boldsymbol{\mu}_v^p$ and in $\boldsymbol{\mu}_e^p$ represents the matching between particles for the pth prediction-measurement couple. Indeed, the particle $\boldsymbol{\mu}_{v_n}^p$ of the pth simulation is matched with the particle $\boldsymbol{\mu}_{e_n}^p$ of the pth experimental measurement counterpart. Hence, by permuting the elements of $\boldsymbol{\mu}_v^p$ or $\boldsymbol{\mu}_e^p$ the assignment between the two N-particles clouds is modified. The aim here is, thus, to find the optimal ordering corresponding to the optimal matching, which minimizes the defined cost, i.e. the optimal function ϕ_p. It can be noted that here the permutation has been applied to the particles of ρ_v^p. However, this is an arbitrary choice since when matching two sets permuting one of them is enough. Then, this matching is conducted for all the P couples, i.e. P optimal matching problems between two N-particles clouds are solved.

3.1.3. Prediction-measurement OT-based difference

Then, the OT-based difference can be calculated. To this purpose, for each prediction-measurement couple, the difference between the coordinates of the matched N particles is computed. It is important to note that every prediction-measurement couple has been matched, i.e. every $\boldsymbol{\mu}_v^p$ has been reorganized to minimize the cost (11). Hence, the difference between coordinates is computed between particles optimally matched. This gives sense to the so called OT-based difference. Thus, the OT-based gap of the nth particle of the pth couple is noted:

$$\boldsymbol{\delta}_n^p = \left[\delta_{n,x}^p, \delta_{n,y}^p\right] = \left[\mu_{v_{n,x}}^p - \mu_{e_{n,x}}^p, \mu_{v_{n,y}}^p - \mu_{e_{n,y}}^p\right], \tag{12}$$

as it is illustrated in Fig. 5. Then, the OT-based difference for the pth couple $\boldsymbol{\delta}^p \in \mathbb{R}^{N\times 2}$ is build as:

$$\boldsymbol{\delta}^p = \begin{bmatrix} \boldsymbol{\delta}_1^p \\ \vdots \\ \boldsymbol{\delta}_N^p \end{bmatrix} \in \mathbb{R}^{N\times 2}. \tag{13}$$

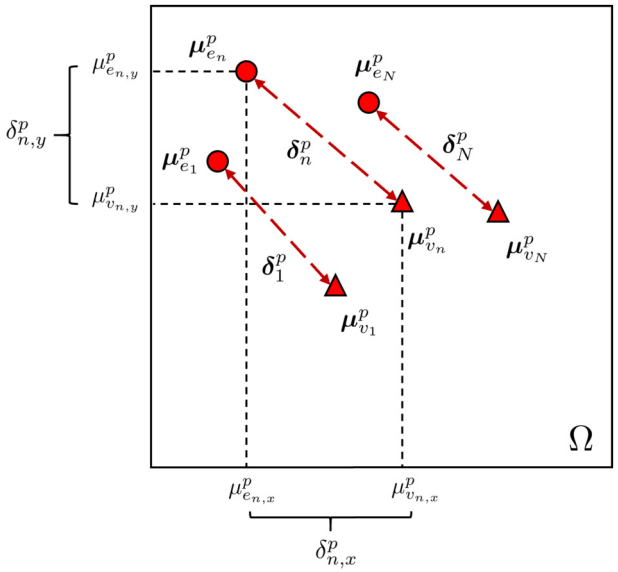


Fig. 5. Scheme of the OT-based difference for the pth prediction-measurement couple. The experimental data decomposed into particles is represented by the circles and its numerical counterpart by triangles. The red double arrows represent the matching between particles. It can be noted this assignment follows the order of the particles (first experimental particle with first numerical particle, second with second and so on) since both particles clouds have been previously matched.

Finally, we compute this OT-based difference $\boldsymbol{\delta}^p$ for all the P couples.

3.1.4. "Digital twin" regression model

Finally, the OT-based "digital twin" can be built relying on a Partial Least-Squares (PLS) regression [1,32,12,14,16]. The PLS method is a latent variable model where principal components are determined such as they best explain the explanatory variables, the response variables and such as they have the strongest possible relationship between both types of variables. The PLS regression is presented in the Appendix A.

Here, the regression is applied over the OT-based differences $\boldsymbol{\delta}^p, p \in [\![P]\!]$ leading to a function that, for a set of parameters from the parametric space $\mathcal{W}(\eta_1, \ldots, \eta_q, \ldots, \eta_Q)$, returns the OT-based gap for each particle.

For each set of parameters p of the training set (i.e. each prediction-measurement couple), there are N OT-based gaps $\boldsymbol{\delta}_n^p, n \in [\![N]\!]$. Therefore, in order to train the PLS regression, the parametric space $\mathcal{W}$ is extended. To each set of parameters p of the training set, we add the coordinates of the N particles of the pth "virtual twin" decomposition: $\boldsymbol{\mu}_{v_x}^p \in \mathbb{R}^N$ and $\boldsymbol{\mu}_{v_y}^p \in \mathbb{R}^N$ respectively.

The new parametric space is noted $\mathcal{W}'(\eta_1, \ldots, \eta_q, \ldots, \eta_Q, \mu_{v_x}, \mu_{v_y})$. This yield to the following $(P \times N) \times (Q + 2)$ matrix X of explanatory variables:

$$X = \begin{bmatrix} \eta_1^1 \ldots \eta_q^1 \ldots \eta_Q^1 & \mu_{v_{1,x}}^1 & \mu_{v_{1,y}}^1 \\ \vdots & \vdots & \vdots \\ \eta_1^1 \ldots \eta_q^1 \ldots \eta_Q^1 & \mu_{v_{N,x}}^1 & \mu_{v_{N,y}}^1 \\ \vdots & \vdots & \vdots \\ \eta_1^p \ldots \eta_q^p \ldots \eta_Q^p & \mu_{v_{1,x}}^p & \mu_{v_{1,y}}^p \\ \vdots & \vdots & \vdots \\ \eta_1^p \ldots \eta_q^p \ldots \eta_Q^p & \mu_{v_{N,x}}^p & \mu_{v_{N,y}}^p \\ \vdots & \vdots & \vdots \\ \eta_p^P \ldots \eta_q^P \ldots \eta_Q^P & \mu_{v_{1,x}}^P & \mu_{v_{1,y}}^P \\ \vdots & \vdots & \vdots \\ \eta_1^P \ldots \eta_q^P \ldots \eta_Q^P & \mu_{v_{N,x}}^P & \mu_{v_{N,y}}^P \end{bmatrix} \in \mathbb{R}^{(P\times N)\times(Q+2)}, \tag{14}$$

where η_q^p corresponds to the qth parameter of the pth prediction-measurement couple.

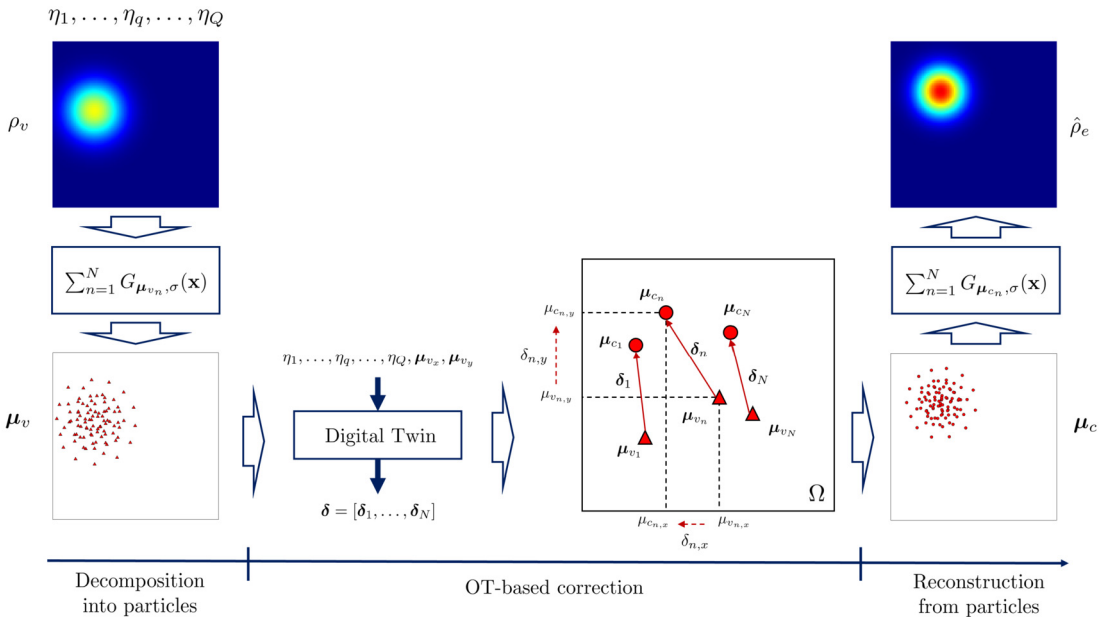


Fig. 6. Scheme of the three steps of the online stage of the methodology.

Moreover, the $(P \times N) \times 2$ matrix Y of response variables is the concatenation of the P OT-based differences $\boldsymbol{\delta}^p, p \in [\![P]\!]$. Hence, the matrix Y writes:

$$Y = \begin{bmatrix} \boldsymbol{\delta}^1 \\ \vdots \\ \boldsymbol{\delta}^p \\ \vdots \\ \boldsymbol{\delta}^P \end{bmatrix} = \begin{bmatrix} \delta^1_{1,x} \delta^1_{1,y} \\ \vdots \\ \delta^1_{N,x} \delta^1_{N,y} \\ \vdots \\ \delta^p_{1,x} \delta^p_{1,y} \\ \vdots \\ \delta^p_{N,x} \delta^p_{N,y} \\ \vdots \\ \delta^P_{1,x} \delta^P_{1,y} \\ \vdots \\ \delta^P_{N,x} \delta^P_{N,y} \end{bmatrix} \in \mathbb{R}^{(P \times N) \times 2}. \tag{15}$$

The PLS regression is trained using the MATLAB function `plsregress` [12] leading to a regression coefficients matrix $B_{PLS} \in \mathbb{R}^{(Q+2)\times 2}$ and a constant $\mathbf{y_0} \in \mathbb{R}^2$ such that:

$$\hat{\mathbf{y}} = \mathbf{x}\ B_{PLS} + \mathbf{y_0}, \tag{16}$$

where $\mathbf{x}$ is a new row of X, i.e. the set of parameters of a new numerical simulation and the coordinates x and y of one of its N particles, and $\hat{\mathbf{y}}$ are the interpolated components of the OT-based correction for this simulation and this particle. It can be noted that the PLS regression has been chosen since it is able to manage the correlation between variables in the matrix X (due to its construction).

The OT-based "digital twin" is thus built. Hence, for any "virtual twin" simulation that has been decomposed in particles, one can compute the OT-based correction to get the expected experimental counterpart. The detailed correction methodology is presented thereafter.

3.2. Online stage: the "virtual twin" correction

The just trained OT-based "digital twin" models the "engineer ignorance" and can be applied over a new numerical simulation from the "virtual twin", from which we do not have access to the experimental data counterpart, leading to the so called "hybrid twin". This OT-based correction can be done in a partially online manner.

Let introduce a new high-fidelity numerical simulation from our parametric space $\mathcal{W}(\eta_1, \ldots, \eta_q, \ldots, \eta_Q)$. The corresponding normalized distribution is called ρ_v. Note that the experimental counterpart ρ_e is unknown. The three steps of the online stage, described thereafter, are illustrated in Fig. 6.

First, the simulation is decomposed into particles following the previously described SPH approach, leading to N Gaussian particles identified by their x and y means (i.e. x and y coordinates):

$$\boldsymbol{\mu}_v = \left[\boldsymbol{\mu}_{v_x}, \boldsymbol{\mu}_{v_y}\right] \in \mathbb{R}^{N\times 2}, \tag{17}$$

such that

$$\rho_v = \sum_{n=1}^{N} G_{\boldsymbol{\mu}_{v_n},\sigma}(\mathbf{x}). \tag{18}$$

It can be noted that the number of particles N, as well as the standard-deviation σ, used to decompose this new numerical simulation are the same that have been fixed as hyperparameters during the "digital twin" training.

The decomposed distribution is therefore described by the set of parameters, $\eta_1, \ldots, \eta_q, \ldots, \eta_Q$, of the solved problem and by the coordinates x and y of the N particles: $\boldsymbol{\mu}_{v_x}$ and $\boldsymbol{\mu}_{v_y}$ respectively. Next, the trained PLS regression applied over this new N inputs in $\mathcal{W}'$ returns the x and y components of the OT-based correction for all the particles: $\boldsymbol{\delta} \in \mathbb{R}^{N\times 2}$. Adding this difference to the SPH-decomposition particles leads to the corrected positions of the particles:

$$\boldsymbol{\mu}_c = \boldsymbol{\mu}_v + \boldsymbol{\delta}. \tag{19}$$

Then, the so expected experimental counterpart, $\hat{\rho}_e$, can be reconstructed by summing all the new Gaussian functions:

$$\hat{\rho}_e = \sum_{n=1}^{N} G_{\boldsymbol{\mu}_{c_n},\sigma}(\mathbf{x}). \tag{20}$$

Finally, in order to recover $\hat{\psi}_e$, the ratio of the total masses of the measurement/prediction couples, $\mathcal{I}_v/\mathcal{I}_e$, is interpolated.

3.3. Model review

The "hybrid twin" approach introduced in this article is now reviewed, as it is illustrated in the Fig. 7. Given $P = 4$ simulation-measurement couples of a problem defined in a parametric space $\mathcal{W}(\eta_1, \ldots, \eta_q, \ldots, \eta_Q)$ relying on Q parameters, the aim is to build an

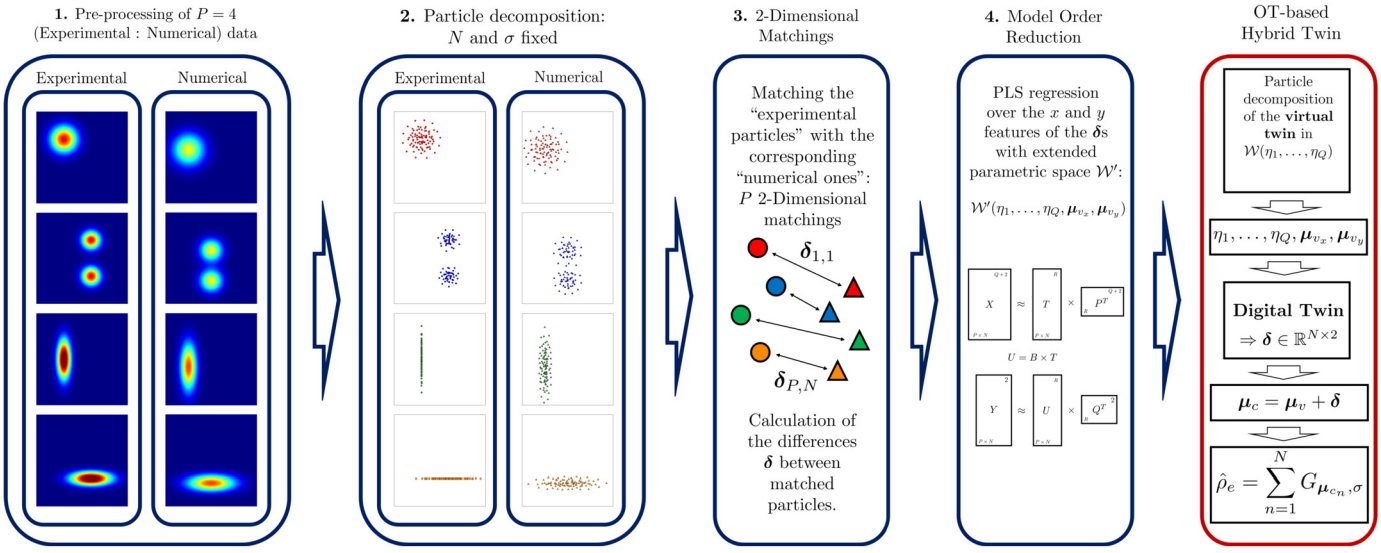


Fig. 7. Summary diagram of the methodology: the training of the "digital twin" is colored in blue and the OT-based "hybrid twin" approach in red.

OT-based "digital twin" combining Optimal Transport theory and PLS regressions. This ignorance model can correct a "virtual twin" simulation, from which we do not have the experimental counterpart, leading to the so called "hybrid twin" approach.

The two stages methodology proceeds as follows. First, the OT-based ignorance model is trained offline based on the simulation-measurement couples following the next steps (colored in blue in the Fig. 7):

1. **Preprocessing:** Normalization of the distributions corresponding to the high-fidelity numerical simulations and the experimental counterparts to obtain unitary integral.
2. **SPH decomposition:** Every measurement and simulation is decompounded into a sum of N identical Gaussian functions of fixed standard deviation σ and mass $1/N$. Remember that the number of particles N and the standard deviation σ of each particle are hyperparameters of our methodology. Therefore, we need to solve $P \times 2$ optimization problems (two for each simulation-measurement couple) in order to place the N particles minimizing the error with respect to the original data. To this purpose a Gradient Descent approach is used.
3. P **2-dimensional matching:** The locations of the N particles for the P numerical simulations and for P experimental measurements are, thus, defined. Next, the Optimal Transport behavior is emulated: for each simulation-measurement couple, each particle from the numerical simulation is matched with one particle from the experimental counterpart. Then, the OT-based gap for each couple is determined by computing difference between the coordinates x and y of the N matched particles.
4. **"Digital twin" training:** Finally, the ignorance model is built using a PLS regression. The PLS model is applied over an extended parametric space as explanatory variables and over the OT-based difference as response variables.

Then, the "digital twin" can be used in a partially online manner to correct a "virtual twin" simulation, from which we do not have the experimental counterpart (colored in red in the Fig. 7):

1. A new high-fidelity simulation, computed in the parametric space $\mathcal{W}$, is decomposed into N particles.
2. Then, the "digital twin" returns the OT-based gap that needs to be added to these particles to obtain the particles of the experimental counterpart.
3. Finally, these N corrected Gaussian functions are added to reconstruct the expected experimental data.

4. Results

4.1. Error evaluation

First of all, the error evaluation methodology is presented. Let us introduce a testing set in the parametric space $\mathcal{W}$. The P_{test} reference solutions, i.e. the experimental data, $\rho_e^p, p \in [\![P_{test}]\!]$ of this set are compared with the corrected numerical solutions $\hat{\rho}_e^p, p \in [\![P_{test}]\!]$ obtained with the OT based "hybrid twin" approach developed. To this purpose, three error metrics are here defined: a maximum value error, a maximum value position error and a L^2-Wasserstein error. It can be note that the reference solution is also compared with the "virtual twin" solution $\rho_v^p, p \in [\![P_{test}]\!]$. In this section, and without loss of generality, the error metrics are presented between the reference and the "hybrid twin" solutions.

First, the maximum value error is calculated as a percentage of the relative difference between the maximum value of the measurement and the maximum value of the OT corrected solution. Hence, the maximum value error for the pth training point writes:

$$\varepsilon_{max}^p = 100 \frac{\left|\max(\rho_e^p) - \max(\hat{\rho}_e^p)\right|}{\max(\rho_e^p)}. \tag{21}$$

Next, the maximum value position error is calculated as the $L2$ norm between the positions in Ω of the maximum value of the experimental data and of the maximum value of the "hybrid twin" solution. Note that this error is normalized with respect to l_Ω, the length of one side of Ω:

$$\varepsilon_{pos}^p = \frac{\left\| \text{argmax}_{\mathbf{x}} \left(\rho_e^p(\mathbf{x})\right) - \text{argmax}_{\mathbf{x}} \left(\hat{\rho}_e^p(\mathbf{x})\right) \right\|_2}{l_\Omega}. \tag{22}$$

Finally, the L^2-Wasserstein metric $W(\rho_e^p, \hat{\rho}_e^p)_2^2$ is calculated between the reference ρ_e^p and the modeled $\hat{\rho}_e^p$ solutions where $p \in [\![P_{test}]\!]$. In order to calculate $W\left(\rho_e^p, \hat{\rho}_e^p\right)$, a Linear Programming methodology is followed.

The Kantorovich OT problem [19] corresponds to an infinite dimensional Linear Program. Indeed, given the distributions ρ_e^p and $\hat{\rho}_e^p$, defined on X and Y, the problem reads

$$W\left(\rho_e^p, \hat{\rho}_e^p\right)_2^2 = \min_{\pi \in \Pi(\rho_e^p, \hat{\rho}_e^p)} \int_{X \times Y} c(x,y) \mathrm{d}\pi(x,y), \tag{23}$$

where $c(x,y) : X \times Y \to \mathbb{R}$ is the cost function and Π the set of transfer plans. The discretized measures ρ_e^p and $\hat{\rho}_e^p$ are defined as weighted sums

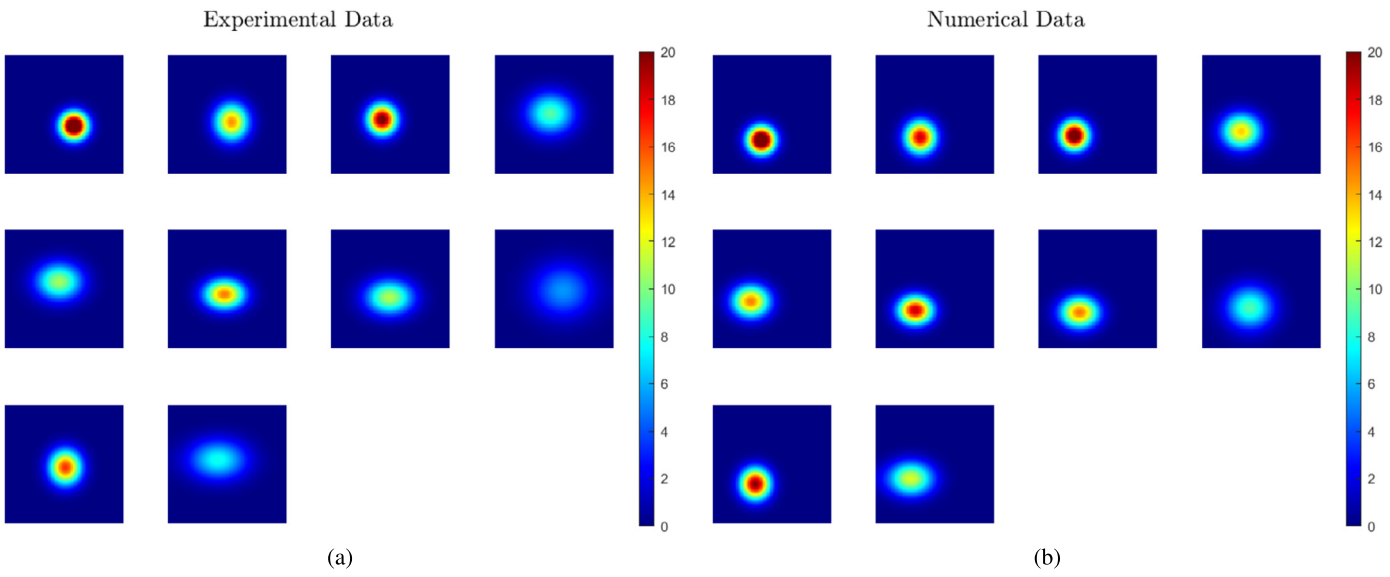


Fig. 8. Test set data points: (a) Finite elements solution of the problem (29) corresponding to the experimental data. (b) Finite elements solution of the problem (30) corresponding to the "virtual twin" simulation.

of Dirac functions. The weights represent the value of the continuous measures evaluated at the corresponding nodes $\mathbf{x}_i$ and $\mathbf{y}_i$ of the mesh. Hence,

$$\rho_e^p = \sum_{i=1}^{D} \left(\rho_e^p\right)_i \delta_{\mathbf{x}_i} \quad \text{and} \quad \hat{\rho}_e^p = \sum_{i=1}^{D} \left(\hat{\rho}_e^p\right)_i \delta_{\mathbf{y}_i}. \tag{24}$$

The discrete cost function is defined as

$$C_{i,j} = c(\mathbf{x}_i, \mathbf{y}_j) = \left\|\mathbf{x}_i - \mathbf{y}_j\right\|_2^2. \tag{25}$$

Therefore, the discrete formulation of the Kantorovich Optimal Transport problem reads

$$W\left(\rho_e^p, \hat{\rho}_e^p\right)_2^2 = \min_{\pi \in \Pi(\rho_e^p, \hat{\rho}_e^p)} \sum_{i,j} C_{i,j} \pi_{i,j}, \tag{26}$$

where $\pi_{i,j}$ represents the quantity of mass transported from $\mathbf{x}_i$ towards $\mathbf{y}_j$. The set of transfer plans reads

$$\Pi\left(\rho_e^p, \hat{\rho}_e^p\right) = \left\{ \pi = (\pi_{ij}) \,\middle|\, \sum_j \pi_{ij} = \left(\rho_e^p\right)_i, \sum_i \pi_{ij} = \left(\hat{\rho}_e^p\right)_j \right\} \tag{27}$$

It should be noted that from now on, in order to analyze the three different errors, the value of the error metric presented corresponds to the mean value of the $p \in [\![P_{test}]\!]$ points of the test set. Hence, the three error metrics are noted:

$$\varepsilon_{max}, \quad \varepsilon_{pos} \quad \text{and} \quad W_2^2. \tag{28}$$

4.2. Heat transfer problem

As it has been presented, in the "hybrid twin" rationale there are two data sources: the "virtual twin" numerical simulation and the experimental counterpart. Since the access to experimental data is very expensive and for operational reasons, here the ignorance gap between measurement and prediction is simulated. To this purpose, a physical problem is solved using a finite elements methodology. On the one hand, the problem is solved following the real physics equations. This data is considered to be the "experimental data". On the other hand, a perturbation coefficient, representing the prediction-measurement ignorance, is introduced in the modeling equations before the system resolution. This data is considered to be the "numerical data", i.e. the "virtual twin" simulation.

In this section, the heat equation is solved in a 2 dimensional domain Ω where the thermal conductivity is defined as an-isotropic. On the domain boundary we impose an homogeneous Neumann condition. The initial condition is defined as a Gaussian heat source that diffuses in time. Therefore, the "experimental data" problem writes:

$$\begin{cases} k_x \frac{\partial^2 T}{\partial x^2} + k_y \frac{\partial^2 T}{\partial y^2} = \rho C_P \frac{\partial T}{\partial t} & \text{in } \Omega \times [0, T_f), \\ T(x, y, t=0) = \frac{1}{\sigma^2 2\pi} \exp^{-\frac{(x-s_x)^2 + (y-s_y)^2}{2\sigma^2}} & \text{in } \Omega, \\ \nabla T \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \tag{29}$$

where ρ is the density, C_P is the specific heat, T_f the final time, k_x and k_y the thermal conductivity along the x and y directions respectively and $\mathbf{n}$ the outward normal from Ω. Thus, the parameters defining the parametric space are s_x, s_y, t, k_x and k_y: $\mathcal{W}(s_x, s_y, t, k_x, k_y) \in \mathbb{R}^5$. Then a perturbation coefficient κ is introduced in the equations. Therefore, the "numerical data" problem writes:

$$\begin{cases} \kappa k_x \frac{\partial^2 T}{\partial x^2} + \kappa k_y \frac{\partial^2 T}{\partial y^2} = \frac{\rho C_P}{\kappa} \frac{\partial T}{\partial t} & \text{in } \Omega \times [0, T_f), \\ T(x, y, t=0) = \frac{1}{\sigma^2 2\pi} \exp^{-\frac{(x-\kappa s_x)^2 + (y-\kappa s_y)^2}{2\sigma^2}} & \text{in } \Omega \\ \nabla T \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{30}$$

First, the finite elements resolution of both problems is presented for the test data set. As it can be seen in Fig. 8, the κ coefficient introduces an error between the so identified experimental and numerical data. Here, the value chosen for κ is 0.7. Indeed, it can be noted that the numerical data is left-down moved and less diffused, which is coherent with how the perturbation coefficient is introduced in the problem (30).

The OT-based "digital twin" is trained and applied to the "virtual twin" simulations of the test set. The results are presented in the Fig. 9. It can be observed that the OT-based correction leads to a solution very close to the experimental data. In order to quantify the remaining error between the corrected "virtual twin" simulations and the measurements, the three error metrics are computed. Moreover, these metrics are also computed between the original "virtual twin" and the experimental data. The error values are presented in the Table 1. It can be observed that the original ignorance gap between the "virtual twin" and the experimental measurements is considerably reduced thanks to the OT-based ignorance model.

Next, the influence of the perturbation coefficient κ is explored. The evolution of the three error metrics for a range of values of κ is pre-

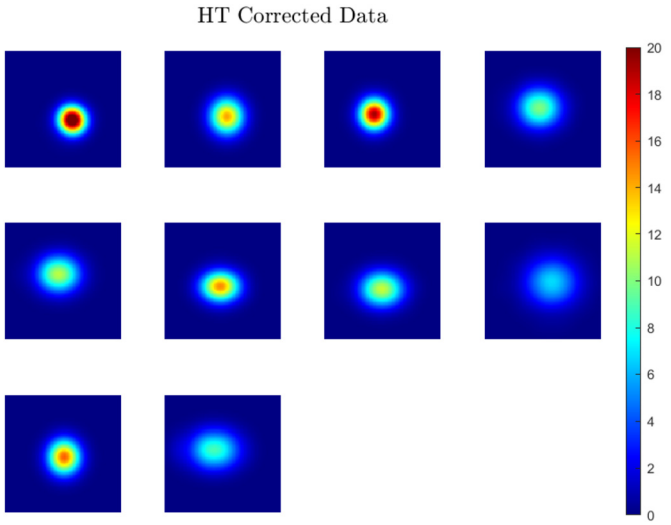


Fig. 9. "Hybrid twin" approach on the test set: "virtual twin" simulations corrected by the OT-based "digital twin".

Table 1
Error metrics for the test set between the experimental and the numerical data, and between the experimental and the OT-based corrected data.

	ϵ_{max} [%]	ϵ_{pos}	W_2^2
Numerical Data	32.4	0.2	0.2
HT Corrected Data	8.1	0.004	0.02

sented in Fig. 10. It can be first observed that the error is null for $\kappa = 1$ since we are comparing a finite element solution with itself. Moreover, note that the three error for both the original and the corrected "virtual twin" simulations are symmetric with respect to 0. This is coherent since the further κ is from 1, the bigger the ignorance gap between simulation and measurement is. Finally, and most importantly, it should be noticed that the corrected ignorance gap is much more smaller than the original simulation-measurement gap, highlighting the performance of the OT-based "hybrid twin" approach.

4.3. Fluid dynamics problem

In this section, the methodology developed is applied to a fluid dynamics problem. Again, since the access to experimental data is very expensive and for operational reasons, the ignorance gap between measurement and prediction is simulated. The same perturbation coefficient strategy, followed in the heat transfer example, is here applied.

A 3D steady laminar jet into a rectangular channel is studied. Indeed, as it is illustrated in the Fig. 11(a), a laminar jet ($Re = 100$) enters a rectangular channel ($\Omega_{channel}$) of length L and base $l \times l$ through a circular hole of radius r (Ω_{inlet}). As indicated in (32), the fluid, considered as incompressible, has a parabolic profile at the inlet domain Ω_{inlet} with a maximum velocity of v_{max}. Moreover, the inlet domain Ω_{inlet} is parameterized by its center x and y coordinates, s_x and s_y respectively, and by its radius r. Therefore, the equation of the circle defining $\Omega_{inlet}(s_x, s_y, r)$ writes:

$$(x - s_x)^2 + (y - s_y)^2 = r^2. \tag{31}$$

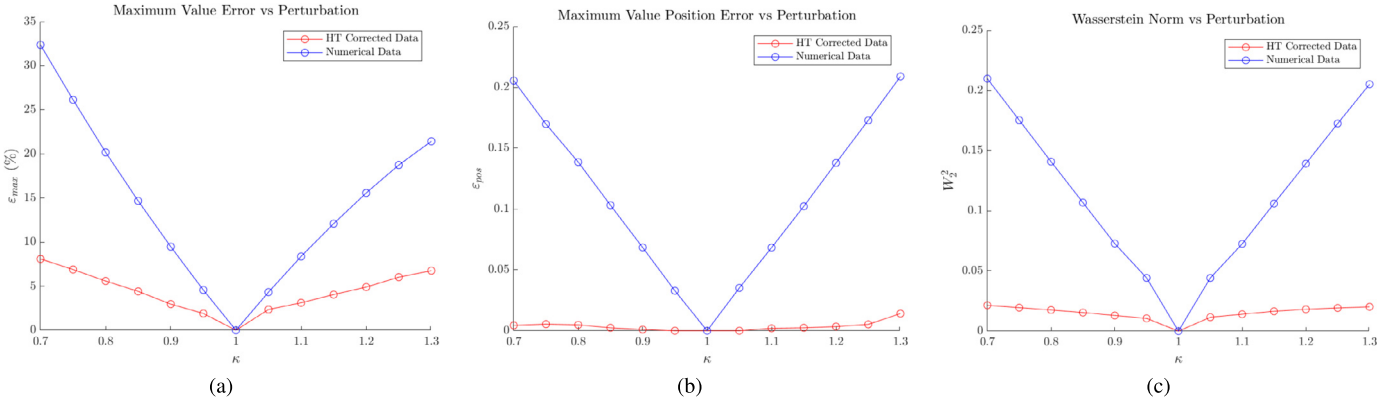
A non slip condition is imposed on the side walls (Ω_{wall}) and on the remaining inlet section wall. Finally, a zero gradient condition is imposed on the outlet section (Ω_{outlet}). Therefore, the "experimental data" problem writes:

$$\begin{cases} \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{v} & \text{in } \Omega_{channel}, \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega_{channel}, \\ \boldsymbol{v}(x, y, z = 0) \\ \quad = v_{max}\left(1 - 0.5\left(\left(\frac{x - s_x}{r}\right)^2 + \left(\frac{y - s_y}{r}\right)^2\right)\right)\boldsymbol{z} & \text{in } \Omega_{inlet}(s_x, s_y, r), \\ \boldsymbol{v} = 0 & \text{on } \Omega_{wall}, \\ \nabla \boldsymbol{v} \cdot \boldsymbol{n} = 0 & \text{on } \Omega_{outlet}, \end{cases} \tag{32}$$

where ρ is the density, ν the kinematic viscosity, $\boldsymbol{n}$ is the outward normal from Ω_{outlet} and $\boldsymbol{z}$ the elementary vector of the z axis. Thus, the parameters defining the parametric space are s_x, s_y, and r: $\mathcal{W}(s_x, s_y, r) \in \mathbb{R}^3$. Then a perturbation coefficient κ is introduced in the problem. This time, the perturbation affects the position of the center of the inlet domain Ω_{inlet} and the maximum velocity of the parabolic inlet profile v_{max}. Therefore, the "numerical data" problem writes:

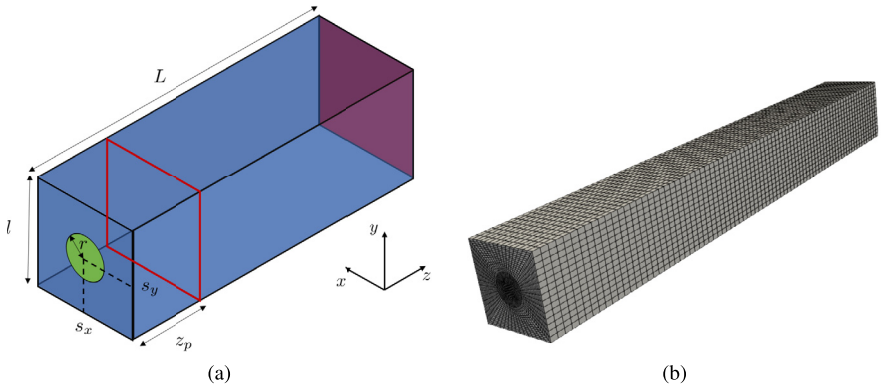
$$\begin{cases} \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{v} & \text{in } \Omega_{channel}, \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega_{channel}, \\ \boldsymbol{v}(x, y, z = 0) \\ \quad = \kappa v_{max}\left(1 - 0.5\left(\left(\frac{x - \kappa s_x}{r}\right)^2 + \left(\frac{y - \kappa s_y}{r}\right)^2\right)\right)\boldsymbol{z} & \text{in } \Omega_{inlet}(\kappa s_x, \kappa s_y, r), \\ \boldsymbol{v} = 0 & \text{on } \Omega_{wall}, \\ \nabla \boldsymbol{v} \cdot \boldsymbol{n} = 0 & \text{on } \Omega_{outlet}. \end{cases} \tag{33}$$

In order to solve both problems, the channel is meshed with an hexahedral mesh as it is shown in the Fig. 11(b). It should be noted that giving the parametric inlet domain $\Omega_{inlet}(s_x, s_y, r)$, a mesh morfing is computed for every point of the design of experiment. Indeed, the shape of the mesh is changed while preserving the connectivity. Only



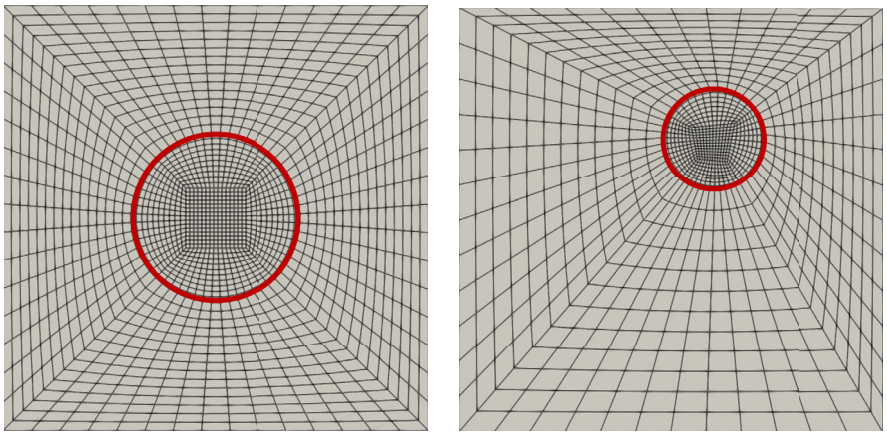
(a) (b) (c)

Fig. 10. Evolution of the three error metrics for a range of values of κ: the error between the original "virtual twin" and the experimental data is plotted in blue and the error between the corrected "virtual twin" and the experimental data is plotted in red.



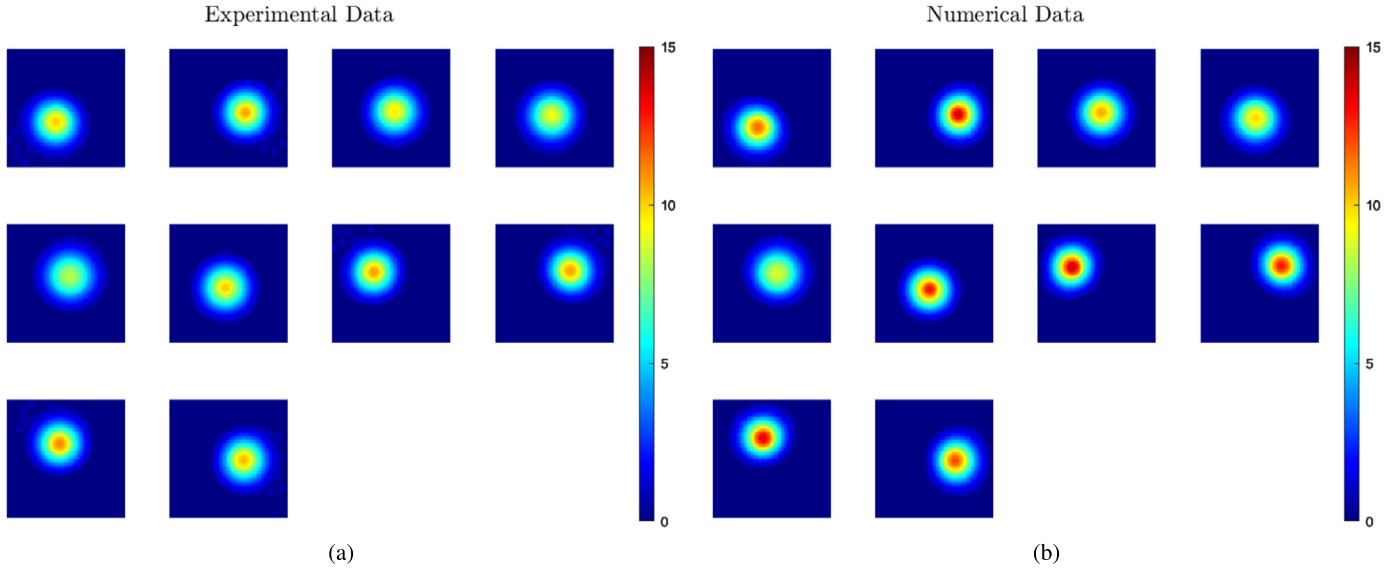
(a) (b)

Fig. 11. (a) Scheme of the rectangular channel geometry: blue surface - Ω_{wall}, green surface - Ω_{inlet}, red surface - Ω_{outlet}, red contour - plane of analysis Ω. (b) Hexahedral mesh of the channel.



(a) (b)

Fig. 12. Mesh of the inlet plane ($z = 0$): (a) Original mesh with Ω_{inlet} at the center of the inlet plane, i.e. $s_x = s_y = 0$, (b) Morfed mesh for a point of the plan of experiment where $s_x \neq 0$, $s_y \neq 0$ and with a different r. The inlet domain Ω_{inlet} has been circled in red.



(a) (b)

Fig. 13. Test set data points: (a) OpenFOAM solution of the problem (32) corresponding to the experimental data. (b) OpenFOAM solution of the problem (33) corresponding to the "virtual twin" simulation.

node positions are updated. An example is showed in the Fig. 12. The Computational Fluid Dynamics OpenFOAM code is used to solve both finite volumes problems. The SimpleFoam solver is selected to solve the Navier-Stokes equations. The z component of the velocity field is monitored on a plane Ω perpendicular to the channel at $z = z_p$, as it is represented in the Fig. 11(a).

First, the resolution of both problems is presented in the plane Ω for the test data set in the Fig. 13. Again, the κ coefficient introduces an

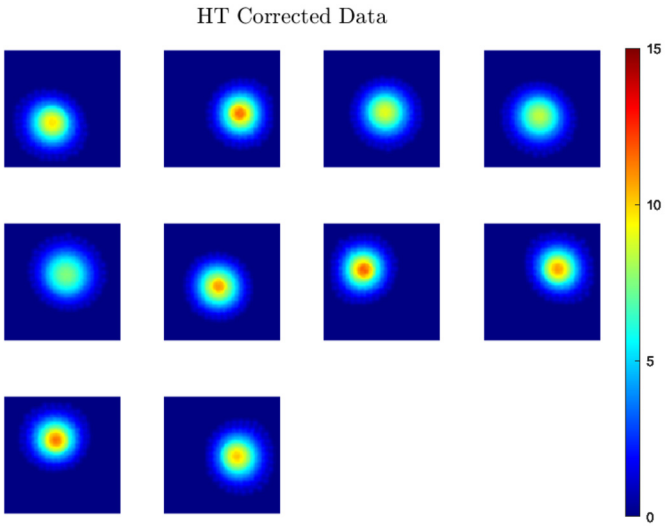


Fig. 14. "Hybrid twin" approach on the test set: "virtual twin" simulations corrected by the OT-based "digital twin".

Table 2
Error metrics for the test set between the experimental and the numerical data, and between the experimental and the OT-based corrected data.

	ϵ_{max} [%]	ϵ_{pos}	W_2^2
Numerical Data	18.2	0.05	0.06
HT Corrected Data	5.6	0.007	0.02

error between the so identified experimental and numerical data. Here, the value chosen for κ is 1.4. Indeed, it can be noted that the numerical data is more distanced from the center of the section and the amplitude of the jet is higher, which is coherent with how the perturbation coefficient is introduced in the problem (33).

Then, the OT-based "digital twin" is trained and applied to the "virtual twin" simulations of the test set. The results are presented in the Fig. 14. It can be observed that the OT-based correction leads to a solution very close to the experimental data. Again, in order to quantify the error, the three error metrics are applied between the original numerical simulations and the experimental data and between the corrected "virtual twin" and the measurements. The error values are presented in the Table 2. The same remark as for the heat problem can be made, the original ignorance gap between the "virtual twin" and the experimental measurements is considerably reduced thanks to the OT-based ignorance model.

5. Conclusion

Data is being widely used in all fields including engineering. From its initial calibration role, data has acquired a much more important role training models able to replace complex engineering systems. To train those models engineers wish to use as much data as possible and as accurately as possible, but this becomes really expensive in many domains. The "hybrid twin" methodology brings a solution to this problem by correcting numerical data to be closer to measurement data. However, in fields such as fluid dynamics the classical "hybrid twin" approach leads to non physical results. Combining the "hybrid twin" rationale with the simplified Optimal Transport Monge problem, our approach leads to an OT-based "digital twin" able to correct "virtual twin" simulations. The SPH decomposition of both the experimental and "virtual twin" data is the key step of our approach, which allows to compute and interpolate the simulation-measurement gap from an Optimal Transport point of view. Therefore, the proposed OT-based "hybrid twin" methodology can correct numerical simulations giving solutions very close to the measurement counterpart data and leading, thus, to a faster and cheaper access to data almost as much accurate as experimental solutions. Finally, for operational reasons and since the access to experimental data is very expensive, the ignorance gap between measurement and prediction for the assessment examples has been simulated. Indeed, the main goal of this paper was to introduce the OT-based "hybrid twin" methodology. However, as noted before, further work is planned and in progress to apply this approach to a real industrial case.

Acknowledgement

We thank Angelo Pasquale of PIMM Laboratory at Arts et Métiers Institute of Technology for providing helpful assistance and advice on the CFD simulations with OpenFOAM.

Appendix A. Partial least squares regression: NIPALS algorithm

Let us introduce some labeled data of, for instance, a complex engineering system. This data is collected in two matrices, X and Y, where each row represents an observation of the system (e.g. at different points in time) and each column a property of the measurement, called variable. The variables in X, called explanatory variables, are supposed to be always available from the system while the variables in Y, called response variables, are not always available and are the ones we want to predict in the future. Let us suppose that there are N observations of the system, K explanatory variables and M response variables.

Principal Components Analysis (PCA) is a methodology consisting of rewriting a rank r matrix X as a sum of r rank 1 matrices. These matrices of rank 1, M_h, can be written as outer products of two vectors, a score one t_h and a loading one p_h^T:

$$X = t_1 p_1^T + \ldots + t_r p_r^T, \tag{34}$$

or in the equivalent form $X = TP^T$, where P^T is built with the loading vectors p_h^T as rows and T with the score vectors t_h as columns. The loading vectors represent the direction vectors of the best-fit lines of the data, i.e. the lines that best explain all the observations with minimum error. These loading vectors are orthonormal. The score vectors are the distances from the origin to the projections of the observations onto these lines [14]. A score/loading couple is also called a latent variable or principal component.

The Non-linear Iterative Partial Last Squares (NIPALS) algorithm is a sequential methodology of determining the principal components [12,16]. The NIPALS method sequentially extracts each component, from the first component, direction of greatest variance, until the user considers that enough components are computed.

Indeed, from the X matrix, the algorithm calculates t_1 and p_1^T. Then the outer product, $t_1 p_1^T$, is subtracted from X and the residual E_1 is determined, which is used to compute t_2 and p_2^T and so on:

$$E_h = E_{h-1} - t_h p_h^T. \tag{35}$$

The NIPALS algorithm follows the next steps:

1. take a column vector x_j from X and call it t_h: $t_h = x_j$
2. calculate p_h^T: $p_h^T = t_h^T X / t_h^T t_h$
3. normalize p_h^T to length 1: $p_h^T = p_h^T / \left\| p_h^T \right\|$
4. calculate t_h: $t_h = X p_h / p_h^T p_h$
5. compare the t_h used in step 2 with the one obtained in step 4. The iteration has converged if they are equal. If not, go again to step 2. Note that once the hth component is determined, the X matrix in 2nd and 4th steps must be substituted by its residual E_{h+1}.

It can be noted that NIPALS gives the same solution than the one resulting from the eigenvector formulae. Nevertheless, introducing the NIPALS methodology is necessary for a good understanding of PLS.

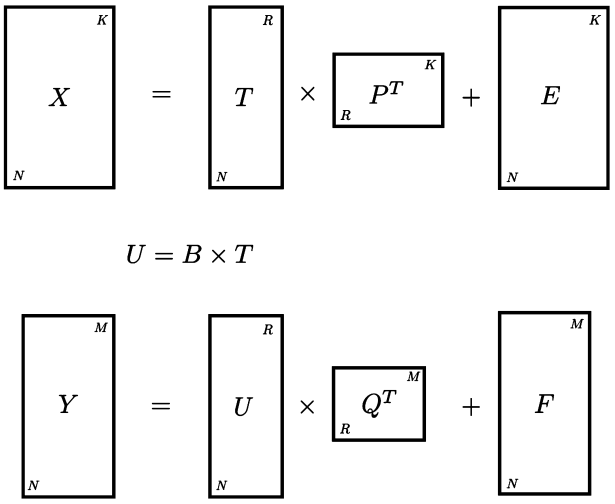


Fig. 15. Partial Least-Squares regression conceptual scheme where N is the number of observations, K the number of explanatory variables, M the number of response variables and R the number of components retained.

Indeed, the PLS model relies on the properties of the NIPALS algorithm. The PLS model consists in outer relations (for the X and Y blocks individually) and an inner relation (linking both blocks). The outer relation for the X block reads:

$$X = TP^T + E, \tag{36}$$

where T and P^T are the sets of scores and loading for the X-space respectively and E the residual matrix. Then, the outer relation for the Y block is:

$$Y = UQ^T + F, \tag{37}$$

where U and Q^T are the sets of scores and loading for the Y-space respectively and F the residual matrix. Finally, the inner relation is a linear one between the scores U and T through the diagonal matrix B:

$$U = BT. \tag{38}$$

Note that X and Y data matrices are assumed scaled and mean-centered. These three relations are graphically presented in Fig. 15 and represent the three simultaneous objectives of the PLS model:

- Best explain the X-space.
- Best explain the Y-space.
- Get the strongest relationship between the X and Y-space.

Indeed, the PLS method extracts the scores U and T such that they have maximal covariance

$$\mathrm{Cov}(t_h, u_h) = \overline{(t_h - \overline{t_h})\,(u_h - \overline{u_h})}, \tag{39}$$

which can be rewritten as:

$$\mathrm{Cov}(t_h, u_h) = \mathrm{Correlation}(t_h, u_h) \times \sqrt{t_h^T t_h} \times \sqrt{u_h^T u_h}, \tag{40}$$

where it can be observed that maximizing the covariance between scores is equivalent to maximizing the three previous objectives.

Therefore, it is important to note that, since the principal components are separately determined for the two spaces, a weak relation will exist between each other. Thus, in order to give information about each other, PLS regression applies the NIPALS method with exchanged scores. Indeed, instead of applying the NIPALS algorithm A to each block and then build a regression between the scores of each space, the PLS-NIPALS method exchanges t_h and u_h in step 2. Moreover, since the calculations order used for the PCA has been modified, the algorithm does not give orthogonal components anymore. To solve this issue, the p_h^T are first replaced by weights w_h^T and an extra loop is included after convergence. The modified NIPALS algorithm with exchanged scores writes:

1. take a column vector y_j from Y and call it u_h: $u_h = y_j$
2. calculate w_h^T: $w_h^T = u_h^T X / u_h^T u_h$
3. normalize w_h^T to length 1: $w_h^T = w_h^T / \left\| w_h^T \right\|$
4. calculate t_h: $t_h = X w_h / w_h^T w_h$
5. calculate q_h^T: $q_h^T = t_h^T Y / t_h^T t_h$
6. normalize q_h^T to length 1: $q_h^T = q_h^T / \left\| q_h^T \right\|$
7. calculate u_h: $u_h = Y q_h / q_h^T q_h$
8. compare the t_h in 4th step with the one in the preceding iteration step. The algorithm has converged if they are equal (within a certain tolerance). If not, go to step 2.
9. calculate X loadings: $p_h^T = t_h^T X / t_h^T t_h$
10. normalize p_h^T: $p_h^T = p_h^T / \left\| p_h^T \right\|$
11. normalize t_h^T: $t_h^T = t_h^T / \left\| p_h^T \right\|$
12. normalize w_h^T: $w_h^T = w_h^T / \left\| p_h^T \right\|$
13. find the regression coefficient b_h for the inner relation: $b_h = u_h^T t_h / t_h^T th$

It can be noted that if the Y block consists only in one variable, steps 5 to 8 can be omitted by fixing $q_h = 1$. Moreover, once the hth component is calculated, X and Y must be replaced by its respective residuals E_{h+1} and F_{h+1}:

$$\begin{aligned} E_h &= E_{h-1} - t_h p_h^T, E_0 = X \\ F_h &= F_{h-1} - b_h t_h q_h^T, F_0 = Y \end{aligned} \tag{41}$$

In practice, the number of components retained R can be chosen, for instance, by analyzing the percentage of variance explained in the response variable as a function of the number of latent variables.

Finally, in the prediction stage, for a new X block the scores are extracted using weights W and loadings P

$$\begin{aligned} \hat{t}_h &= E_{h-1} w_h \\ E_h &= E_{h-1} - \hat{t}_h p_h^T, \end{aligned} \tag{42}$$

and the new block Y is predicted

$$\hat{Y} = \sum b_h \hat{t}_h q_h^T, \tag{43}$$

where the sum is over h and for all the principal components that one wants to keep. The PLS prediction model can also be written as:

$$\hat{Y} = XW(P^T W)^{-1} \mathrm{diag}(b) Q^T. \tag{44}$$

References

[1] H. Abdi, Partial least squares (pls) regression, in: Encyclopedia of Social Sciences Research Methods, 2003.
[2] M. Agueh, G. Carlier, Barycenters in the Wasserstein space, SIAM J. Math. Anal. 43 (2) (2011).
[3] J.D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem, Numer. Math. 84 (3) (2000) 375–393.
[4] N. Bonneel, M. van de Panne, S. Paris, W. Heidrich, Displacement interpolation using Lagrangian mass transport, ACM Trans. Graph. 30 (6) (2011) 1–12.
[5] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, Society for Industrial and App. Math, 2009.
[6] L. Carius, R. Findeisen, The impact of experimental data quality on computational systems biology and engineering, IFAC-PapersOnLine 49 (26) (2016) 140–146.
[7] Q. Chatenet, A. Tahan, M. Gagnon, J. Chamberland-Lauzon, Numerical model validation using experimental data: application of the area metric on a Francis runner, IOP Conf. Ser. Earth Environ. Sci. 49 (2016).
[8] B. Chen, G. Becigneul, O.E. Ganea, R. Barzilay, T. Jaakkola, Optimal transport graph neural networks, arXiv preprint, arXiv:2006.04804, 2020.
[9] F. Chinesta, E.G. 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. (2019).

[10] E.G. Cueto, D. Gonzalez, A. Badias, F. Chinesta, N. Hascoet, J.L. Duval, Hybrid twins. Part ii. Real-time, data-driven modeling, in: International ESAFORM Conference on Material Forming, 2021.
[11] M. Cuturi, Sinkhorn distances: lightspeed computation of optimal transport, Adv. Neural Inf. Process. Syst. (2013).
[12] S. de Jong Simpls, An alternative approach to partial least squares regression, Chemom. Intell. Lab. Syst. 18 (3) (1993).
[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] K. Dunn, Process Improvement Using Data, 2021.
[15] C.J. Freitas, The issue of numerical uncertainty, Appl. Math. Model. 26 (2002) 237–248.
[16] P. Geladi, B.R. Kowalski, Partial least-squares regression: a tutorial, Anal. Chim. Acta 185 (1986) 1–17.
[17] T. Hey, S. Tansley, K. Tolle, The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, Redmond, 2009.
[18] F.S. Hillier, G.J. Lieberman, Introduction to Operations Research, McGraw-Hill, 1990.
[19] L. Kantorovich, On the transfer of masses, Dokl. Akad. Nauk 37 (2) (1942) 227–229 (in Russian).
[20] S. Lind, B. Rogers, P. Stansby, Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling, Proc. R. Soc. A 476 (2020).
[21] Y. Lu, J. Lu, A universal approximation theorem of deep neural networks for expressing probability distributions, arXiv preprint, arXiv:2004.08867, 2020.
[22] B. Lévy, E.L. Schwindt, Notions of optimal transport theory and how to implement them on a computer, Comput. Graph. 72 (2018) 135–148.
[23] A.V. Makkuva, A. Taghvaei, J.D. Lee, S. Oh, Optimal transport mapping via input convex neural networks, arXiv preprint, arXiv:1908.10962, 2019.
[24] R.J. McCann, A convexity principle for interacting gases, Adv. Math. 128 (1997) 153–179.
[25] A. Mensch, G. Peyré, Online Sinkhorn: Optimal Transport Distances from Sample Streams, vol. 33, Curran Associates, Inc., 2020.
[26] G. Monge, Mémoire sur la théorie des déblais et des remblais, Histoire de l'Académie Royale des Sciences de Paris, 1781, pp. 666–704.
[27] Q. Mérigot, A multiscale approach to optimal transport, Comput. Graph. Forum 30 (5) (2011).
[28] W.L. Oberkampf, S.M. DeLand, B.M. Rutherford, K.V. Diegert, K.F. Alvin, Error and uncertainty in modeling and simulation, Reliab. Eng. Syst. Saf. 75 (2002) 333–357.
[29] G. Peyré, M. Cuturi, Computational optimal transport, Found. Trends Mach. Learn. 11 (5–6) (2019) 355–607.
[30] R. Sinkhorn, A relationship between arbitrary positive matrices and doubly stochastic matrices, Ann. Math. Stat. 35 (1964).
[31] V. Timchenko, S.A. Tkachenko, J. Reizes, G.E. Lau, G.H. Yeoh, Is comparison with experimental data a reasonable method of validating computational models?, J. Phys. Conf. Ser. 745 (2016).
[32] R.D. Tobias, An introduction to partial least squares regression, 1996.
[33] S. Torregrosa, V. Champaney, A. Ammar, V. Herbert, F. Chinesta, Surrogate parametric metamodel based on optimal transport, Math. Comput. Simul. (2021).
[34] C. Villani, Topics in Optimal Transportation, vol. 58, American Mathematical Soc., 2003.
[35] C. Villani, Optimal Transport, Old and New, Springer, 2006.
[36] J. Weed, F. Bach, Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance, Bernouilli 25 (4A) (2019) 2620–2648.