

WASSERSTEIN MODEL REDUCTION APPROACH FOR PARAMETRIZED FLOW PROBLEMS IN POROUS MEDIA

BEATRICE BATTISTI¹, TOBIAS BLICKHAN², GUILLAUME ENCHERY³, VIRGINIE EHLRACHER⁴, DAMIANO LOMBARDI⁵ AND OLGA MULA⁶

Abstract. The aim of this work is to build a reduced order model for parametrized porous media equations. The main challenge of this type of problems is that the Kolmogorov width of the solution manifold typically decays quite slowly and thus makes usual linear model order reduction methods inappropriate. In this work, we investigate an adaptation of the methodology proposed in [Ehrlicher et al., *Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces*, ESAIM: Mathematical Modelling and Numerical Analysis (2020)], based on the use of Wasserstein barycenters [Agueh & Carlier, *Barycenters in the Wasserstein Space*, SIAM Journal on Mathematical Analysis (2011)], to the case of non-conservative problems. Numerical examples in one-dimensional test cases illustrate the advantages and limitations of this approach and suggest further research directions that we intend to explore in the future.

Résumé. Le but de ce travail est de construire un modèle réduit pour des problèmes d'écoulements en milieux poreux paramétrés. La difficulté principale de ce type de problèmes est que la distance de Kolmogorov de l'ensemble de solutions décroît lentement, rendant ainsi les méthodes de réduction de modèles linéaires usuelles inefficaces. Ici, nous proposons une adaptation de la méthodologie proposée dans [Ehrlicher et al., *Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces*, ESAIM: Mathematical Modelling and Numerical Analysis (2020)], utilisant des barycentres de Wasserstein [Agueh & Carlier, *Barycenters in the Wasserstein Space*, SIAM Journal on Mathematical Analysis (2011)], au cas de problèmes non conservatifs. Des tests numériques en dimension 1 permettent d'illustrer les avantages et les limitations de cette approche et d'identifier des pistes de recherche que nous souhaiterons aborder dans un futur travail.

INTRODUCTION

Model-order reduction methods have proved to be extremely useful tools in order to accelerate parametric studies in various contexts. Given a Partial Differential Equation (PDE) involving parameters, the main goal of model-order reduction is to provide a fast approximation of the parameter-to-solution map. Classical strategies are linear in the sense that they rely on approximating the set of solutions with linear or affine spaces. This approach is the backbone of most existing methods among which stand the reduced basis method (see [22, 30]),

¹ Politecnico di Torino, Université de Bordeaux

² Max-Planck-Institute for Plasma Physics & Technische Universität München

³ IFPEN

⁴ Ecole Nationale des Ponts et Chaussées INRIA

⁵ INRIA

⁶ Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600 MB, Eindhoven, Netherlands.

the empirical interpolation method and its generalized version (G-EIM, see [4, 21, 25, 26]), Principal Component Analysis (PCA, see [5, Chapter 1]), polynomial-based methods like [13, 14] or low-rank methods [23]. The efficiency of such approaches is linked to the decay of the so-called Kolmogorov width of the solution manifold, which informs about the best possible approximation performance of the set of solutions when working with linear spaces. In cases where this width decays quickly with the dimension of the reduced space, significant gains in computational time can be achieved with respect to working with a direct, full-order computation of the solutions with classical discretization methods such as finite elements, or finite volumes. Families of PDEs where the Kolmogorov width presents a fast decay are elliptic or parabolic problems (see [12]).

However, in situations where the decay of the Kolmogorov width is slow with respect to the dimension of the reduced space, linear model-order reduction methods are inherently doomed to fail. This is the case, for instance, of parametrized flow problems in porous medium equations, which is the object of the present work. The acceleration of parametric studies of this type of problems requires the use of *nonlinear* model-order reduction methods. This direction is a very active research topic, and numerous nonlinear strategies have been proposed (see [1, 2, 11, 20, 24, 28] to name a few). To deal with advection dominated problems, some studies rely on snapshots remapping: [10, 27, 35, 38]. In this work, we consider an adaptation of a methodology based on the use of Wasserstein barycenters, which was originally proposed in [16] for the reduction of conservative flow problems (see also [6], [33] for related work in computer graphics and computational anatomy). Note that the use of Wasserstein metrics to build reduced-order models for transport-dominated problems was also investigated in [31]. In our present context, we wish to investigate methods for the reduction of parametrized flow problems in porous media problems, which are not conservative. The fact that the total mass of each fluid phase in the system of interest may vary with respect to time and other parameter values yields a difficulty compared to the cases which were studied in [16], where only conservative flows were considered. We propose here an approach in order to circumvent them. In this preliminary work, we only investigate one-dimensional problems, and leave the extension of the approach for higher-dimensional problems for future research. Note that, for conservative flows, a higher dimensional extension is investigated in [15].

The paper is structured as follows: first, we present the parametrized two-phase flow problem in porous media which we focus on in this work in Section 1. Then, we recall some basic properties of Wasserstein spaces and Wasserstein barycenters in dimension 1 in Section 2 and we describe the proposed model-order reduction methodology in Section 4. Finally, numerical results illustrating the strengths and weaknesses of our approach are shown in Section 5.

1. PROBLEM SETTING: TWO-PHASE FLOW PROBLEM IN POROUS MEDIA

1.1. The physical model

We consider a simplified model of two-phase flow through a porous rock. This type of model is often used in geosciences to describe the displacement of two non-miscible fluid phases. The wetting phase often corresponds to the water component whereas the second one depends on the application: air in hydrogeology, CO₂ for underground gas storage, etc.

In the following, we denote by $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}^*$ the spatial domain occupied by the porous medium. The domain Ω has a total volume

$$V_t = V_v + V_s > 0$$

which is the sum of the volume V_v occupied by the void, and the volume V_s occupied by the solid material. We can thus characterize the medium by its porosity

$$\phi := \frac{V_v}{V_t} \in [0, 1]$$

which is the ratio of the volume of void space V_v over the total volume of the material V_t . We assume here that the porosity of the medium is constant in time and characterized by a function $\phi : \Omega \rightarrow [0, 1]$. The volume

V_v of the void space is here assumed to be completely saturated by two incompressible and non-miscible fluid phases, called wetting (w) and non-wetting (nw), with respective volumes V_w and V_{nw} which are such that

$$V_v = V_w + V_{nw}.$$

Their volume fraction is given by the saturation

$$s_\alpha := V_\alpha/V_v, \quad \alpha \in \{w, nw\}$$

and we have that

$$s_{nw} + s_w = 1.$$

Due to incoming/outcoming flows from outside the domain Ω (runoff waters from the surface, injecting/pumping wells, etc.), both saturations s_{nw} and s_w depend on space and time. For a given time interval $[0, T]$, they are solutions to an equation stating the conservation of their phase volume,

$$\phi(x) \partial_t s_\alpha(x, t) + \nabla \cdot v_\alpha(x, t) = 0, \quad \forall (x, t) \in \Omega \times (0, T), \quad \forall \alpha \in \{w, nw\}. \quad (1)$$

In equation (1), v_α denotes the generalized Darcy velocity. Neglecting the gravity and capillary effects, the velocity is defined for all $(x, t) \in \Omega \times (0, T)$ as

$$v_\alpha(x, t) := -\lambda_\alpha(x, t) K(x) \nabla P(x, t), \quad \forall \alpha \in \{w, nw\}, \quad (2)$$

where P is the fluid pressure. The pressure depends on space and time, and it is assumed to be the same for both phases. The coefficient $K : \Omega \rightarrow \mathbb{R}_+$ is the absolute permeability of the porous medium and reflects the ability of one fluid to flow through that medium. The phase mobility λ_α is also involved in the definition of the Darcy velocity. It is defined as

$$\lambda_\alpha(x, t) := \frac{k_{r_\alpha}(x, t)}{\mu_\alpha}$$

where $\mu_\alpha \in \mathbb{R}_+$ is the phase viscosity. k_{r_α} is the relative permeability which is an increasing function of s_α with values in $[0, 1]$. In hydrogeology or in reservoir engineering applications, this law is often defined thanks to the Brooks-Corey model [9], which we simplify here to a power law

$$k_{r_\alpha}(x, t) = s_\alpha^\beta(x, t),$$

with $\beta > 0$. The dependency of k_{r_α} on s_α induces a nonlinear coupling between equations (1) and equation (2) for the unknowns are s_w , s_{nw} and p . In this work, the system is closed with boundary conditions

$$P(t, x) = P_D(t, x), \quad \forall (x, t) \in \partial\Omega \times [0, T], \quad (3)$$

$$s_w(t, x) = 1, \quad \forall (x, t) \in \partial\Omega \times [0, T] \quad \text{s.t.} \quad v(t, x) \cdot n(x) < 0, \quad (4)$$

where n is the outward unit normal of $\partial\Omega$, and initial conditions

$$s_w(x, t = 0) = s_w^0, \quad \forall x \in \Omega. \quad (5)$$

In general, the problem is often reformulated in terms of pressure P and the saturation s_w . By introducing the total Darcy velocity

$$v := v_w + v_{nw} = -(\lambda_w + \lambda_{nw})k\nabla P, \quad (6)$$

the previous system of equations is equivalent to

$$\begin{cases} \operatorname{div}(v) = 0 \\ \phi \partial_t s_w + \nabla \cdot (f_w(s_w)v) = 0 \\ s_{nw} = 1 - s_w, \end{cases} \quad (7)$$

where

$$f_\alpha(s_\alpha) := \frac{\lambda_\alpha(s_\alpha)}{\lambda_w(s_w) + \lambda_{nw}(s_{nw})}, \quad \alpha \in \{w, nw\}$$

is the fractional flow.

Problem (7) with conditions (3)-(4)-(5) is solved using a finite-volume discretization in space and an IMPES (Implicit for Pressure, Explicit for Saturation) scheme in time. This latter scheme corresponds to an Euler implicit scheme for the pressure in the first equation of (7) and to an explicit scheme for the saturation in the second equation of (7) where the phase mobilities in (6) are evaluated explicitly too. A similar discrete system, made of a two-point scheme for the pressure gradient and an upwind scheme for the saturation, has already been considered in [8] to study the reductibility of the first equation of (7) with respect to some rock and fluid properties. The only difference between the present discretization and that study lies in the time discretization used for the saturation equation, which was implicit in this previous work and could allow the use of larger time steps. We refer to that work and to the references cited therein for more details on these discretizations. The IMPES scheme enables to decouple the resolution of (7), starting from the pressure, updating the total Darcy fluxes and finally computing the saturations.

1.2. Parametric Problem and Manifold set of solutions

Let $\mathcal{Y} \subset \mathbb{R}^{n_p}$ be a compact set of parameter values for some $n_p \in \mathbb{N}^*$. For all $y \in \mathcal{Y}$, let $K^y : \Omega \rightarrow \mathbb{R}_+$, $\phi^y : \Omega \rightarrow [0, 1]$, $\beta^y \in \mathbb{R}_+^*$ and $\mu_\alpha^y \in \mathbb{R}_+$ for $\alpha \in \{w, nw\}$. Moreover, let us denote by $s^y = s_w^y$ the solution s_w to system (7) with data $K = K^y$, $\phi = \phi^y$, $\beta = \beta^y$ and $\mu_\alpha = \mu_\alpha^y$.

One may need, for uncertainty quantification for instance, to evaluate solutions to (7) for many instances of parameter values $y \in \mathcal{Y}$, and thus to reduce the computational time to evaluate the parameter-to-solution map because solving (7) may be too computationally demanding.

More precisely, we wish to approximate the set of solutions

$$\mathcal{M} := \{s^y(\cdot, t) : (y, t) \in \mathcal{Y} \times [0, T]\}. \quad (8)$$

Note that time is treated as parameter in \mathcal{M} . Thus defining $\mathcal{Z} := \mathcal{Y} \times [0, T]$ and denoting by s^z the function $s^y(t, \cdot)$ for all $z = (y, t) \in \mathcal{Z}$, the solution set \mathcal{M} can be rewritten as

$$\mathcal{M} := \{s^z : z \in \mathcal{Z}\}. \quad (9)$$

In the following, we develop a model reduction approach to approximate elements of the solution set \mathcal{M} based on the use of Wasserstein barycenters. For the sake of simplicity, we will assume from now on that $d = 1$ with one-dimensional problems, for which certain computations in this space are greatly facilitated and that $\Omega = (x_{\min}, x_{\max}) \subset \mathbb{R}$, with $-\infty \leq x_{\min} < x_{\max} \leq \infty$. Section 2 presents the essential notions of Wasserstein spaces and barycenters, and Section 4 presents our model reduction algorithm. Note that our proposed approach enables to deal with parametric solutions of the PDE of interest s^z that belong to $L^1(\Omega)$ (and not necessarily $L^2(\Omega)$ or $L^\infty(\Omega)$).

2. WASSERSTEIN BARYCENTERS IN DIMENSION 1

In this section, we recall some basic properties of the L^2 -Wasserstein space in dimension 1 and introduce the notion of Wasserstein barycenters.

2.1. Definition of the L^2 -Wasserstein space in one dimension

Let $\mathcal{P}_2(\Omega)$ denote the set of probability measures on $\Omega \subseteq \mathbb{R}$ with finite second-order moments. For all $u \in \mathcal{P}_2(\Omega)$, we denote

$$\text{cdf}_u : \begin{cases} \Omega & \rightarrow & [0, 1] \\ x & \mapsto & \text{cdf}_u(x) := \int_{x_{\min}}^x du \end{cases} \quad (10)$$

its cumulative distribution function (cdf), and

$$\text{icdf}_u : \begin{cases} [0, 1] & \rightarrow \Omega \\ p & \mapsto \text{cdf}_u^{-1}(p) := \inf\{x \in \Omega, \text{cdf}_u(x) \geq p\} \end{cases} \quad (11)$$

the generalized inverse of the cdf (icdf). The L^2 -Wasserstein distance is defined by

$$W_2(u, v) := \inf_{\pi \in \Pi(u, v)} \left(\int_{\Omega \times \Omega} (x - y)^2 d\pi(x, y) \right)^{1/2}, \quad \forall (u, v) \in \mathcal{P}_2(\Omega) \times \mathcal{P}_2(\Omega),$$

where $\Pi(u, v)$ is the set of probability measures on $\Omega \times \Omega$ with marginals u and v . In the particular case of one dimensional marginal domains, it can be equivalently expressed using the inverse cumulative distribution functions as

$$W_2(u, v) = \|\text{icdf}_u - \text{icdf}_v\|_{L^2([0,1])}. \quad (12)$$

The space $\mathcal{P}_2(\Omega)$ endowed with the distance W_2 is a metric space, usually called L^2 -Wasserstein space (see [36] for more details).

2.2. Wasserstein barycenters

Let $n \in \mathbb{N}^*$ and let

$$\Sigma_n := \left\{ (\lambda_1, \dots, \lambda_n) \in [0, 1]^n, \sum_{i=1}^n \lambda_i = 1 \right\}$$

be the probability simplex of dimension $n - 1$.

For any family of probability measures $U_n = (u_i)_{1 \leq i \leq n} \in \mathcal{P}_2(\Omega)^n$ and barycentric weights $\Lambda_n = (\lambda_i)_{1 \leq i \leq n} \in \Sigma_n$, there exists a unique minimizer to

$$\text{Bar}(U_n, \Lambda_n) = \arg \min_{v \in \mathcal{P}_2(\Omega)} \sum_{i=1}^n \lambda_i W_2(v, u_i)^2, \quad (13)$$

which is the barycenter with respect to U_n and Λ_n . The barycenter can be easily characterized in terms of its inverse cumulative distribution function since (13) implies that

$$\text{icdf}_{\text{Bar}(U_n, \Lambda_n)} = \underset{f \in L^2([0,1])}{\text{argmin}} \sum_{i=1}^n \lambda_i \|\text{icdf}_{u_i} - f\|_{L^2([0,1])}^2, \quad (14)$$

which yields

$$\text{icdf}_{\text{Bar}(U_n, \Lambda_n)} = \sum_{i=1}^n \lambda_i \text{icdf}_{u_i}. \quad (15)$$

For any given function $u \in \mathcal{P}_2(\Omega)$, one can define a notion of best barycenter $b(u, U_n) \in \mathcal{P}_2(\Omega)$ approximating u among the family of barycenters

$$\mathcal{B}(U_n) = \{\text{Bar}(\Lambda_n, U_n) : \Lambda_n \in \Sigma_n\}.$$

The best barycenter is characterized in terms of its optimal weights

$$\Lambda_n^{\text{opt}} = \arg \min_{\Lambda_n \in \Sigma_n} W_2^2(u, \text{Bar}(U_n, \Lambda_n)), \quad (16)$$

which are sometimes called the *barycentric weights*. This allows us to define the optimal barycenter as

$$b(u, U_n) = \text{Bar}(U_n, \Lambda_n^{\text{opt}}).$$

It can be easily characterized in terms of its inverse cumulative distribution function. This is due to the fact that for all $\Lambda_n \in \Sigma_n$ and all $u \in \mathcal{P}_2(\Omega)$,

$$W_2^2(u, \text{Bar}(U_n, \Lambda_n)) = \left\| \text{icdf}_u - \sum_{i=1}^n \lambda_i \text{icdf}_{u_i} \right\|_{L^2([0,1])}^2$$

so the computation of the optimal weights Λ_n^{opt} in problem (16) is a simple cone quadratic optimization problem. Note that at least one minimizer Λ_n^{opt} exists but uniqueness is not guaranteed.

3. THE GREEDY BARYCENTER ALGORITHM

We detail in this section the greedy barycenter algorithm we use here so as to build a reduced-order model to efficiently compute approximations of elements of \mathcal{M} .

3.1. Offline phase: greedy algorithm

For all $s \in \mathcal{M}$, we denote by $m(s) := \int_{\Omega} s$ and by

$$\mathcal{U} := \left\{ u^z := \frac{s^z}{m(s^z)}, z \in \mathcal{Z} \right\}.$$

From now on, we make the assumption that $\mathcal{U} \subset \mathcal{P}_2(\Omega)$ and denote by $f^z := \text{icdf}_{u^z} \in L^2(0, 1)$ for all $z \in \mathcal{Z}$.

The Greedy Barycenter algorithm is an iterative procedure which aims at selecting, after n iterations of the algorithm, a family of parameters $(z_1, \dots, z_n) \subset \mathcal{Z}$, and hence a family $U_n = (u^{z_i})_{1 \leq i \leq n} \subset \mathcal{U}$, such that any element $u \in \mathcal{U}$ can be approximated with good accuracy in the Wasserstein norm by a Wasserstein barycenter $\text{Bar}(U_n, \Lambda_n)$ for some $\Lambda_n \in \Sigma_n$.

Let $\mathcal{Z}_{\text{train}} \subseteq \mathcal{Z}$ be a finite subset of \mathcal{Z} and let $\mathcal{U}_{\text{train}} := \{u^z : z \in \mathcal{Z}_{\text{train}}\}$ and

$$\mathcal{F}_{\text{train}} := \{f^z = \text{icdf}_{u^z} \in L^2(0, 1) : z \in \mathcal{Z}_{\text{train}}\}. \quad (17)$$

In practice, the strategy consists in finding a good collection of *atoms* $A_n = (a_i := f^{z_i})_{1 \leq i \leq n}$ such that for all $z \in \mathcal{Z}$, $f^z = \text{icdf}_{u^z}$ is approximated with good accuracy in the $L^2(0, 1)$ norm by a convex combination of elements of A_n , following similar guidelines as in [16].

Greedy Barycenter Algorithm:

- **Initialisation:** find $(z_1, z_2) \in \mathcal{Z}_{\text{train}}$ solution to

$$(z_1, z_2) \in \operatorname{argmax}_{z, z' \in \mathcal{Z}_{\text{train}}} \left\| f^z - f^{z'} \right\|_{L^2(0,1)}^2.$$

Set $U_2 := (u^{z_1}, u^{z_2})$ and $A_2 := (f^{z_1}, f^{z_2})$.

- **Iteration $n \geq 3$:** find $z_n \in \mathcal{Z}_{\text{train}}$ such that

$$z_n \in \operatorname{argmax}_{z \in \mathcal{Z}_{\text{train}}} \min_{(\lambda_i)_{1 \leq i \leq n-1} \in \Sigma_{n-1}} \left\| f^z - \sum_{i=1}^{n-1} \lambda_i f^{z_i} \right\|_{L^2(0,1)}^2. \quad (18)$$

Set $U_n := (u^{z_1}, u^{z_2}, \dots, u^{z_n})$ and $A_n := (f^{z_1}, f^{z_2}, \dots, f^{z_n})$.

The obtained set of atoms $A_n = (f^{z_1}, \dots, f^{z_n}) = (a_1, \dots, a_n)$ is the output of the greedy algorithm.

As stated in Section 2.2, solving (18) amounts to solving $|\mathcal{Z}_{\text{train}}|$ quadratic optimization problems on unit simplices in \mathbb{R}^n . These problems are solved using the COSMO package [19]. While the number of optimization problems to be solved in each greedy iteration is large, they are independent of each other and could be easily parallelized. More details on the optimization problem will be discussed in Section 4.3.

The greedy algorithm terminates with one of the following three termination criteria:

- (1) when an absolute tolerance $\varepsilon_{abs} \geq 0$ is reached, i.e.

$$\Delta_n^2 := \max_{z \in \mathcal{Z}_{\text{train}}} \min_{(\lambda_i)_{1 \leq i \leq n} \in \Sigma_n} \left\| f^z - \sum_{i=1}^n \lambda_i f^{z_i} \right\|_{L^2(0,1)}^2 < \varepsilon_{abs}^2, \quad (19)$$

- (2) when the addition of a supplementary atom to the dictionary does not reduce the error beyond a given relative tolerance ε_{rel} , i.e.

$$\Delta_n - \Delta_{n+1} < \varepsilon_{rel} \Delta_n, \quad (20)$$

- (3) when a specified maximum number n_{max} of atoms is reached.

3.2. Online phase: interpolation

Let us assume here that we know the values of $m(s^z)$ for all $z \in \mathcal{Z}_{\text{train}}$. We then denote by $\mathcal{I}_m : \mathcal{Z} \rightarrow \mathbb{R}$ an interpolation function such that

$$\forall z \in \mathcal{Z}_{\text{train}}, \quad \mathcal{I}_m(z) = m(s^z).$$

Furthermore, let us assume that for all $z \in \mathcal{Z}_{\text{train}}$, we know the values $\Lambda^{\text{opt}}(z) = (\lambda_1^{\text{opt}}(z), \dots, \lambda_n^{\text{opt}}(z)) \in \Sigma_n$ solution to the optimization problem

$$\Lambda_n^{\text{opt}}(z) \in \underset{(\lambda_i)_{1 \leq i \leq n} \in \Sigma_n}{\operatorname{argmin}} \left\| f^z - \sum_{i=1}^n \lambda_i a_i \right\|_{L^2(0,1)}^2.$$

We then denote, for all $1 \leq i \leq n$, by $\mathcal{I}_i : \mathcal{Z} \rightarrow \Sigma_n$ an interpolation function such that

$$\forall z \in \mathcal{Z}_{\text{train}}, \quad \mathcal{I}_i(z) = \lambda_i^{\text{opt}}(z).$$

The online phase of the reduced-order modeling method then works as follows:

Online phase:

- (1) Given a new parameter value $z^* \in \mathcal{Z}$, we compute for all $1 \leq i \leq n$, $\tilde{\lambda}_i^* = \mathcal{I}_i(z^*)$ and $\tilde{m}^* = \mathcal{I}_m(z^*)$.
- (2) We compute Λ^* as the projection on Σ_n of $\tilde{\Lambda}^* := (\tilde{\lambda}_1^*, \dots, \tilde{\lambda}_n^*)$ using Algorithm 1 from [37] and define $m^* = [\tilde{m}^*]_+^1$.
- (3) Compute $f^* := \sum_{i=1}^n \lambda_i^* a_i$.
- (4) By construction, f^* is then equal to $\operatorname{icdf}_{u^*}$ for some unique $u^* \in \mathcal{P}_2(\Omega)$. Compute u^* .
- (5) The reduced-order modeling approximation s_n^{gB, z^*} of s^{z^*} is then defined as $s_n^{\text{gB}, z^*} = m^* u^*$.

We use piece-wise linear interpolation in the numerical experiments. Since the interpolants are rather simple functions (see Figures 3 and 8), this appears sufficient. Further work could be done experimenting with higher order interpolation schemes.

¹ For convenience, we repeat the algorithm here:

- (a) Sort $\tilde{\Lambda}^*$ such that $\tilde{\lambda}_1^* \geq \dots \geq \tilde{\lambda}_n^*$.
- (b) Find $\bar{j} := \max\{1 \leq j \leq n : \{\lambda_j^* + j^{-1}(1 - \sum_{i=1}^j \lambda_i^*) > 0\}\}$
- (c) Let $c := \bar{j}^{-1}(1 - \sum_{i=1}^{\bar{j}} \lambda_i^*)$
- (d) Return Λ^* where $\Lambda_i^* = \max\{\tilde{\lambda}_i^* + c, 0\}$ for $i = 1, \dots, n$.

4. COMPUTATIONAL ASPECTS

The aim of this section is to present the methodology used in order to build a reduced-order model for the parametrized porous media flow problem presented in the preceding section. We also discuss some choices we make in our implementation. The code used to generate the examples shown here is available at

<https://github.com/ToBlick/MorporJ>

The link also contains certain interactive examples to help gain understanding on the behavior of weighted Wasserstein barycenters as one varies the weights.

4.1. Computing cumulative distribution functions and their inverse

In practice, a snapshot $s \in \mathcal{M}$ (i.e. the wetting saturation s_w) is provided, for a given value of the set of parameters, as a piece-wise constant function on a uniform grid of size $N = 1000$. We denote this discrete representation as $\hat{\mathbf{s}} \in \mathbb{R}^N$ and use typewriter font for all discrete distributions, cumulative distribution functions, and their inverse.

In our one-dimensional numerical examples (with $x_{\min} = 0$ and $x_{\max} = 1$ for the sake of simplicity), we always consider that the saturation s has to satisfy the boundary condition $s(x_{\min}) = 1$. This is the reason why, from $\hat{\mathbf{s}}$, we compute another vector $\mathbf{s} := (\mathbf{s}_i)_{1 \leq i \leq N+2} \in \mathbb{R}^{N+2}$ such that $\mathbf{s}_1 = 0.0$, $\mathbf{s}_2 = 1.0$ and $\mathbf{s}_{i+2} = \hat{\mathbf{s}}_i$ for all $1 \leq i \leq N$. The value of \mathbf{s}_2 is chosen so that to enforce the boundary condition mentioned above, and the value of \mathbf{s}_1 is chosen so that the algorithm we use to compute a discrete approximation of the icdf of $u = \frac{s}{m(s)}$ is more numerically stable.

Let us describe this procedure here. First, from \mathbf{s} , an approximation of u is given by the vector $\mathbf{u} = (\mathbf{u}_i)_{1 \leq i \leq N+2} \in \mathbb{R}^{N+2}$ defined by

$$\forall 1 \leq i \leq N + 2, \mathbf{u}_i = \frac{\mathbf{s}_i}{\sum_{j=1}^{N+2} \mathbf{s}_j}.$$

Second, an approximation of cdf u is given by the vector $\mathbf{cdf} = (\mathbf{cdf}_i)_{1 \leq i \leq N+2} \in \mathbb{R}^{N+2}$ defined by

$$\forall 1 \leq i \leq N + 2, \quad \mathbf{cdf}_i = \sum_{j=1}^i \mathbf{u}_j.$$

Note that the definition of \mathbf{s} guarantees that $\mathbf{cdf}_1 = 0$ and $\mathbf{cdf}_{N+2} = 1$.

The discrete approximation of icdf u will be given as a piecewise constant function on the interval $[0, 1]$. Let $\mathbf{x}_1 = x_{\min} < \mathbf{x}_2 < \dots < \mathbf{x}_{N+2} = x_{\max}$ be a uniform discretization grid of (x_{\min}, x_{\max}) . Let also $M \in \mathbb{N}^*$ (in practice M is chosen to be equal to $N + 2$ in the numerical experiments) and $\mathbf{p}_1 = 0 < \mathbf{p}_2 < \dots < \mathbf{p}_M = 1$ be a uniform discretization grid of the interval $(0, 1)$. The discrete approximation of icdf u is then given as a vector $\mathbf{icdf} = (\mathbf{icdf}_j)_{1 \leq j \leq M} \in \mathbb{R}^M$ the components of which are defined as follows: for all $1 \leq j \leq M$, let $i \geq 2$ be the smallest integer such that $\mathbf{cdf}_i \geq \mathbf{p}_j$. Then,

$$\mathbf{icdf}_j = \mathbf{x}_{i-1} + (\mathbf{x}_i - \mathbf{x}_{i-1})(\mathbf{p}_j - \mathbf{cdf}_{i-1}) / (\mathbf{cdf}_i - \mathbf{cdf}_{i-1}). \quad (21)$$

Note that this can be done in a single pass over j , since the cdf is monotonically increasing. To avoid numerical instabilities whenever $(\mathbf{cdf}_i - \mathbf{cdf}_{i-1}) \approx 0$, we let $\mathbf{icdf}_j = \mathbf{x}_{i-1}$ whenever $(\mathbf{cdf}_i - \mathbf{cdf}_{i-1})$ is smaller than a prescribed tolerance (10^{-12} in the numerical experiments).

If we apply the same procedure to the resulting icdf, with the roles of \mathbf{x} and \mathbf{p} swapped, we obtain a numerical approximation of the inverse of the inverse cumulative distribution function, which is given as a vector $\mathbf{icicdf} \in \mathbb{R}^{N+2}$. Up to the error introduced by the finite grid size and the linear interpolation, $\mathbf{icicdf} \approx \mathbf{cdf}$. On average, this error lies between 10^{-4} and 10^{-3} in the discrete L^2 norm for the data from our numerical

experiments.

As mentioned at the end of Section 2.2, the solution to the minimization problem yields the icdf of a normalized snapshot. We invert it to obtain the iicdf and perform simple first order backwards finite differences to compute the corresponding pdf. This guarantees that the reconstructed pdf integrates to one.

Clearly, both the inversion and differentiation could be carried out using higher-order methods such as a spline interpolant of the discrete pdf. We choose the simple linear interpolants as they prove to be robust when applied to the snapshots we are considering in this work, which feature jump discontinuities and, in some cases, resemble even discrete Dirac distributions. This, of course, limits the numerical accuracy of our method, as will be discussed later.

4.2. Sources of Error

Given a target $s^* = s^{z^*}$ for some $z^* \in \mathcal{Z}$ and the reduced-order model approximation obtained $s_n^{\text{gB}*} = s_n^{\text{gB},z^*}$, we can identify three sources of error.

- (1) Firstly, s^* might not be close to any s_n^{gB} that can be obtained by means of inverting and differentiating $\sum_{i=1}^n \mathcal{L}_i(z^*) f^{z_i}$ using the given atoms and interpolations. This error describes how rich the range of the collection of atoms is. The decay of this error as n increases is of prime interest to evaluate the quality of the proposed method.
- (2) Secondly, the optimization may fail to find the optimal barycentric weight Λ_n^{opt} . The properties of the system that has to be solved is addressed in Section 4.3.
- (3) Thirdly, there is a numerical error introduced simply by following the pdf \mapsto cdf \mapsto icdf \mapsto iicdf \mapsto pdf pipeline as outlined in the beginning of this section. Naturally, this sets a lower bound on the error we can obtain, even with arbitrary large n . For all our numerical experiments, however, this error is at least one order of magnitude smaller than the average errors we observe in the numerical experiments, so it is not the limiting factor in these cases.

Note that the proposed method operates entirely in the W_2 space and on the icdfs of the saturations. This error in W_2 need not reflect the errors in L^1 between the reconstructed saturations themselves, which we chose as a reference.

For example, consider the distributions $s_1 = 10 \times \mathbb{1}_{[0.0,0.1]}$, $s_2 = 2 \times \mathbb{1}_{[0.0,0.5]}$ and $s_3 = \mathbb{1}_{[0.0,1.0]}$. While $\|s_1 - s_2\|_{L_1} = 1.6 > 1.0 = \|s_2 - s_3\|_{L_1}$, we find $W_2(s_1, s_2) \approx 0.23 < 0.29 \approx W_2(s_2, s_3)$. Note that the distributions in this example are shaped similarly to the ones in our numerical experiments. As a result, reducing the training error in W_2 through the addition of an atom to the dictionary through the greedy algorithm may not reflect in the L_1 norm.

The properties of the interpolation function are not adding to the error in our experiments, as we are only looking at the training error. This decision is made to keep the presentation brief as the generalization error depends largely on the selection of the training and test data sets, which we do not address.

4.3. Short comments on computational cost

A thorough study of the run-times for the proposed method is beyond the scope of this article. The numerical inversion of a discrete cdf or icdf depends only linearly on N , which allows the computation of Wasserstein barycenters for given weights in real time, as can be seen in the `demo` notebook accompanying the code provided along with this work.

Selecting atoms with the greedy algorithm involves solving quadratic systems of the form $\min \|\Lambda \mathbf{A} - \mathbf{f}\|^2$ with constraints $\Lambda \geq 0$ element-wise and $\sum_i \lambda_i = 1$. \mathbf{A} is an $N \times n$ matrix with the discrete atoms as its columns, thus n increases with every iteration of the algorithm. Λ is the vector of barycentric weights to be optimized and \mathbf{f} is the icdf of the saturation we aim to approximate. As stated, this system has to be solved $|\mathcal{M}_{\text{train}}|$ times in every iteration and we observe that $\mathbf{A}^T \mathbf{A}$ becomes very ill-conditioned with growing iteration number, see Figure 11 below. We cannot orthogonalize the columns of \mathbf{A} , since the resulting atoms would no longer be

discrete icdfs (for example, they will most likely no longer be strictly positive). This point is discussed further in Section 5.3.

Since \mathbf{A} does not change within greedy iterations, significant computational time can be saved by assembling the system once and then modifying \mathbf{f} in-place. Furthermore, we initialize Λ inversely proportional to the W_2 distance between each atom in the dictionary and the distribution \mathbf{f} that is to be approximated - that is, if \mathbf{f} is very close to the i -th atom, the corresponding $\lambda[i]$ will be initialized close to one.

5. NUMERICAL EXAMPLES

In this section, we compare the performance of the method for two test-cases to a POD representation. In the latter case, we project every element of $\mathcal{M}_{\text{train}}$ to the POD basis and back. We choose the relative L^1 error to compare the methods. To be precise, let Ψ be the $N \times n$ matrix that has as columns the first n POD modes. The POD reconstruction of a solution s is then $\mathbf{s}^{\text{POD}} := \Psi_{\text{POD}}^T \Psi_{\text{POD}} \mathbf{s}$ and the corresponding error $\|s - s^{\text{POD}}\|_{L^1} / \|s\|_{L^1}$. The reconstruction error for the barycentric method is given by $\|s - \bar{s}^{gB}\|_{L^1} / \|s\|_{L^1}$, where \bar{s}^{gB} is obtained by applying the method as it was described in Section 2.

In the following examples, $\Omega = (0, 1)$, corresponding to a physical domain of length 1 km and the grid is composed of $N + 2 = 1002$ cells. The time interval, which covers several years, is discretized thanks to time steps satisfying the CFL condition given in [17]. The boundary and initial conditions are: $p_D(0, t) = 4.137 \times 10^7$ Pa, $p_D(L, t) = 2.758 \times 10^7$ Pa, $s(0, t) = 1$ for all $t \in [0, T]$, $s_w(x, 0) = 0$ for all $x \in (0, L)$.

5.1. Example 1: Homogeneous medium with a varying exponent of the relative permeability law and various viscosity ratios

For the first example we consider the case of a homogeneous medium with porosity $\phi = 0.1$ and permeability $k = 10^{-13} \text{ m}^2$. The wetting viscosity is kept constant at a value of $\mu_{nw} = 0.003$ Pa s, but the non-wetting viscosity is changed in such a way that $\mu = \mu_{nw} / \mu_w$ ranges from 1 to 25. The sampling is chosen analogous to a geometric progression, so that $\mu \in \{1, 2, 3, 6, 12, 25\}$. The exponent β of the relative permeability varies uniformly from 2 to 6, that is $\beta \in \{2, 3, 4, 5, 6\}$. The evolution of the water saturation is calculated over 5 years, saving snapshots at a fixed time-step, $t \in \{0.2, 0.4, \dots, 5.0\}$ years, leading to $|\mathcal{Z}_{\text{train}}| = 750$ and $z = (t, \mu, \beta)$. Some snapshots of the training set at the final time are shown in Figure 1.

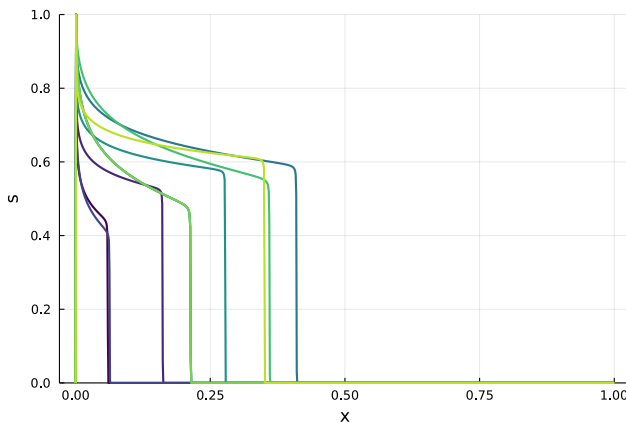


FIGURE 1. Randomly selected snapshots at final time from example 1.

The methodology described in Sections 3-4 is then applied. A few selected atoms selected by the greedy algorithm are shown in Figure 2. Figure 3 shows cross-sections of the interpolating functions from the parameter space for the case $n = 3$. Note how these are simple functions showing a smooth mapping from parameter space

to the reduced coordinates. The average reconstruction error, measured in the L_1 norm, is already below 5% for $n = 3$ (see Table 1).

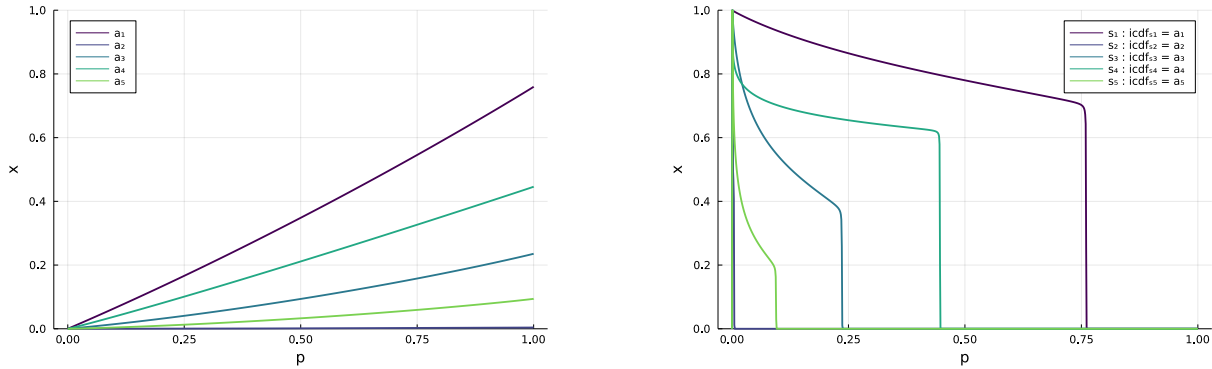


FIGURE 2. Dictionary atoms (left) and corresponding densities (right) for the first five atoms chosen by the greedy algorithm for example 1.

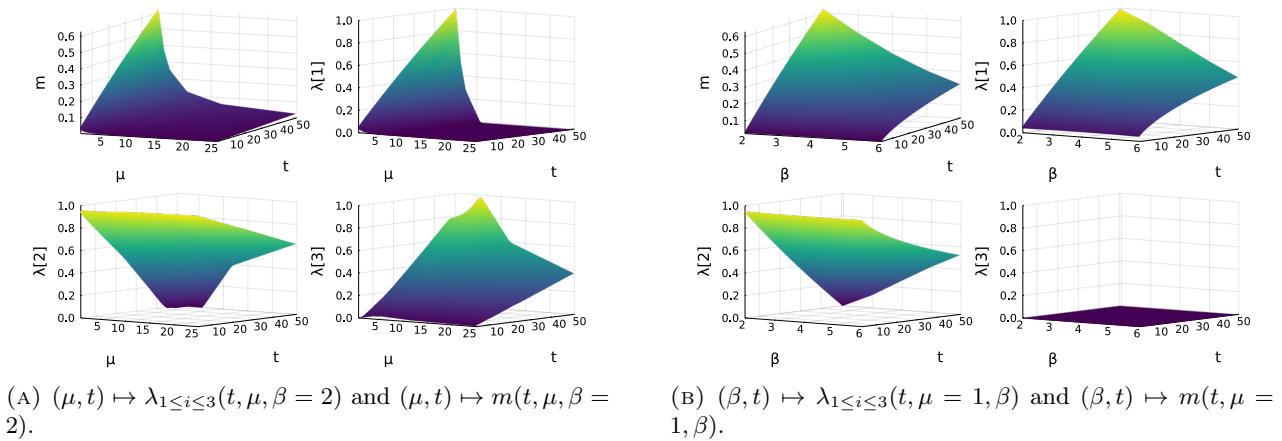


FIGURE 3. Interpolations from parameter space to barycentric weights and mass in Example 1.

The decay of the maximum and average Wasserstein error on $\mathcal{U}_{\text{train}}$ as more atoms are added is shown in Figure 4. Note that there is an increase in maximum error at the addition of the 29th atom. This can be explained by the iterative solver not converging before reaching its maximum iteration number while determining the optimal weights for one of the training snapshots. While the addition of atoms should only decrease the reconstruction error, we observe in practice that many atoms may make it harder to solve for the optimal barycentric weights (c.f. Section 5.3).

Table 1 shows that decent reconstructions can already be achieved with very few atoms. This highlights the strength of the method and its applicability to the example at hand. As expected, the POD reconstruction is ill-suited: to obtain an L^1 -error of one percent, one needs over 230 modes, which is almost one third of the size of the training set - barely any reduction is possible. This is mainly due to oscillations occurring in the POD reconstruction around the jump discontinuity as shown in Figure 5.

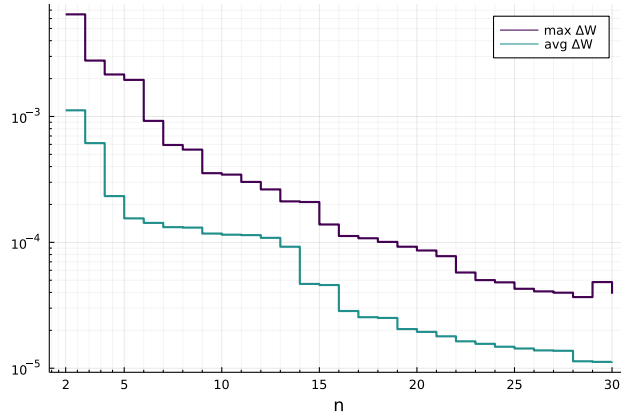


FIGURE 4. Decay of the W_2 error through the iterations of the greedy algorithm for example 1.

ε	0.1	0.05	0.01	0.005
$n_{gBar}(\varepsilon)$	2	3	16	29
$n_{POD}(\varepsilon)$	41	77	231	$\gg 250$

TABLE 1. Atoms and POD modes needed to reach a given tolerance ε , in L^1 norm, for the snapshots used in example 1.

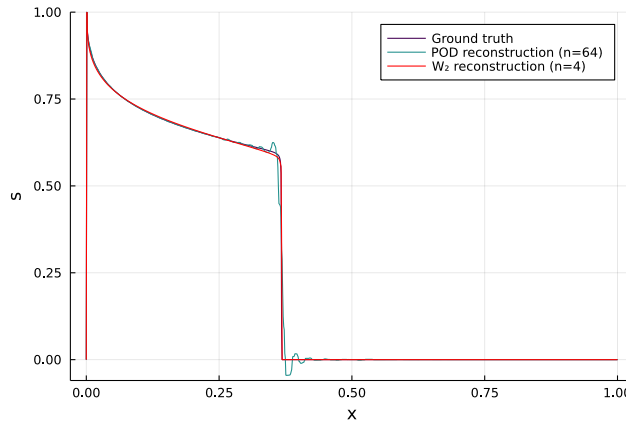


FIGURE 5. Example 1: Snapshot and its reconstruction using the gBar algorithm and a POD basis.

5.2. Example 2: Porous medium made of two rock types with varying permeability and position of the heterogeneity

For the second example, a heterogeneous medium is tested, composed of two different rock types: a highly permeable one (HP) characterized by $\phi_{HP} = 0.1$ and $k_{HP} = 10^{-13} \text{ m}^2$, and a low permeable one (LP) with fixed porosity $\phi_{LP} = 0.01$ but uniformly varying permeability $k_{LP} \in \{5, 6, 7, 8, 9\} \times 10^{-14} \text{ m}^2$. In this case, the viscosities are kept constant at $\mu_w = 0.003 \text{ Pa s}$ and $\mu_{nw} = 0.03 \text{ Pa s}$ (giving a ratio of $\mu = 10$), and so is the

relative permeability exponent, $\beta = 2$.

The location of the interface between the two rock types is denoted by γ . For $x < \gamma$, the rock type is HP, and LP otherwise. γ , which is sampled at irregular intervals, with a finer sampling around $x = 0$: $\gamma \in \{0.0, 0.05, 0.1, 0.15, 0.2, 0.4, 0.6\}$. The evolution of the water saturation is calculated over 10 years and snapshots are regularly taken at $t \in \{0.5, 1.0, \dots, 10.0\}$ years, leading to $z = (t, k_{LP}, \gamma)$ and $|\mathcal{Z}_{\text{train}}| = 700$.

From Figure 6, where some snapshots at the final time are shown, we see that the solution is quite different to the homogeneous case: when the saturation front reaches the permeability interface, the velocity of propagation changes, leading to a kink in the saturation profile. In some cases, the front does not reach the jump and the evolution of the saturation still behaves as in an homogeneous medium.

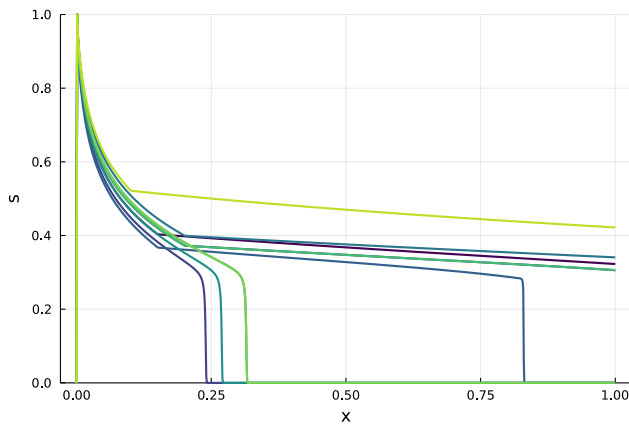


FIGURE 6. Randomly selected snapshots at final time from example 2.

The first atoms selected by the greedy algorithm are shown in Figure 7 and cross-sections of the interpolating functions for $\lambda_{1 \leq i \leq 3}(t, \gamma, k_{LP})$ and $m(t, \gamma, k_{LP})$ in Figure 8.

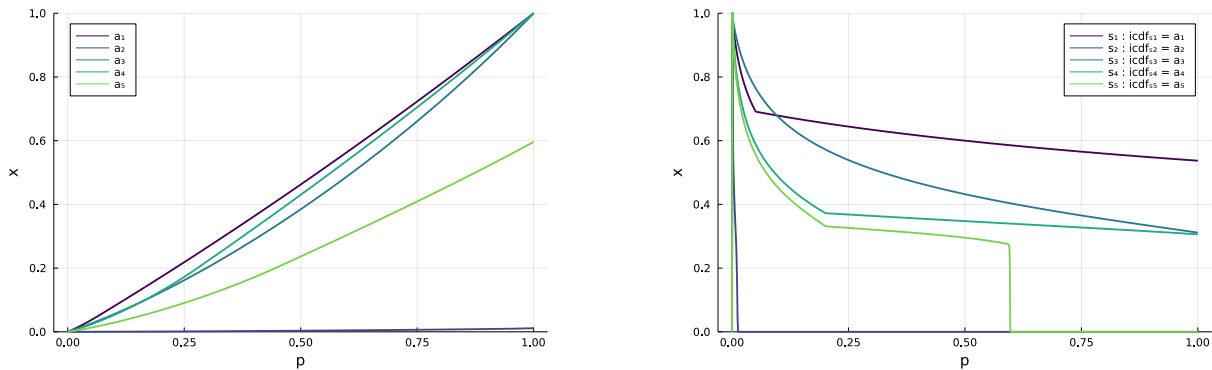


FIGURE 7. Dictionary atoms (left) and corresponding densities (right) for the first five atoms chosen by the greedy algorithm for example 2.

Letting the greedy algorithm add atoms allows the W_2 -error to decrease in a similar way with respect to the homogeneous case (Figure 9). Note, however, that both maximum and average errors in W_2 are about one order of magnitude above the values from Example 1. This also reflects in the errors as measured in the L_1 norm,

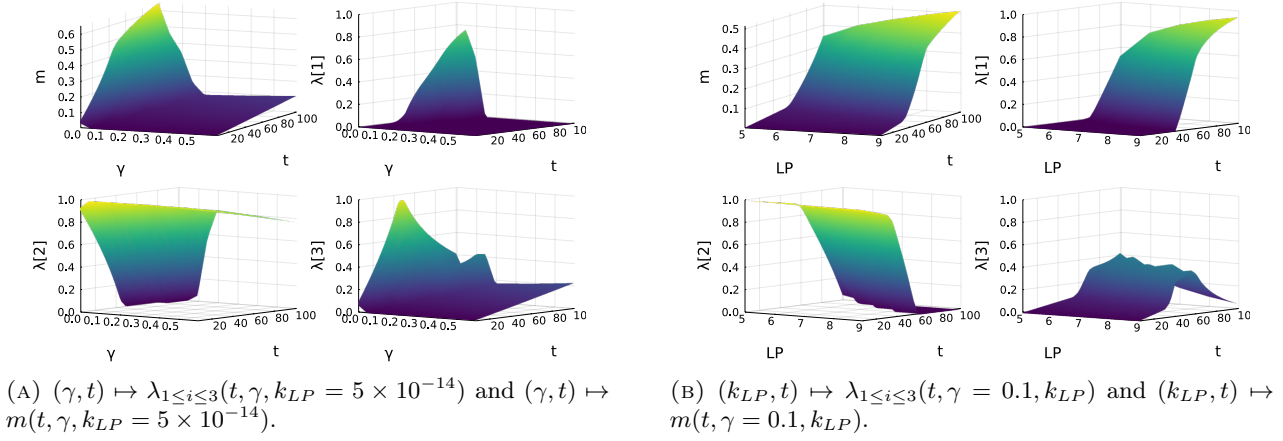


FIGURE 8. Interpolations from parameter space to barycentric weights and mass in example 2.

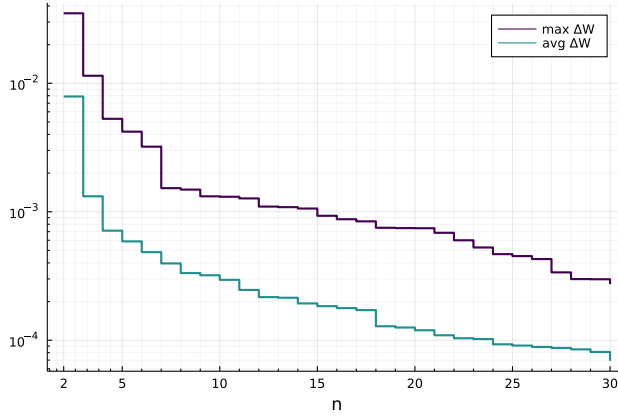


FIGURE 9. Decay of the W_2 error through the iterations of the greedy algorithm for example 2.

ε	0.1	0.05	0.01	0.005
$n_{gBar}(\varepsilon)$	3	3	14	-
$n_{POD}(\varepsilon)$	20	35	118	173

TABLE 2. Atoms and POD modes needed to reach a given tolerance ε , in L^1 norm, for the snapshots used in example 2.

shown in Table 2. It was not possible to obtain an accuracy of 0.5% for this test-case before the barycentric algorithm became unstable and the iterative solver failed to converge to optimal barycentric weights.

The reconstruction of a snapshot, plotted in Figure 10, shows that the proposed method does not accurately capture the kink of the solution at the interface of the HP/LP regions at $x \approx 0.2$. On the other hand, the POD reconstruction shows an instability at the solution front. Several elements of the training set no longer show

this jump discontinuity, however, and in these cases the POD method performs better, which reflects also in the second row of Table 2.

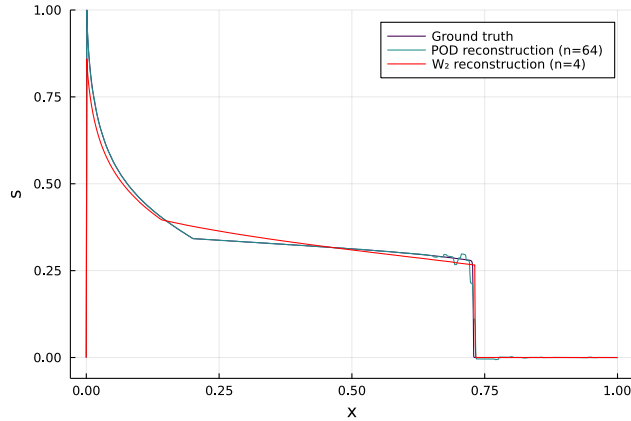


FIGURE 10. Example 2: Snapshot and its reconstruction using the gBar algorithm and a POD basis. Note how the barycentric interpolation does not capture the kink located at the HP/LP transition well.

5.3. Properties of the minimization problem

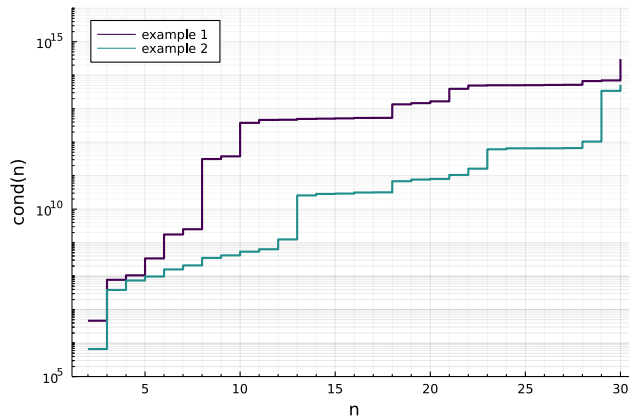


FIGURE 11. Conditioning number of the matrix $\mathbf{A}^T \mathbf{A}$ as n increases.

In this section, we want to provide some insights into the properties of the optimization problem $\min_{\Lambda} \|\mathbf{A}\Lambda - \mathbf{f}\|$, i.e. the approximation of a discrete icdf \mathbf{f} as a convex combination of a collection of atoms (the columns of \mathbf{A}). Figure 11 shows the conditioning number of the matrix $\mathbf{A}^T \mathbf{A}$ throughout the greedy algorithm iterations for examples one and two. In both cases, the conditioning number grows fast with increasing n , making it harder to solve for the optimal Λ . Note that a bad conditioning of the matrix $\mathbf{A}^T \mathbf{A}$ does not necessarily mean that the optimization problem has no solution. For example, for $s_1 = \delta_{x_1}$ and $s_2 = \delta_{x_2}$, we find $\text{icdf}_1 \equiv x_1$ and $\text{icdf}_2 \equiv x_2$ and $\mathbf{A}^T \mathbf{A}$ is singular. Nevertheless, the optimization problem for the optimal barycenter to approximate a given u by these atoms is well posed - it is solved by projecting the mean of u onto $\{(1 - \lambda)x_1 + \lambda x_2 : 0 \leq \lambda \leq 1\}$.

Figure 12 shows the W_2 error between two target icdfs and the barycenter of atoms for different n and Λ in log-scale. We display elements of the $n - 1$ -dimensional simplex as points in a convex polygon. For $n = 3$, this corresponds to the well-known barycentric coordinates within a triangle, depicted in Figures 12a and 12d. For $n > 3$, we use generalized barycentric coordinates (in particular, so-called Wachspress coordinates) within polygons as described in [18]. These coordinates associate to every point $x_{\text{poly}} \in \{n - \text{Polygon}\} \subset \mathbb{R}^2$ a vector of barycentric weights $\lambda_{\text{gbc}} \in \Sigma_n \subset \mathbb{R}^n$. The i -th vertex of a polygon corresponds to barycentric weights with only one non-zero entry for the i -th atom, while points along the edge between the vertices corresponding to a_i and a_j represent $(1 - \lambda)a_i + \lambda a_j$ where $0 < \lambda < 1$ along the edge. The center of each polygon corresponds to the point where $\lambda_{\text{gbc},i} = 1/n \forall i$. The mapping $x_{\text{poly}} \mapsto \lambda_{\text{gbc}}$ is not surjective, so the second and third column of Figure 12 show only a specific cut through the true energy landscape defined on Σ_4 and Σ_8 . For example, from Figure 12b, we can see that approximations with W_2 error below 10^{-3} lie close to a line that enters the tetrahedron Σ_4 at the a_1 - a_2 edge and exits at the a_3 - a_4 edge. The minimum lies on the a_1 - a_3 - a_4 face, very close to the a_3 - a_4 edge. In the case $n = 8$, it naturally is quite hard to faithfully represent this high-dimensional setting with a two-dimensional plot. Recall that the objective is a quadratic function constrained to a convex set. Therefore, no disjoint minima are possible.

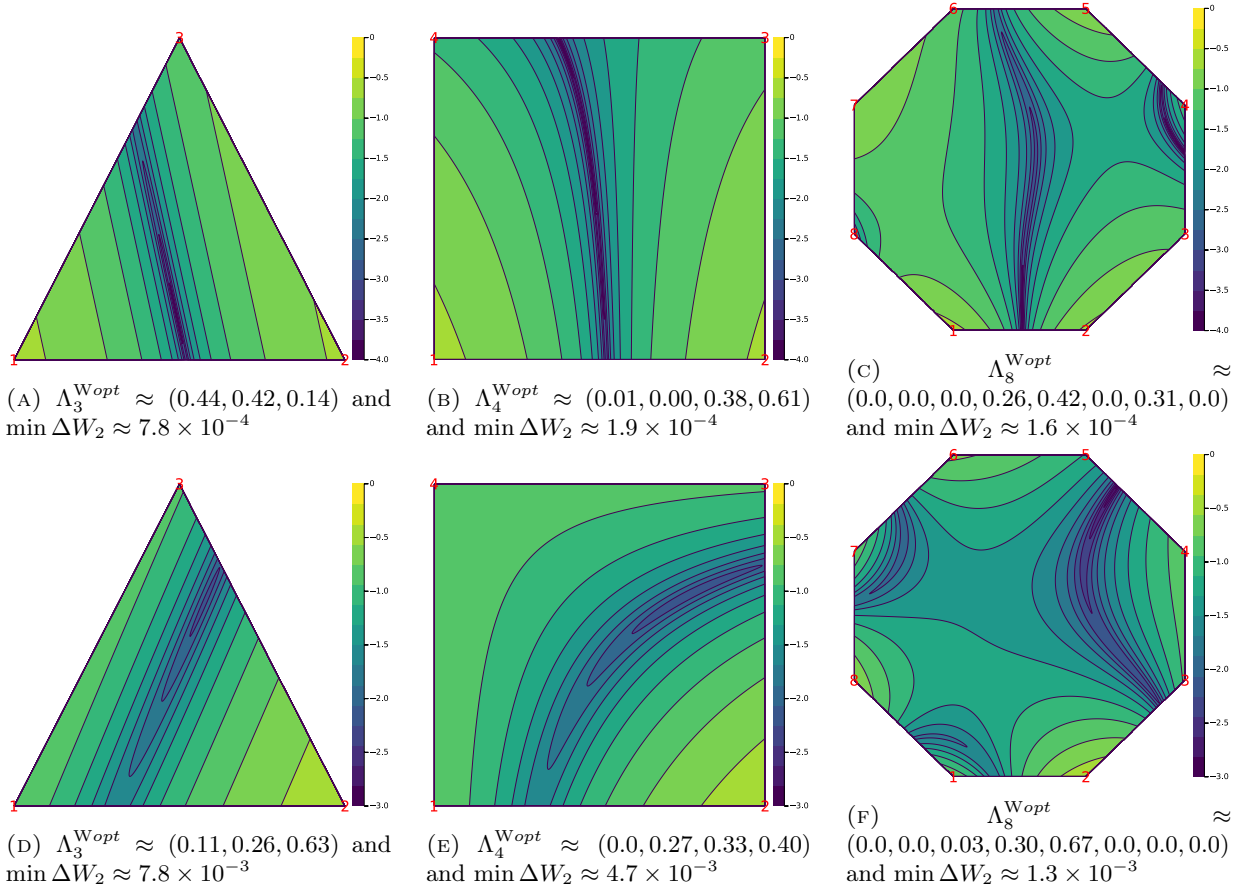


FIGURE 12. \log_{10} of the W_2 distance between barycenter reconstructions and the snapshot from Figure 5 (top row) and Figure 10 (bottom row) for different n as a function of $\lambda_1, \dots, \lambda_n$. The vertices of the polygons mark the atoms a_1, \dots, a_n .

In all depicted cases, the contour lines of the energy are very eccentric, a consequence of the poor conditioning of the matrix $\mathbf{A}^T \mathbf{A}$. In both examples and for large n , optimal barycentric weights lie on the faces of the simplices, with several zero entries. This sparsity is seen throughout the elements of $\mathcal{M}_{\text{train}}$ and has also been observed in [6]. It is an interesting question for future work how one could choose and/or modify the atoms in order to avoid this and thereby moving the minimum of the objective into the interior of the simplex.

The existence of a mapping from the Wasserstein space to \mathbb{R}^N (namely, $\text{pdf}_s \mapsto \text{icdf}_s$) is a special case of one spatial dimension and does not hold in higher-dimensional case ([29], Proposition 8.2). To get an understanding if we are adding an atom to the collection in step $n+1$ that is already close to a convex combination of the atoms $\{a_1, \dots, a_n\}$, we can exploit this. Using the distance matrix $(\|a_i - a_j\|_{L_2}^2)_{1 \leq i, j \leq n}$, we calculate the volume of the simplex with vertices $\{a_i\}_{1 \leq i \leq n}$ as a function of n using the Cayley–Menger determinant [7]. We normalize this value with the volume of the unit n -simplex. The results are shown in Figure 13.

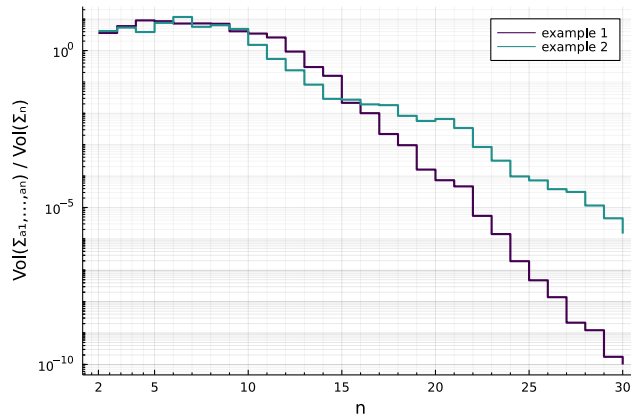


FIGURE 13. Normalized volume of the simplex with vertices a_1, \dots, a_n as n increases.

Adding another atom that is close to the simplex with vertices a_1, \dots, a_n will lead to a small volume for the resulting simplex with vertices a_1, \dots, a_{n+1} . For $n = 2$, this would be a very slim triangle. We observe that up until $n \approx 10$, in both examples, the simplices relative volume remains almost constant, which we interpret as every additional atom providing incremental information. For larger n , the relative volume decreases exponentially as the simplices become more degenerate, i.e. some of the a_i 's are almost affinely dependent.

5.4. Conclusions from the numerical experiments

The numerical experiments exemplify that the presented method is able to represent the solution space of a complex non-linear equation using only very few atoms. In the first test-case considered, using only $n = 3$ atoms allows to reconstruct every element in training data set with an accuracy of 95%. The data depend on three parameters and the solutions vary considerably with those parameter values - note, for example, how slowly the front of the wetting phase moves at high values of μ and β . The map from the parameter space to the barycentric weights for only a small number of atoms has a simple structure as seen in Figure 3a and 3b. The second test-case features a collection of solutions that are both more challenging for the presented method and easier for the POD method to represent. In this case, several elements of $\mathcal{M}_{\text{train}}$ are supported on the entire domain, which removes the biggest challenge for the POD representation, i.e. the jump discontinuity. At the same time, the barycentric interpolation does not preserve the characteristic kink located at the HP/LP interface, reducing the quality of the barycentric reconstruction. Ultimately, the barycentric method still performs well even with very few atoms, but cannot reach the same level of accuracy from test-case 1 before the dictionary becomes too ill-conditioned and the greedy algorithm terminates.

It is important to note, when comparing the presented method to POD, that there is no straightforward online method for the latter. We are merely comparing the encoding and reconstruction properties of the two methods. The online method for the barycentric representation is purely interpolating and comes at a very small computational cost (that is: evaluation of the functions $\mathcal{I}_{1,\dots,n}$, calculation of the convex combination, inversion and differentiation of the icdf) that is linear in N , the dimension of the full-order problem.

Lastly, we provide some run time values. Naturally, these are strongly dependent on the implementation, size of the problem, solver tolerances chosen, and many other factors, but we believe despite those caveats they might be of interest. The full order model run time depends on the parameters, as it employs a variable time-step size to deal with the stiffness of the equation. On an all-purpose personal notebook, it takes between 6 to 66 seconds to generate a time-series of snapshots for example one, and between 120 and 610 seconds to generate one for example two. Running the greedy algorithm to obtain $n = 5, 10$ and 30 atoms in the offline phase takes around 11, 16, and 140 seconds respectively for example one and 102, 115, and 586 seconds for example two. Reconstructing one snapshot in the online phase takes around 1.6×10^{-5} , 3.4×10^{-5} , and 8.9×10^{-5} seconds in example one and 1.6×10^{-5} , 3.4×10^{-5} , and 8.4×10^{-5} seconds in example two.

6. FUTURE RESEARCH DIRECTIONS AND EXTENSIONS

The computation of Wasserstein distances and barycenters through the inverse cumulative distribution functions is a special case limited to one-dimensional problems. There has been a lot of progress in the development of algorithms for arbitrary dimensions in recent years which can be used to extend the present work for these cases [29], [34]. We intend to address this in upcoming work. It can be worthwhile to allow for the inclusions of atoms into the dictionary that are not snapshots of the solution themselves. This approach is taken in [33], where the atoms are initialized as uniform distributions and then optimized alongside the barycentric weights, i.e. one solves

$$\min_{U_n \in \mathcal{P}_2(\Omega)^n} \min_{\Lambda_n \in \Sigma_n} \sum_{u \in \mathcal{U}_{\text{train}}} W_2^2(u, \text{Bar}(U_n, \Lambda_n)). \quad (22)$$

We would further like to explore ideas on how to improve the conditioning of the matrix arising from the atoms. Instead of adding new atoms to the dictionary as they come, one could look for a way of adding a modified atom which is - in a sense that has to be made precise - fully outside the current range of the dictionary. In classical linear reduced basis methods, this is ensured by projecting any new element to the orthogonal complement of the present basis but such an operation is not appropriate for the current framework. Greedy algorithms developed for the cone of positive functions such as the Enlarged Nonnegative Greedy algorithm introduced in [3] could be an interesting starting point for our purposes.

It would also be very interesting to see if the approach proposed in [31] could also be adapted in the present non conservative context.

Lastly, the proposed method is purely interpolating in the online phase. It would be very interesting to utilize a more intrusive method, perhaps one that directly works with the residual of the equation the way a projection-based ROM does. This is made difficult by the fact that this residual is generally given as a function of the pdf, while the reduced representation by atoms in our method is formulated in terms of icdfs. Moving back and forth between those two representations is costly (in particular, it depends on the size N of the high fidelity model). Of particular interest could be functions that are expressed as Wasserstein gradient flows of the (time-discrete) form $f(t + \Delta t) \in \arg \min_f W_2^2(f(t), f) + 2\Delta t V(f)$ for a given potential V [32]. In this case, the first term can be cheaply evaluated in terms of icdfs in the one dimensional case. As a perspective, it would also be interesting to compare our approach with other nonlinear intrusive methods, like [35], or the transformed snapshot interpolation from [38].

FUNDING AND ACKNOWLEDGMENTS

This work was initiated and substantially realized during the six weeks of the CEMRACS 2021 event held at CIRM, Luminy. The authors' stay at CIRM has been funded by IFPEN, ANR ADAPT (ANR-18-CE46-0001), ANR project COMODO (ANR-19-CE46-0002), and the Emergences Project grant "Models and Measures" from the Paris City Council. The authors would like to thank Sébastien Boyaval for fruitful discussions.

REFERENCES

- [1] D. AMSALLEM AND B. HAASDONK, *Pebl-rom: Projection-error based local reduced-order models*, Advanced Modeling and Simulation in Engineering Sciences, 3 (2016), p. 6.
- [2] D. AMSALLEM, M. J. ZAHR, AND C. FARHAT, *Nonlinear model order reduction based on local reduced-order bases*, International Journal for Numerical Methods in Engineering, 92 (2012), pp. 891–916.
- [3] A. BAKHTA, T. BOIVEAU, Y. MADAY, AND O. MULA, *Epidemiological forecasting with model reduction of compartmental models. application to the covid-19 pandemic*, Biology, 10 (2021), p. 22, <https://doi.org/https://doi.org/10.3390/biology10010022>.
- [4] M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An Empirical Interpolation Method: application to efficient reduced-basis discretization of partial differential equations.*, C. R. Acad. Sci. Paris, Série I., 339 (2004), pp. 667–672, <https://doi.org/https://doi.org/10.1016/j.crma.2004.08.006>.
- [5] P. BENNER, A. COHEN, M. OHLBERGER, AND K. WILLCOX, *Model Reduction and Approximation: Theory and Algorithms*, vol. 15, SIAM, 2017, <https://doi.org/https://doi.org/10.1137/1.9781611974829>, <https://doi.org/10.1137/1.9781611974829>.
- [6] N. BONNEEL, G. PEYRÉ, AND M. CUTURI, *Wasserstein barycentric coordinates: histogram regression using optimal transport.*, ACM Trans. Graph., 35 (2016), pp. 71–1.
- [7] J. C. BOWERS AND P. L. BOWERS, *A Menger Reduc: embedding metric spaces isometrically in Euclidean space*, The American Mathematical Monthly, 124 (2017), pp. 621–636, <https://doi.org/10.4169/amer.math.monthly.124.7.621>, <https://www.tandfonline.com/doi/abs/10.4169/amer.math.monthly.124.7.621>, <https://arxiv.org/abs/https://www.tandfonline.com/doi/pdf/10.4169/amer.math.monthly.124.7.621>.
- [8] S. BOYAVAL, G. ENCHÉRY, R. SANCHEZ, AND Q. H. TRAN, *A reduced-basis approach to two-phase flow in porous media*, in Finite Volumes for Complex Applications VIII - Hyperbolic, Elliptic and Parabolic Problems, 2017, pp. 477–486.
- [9] R. H. BROOKS AND A. T. COREY, *Hydraulic properties of porous media*, Hydrology Paper, 7 (1964), pp. 26–28.
- [10] N. CAGNIART, Y. MADAY, AND B. STAMM, *Model order reduction for problems with large convection effects*, in Contributions to partial differential equations and applications, Springer, 2019, pp. 131–150.
- [11] K. CARLBERG, *Adaptive h-refinement for reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1192–1210.
- [12] A. COHEN AND R. DEVORE, *Kolmogorov widths under holomorphic mappings*, IMA Journal of Numerical Analysis, 36 (2016), pp. 1–12, <https://doi.org/10.1093/imanum/dru066>, <http://dx.doi.org/10.1093/imanum/dru066>, https://arxiv.org/abs/oup/backfile/content_public/journal/imajna/36/1/10.1093_imanum_dru066/3/dru066.pdf.
- [13] A. COHEN, R. DEVORE, AND C. SCHWAB, *Convergence rates of best n-term galerkin approximations for a class of elliptic spdes*, Foundations of Computational Mathematics, 10 (2010), pp. 615–646.
- [14] A. COHEN, R. DEVORE, AND C. SCHWAB, *Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's*, Analysis and Applications, 09 (2011), pp. 11–47, <https://doi.org/10.1142/S0219530511001728>, <https://arxiv.org/abs/https://doi.org/10.1142/S0219530511001728>.
- [15] H. DO, J. FEYDY, AND O. MULA, *Structured prediction with sparse wasserstein barycenters*, to be submitted in the forthcoming weeks 2022.
- [16] V. EHRLACHER, D. LOMBARDI, O. MULA, AND F.-X. VIALARD, *Nonlinear model reduction on metric spaces. application to one-dimensional conservative pdes in wasserstein spaces*, ESAIM: Mathematical Modelling and Numerical Analysis, 54 (2020), pp. 2159–2197.
- [17] G. ENCHÉRY, R. MASSON, S. WOLF, AND R. EYMARD, *Mathematical and numerical study of an industrial scheme for two-phase flows in porous media under gravity*, Computational Methods in Applied Mathematics, 2 (2002), pp. 325–353.
- [18] M. S. FLOATER, *Generalized barycentric coordinates and applications*, Acta Numerica, 24 (2015), pp. 161–214, <https://doi.org/10.1017/S0962492914000129>, https://www.cambridge.org/core/product/identifier/S0962492914000129/type/journal_article (accessed 2022-03-28).
- [19] M. GARSTKA, M. CANNON, AND P. GOULART, *COSMO: A conic operator splitting method for convex conic problems*, arXiv:1901.10887 [math], (2020), <https://doi.org/10.1007/s10957-021-01896-x>, <http://arxiv.org/abs/1901.10887> (accessed 2022-03-24). arXiv: 1901.10887.
- [20] F. J. GONZALEZ AND M. BALAJEWICZ, *Learning low-dimensional feature dynamics using deep convolutional recurrent autoencoders*, arXiv preprint arXiv:1808.01346, (2018).

- [21] M. GREPL, Y. MADAY, N. NGUYEN, AND A. PATERA, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations.*, ESAIM, Math. Model. Numer. Anal., 41(3) (2007), pp. 575–605.
- [22] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Mathematics, (2015).
- [23] B. N. KHOROMSKIJ AND C. SCHWAB, *Tensor-structured galerkin approximation of parametric and stochastic elliptic pdes*, SIAM Journal on Scientific Computing, 33 (2011), pp. 364–385.
- [24] K. LEE AND K. CARLBERG, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, arXiv preprint arXiv:1812.08373, (2018).
- [25] Y. MADAY AND O. MULA, *A Generalized Empirical Interpolation Method: application of reduced basis techniques to data assimilation*, in Analysis and Numerics of Partial Differential Equations, F. Brezzi, P. Colli Franzone, U. Gianazza, and G. Gilardi, eds., vol. 4 of Springer INdAM Series, Springer Milan, 2013, pp. 221–235, https://doi.org/http://dx.doi.org/10.1007/978-88-470-2592-9_13, <http://arxiv.org/pdf/1512.00683.pdf>.
- [26] Y. MADAY, O. MULA, AND G. TURINICI, *Convergence analysis of the generalized empirical interpolation method*, SIAM Journal on Numerical Analysis, 54 (2016), pp. 1713–1731, <https://doi.org/10.1137/140978843>, <http://dx.doi.org/10.1137/140978843>.
- [27] S. MOWLAVI AND T. P. SAPSIS, *Model order reduction for stochastic dynamical systems with continuous symmetries*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1669–A1695.
- [28] B. PEHERSTORFER, D. BUTNARU, K. WILLCOX, AND H.-J. BUNGARTZ, *Localized discrete empirical interpolation method*, SIAM Journal on Scientific Computing, 36 (2014), pp. A168–A192.
- [29] G. PEYRÉ AND M. CUTURI, *Computational Optimal Transport*, arXiv:1803.00567 [stat], (2020), <http://arxiv.org/abs/1803.00567> (accessed 2022-03-30). arXiv: 1803.00567.
- [30] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced basis methods for partial differential equations: an introduction*, vol. 92, Springer, 2015, <https://doi.org/10.1007/978-3-319-15431-2>.
- [31] D. RIM, B. PEHERSTORFER, AND K. MANDLI, *Manifold approximations via transported subspaces: Model reduction for transport-dominated problems*, (2019), <https://doi.org/10.48550/arXiv.1912.13024>. Preprint, arXiv:1912.13024.
- [32] F. SANTAMBROGIO, *{ Euclidean, Metric, and Wasserstein } Gradient Flows: an overview*, Sept. 2016, <http://arxiv.org/abs/1609.03890> (accessed 2022-08-17). arXiv:1609.03890 [math].
- [33] M. A. SCHMITZ, M. HEITZ, N. BONNEEL, F. NGOLE, D. COEURJOLLY, M. CUTURI, G. PEYRÉ, AND J.-L. STARCK, *Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning*, SIAM Journal on Imaging Sciences, 11 (2018), pp. 643–678.
- [34] T. SÉJOURNÉ, J. FEYDY, F.-X. VIALARD, A. TROUVÉ, AND G. PEYRÉ, *Sinkhorn Divergences for Unbalanced Optimal Transport*, arXiv:1910.12958 [cs, math, stat], (2021), <http://arxiv.org/abs/1910.12958> (accessed 2022-03-30). arXiv: 1910.12958.
- [35] T. TADDEI, *A registration method for model order reduction: data compression and geometry reduction*, SIAM Journal on Scientific Computing, 42 (2020), pp. A997–A1027.
- [36] C. VILLANI, *Topics in optimal transportation*, no. 58, American Mathematical Soc., 2003.
- [37] W. WANG AND M. A. CARREIRA-PERPINAN, *Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application*, arXiv:1309.1541, (2013), <http://arxiv.org/abs/1309.1541> (accessed 2022-03-16).
- [38] G. WELPER, *h and hp-adaptive interpolation by transformed snapshots for parametric and stochastic hyperbolic pdes*, arXiv preprint arXiv:1710.11481, (2017).