Reconstruction of blood flow velocity with deep learning information fusion from spectral ct projections and vessel geometry

Shusong Huang, Monica Sigovan, Bruno Sixou

Reconstruction of blood flow velocity with deep learning information fusion from spectral CT projections and vessel geometry

Shusong HUANG[a], Monica SIGOVAN[a] and Bruno SIXOU[a]

[a] CREATIS, CNRS UMR 5220, Inserm U630, INSA de Lyon, Universite de Lyon, F-69621

ARTICLE HISTORY
Compiled October 25, 2024

ABSTRACT
In this work, we investigate a new deep learning reconstruction method of blood flow velocity within deformed vessels from contrast enhanced X-ray projections and vessel geometry. The principle of the method is to perform linear or nonlinear dimension reductions on the Radon projections and on the mesh of the vessel. These low dimensional projections are then fused to obtain the velocity field in the vessel. The accuracy of the reconstruction method is proved using various neural network architectures with realistic unsteady blood flows. The approach leverages the vessel geometry information and outperforms the simple PCA-net.

KEYWORDS
deep learning; inverse problems; hemodynamics; dimension reduction

1. Introduction

It is well-known that cardiovascular diseases are influenced by blood flow patterns [1–3]. Blood is a complex example of particulate suspension nanofluid, that has attracted much attention recently [4,5] . Classical imaging techniques like magnetic resonance and ultrasound have a low spatio-temporal resolution and the hemodynamic parameters like shear stresses are difficult to estimate with accuracy [6–9]. The evolution with time of the concentration of a tracer injected in the vessel can be exploited with X-ray CT measurements to evaluate the velocity field[10–13]. The propagation of the tracer in the vessel can be described with a transport partial differential equation. In a recent work, we have investigated a method to reconstruct the velocity field from Radon projections perpendicular to the vessel that leverages this equation as a regularization term [14]. The computational times to apply this variational inversion with the adjoint method remain very high.

Recent advances in deep learning have demonstrated success in addressing complex inverse problems, particularly those involving parametrized Partial Differential Equations [15]. Emerging methodologies for inverse problems involve fundamental governing laws in the loss function significantly reducing the parameters search complexity [16–19]. Operator approximations with deep learning techniques that have the property of discretization invariance have been proposed in recent works [20]. Motivated by recent novel deep learning approaches and aiming to enhance the efficiency of our

CONTACT Bruno SIXOU. Email: bruno.sixou@insa-lyon.fr

former variational reconstruction scheme, we have explored a new inversion method integrating Proper Orthogonal Decomposition (POD) or Principal Component Analysis (PCA) with deep learning techniques [21]. The principle of our new method is to map dimensionality-reduced representations of the Radon projections and blood velocity field with a neural network architecture to estimate the inverse operator. The inputs and outputs of the networks are expanded in terms of basis functions spanning linear spaces. They are calculated offline from snapshots obtained with a sampling of the parametric Radon projection or velocity spaces. The method outperforms the variational approaches developed in our former work in terms of accuracy and online computation times. We have demonstrated the efficiency of the framework through realistic simulations carried out with stationary velocity fields. This work is in line with many recent works devoted to the proof-of-concept data-driven modeling of cardiovascular flows [22]. Some deep learning models for 3D aortic pressure and velocity flow fields have been described in other works [23,24].

Yet, the proposed method has some limitations. The shape of the vessel is fixed and we have only considered steady velocity fields. The artery flow field is also determined by the inflow velocity, which changes throughout the cardiac cycle and it is crucial to systematically account for the unsteady velocity variations. It is well-known that shape variations are crucial factors affecting velocity fields and the derived hemodynamic parameters. Velocity, shear stress and pressure drop strongly scale with the diameter. The aortic flow fields are highly sensitive to geometric and topological variations [25].

In this work, our aim is to generalize our former deep learning approach to varying shapes and unsteady flows. The principle of the method is to use dimension reductions for the input mesh and Radon projections with linear or nonlinear PCA to obtain the time dependent velocity field. Then we investigate several methods to fuse the information from the mesh and Radon projections. High-fidelity studies require a high spatio-temporal resolution [26] and it is time-consuming and challenging to obtain a large dataset. We thus test the approach on scarse and large datasets with different resolutions. We compare several neural network architectures to solve our inverse problem and we show the effectiveness of the fusion methods.

We structure this paper as follows. In section 2, following the introduction, the methodological tools will be presented: the inverse problems with a PDE constraint, the linear and nonlinear dimension reduction techniques, and the deep learning approaches to approximate the unknown velocity field with the fusion of the reduced coefficients are described. In section 3, simulation details are presented with the mesh deformation scheme, finite elements tools, neural network architectures and optimization strategies. The results for blood flow reconstruction leveraging these methods to tackle the inverse problem are elaborated in Section 4, followed by a discussion of the findings and a conclusion.

2. General methodological outline

In this work, we develop a proof-of-concept pipeline that uses the Radon projections and the mesh coordinates to infer the velocity field in a new vessel. The method is based on linear and nonlinear dimension reduction methods like PCA and kernel PCA described in the following. The pipeline uses randomised perturbations of a mesh and parametrized velocity fields to produce a large data set of geometries and simulated blood flows.

2.1. Inverse problem statement

We consider the inverse problem of the blood flow reconstruction which takes advantage of the Radon projections perpendicular to the vessel as elaborated in [21]. In this work, a set of model parameters μ in the set $\mathcal{M}$ is used to parametrize the flow field. Two-dimensional projections are acquired perpendicularly to the z axis which is the main vessel axis direction. The lumen of the vessel and the simulation time are referred to $\Omega \in \mathbb{R}^3$ and $[0, T]$. We consider then in this domain a convection-diffusion for the density of the tracer $f(\mathbf{x}, t)$:

$$e(f, \mathbf{V}, \mu) = \frac{\partial f(\mathbf{x}, t)}{\partial t} + \mathbf{V} \cdot \nabla f(\mathbf{x}, t) - D \triangle f(\mathbf{x}, t) = 0, \tag{1a}$$

$$f(\mathbf{x}, t) = f_{in} \quad \forall x \in (\mathcal{D}) \;\; \forall t \in [0, T], \tag{1b}$$

$$f(\mathbf{x}, 0) = 0 \quad \forall x \in \Omega - (\mathcal{D}). \tag{1c}$$

In this model, $\mathbf{V} = (u, v, w) : [0, T] \times \Omega \to \mathbb{R}^3$ represents the velocity field obtained with the Navier-Stokes (NS) equations. The inlet boundary where the contrast agent enters the vessel with the density f_{in} is a disk $(\mathcal{D})$ in the plane $z = 0$.

The modelization of a realistic blood velocity field is carried out with the Navier-Stokes equation. The equations have been studied extensively. The incompressible unsteady Navier-Stokes equation can be written as [27]

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V}.\nabla)\mathbf{V} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{V} \;\; in \;\; \Omega \times [0, T], \tag{2a}$$

$$\nabla \mathbf{V} = 0 \;\; in \;\; \Omega \times [0, T]. \tag{2b}$$

where p the pressure and $Re = \frac{UL}{\nu}$ is the Reynolds number defined by a characteristic length L, U a characteristic velocity and ν the kinematic viscosity. Initial and boundary conditions must be considered to define the solution. For the boundary solution, will use a zero pressure outlet condition, and a parabolic or flat inlet velocity $\mathbf{V}(z = 0, t)$ parametrized by the parameter μ_1, as detailed in the following.

2.2. Data reduction with PCA and kernel PCA

The algorithmic scheme and the numerical implementation of the deep learning method are detailed in the following section. The first step is a dimension reduction applied to the Radon projections Rf or to the mesh nodes coordinates with Proper Orthogonal Reduction (POD), named also Principal Component Analysis (PCA). We also investigate the nonlinear kernel PCA technique for the mesh geometries. Dimensionality reduction refers to the problem to map high-dimensional data points to low dimensions such that as much structure as possible is preserved.

2.2.1. Linear dimension reduction with PCA

Let g denote the scalar field, depending on the parameter μ, on which we want to perform the linear dimension reduction. This field corresponds to the Radon projections, the components of the velocity field or the mesh coordinates. The vectors $g(\mu)$ are vectors of $\mathbb{R}^m$ where m is the number of coefficients of the Radon projections or the number of points of the Finite Element (FE) approximation and it depends on a parameter $\mu \in \mathcal{M}$. The first idea is to characterize the manifold

$$\mathcal{M} = \{g(\mu), \mu \in \mathcal{M}\} \tag{3}$$

as low dimensional linear subspace [28]. In order to characterize the manifold, it is populated by N_s samples or snapshots. The corresponding solutions $\{y_j\}_{1\leq j\leq N_s} = \{g(\mu_j)\}_{1\leq j\leq N_s}$ are computed solving the full-order system. These solutions constitute a training set that is going to be used to construct the reduced order model. The calculated basis vectors giving the essential information on the data are spanning linear finite dimensional spaces that minimize the empirical projection errors, with dimensions N_R, N_V^3, N_m for the Radon projections, the velocity field and the mesh coordinates respectively.

The snapshots in the training set are collected in a matrix, namely $Y = [y_1, ...y_{Ns}] \in \mathbb{R}^{m\times N_s}$. In order to eliminate the redundancies and to further reduce the dimension of the linear subspace where the solution is sought, the principal component analysis (PCA) technique is used, based on the singular value decomposition (SVD) of Y [29]. It produces two unit matrices $\Phi_g \in \mathbb{R}^{N_s\times N_s}$ and $\Psi_g \in \mathbb{R}^{m\times m}$ and a diagonal matrix $D = \mathsf{diag}(\sigma_1, ..., \sigma_d) \in \mathbb{R}^{d\times d}$ such that:

$$\Psi_g^t Y \Phi_g = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} = \Sigma \in \mathbb{R}^{m\times N_s} \tag{4}$$

The singular values are sorted in descendent order, $\sigma_1 \geq \sigma_2 \geq .... \geq 0$ Considering the first N_g columns of the matrix Ψ_g, the space span$(\psi_{g,1}, \psi_{g,i}, ..., \psi_{g,N_g}\ )$ is an approximation to the manifold collecting most of the information. The PCA enables the data to be represented using only the most significant modes. In the following, we have used the number of modes that capture cumulatively more than 90% of the variance of flow features, Radon projection features, or mesh variations.

2.2.2. Nonlinear dimension reduction with kernel PCA

In the former section, we have presented the proper orthogonal decomposition method. The number of degrees of freedom is readily reduced to the dimension of the linear space using a reduced basis technique obtained with the set of solutions of the parametric problem. Nonlinear dimension reduction techniques have been used recently in works devoted to data-driven modeling of cardio-vascular flows [30]. It is found that the nonlinear reduction methods like autoencoder architectures achieve superior results. For mesh geometries, nonlinear reduction schemes are leveraged in [22]. In this work, t-Stochastic Neighbor Embedding (t-SNE) method has been used for mesh reduction. The (t-Stochastic Neighbor Embedding) t-SNE is a statistical method for visualising high-dimensional data by embedding each N-dimensional data point in a reduced space, typically of two or three dimensions [31]. This technique belongs to the class of non-parametric techniques: it provides a mapping of the given data points

only, without an explicit mapping prescription on how to project further points which are not contained in the data set to low dimensions.

For complex data, the PCA strategy can be extended to use a nonlinear dimensionality reduction technique, namely, the kernel principal component analysis (kPCA), in order to find a nonlinear manifold, with an expected lower dimension [32–35]. kPCA is an algorithmically simple approach, proposing an explicit and analytical forward mapping from the full-order space to the reduced-order space. More specifically, kernel principal component analysis (kPCA) is a nonlinear dimensionality reduction technique, based on applying PCA to a transformed training set, with transformation defined by a kernel $k(.,.)$. The kernel is used to build a Gram matrix $G \in \mathbb{R}^{N_s \times N_s}$ with $G_{ij} = k(y_i, y_j)$ for $i, j = 1, ..., N_s$ from the snapshots $(y_i)_{1 \leq i \leq N_s} = (g(\mu_i))_{1 \leq i \leq N_s}$. G is symmetric positive definite, its SVD decomposition can be written as:

$$G = V \Sigma V^t \tag{5}$$

The square diagonal matrix Σ has entries $(\sigma_i^2)_{1 \leq i \leq N_s}$. The dimensionality reduction is performed by selecting the reduced dimension N_g. Matrix $V^* \in \mathbb{R}^{N_s \times N_g}$ is taken as containing the first N_g columns of the matrix V. Similarly as with PCA, we define the maping $g : \mathbb{R}^m \to \mathbb{R}^{N_s}$ with $g(x)_i = k(x_i, x)$ for $i = 1, ...N_s$. The kPCA forward mapping can be defined as

$$F : \mathbb{R}^m \to \mathbb{R}^{N_g}, \tag{6a}$$

$$x \mapsto z = (V^*)^t g(x). \tag{6b}$$

The choice of an appropriate kernel function is relevant to guarantee the efficiency and the accuracy of the method.

2.3. Deep learning approaches

2.4. Blood flow reconstruction with PCA-net

In our former work, we have used a fully connected network to connect the PCA coefficient of the steady velocity field and the PCA coefficients of the Radon projections. The approach to obtain an approximate solution of the inverse problem is detailed in [21]. The neural architecture is displayed in Figure 1. The methodology integrates dimension reduction with deep neural architectures to learn data-driven maps between the input and output function spaces, the Hilbert space of the velocity $\mathcal{L}_V = L_2(\Omega \to \mathbb{R}^3)$ and the Hilbert space of the Radon projections $\mathbf{V}$ and $\mathcal{L}_p = L_2([0, T] \times [-a, a] \times [0, z_{max}] \to \mathbb{R})$ where a and z_{max} are transverse and longitudinal lengths. The Hilbert space $\mathcal{L}_V$ is decomposed into the product of three Hilbert spaces $\mathcal{L}_V = \mathcal{L}_V^x \times \mathcal{L}_V^y \times \mathcal{L}_V^z$ for the components. The principle of the method is to approximate the nonlinear inverse operator $\Psi = \Psi^x \times \Psi^y \times \Psi^z : \mathcal{L}_p \to \mathcal{L}_V$ from velocity and Radon projection samples. In this setting, the velocity components are $V_x = \Psi^x(p)$, $V_y = \Psi^y(p)$ and $V_z = \Psi^z(p)$, for Radon projections p . The approximation for this operator is a neural network $\Psi_\theta = \Psi_\theta^x \times \Psi_\theta^y \times \Psi_\theta^z : \mathcal{L}_p \to \mathcal{L}_V$, where $\theta \in \mathbb{R}^D$ corresponds to the D neural network parameters. In order to estimate the optimal

parameters θ^*, we consider the loss function function $\mathcal{L} : \mathbb{R}^D \to \mathbb{R}$:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \frac{E_n^x + E_n^y + E_n^z}{\|\Psi^x(p_n)\|_2 + \|\Psi^y(p_n)\|_2 + \|\Psi^z(p_n)\|_2} \tag{7}$$

where N denotes the amount of data, and $E_n^x = \|V_x(p_n) - \Psi_\theta^x(p_n, \theta)\|_{L_2}^2$, $E_n^y = \|V_y(p_n) - \Psi_\theta^y(p_n, \theta)\|_{L_2}^2$ and $E_n^z = \|V_z(p_n) - \Psi_\theta^z(p_n, \theta)\|_{L_2}^2$ the reconstruction errors incurred by using the network along x, y and z respectively. The approach investigated is similar to the general methodology developed in [20] for data-driven approximation of input-output maps between infinite-dimensional spaces. We define the PCA subspaces approximating $\mathcal{L}_p$ and $\mathcal{L}_V$ and spanned with the orthonormal bases $\{\psi_{Rf,i}\}_{1\le i\le N_R}$ for $\mathcal{L}_p$ and $\{\psi_{V,i}\}_{1\le i\le N_V^3} = \{\psi_{V,ix}^x\}_{1\le ix\le N_V} \times \{\psi_{V,iy}^y\}_{1\le iy\le N_V} \times \{\psi_{V,iz}^z\}_{1\le iz\le N_V}$ for $\mathcal{L}_V$ of rank N_R and N_V for each velocity component. Then we construct a mapping α that takes as input the PCA coefficients of the Radon projections and outputs the basis coefficients of the velocity. The input-to-output map is based on a fully connected network (FC) and it will be referred to as PCA-FC in the following. For the component along x of the solution map Ψ_θ, it is possible to use the expansion:

$$\Psi_\theta^x(p, \theta)(M) = \sum_{j=1}^{N_V} \alpha_j^x(F_{\mathcal{L}_p} p, \theta)\psi_{V,j}^x \tag{8}$$

The function $\alpha_j^x : \mathbb{R}^{N_R} \times \mathbb{R}^D \to \mathbb{R}$, for $1 \le j \le N_V$, maps the PCA basis coefficients of the approximation of p , $F_{\mathcal{L}_p} p = \{(\psi_{Rf,i}, p)\}_{1\le i\le N_R}$ and the ones corresponding to the velocity along x, $F_{\mathcal{L}_V}^x V$. Similar expansions with data-driven bases hold for the other directions.

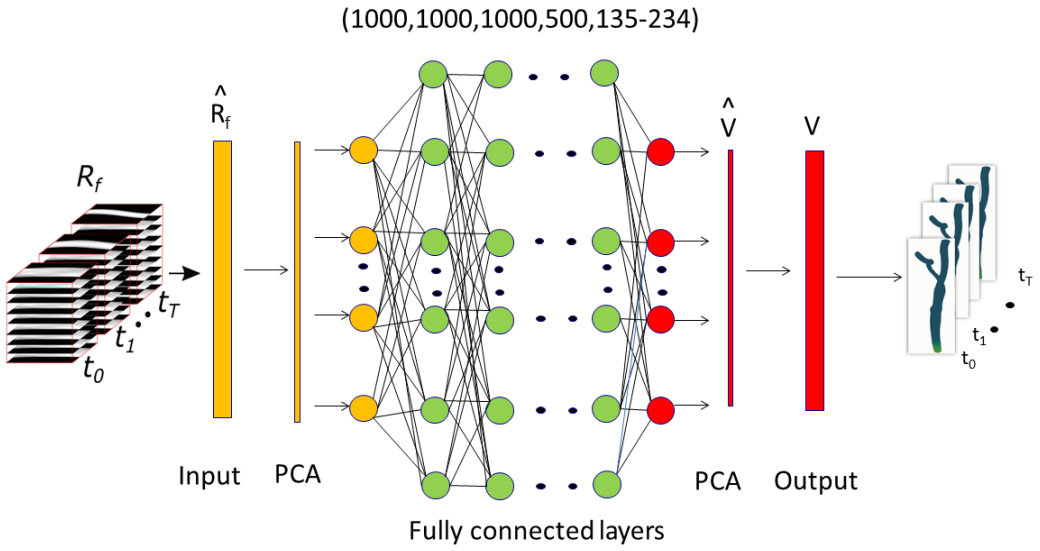


Figure 1. Fully connected network connecting the time dependent Radon projections and the time dependent velocity field.

.

2.5. *Information fusion with deep learning architectures*

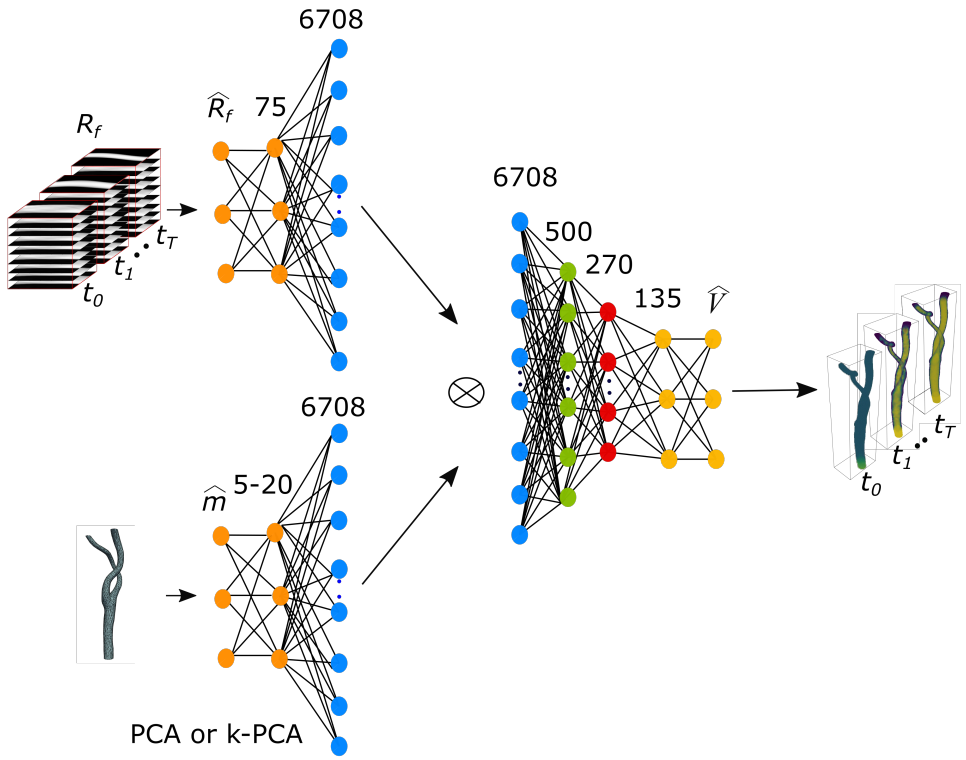


Figure 2. Fully connected network connecting the time dependent Radon projections, the mesh and the time dependent velocity field: Fusenet 1.

.

The objective of this work is to leverage the mesh information and to fuse the reduced features from the Radon projections and from the mesh to estimate the velocity field. Information fusion has been investigated with deep learning techniques [36]. We will use combination of linear or nonlinear dimension reductions, concatenation of features, features dot products and fully connected networks for an effective information fusion. The deep learning architectures used for information fusion are presented in Figure 2 and Figure 3. The deep learning architectures will be described precisely in the numerical section. We summarize in this subsection the main principles of the methods. In the diagrams shown in Figures 2 and 3, the inputs are the Radon transform and the coordinates of the meshes. Some feature dimension reduction is performed with PCA or kernel PCA for the mesh and with PCA for the Radon projections. To obtain the PCA coefficients of the velocity field, we have used several classical techniques to fuse the features obtained with the former embeddings to generate a single representation based on dot product or concatenation. On these diagrams, the reduced coordinates of the Radon projection, velocity field and mesh are denoted as $\hat{Rf}$, $\hat{V}$ or $\hat{m}$ respectively. Let N_R, N_V and N_m denote the number of modes considered to describe the Radon projections, each velocity field component and the simulation mesh repectively. The PCA coefficients are sampled for N_s times regularly spaced in the time interval [0,T], with a time step Δt. At the end of this post-processinng step, N_m, $N_R \times N_s$ and $3 \times N_V \times N_s$ coefficients are concatenated as vectors for the input and ouput of the network.

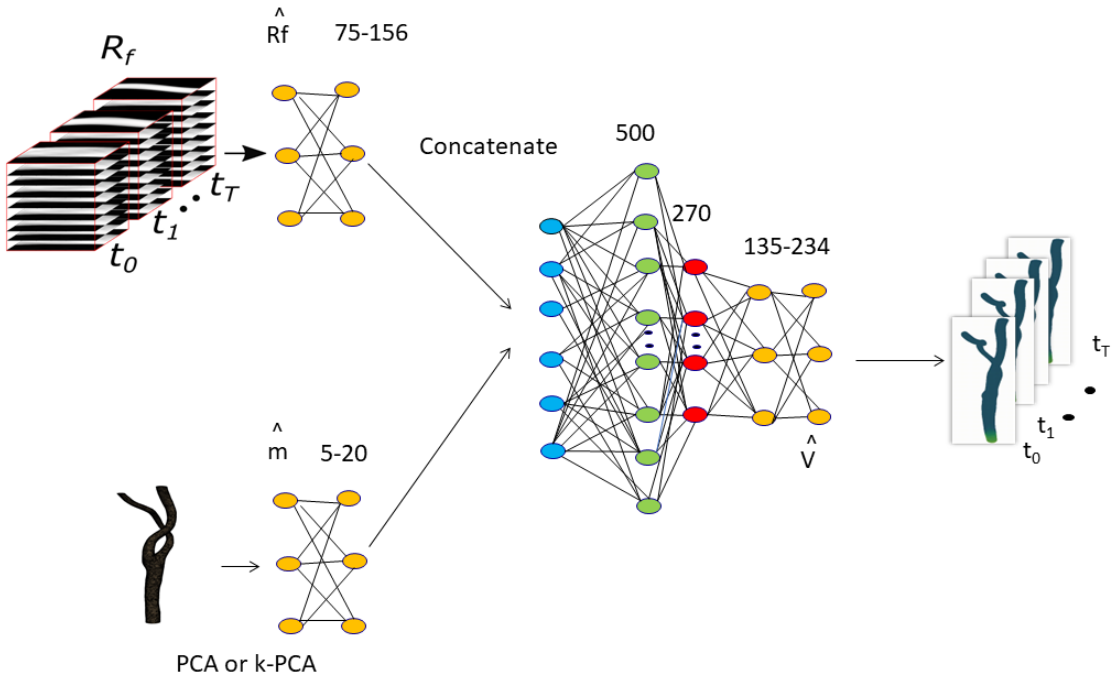


Figure 3. Fully connected network connecting the time dependent Radon projections, the mesh and the time dependent velocity field: Fusenet 2.

.

The two fusion architectures are referred to as Fusenet1 and Fusenet2.
FuseNet1: In the first method denoted as FuseNet1 and displayed in Figure 2, we use a pairwise dot product of the vector features obtained with the two embeddings. The size of the dot product vector corresponds to the number of nodes in the mesh. Then a fully connected network is used to output the reduced velocity coordinates and a reverse PCA is performed to obtain the velocity field. This neural network architecture has been used when the number of vertices in the mesh is small. For simulations with a high spatial discretizations and many nodes, the number of parameters to tune grows, this model does not perform well and the network is prone to overfitting.
FuseNet2: In the second method denoted as FuseNet2 and displayed in Figure 4, we concatenate the vectors obtained with the mesh or Radon projection embeddings. Then fully connected layers are used to to generate the PCA coefficients of the velocity field. These PCA coefficients are then used to produce the flow field solutions for the considered geometry. Let m a mesh in the mesh manifold $\mathcal{M}$, the information fusion operator for both methods can be written as:

$$\Psi^x_\theta(p, m, \theta)(M) = \sum_{j=1}^{N_V} \alpha^x_j(F_{\mathcal{L}_p}(p), F_{\mathcal{M}}(m), \theta)\psi^x_{V,j}(M) \tag{9}$$

where $F_{\mathcal{M}}$ denotes the mesh coordinates dimension reduction operator. We focus here on simple fusion information methods, more elaborate approaches based on attention

networks may also be considered [36].

3. Simulations details

3.1. Mesh deformation scheme

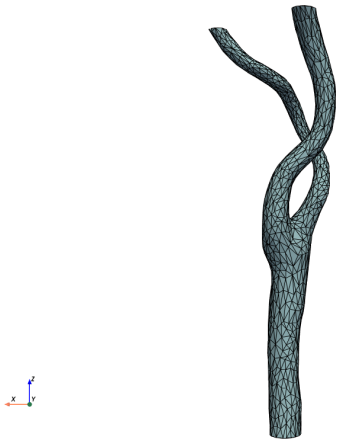
The 3D vessels displayed in Figure 4.a, 4.d were used to generate synthetic data by applying random spatial perturbations to the original mesh. To prevent unnatural, discontinuous geometric differences within each mesh phantom, the perturbations are based on the amplitude of a sinusoid, which distributes the perturbation lengthwise. Variations in shape are due to synthetic perturbations applied to the artery diameter, the function of which is a sinusoid with randomised amplitude, frequency, phase and vertical displacement. This ensures smooth, continuous variation along the length of the artery. The scaling ratio s for the coordinates of the surface points can be written as:

$$s(z) = 0.9 + 0.2 * r * cos(2\pi z/L + \phi) \tag{10}$$

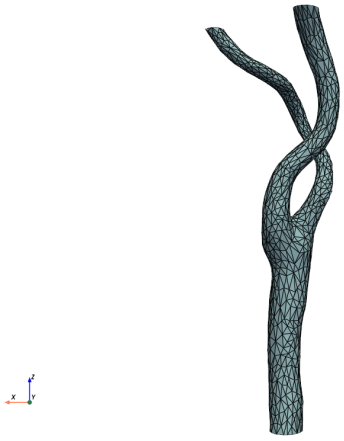
where r is a scalar randomly sampled between 0 and 1, L is a random length sampled between in the range [0.4,0.8] times the vessel length, and the phase ϕ randomly sampled between [0, $\pi/2$]. The mesh manifold samples are thus parametrized with $\mu_2 = (r, L, \phi)$. Starting for an initial mesh geometry, we have used a mesh deformation scheme based on radial basis functions (RBF) [37–39]. RBF shape parametrization technique is based on the definition of a map, $\mathcal{M}(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}^m$, defined as $\mathcal{M}(\boldsymbol{x}) = p(\boldsymbol{x}) + \sum_{i=1}^{\mathcal{N}_C} \gamma_i \varphi(\|\boldsymbol{x} - \boldsymbol{x_{C_i}}\|)$ where $\mathcal{N}_C$ the number of control points, $p(\boldsymbol{x})$ is a low degree polynomial term, $(\gamma_i)_{1 \leq i \leq \mathcal{N}_C}$ are some weights and $\varphi(\|\boldsymbol{x} - \boldsymbol{x_{C_i}}\|)$ a radial function based on the Euclidean distance between the control points position $\boldsymbol{x_{C_i}}$ and $\boldsymbol{x}$. In the following, we have considered Gaussian splines. The principle of the method is that after the computation of the weights and the polynomial terms from the coordinates of the control points before and after the deformation, we can deform all the points of the mesh accordingly. The proposed mesh deformation strategy was implemented with the Pygem library [40]. Some examples of meshes are displayed in Figure 4.b and Figure 4.c for the very coarse case. With this method, 1000 phantom 3D meshes were generated with a coarse discretization, but two different refinement levels, as detailed in the following.

3.2. Finite elements simulation details

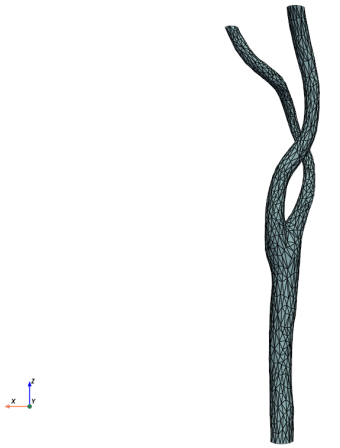
In order to evaluate the performance of our approach, we generate the reference numerical solutions by using the finite element Fenics library. The numerical experiments have been performed with the type of generated realistic phantoms generated with the previous method and displayed on Figure 4. The computational domain is a vessel with a total streamwise length of 10 cm with a bifurcation and an average diameter of approximately 10 millimeters. We have used the Fenics software [41,42] to perform the finite elements simulations. In order to test the method, we have created two datasets. The first one, denoted as Dataseta in the following, is a large set with a coarse spatio-temporal resolution. The second dataset, referred to as Datasetb, is scarse and corresponds to a less coarse spatio-temporal resolution. The spatial discretization remains coarse in comparison with the resolution standards in the field.



(a) Original Coarse mesh.



(b) Deformed Coarse mesh, exemple 1.



(c) Deformed Coarse mesh, exemple 2.

(d) Original mesh with higher spatial resolution.

Figure 4. Original meshes and deformed meshes obtained with the deformation scheme .

Dataseta

This dataset corresponds to a coarse resolution.The time step is set as $dt = 0.01s$ with a total run time of $T = 0.5s$ and $N_t = 50$ time steps. The velocity fields to be reconstructed $\mathbf{V}^*(\mathbf{x}, t)$ are calculated with the non-stationary incompressible Navier-Stokes equations with Re=200 [21]. The simulation domain is discretized with $P1$ finite elements and 2236 vertices. The time step and the number of nodes in this simulation are small but similar to the ones used for simple CFD simulations [30].

Standard non-slip conditions were applied at the wall. Boundary conditions included a fixed, parabolic or flat velocity inlet and zero pressure outlet condition for all cases to produce realistic data. The inflow velocity was set as follows:

$$\begin{cases} \mu_1 + \mu_1 sin(\pi t) & for \quad t \quad \leq \quad 0.5 \\ \mu_1 + \mu_1(3/2 - 0.5cos(2\pi(t - 0.5))) & for \quad t \quad \geq \quad 0.5 \end{cases} \tag{11}$$

with a velocity parameter μ_1 which will be sampled as described in the following. It is well-known that stable spatial discretizations are required for Navier-Stokes sim-

ulations. The Taylor-Hood ($P2-P1$) finite elements [43] are used in this study. The algorithm used for the non-stationary Navier-Stokes velocity field is the Incremental Pressure Correction scheme in Fenics[41,42]. This space-time grid generates a set of coarse synthetic velocity fields.

Datasetb

The spatial discretization grid includes 64000 nodes with P1 finite elements similar to the one used in [14]. More refined meshes should be considered for very high-resolution simulations [26] . The time discretization is chosen such that dt=0.002 which is similar to the time step used in [44]. Following [45], the inlet velocity is defined as the waveform:

$$\begin{cases} \mu_1 sin(\pi t/T) \quad for \quad t \quad \leq \quad T \\ \mu_1 \frac{\pi}{T}(t-T)exp(-\kappa(t-T)) \quad for \quad T_c > t \quad > \quad T \end{cases} \tag{12}$$

where μ_1 is a constant amplitude, T is the opening-time of the valve, T_c the total duration of the cardiac cycle and $1/\kappa$ represents the typical time for the closing of the valve. In the following, the following values have been chosen $T = 0.4s$, $T_c = 0.8s$ and $\kappa = 70s^{-1}$. In this case, the number of time steps is thus $N_t = 400$.

The simulation of the transport equation was performed as explained in [21]. The inlet tracer concentration is set to $1kg/m^3$ with a diffusion coefficient $1e-5$. The method developed yields similar results for diffusion coefficients in the range [1e-5,1e-4] but it does not perform well for high Peclet numbers. The Radon projections perpendicular to the main vessel direction are evaluated with the Scipy library [46,47] for Nz=128 values equally spaced. The Radon projections are measured for N_p =100 values, for $N_\theta = 180$ projection angles in the angular range [0,π], for each time step $t_p = pT/N_t$ in the time domain, [0, T]. A Gaussian white noise with peak-to-peak signal-to-noise ratio (PSNR) between 0 and 20 dB is used to obtain noisy projections, which is defined as:

$$PSNR = 20log(\frac{S_{max}}{n_{max}}) \tag{13}$$

where S_{max} is the maximum signal amplitude and n_{max} the maximum noise amplitude.

3.3. Dataset generation and network architecture

In this subsection, we present in detail the dataset generation and the network architecture. In the first step, 1000 synthetic meshes were generated using the mesh deformation scheme presented above and various μ_2 values. PCA or nonlinear kPCA reduction was then performed and 10 synthetic meshes were sampled uniformly from the first 5 PCA or kPCA coefficients. With these ten sampled geometries, the non-stationary Navier-Stokes equations were solved with an inflow velocity field parametrized by the parameter μ_1 randomly sampled between 0 and 0.6m/s. For the coarse meshes and Dataseta, for each geometry, 130 velocity fields were generated. The total number of velocity field snapshots is 1300. For the higher resolution meshes and Datasetb, for each geometry, 13 velocity fields were generated. The total number of velocity field snapshots is 130. This scarse dataset was generated because of the larger computational cost required by higher spatio-temporal resolutions. As detailed in the following, the computational time needed to generate the samples of the two datasets is similar. This computational fluid dynamics and Radon projections datasets were then post-

processed to obtain eigen functions for $N_m = 5$, $N_R = 6$, $N_{Vx} = 3$, $N_{Vy} = 3$, $N_{Vz} = 3$ modes for the mesh, the Radon projections, and velocity components respectively. The PCA modal basis coefficients are estimated every $\Delta t = 0.03s$ and vectors of size $(N_R \times N_s, 3 \times N_V \times N_s) = (75, 135)$ and $(N_R \times N_s, 3 \times N_V \times N_s) = (156, 234)$ are obtained to describe the evolution of the Radon projections and velocity field for simulations performed with the Dataseta or the Datasetb respectively.

The kernel PCA was implemented with the Scikit library [47] with a radial Gaussian kernel function with $\gamma = 0.1$ as hyper-parameter and 5 components. The γ parameter was optimized after numerical experimentation, trying to accumulate a significant amount of variance in the data. The number of PCA modes for each variable, velocity, Radon and mesh geometry, have been chosen to approximate more than 95% of the variance. The nonlinear kernel PCA captures approximately 2% more variance than the linear PCA. When more basis functions are included in the PCA approach, the errors stagnates and the basis can be considered as nearly optimal for the inversion for the range of the flow patterns investigated. The training/validation dataset was composed of 1000 randomly selected meshes, Radon and velocity fields, with the remaining 300 going into the test set for the Dataseta. For the Datasetb, the numbers of training/test samples were 100/30. The training was implemented using the open-source library Keras 2.2.5 with Tensorflow backbone and performed on an HPC cluster with 4 processors. Model training lasted 100 epochs and training/validation loss was monitored to prevent overfitting. For the simple PCA net, we designed the neural network architecture with 4 hidden layers of width $(500, 1000, 1000, 1000)$ (Figure 1). For the fusion networks, we have tested several architectures to obtain the best results with the minimum number of training parameters. The best results have been obtained with the fully connected neural networks displayed in Figure 2 and Figure 3. The numbers of units in the hidden layers are indicated on these figures. The velocity field, Radon projection or mesh reduced coordinates are denoted as $\hat{V}$, $\hat{Rf}$, $\hat{m}$ respectively. Relu activation functions are used in the layers, except in the last layer. For the neural architecture used with the Dataseta, the number of training parameters are 3238195 for the classical PCA net already studied in [21], 2385871 for Fusenet1 and 225975 for Fusenet2 respectively with a reduced dimension of 5 for the mesh. The network Fusenet1, displayed in Figure 2, is used for the Dataseta with a coarse spatial resolution. An increase in the number of modes in the mesh leads to an increase of the number of parameters in the network, and the model does not perform well. For the networks used for the Datasetb, the number of trainable parameters are 3473716 and 378825 for the classical PCA net and Fusenet2 repectively. The number of hidden layers, the number of weights per hidden layer, the nonlinear activation functions, the batch size for mini-batch gradient descent, the number of epochs to perform training have been optimized. The network is trained on 100 epochs with an Adam optimizer, and a decreasing learning rate between 10^{-3} and 10^{-4}.

4. Results and discussion

In this section, we present the numerical results and we compare the fusion methods against former PCA-net method. We will test the proposed schemes on non-stationary velocity fields. We will visualize the recovered solutions and analyse the reconstruction errors of the various approaches.

4.1. Results

In order to illustrate qualitatively the results, we first display in Figure 5 some horizontal sections of the density of the tracer which is exploited to reconstruct the velocity field with the less coarse spatial resolution simulations. We show examples of the reconstructed velocity components in Figure 6 to 11 obtained with the Fusenet2 method based on kernel PCA for some test velocity fields and some test mesh from the dataset Datasetb. Cross-sections of the reconstructed and of the true velocity are displayed on these figures, for the selected simulation times and position along the z axis. We also present the error maps on these figures. Some streamlines for the ground truth and reconstructed velocity fields can be seen on Figure 12. The Fusenet1 and Fusenet2 models performs also well for the coarse resolution. These figures show qualitatively that good reconstruction results are obtained with the proposed approaches.

Table 1. Reconstruction errors for the velocity components and the velocity norm for the methods investigated and the coarse spatio-temporal resolution Dataseta, PSNR=20 dB.

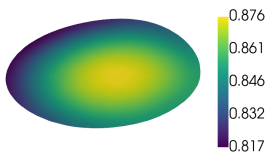
Method	V	Vx	Vy	Vz
PCA-net	0.0377(0.0081)	0.0878 (0.0220)	0.1235 (0.0297)	0.0222 (0.0047)
FuseNet1, PCA	0.0079 (0.0039)	0.0191 (0.0100)	0.0225 (0.011)	0.0049 (0.0024)
FuseNet1, kPCA	0.0069 (0.0033)	0.0179 (0.0087)	0.0191 (0.0091)	0.0040 (0.002)
FuseNet2 PCA	0.0129 (0.0035)	0.0314 (0.0106)	0.0377 (0.0086)	0.0079 (0.0021)
FuseNet2, kPCA	0.0094 (0.0043)	0.0250 (0.0123)	0.0225 (0.0094)	0.0058 (0.0027)

Table 2. Reconstruction errors for the velocity components and the velocity norm for the methods investigated and the finer spatio-temporal resolution Datasetb, PSNR=20 dB.

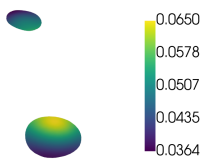
Method	V	Vx	Vy	Vz
PCA-net	0.0491(0.0205)	0.1141 (0.0552)	0.1605 (0.0742)	0.0288 (0.0118)
FuseNet2 PCA	0.0167 (0.0072)	0.0490 (0.0222)	0.0408 (0.0180)	0.0103 (0.0044)
FuseNet2, kPCA	0.0127 (0.0090)	0.0338 (0.0240)	0.0305 (0.0188)	0.0078 (0.0055)

Table 3. Reconstruction errors for the velocity components and the velocity norm for Fusenet1 architecture and kPCA and different noise levels for the Dataseta.

PSNR	V	Vx	Vy	Vz
20 dB	0.0069 (0.0033)	0.0179 (0.0087)	0.0191 (0.0091)	0.0040 (0.0020)
0 dB	0.0653 (0.0540)	0.0697 (0.0536)	0.0698 (0.0524)	0.0642 (0.0549)



(a) Tracer density for $z = 0.15$.

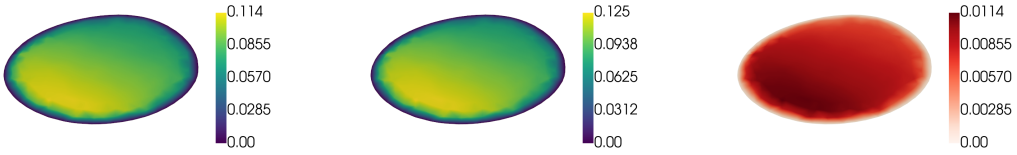


(b) Tracer density for $z = 0.17$.

Figure 5. Examples of tracer denisty field cross-sections, t=0.15s, inlet tracer concentration 1.0kg/m^3, $\mu_1 = 0.5m/s$.

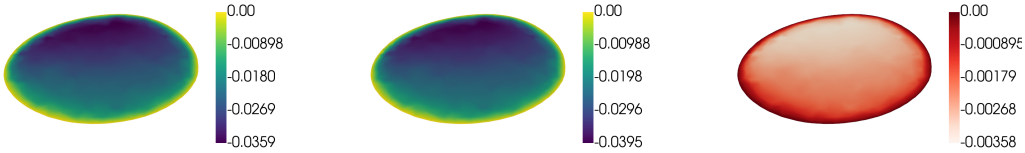
Table 4. Reconstruction errors for the velocity components and the velocity norm for Fusenet2 architecture and kPCA and different noise levels for the Datasetb.

PSNR	V	Vx	Vy	Vz
20 dB	0.0096 (0.0066)	0.0271 (0.0182)	0.0251 (0.0189)	0.0057 (0.004)
10 dB	0.0915 (0.0820)	0.0976 (0.0804)	0.0978 (0.0786)	0.0902 (0.0823)



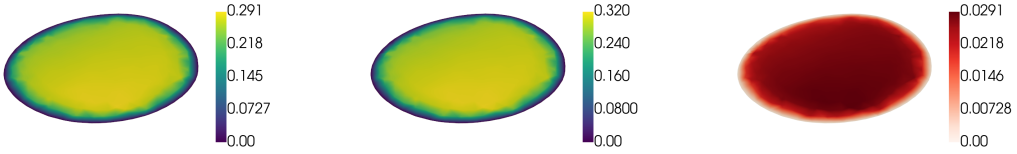
(a) Ground truth velocity along x. (b) Reconstructed velocity along x. (c) Error map for velocity along x.

Figure 6. Distribution along x of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.155m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.



(a) Ground truth velocity along y. (b) Reconstructed velocity along y. (c) Error map for velocity along y.

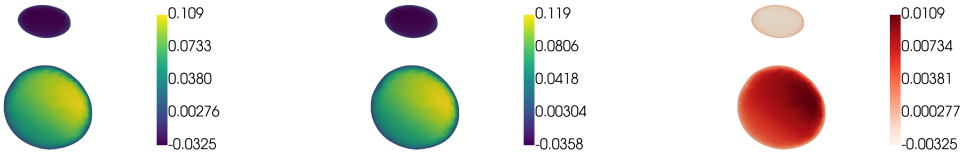
Figure 7. Distribution along y of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.155m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.



(a) Ground truth velocity along z. (b) Reconstructed velocity along z. (c) Error map for velocity along z.

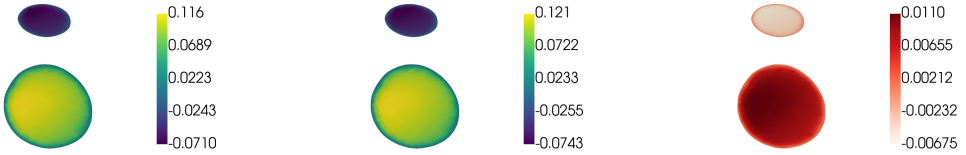
Figure 8. Distribution along z of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.155m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.

The reconstruction errors are summarized in table 1 and table 2 for the velocity components and the velocity norm for the various reconstruction strategies for a low



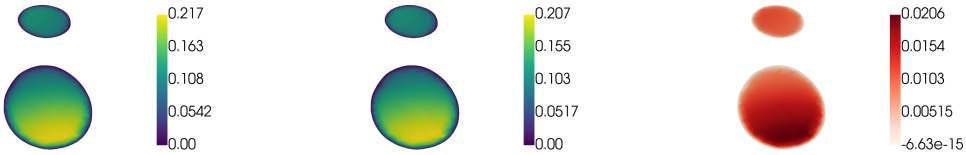
(a) Ground truth velocity along x. (b) Reconstructed velocity along x. (c) Error map for velocity along x.

Figure 9. Distribution along x of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.165m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.



(a) Ground truth velocity along y. (b) Reconstructed velocity along y. (c) Error map for velocity along y.

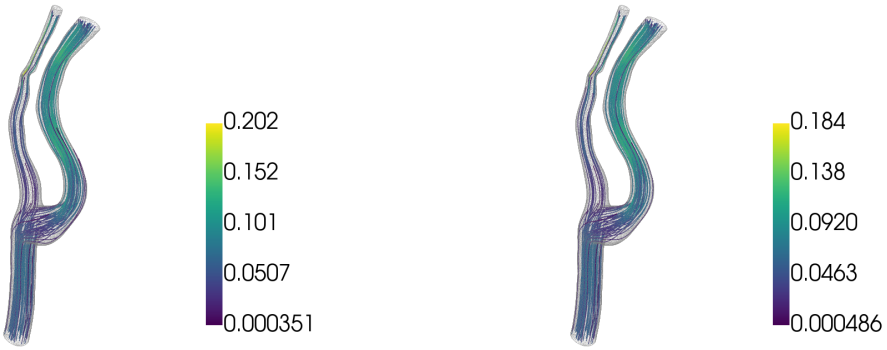
Figure 10. Distribution along y of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.165m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.



(a) Ground truth velocity along z. (b) Reconstructed velocity along z. (c) Error map for velocity along z.

Figure 11. Distribution along z of ground truth velocity, reconstructed velocity of Fusenet2 method and its error map: slice location z=0.165m, t=0.05s, $\mu_1 = 0.25m/s$, inlet tracer concentration 1.0kg/m^3, diffusion coefficient 1e-5, PSNR=20 dB.

noise level with PSNR=20 dB, for the two datasets. The tables compare the errors for simple PCA-net, FuseNet1 and FuseNet2, for linear or nonlinear reductions for the mesh coordinates. The reconstruction errors for various values of the PSNR are displayed in table 3 and 4 for the Fusenet1 and Fusenet2 with kernel PCA method for the Dataseta and Datasetb respectively. With the fusion approaches, the reconstruction errors for the velocity show large reductions and the methods are able to give a good approximation of these unknowns fields. Table 1 and 2 show that leveraging the reduced mesh information, the reconstruction results are improved.The proposed



(a) Ground truth velocity field. (b) Reconstructed velocity field.

Figure 12. Streamlines with velocity norm of the ground truth and reconstructed velocity fields with Fusenet2 method t=0.3s, $\mu_1 = 0.5m/s$, PSNR=0 dB.

approach outperforms the simple reconstruction method proposed in our former work (Simple PCA-net) with a fully connected network connecting the PCA coefficients of the velocity and of the Radon projections, that has been used for a stationary velocity field, without the mesh information. Moreover, our results show that the better reconstruction errors are achieved with Fusenet1 based on dot product for the coarse spatio-temporal resolution and with Fusenet2 with concatenation of velocity and mesh reduced information for the less coarse one. We obtain also good reconstruction results for the transverse components of the velocity. The reconstruction degrades with the increase of the noise level. The results obtained for the less coarse spatio-temporal resolution degrade slightly with respect to the ones achieved with the coarse one with a higher standard deviation. For the same offline computational time, less snapshots have been considered in this work and the reduced order model is less accurate. The results of this section thus clearly demonstrate the effectiveness of the proposed fusion methodologies to solve our inverse problem for a non stationary velocity field and varying geometries. It should be noted that these results are obtained with fusion networks with less parameters than the simple PCA-net. The results with the nonlinear kernel PCA outperform slightly the ones obtained the linear PCA for the same number of coefficients. With the linear PCA method, 10 coefficients are necessary to reach the same level of accuracy as the one obtained with the nonlinear reduction method with 5 coefficients.

4.2. Discussion

4.2.1. Results discussion

Many studies have emphasized the impact of blood flow and stress distribution for the study of atherosclerosis diseases. In our previous research work, we proved the efficacy of a novel inversion method in reaching rich insights into blood velocity patterns, with notable enhancement in spatial resolution when compared with common medical imaging techniques like NME or ultrasound. This innovative approach is based on integration of deep learning techniques with dimension reduction schemes applied to either Radon projections or blood flow velocity fields. The proposed approach can also

achieve an accurate reconstruction of the transverse components of the blood velocity field. As detailed in our former work, the proposed method based on a coupling of PCA and deep learning significantly outperforms the classical variational methods. The classical formlation of the inverse problem with the Navier-Stokes equation as constraint is based on the adjoint method [14]. It is very time consumming, does not leverage the regularizing effect of dimension reduction methodologies and yields less robust and accurate reconstructions. They are examples of physics informed deep learning approaches. Yet, this study was restricted to stationary velocity fields and does not leverage the mesh information. The aim of this work is to propose a new inverse problem formulation capable of tackling time dependent Radon projections and vessel domains. In this new proof-of-concept study, we show that the blood flow in vessels can be reconstructed using deep learning techniques, Radon measurements, and mesh information in an automatic process, and with a good accuracy. The proposed method with a fusion of mesh information and Radon measurements information achieves lower reconstruction errors compared to the former scheme. The numerical trials conducted on practical cases proved the effectiveness of the proposed approach for robust flow estimations and in tackling complex nonlinear dynamic inverse problems. We also outline the interest of deep learning methods in solving these inverse problems characterized by parametric partial differential equation constraints and diverse geometries. Notably, this approach performs well with noisy data. Two architectures have been proposed. A choice has to be done depending on the spatio-temporal resolution and data set size. The Fusenet2 architecture with a concatenation of reduced latent variables representing Radon and mesh geometry information is more robust, easier to train and accurate for high-dimensional and scarse data, which are common in hemodyanmics. The datasets considered are small and may not be representative of all possible dynamics in the blood flow investigated. A larger and more diverse dataset could enhance the generalizability of the findings. Yet, for the very scarse dataset Datasetb, more latent coefficients are leveraged, the amount of extracted information from the data is higher and the model performance degrade only slightly. A more comprehensive representation of the underlying data is considered and meaningful flow patterns can be learned from the limited dataset. As shown on Table 4, lower noise levels have to be considered to achieve good reconstruction for the Datasetb. A small dataset exacerbates the sensivity to noise, the model may memorize the noise or outliers present in the limited training samples. In contrast, a larger dataset helps mitigate overfitting by exposing the model to a more comprehensive representation of the underlying data distribution, resulting in improved generalization to unseen examples. The ability of the proposed approach to generalize to unseen data for more complex fluid problems in the low data regime remains an open question. The approach does not require a precise knowledge of boundary and initial conditions and material properties which are required for complex biomedical Computational Fluid Dynamics simulations but very challenging to determine. The proposed data-driven method does not require the knowledge of the PDE forward mapping based on the underlying physics and the precise knowledge of the mesh geometry. In our former work, laminar, steady-state flow conditions were enforced. The study was here generalized to time dependent blood flows. We have considered not only stationary velocity fields but extended the method to more complex non stationary velocity fields with time dependent inflow waveforms. Another contribution is that the methods can account for varying blood vessels. We have extended the former approach with geometrical parametrization and with a dimension reduction applied to moving domains and meshes. The results obtained when the mesh information is taken into account outperform the ones obtained with the

minimum information based on Radon projections.

In this work, we have used dimensionality reduction techniques to produce dimensionality-reduced representations of vessel shapes, Radon projections and velocity fields. PCA and kernel PCA are suitable tools for dimensionality-reduced representations of 3D shapes. Our results suggest that linear projection-based reduced order methods may not be optimal for dimensionality reduction for complex geometries because the number of required modes increase significantly. Manifolds for real data are expected to be strongly nonlinear and one needs to make use of nonlinear techniques. Nonlinear autoencoders are known to provide a great flexibility and better approximation results than linear projection techniques for nonlinear manifolds [48–50]. In this work, we have investigated a simple dimension nonlinear dimension with kernel PCA for the meshes manifold and we obtain good reconstruction results. In the future, other approaches for creating dimensionality-reduced representations of the varying mesh structures could be explored with autoencoders [23]. Moreover, we have investigated the variation of one mesh from one patient. It could be interesting to have several meshes for different subjects in the dataset and study other nonlinear dimension reductions or extension of basic PCA with a mixing of linear subspaces [51,52]. The nonlinear reduction techniques could also be exploited for the velocity field. In [30], the authors compares various nonlinear dimension reduction techniques in cardiovascular flows. It is shown that for the spatial reduction, nonlinear autocencoders achieves the best overall performance. In this work, for the sake of simplicity, we have focussed on PCA for the velocity reduction, which does not require the investigation of multiple architectures. The motivation to use PCA is that it is the simplest dimension reduction approach with no hyperparameters and we can extend our former work based on a simple PCA-net. The main objective of the work was to fuse the Radon and mesh geometry information by building on the already proposed approach. PCA has the advantage to remains interpretable through its formulation in terms of a modal basis expansion. Nonlinear embeddings for the velocity field may enhance our results and they will be investigated in future research. They are sometimes criticized for their limited interpretability and susceptibility to overfitting and they require to tune more parameters. Mode decomposing autoencoders [30] could be useful to obtain interpretable spatial modes.

4.2.2. Method limitations

There are several drawbacks related to the proposed approach. The deep learning technique proposed has some offline computational cost. which may make the use of these methods challenging in practice. The CPU time for the generation of the snapshots for the tracer concentration and velocity is around 150-200h for one CPU for both the Dataseta with coarse spatio-temporal resolution and the scarse Datasetb with a less coarse one. The training time of the deep learning models ranges between 2393s and 4925s for one CPU. The reconstruction times are below 0.7s for the various networks investigated. The online runtime of the deep learning methods is much smaller than the one of the classical variational methods. The first limitation of the method is that the generation of the CFD training dataset is time consuming for good spatial and time resolutions. Moreover, the meshes investigated are coarse and in order to go beyond model development, it will be crucial to improve the spatial resolution. A range of values for the Reynolds number must be investigated to replace an average Reynolds number. Parallel simulations could be performed to sample efficiently the CFD manifold. Yet, depending on the a priori knowledge of the manifold where the velocity field

lives, for example if there are some informations about the inlet or outlet boundary conditions, the number of snapshots could be reduced. In the case of limited training data, it could also be interesting to develop bayesian neural network based approaches that provide uncertainty quantification. These methods are practical methods when needing to compute several solutions in parallel or several times or for clinical applications. The main conclusion of the work is that the reconstruction results are improved when the geometry of the mesh is fused with the Radon projection information. Yet, the full understanding of the performance of various types of deep learning models on such problems is still an open question.

The presented method has been evaluated by leveraging synthetically generated data with a good knowledge of their structure and complexity. It has to be further explored considering real-world cardiovascular datasets and 3D patient-specific blood flows. Despite these limitations, it can be shown that the approach is successful for the modeling of blood flows from Radon projections and mesh geometry. It allows for great flexibility in personalization of blood flows models.

5. Conclusion

We have explored an innovative inversion approach to reconstruct the blood flow leveraging Radon projections orthogonal to the vessel direction alongside with mesh geometry data. The method presented couples linear or nonlinear reductions with deep learning networks. Reduced bases for the velocity field, the Radon projections and the mesh are extracted from a collection of high-fidelity solutions via PCA or nonlinear PCA and employs fully connected neural networks to accurately estimate the coefficients of the velocity field to be reconstructed with the input reduced features. Numerical results on unsteady velocity fields and deformed meshes show the accuracy of the proposed method with speed up at the online stage compared to the traditional variational methods and lower reconstruction errors with respect to the simple PCA-net already studied.

6. Acknowledgment

The authors acknowledge financial support of the China Scholarship Council(CSC).

References

[1] Shi Y, Lawford P, Hose R. Review of zero-d and 1-d models of blood flow in the cardiovascular system. Biomedical engineering online. 2011;10:1–38.

[2] Crosetto P, Reymond P, Deparis S, et al. Fluid–structure interaction simulation of aortic blood flow. Computers & Fluids. 2011;43(1):46–57.

[3] Quarteroni A, Manzoni A, Vergara C. The cardiovascular system: mathematical modelling, numerical algorithms and clinical applications. Acta Numerica. 2017;26:365–590.

[4] Abdelsalam S, Zaher A. Biomimetic amelioration of zirconium nanoparticles on a rigid substrate over viscous slime a physiological approach. Appl Math Mech. 2023;44:1563–1576.

[5] Abdelsalam S. Revolutionizing bioconvection: artificial intelligence-powered nanoencapsulation with oxytactic microorganisms. Engineering Applications of Artificial intelligence. 2024;137.

[6] Jiang J, Kokeny P, Ying W, et al. Quantifying errors in flow measurement using phase contrast magnetic resonance imaging: comparison of several boundary detection methods. Magnetic resonance imaging. 2015;33(2):185–193.
[7] Pelc NJ, Sommer FG, Li KC, et al. Quantitative magnetic resonance flow imaging. Magnetic resonance quarterly. 1994;10(3):125–147.
[8] MacDonald ME, Frayne R. Cerebrovascular mri: a review of state-of-the-art approaches, methods and techniques. NMR in Biomedicine. 2015;28(7):767–791.
[9] Markl M, Schnell S, Wu C, et al. Advanced flow mri: emerging techniques and applications. Clinical radiology. 2016;71(8):779–795.
[10] Korporaal JG, Benz MR, Schindera ST, et al. Contrast gradient-based blood velocimetry with computed tomography: theory, simulations, and proof of principle in a dynamic flow phantom. Investigative radiology. 2016;51(1):41–49.
[11] Bouillot P, Brina O, Chnafa C, et al. Robust cerebrovascular blood velocity and flow rate estimation from 4d-cta. Medical physics. 2019;46(5):2126–2136.
[12] Barfett JJ, Velauthapillai N, Fierstra J, et al. Intra-vascular blood velocity and volumetric flow rate calculated from dynamic 4d ct angiography using a time of flight technique. The international journal of cardiovascular imaging. 2014;30:1383–1392.
[13] Daly SM, Leahy MJ. 'go with the flow': a review of methods and advancements in blood flow imaging. Journal of biophotonics. 2013;6(3):217–255.
[14] Huang S, Sigovan M, Sixou B. Reconstruction of vascular blood flow in a vessel from tomographic projections. Biomedical Physics & Engineering Express. 2021;7(6):065032.
[15] Arridge S, Maass P, Öktem O, et al. Solving inverse problems using data-driven models. Acta Numerica. 2019;28:1–174.
[16] Jia X, Willard J, Karpatne A, et al. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. ACM/IMS Transactions on Data Science. 2021;2(3):1–26.
[17] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics. 2019;378:686–707.
[18] Zhu Y, Zabaras N, Koutsourelakis PS, et al. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics. 2019;394:56–81.
[19] Geneva N, Zabaras N. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. Journal of Computational Physics. 2020;403:109056.
[20] Bhattacharya K, Hosseini B, Kovachki NB, et al. Model reduction and neural networks for parametric pdes. The SMAI journal of computational mathematics. 2021;7:121–157.
[21] Huang S, Sigovan M, Sixou B. Deep learning methods for blood flow reconstruction in a vessel with contrast enhanced x-ray computed tomography. International Journal for Numerical Methods in Biomedical Engineering. 2024;40(1):e3785.
[22] Morgan B, Murali AR, Preston G, et al. A physics-based machine learning technique rapidly reconstructs the wall-shear stress and pressure fields in coronary arteries. Frontiers in Cardiovascular Medicine. 2023;10:1221541.
[23] Liang L, Mao W, Sun W. A feasibility study of deep learning for predicting hemodynamics of human thoracic aorta. Journal of biomechanics. 2020;99:109544.
[24] Pajaziti E, Montalt-Tordera J, Capelli C, et al. Shape-driven deep neural networks for fast acquisition of aortic 3d pressure and velocity flow fields. PLoS Computational Biology. 2023;19(4):e1011055.
[25] Tse KM, Chang R, Lee HP, et al. A computational fluid dynamics study on geometrical influence of the aorta on haemodynamics. European Journal of Cardio-Thoracic Surgery. 2013;43(4):829–838.
[26] Valen-Sendstad K, Mortensen M, Langtangen HP, et al. Implementing k-epsilon model in the fenics finite element programming environment. Tapir; 2009. p. 583–588.
[27] Roubíček T. Nonlinear partial differential equations with applications. Vol. 153. Springer Science & Business Media; 2013.

[28] Strazzullo M, Ballarin F, Mosetti R, et al. Model reduction for parametrized optimal control problems in environmental marine sciences and engineering. SIAM Journal on Scientific Computing. 2018;40(4):B1055–B1079.
[29] Volkwein S. Proper orthogonal decomposition: Theory and reduced-order modelling. Lecture Notes, University of Konstanz. 2013;4(4):1–29.
[30] Csala H, Dawson ST, Amirhossein A. Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. Phys Fluids. 2022;34:443–482.
[31] Van der Maaten L, Hinton G. Visualizing data using t-sne. Journal of machine learning research. 2008;9(11).
[32] Díez P, Muixí A, Zlotnik S, et al. Nonlinear dimensionality reduction for parametric problems: A kernel proper orthogonal decomposition. International Journal for Numerical Methods in Engineering. 2021;122(24):7306–7327.
[33] Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation. 1998;10(5):1299–1319.
[34] Schölkopf B, Smola A, Müller KR. Kernel principal component analysis. In: International conference on artificial neural networks; Springer; 1997. p. 583–588.
[35] García-González A, Huerta A, Zlotnik S, et al. A kernel principal component analysis (kpca) digest with a new backward mapping (pre-image reconstruction) strategy. arXiv preprint arXiv:200101958. 2020;.
[36] Nagrani A, Yang S, Arnab A, et al. Attention bottlenecks for multimodal fusion. Advances in neural information processing systems. 2021;34:14200–14213.
[37] Forti D, Rozza G. Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems. International Journal of Computational Fluid Dynamics. 2014;28(3-4):158–169.
[38] Tezzele M, Ballarin F, Rozza G. Combined parameter and model reduction of cardiovascular problems by means of active subspaces and pod-galerkin methods. Mathematical and numerical modeling of the cardiovascular system and applications. 2018;:185–207.
[39] Salmoiraghi F, Ballarin F, Corsi G, et al. Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: overview and perspectives. In: Proceedings of the ECCOMAS Congress 2016, 7th European Conference on Computational Methods in Applied Sciences and Engineering, Crete Island, Greece, June 5-10, 2016; Vol. 1; Institute of Structural Analysis and Antiseismic Research School of Civil . . . ; 2016. p. 1013–1031.
[40] Tezzele M, Demo N, Mola A, et al. Pygem: Python geometrical morphing. Software impacts. 2021;7:100047.
[41] Goda K. A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. Journal of Computational Physics. 1979;30:76–95. Available from: https://api.semanticscholar.org/CorpusID:121259769.
[42] Logg A, Mardal KA, Wells G. Automated solution of differential equations by the finite element method: The fenics book. Vol. 84. Springer Science & Business Media; 2012.
[43] Taylor C, Hood P. A numerical solution of the navier-stokes equations using the finite element technique. Computers & Fluids. 1973;1(1):73–100.
[44] Jiang J, Johnson K, Valen-Sendstad K, et al. Flow characteristics in a canine aneurysm model: A comparison of 4d accelerated phase-contrast mr measurements and computational fluid dynamics simulations. MedPhys. 2011;38(11):6300–6311.
[45] Garay J, Nolte D, Locke M, et al. Parameter estimation in fluid flow models from aliased velocity measurements. Inverse Problems. 2022;38:095002.
[46] Virtanen P, Gommers R, Oliphant TE, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods. 2020;17(3):261–272.
[47] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research. 2011;12:2825–2830.
[48] Dong G, Liao G, Liu H, et al. A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. IEEE Geoscience

and Remote Sensing Magazine. 2018;6(3):44–68.
[49] Gupta R, Jaiman R. Three-dimensional deep learning-based reduced order model for unsteady flow dynamics with variable reynolds number. Physics of Fluids. 2022;34(3).
[50] Lee K, Carlberg KT. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. Journal of Computational Physics. 2020;404:108973.
[51] Tipping ME, Bishop CM. Mixtures of probabilistic principal component analyzers. Neural computation. 1999;11(2):443–482.
[52] Romaszko L, Lazarus A, Gao H, et al. Massive dimensionality reduction for the left ventricular mesh; 2019.