

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/319750286>

Nonlinear Methods for Design–Space Dimensionality Reduction in Shape Optimization

Chapter · September 2017

DOI: 10.1007/978-3-319-72926-8_11

CITATIONS

3

READS

198

4 authors, including:



Danny D'Agostino

Sapienza University of Rome

10 PUBLICATIONS 26 CITATIONS

[SEE PROFILE](#)



Andrea Serani

Italian National Research Council

59 PUBLICATIONS 348 CITATIONS

[SEE PROFILE](#)



Matteo Diez

Italian National Research Council

142 PUBLICATIONS 1,124 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



FRIDA - FRamework for Integrated Design of Aircraft [View project](#)



Multifidelity metamodels and adaptive grid refinement for shape optimisation [View project](#)

Nonlinear Methods for Design-Space Dimensionality Reduction in Shape Optimization

(Pre-print)

Danny D’Agostino^{1,2}, Andrea Serani¹, Emilio F. Campana¹, and Matteo Diez¹

¹ CNR-INSEAN, Natl. Research Council–Marine Tech. Research Inst., Rome, Italy
`matteo.diez@cnr.it`

² Department of Computer, Control, and Management Engineering “A. Ruberti”,
Sapienza University of Rome, Rome, Italy

Abstract. In shape optimization, design improvements significantly depend on the dimension and variability of the design space. High dimensional and variability spaces are more difficult to explore, but also usually allow for more significant improvements. The assessment and breakdown of design-space dimensionality and variability are therefore key elements to shape optimization. A linear method based on the principal component analysis (PCA) has been developed in earlier research to build a reduced-dimensionality design-space, resolving the 95% of the original geometric variance. The present work introduces an extension to more efficient nonlinear approaches. Specifically the use of Kernel PCA, Local PCA, and Deep Autoencoder (DAE) is discussed. The methods are demonstrated for the design-space dimensionality reduction of the hull form of a USS Arleigh Burke-class destroyer. Nonlinear methods are shown to be more effective than linear PCA. DAE shows the best performance overall.

Keywords: Shape optimization, hull-form design, nonlinear dimensionality reduction, kernel methods, deep autoencoder

1 Introduction

The simulation-based design (SBD) paradigm has demonstrated its capability of supporting the design decision process, providing large sets of design options and reducing time and costs of the design process. The recent development of high performance computing (HPC) systems has driven the SBD towards its integration with optimization algorithms, moving the SBD paradigm further, to automatic SBD optimization (SBDO). In shape optimization, SBDO consists of three main elements: (i) a simulation tool, (ii) an optimization algorithm, and (iii) a shape modification tool, which need to be integrated efficiently and robustly. In this context, design improvements significantly depend on the dimension and extension of the design space: high dimensional and variability spaces are more difficult and computationally expensive to explore but, at the same time, potentially allow for bigger improvements. The assessment and breakdown

of the design-space dimensionality and variability are therefore a key element for the success of the SBDO [1].

Online linear dimensionality reduction techniques have been developed, requiring the evaluation of the objective function or its gradient. As an example, principal component analysis (PCA) or proper orthogonal decomposition (POD) methods have been applied for reduced-dimensionality local representations of feasible design regions [2]. A PCA/POD based approach is used in the active subspace method (ASM) [3] to discover and exploit low-dimensional and monotonic trends in the objective function, based on the evaluation of its gradient. Online methods improve the shape optimization efficiency by basis rotation and/or dimensionality reduction. Nevertheless, they do not provide an assessment of the design space and the associated shape parametrization before optimization is performed or objective function and/or gradient are evaluated.

Offline linear methodologies have been developed with focus on design-space variability and dimensionality reduction for efficient optimization procedures. A method based on the Karhunen-Loève expansion (KLE) has been formulated for the assessment of the shape modification variability and the definition of a reduced-dimensionality global model of the shape modification vector in [1]. No objective function evaluation nor gradient is required by the method. The KLE is applied to the continuous shape modification vector, requiring the solution of a Fredholm integral equation of the second kind. Once the equation is discretized, the problem reduces to the PCA of discrete data. Offline linear methods improve the shape optimization efficiency by reparametrization and dimensionality reduction, providing the assessment of the design space and the shape parametrization before optimization and/or performance analysis are performed. The assessment is based on the geometric variability associated to the design space of the shape optimization. Although linear methods have been successfully applied for a wide range of problems, they may be not efficient when complex non linear relationship are involved in the performance analysis and optimization.

In the last years researchers have developed nonlinear methods for data dimensionality reduction. Nonlinear dimensionality reduction (NLDR) methods generalize linear methods to address data with nonlinear structures. Kernel PCA (KPCA) solves a PCA eigenproblem in a new space (called feature space) by using kernel methods [4]. Local PCA (LPCA) divides the initial design space in k clusters and a PCA is applied for each of them, supposing that the data in each cluster has an approximate linear structure. LPCA techniques may be differentiated based on the clustering method, which may follow k -means [5] or spectral approaches [6]. Artificial neural networks (ANN) have been also used to reduce data dimensionality, by performing both encoder and decoder tasks (the method is also known as autoencoder).

The objective of the present work is to combine NLDR techniques with shape parametrization in SBDO for ship hydrodynamics. Specifically KPCA, LPCA with k -means (LPCA-KM), LPCA with spectral clustering (LPCA-SC), and Deep Autoencoder (DAE) are used to build a reduced-dimensionality design-

space, resolving at least the 95% of the original design variability based on the concept of geometric variance [1]. The methods are demonstrated for the design-space dimensionality reduction of the hull form of USS Arleigh Burke-class destroyer, namely the DTMB 5415 model, an early and open to public version of the DDG-51. The effectiveness of the NLDR techniques is shown and discussed, comparing the results to the linear KLE/PCA method from earlier work [1].

2 Dimensionality Reduction Methods

General definitions and assumptions for the current problem are presented in the following, along with linear and nonlinear dimensionality reduction methods.

2.1 General Definitions and Assumptions

Consider a geometric domain \mathcal{G} (which identifies the initial shape) and a set of coordinates $\mathbf{x} \in \mathcal{G}$.

Assume that $\mathbf{u} \in \mathcal{U}$ is the design variable vector, which defines a continuous shape modification vector $\delta(\mathbf{x}, \mathbf{u})$. Consider the design variables \mathbf{u} as a random field defined over a domain \mathcal{U} , with associated probability density function $p(\mathbf{u})$. The associated mean shape modification is evaluated as

$$\langle \delta \rangle = \int_{\mathcal{U}} \delta(\mathbf{x}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u} \quad (1)$$

If one defines the internal product in \mathcal{G} as

$$(\mathbf{f}, \mathbf{g}) = \int_{\mathcal{G}} \mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) d\mathbf{x} \quad (2)$$

with associated norm $\|\mathbf{f}\| = (\mathbf{f}, \mathbf{f})^{1/2}$, the variance associated to the shape modification vector (geometric variance) may be defined as

$$\sigma^2 = \langle \|\hat{\delta}\|^2 \rangle = \int_{\mathcal{U}} \int_{\mathcal{G}} \hat{\delta}(\mathbf{x}, \mathbf{u}) \cdot \hat{\delta}(\mathbf{x}, \mathbf{u}) p(\mathbf{u}) d\mathbf{x} d\mathbf{u} \quad (3)$$

where $\hat{\delta} = \delta - \langle \delta \rangle$, and $\langle \cdot \rangle$ denotes the ensemble average over \mathbf{u} . Generally, $\mathbf{x} \in \mathbb{R}^n$ with $n = 1, 2, 3$, $\mathbf{u} \in \mathbb{R}^M$ with M number of design variables, and $\delta \in \mathbb{R}^m$ with $m = 1, 2, 3$ (with m not necessarily equal to n). Figure 1 shows an example with $n = 1$ and $m = 2$. Ensemble averages $\langle \cdot \rangle$ over $\mathbf{u} \in \mathcal{U}$ may be evaluated by

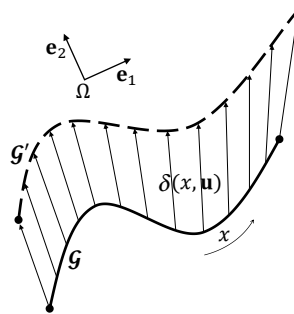


Fig. 1: Scheme and notation for the current formulation, showing an example for $n = 1$ and $m = 2$

Monte Carlo (MC) sampling using a statistically convergent number of random realizations S , $\{\mathbf{u}_k\}_{k=1}^S \sim p(\mathbf{u})$. These are collected in a $[S \times L]$ matrix

$$\mathbf{D} = \left[\begin{array}{c|c|c} \mathbf{d}(\mathbf{u}_1) & \dots & \mathbf{d}(\mathbf{u}_S) \end{array} \right]^T \quad (4)$$

representing the (MC sampled) original design space, where $\mathbf{d}(\mathbf{u}_k) = \{\mathbf{d}_q(\mathbf{u}_k)\}_{q=1}^m$ is the deviation from the mean of the shape modification vector and its q -th component is evaluated at discrete coordinates \mathbf{x}_t , $t = 1 \dots, T$, as

$$\mathbf{d}_q(\mathbf{u}_k) = \left\{ \begin{array}{c} \delta_q(\mathbf{x}_1, \mathbf{u}_k) \\ \vdots \\ \delta_q(\mathbf{x}_T, \mathbf{u}_k) \end{array} \right\} - \frac{1}{S} \sum_{k=1}^S \left\{ \begin{array}{c} \delta_q(\mathbf{x}_1, \mathbf{u}_k) \\ \vdots \\ \delta_q(\mathbf{x}_T, \mathbf{u}_k) \end{array} \right\} \quad (5)$$

with $\delta_q = \boldsymbol{\delta} \cdot \mathbf{e}_q$, where $\{\mathbf{e}_q\}_{q=1}^m \in \mathbb{R}^m$ is a basis of orthogonal unit vector. Note that $L = mT$.

A reduced-dimensionality representation of \mathbf{D} is sought after for later use in the SBDO.

2.2 Principal Component Analysis

PCA allows to reduce the input dimensionality of the data, performing a projection of the points in a new linear subspace, defined by the eigenvectors of the $[L \times L]$ covariance matrix $\mathbf{C} = \mathbf{D}^T \mathbf{D} / S$. These eigenvectors have the properties to maximize the variance of points projected on them and to minimize the mean squared distance between the original points and the relative projections [7]. The principal components are defined by the solution of the eigenproblem

$$\mathbf{C}\mathbf{z} = \lambda\mathbf{z} \quad (6)$$

The solutions $\{\mathbf{z}_i\}_{i=1}^L$ of the Eq. 6 are used to build a reduced-dimensionality space for the shape modification vector \mathbf{d} as

$$\mathbf{d} \approx \sum_{i=1}^N \alpha_i \mathbf{z}_i = \hat{\mathbf{d}} \quad (7)$$

where α_i is the i -th component of the new design variable vector $\boldsymbol{\alpha} \in \mathbb{R}^N$. Equation 7 may be truncated to the N -th order, preserving a desired level of confidence β ($0 < \beta \leq 1$), provided that

$$\sum_{i=1}^N \lambda_i \geq \beta \sum_{i=1}^L \lambda_i = \beta \sigma^2 \quad (8)$$

assuming $\lambda_i \geq \lambda_{i+1}$. Only M eigenvalues are expected to be non zeros.

2.3 Kernel Principal Component Analysis

The kernel PCA (KPCA) method [4] is a nonlinear extension of PCA. It finds directions of maximum variance in a higher (possibly infinite) dimensional feature space \mathcal{F} , mapping the points from the input space \mathcal{I} by a possible nonlinear function $\Phi : \mathcal{I} \rightarrow \mathcal{F}$ as

$$\mathbf{d}_k \rightarrow \Phi(\mathbf{d}_k), \quad \forall k = 1, \dots, S \quad (9)$$

where, for the sake of simplicity, the $\mathbf{d}(\mathbf{u}_k)$ of Eq. 4 is here simplified in \mathbf{d}_k . Then PCA is computed in the feature space \mathcal{F} . Assuming that $\sum_k \Phi(\mathbf{d}_k) = 0$, the kernel principal component $\{\mathbf{z}_p\}_{p=1}^P$ can be found solving the eigenproblem

$$\Sigma_{\Phi} \mathbf{z}_p = \lambda_p \mathbf{z}_p \quad (10)$$

where Σ_{Φ} is the $[P \times P]$ covariance matrix in the feature space \mathcal{F} , defined as

$$\Sigma_{\Phi} = \frac{1}{S} \sum_{k=1}^S \Phi(\mathbf{d}_k) \Phi(\mathbf{d}_k)^T \quad (11)$$

KPCA allows the solution of Eq. 10 without computing explicitly the Eq. 9, since it appears only within an inner product [8], which can be computed efficiently by a kernel function $K(\mathbf{d}_i, \mathbf{d}_k) = \Phi(\mathbf{d}_i)^T \Phi(\mathbf{d}_k)$. Defining \mathbf{z}_p as a linear expansion of $\Phi(\mathbf{d}_k)$

$$\mathbf{z}_p = \sum_{k=1}^S c_{pk} \Phi(\mathbf{d}_k) \quad (12)$$

the Eq. 10 can be recasted as

$$\mathbf{K} \mathbf{c}_p = \lambda_p S \mathbf{c}_p \quad (13)$$

where \mathbf{K} is the symmetric and positive-semidefinite $[S \times S]$ kernel matrix, with $\mathbf{K}_{ik} = K(\mathbf{d}_i, \mathbf{d}_k)$. The length of the S -component vector \mathbf{c}_p is chosen such that $\mathbf{z}_p^T \mathbf{z}_p = \lambda_p S \mathbf{c}_p^T \mathbf{c}_p = 1$. Once the eigenproblem in Eq. 13 is solved, the new design variables can be found projecting $\Phi(\mathbf{d})$ on \mathbf{z}_p as

$$\alpha = \Phi(\mathbf{d}) \mathbf{z}_p = \sum_{k=1}^S c_{pk} \Phi(\mathbf{d})^T \Phi(\mathbf{d}_k) = \sum_{k=1}^S c_{pk} K(\mathbf{d}, \mathbf{d}_k) \quad (14)$$

The reconstruction of the original data from the feature space \mathcal{F} in KPCA is more problematic than PCA, since it needs to find, for every point $\Phi(\mathbf{d}_k)$, the relative pre-image \mathbf{d}_k in the input space \mathcal{I} . In this paper, approximate pre-images technique proposed in [9] is used.

2.4 Local Principal Component Analysis

Local PCA (LPCA) performs a PCA for every different disjoint region of the input space \mathcal{I} , assuming that, if the local regions are small enough, the data manifold will not curve much over the extent of the region and the linear model will be a good fit [5].

The first step in LPCA is to cluster the data in k sets, applying a clustering algorithm, such that $\mathbf{D} = \{\mathbf{D}_1, \dots, \mathbf{D}_i\}_{i=1}^k$. Herein, LPCA is performed with two clustering techniques: the k -means (LPCA-KM) algorithm [10] and a spectral clustering (LPCA-SC) [11]. The k -means clustering algorithm is described in Alg. 1.

Algorithm 1 k -means clustering algorithm

Require: Random k centroids as representative points of each cluster $\mathbf{D}_i \forall i = 1, \dots, k$.

- 1: **repeat**
- 2: Assign each point \mathbf{d}_j to the nearest centroid $\boldsymbol{\mu}_i$ using the Euclidean distance as similarity measure.
- 3: Update the centroids according to: $\boldsymbol{\mu}_i = \frac{1}{|\mathbf{D}_i|} \sum_{\mathbf{d}_j \in \mathbf{D}_i} \mathbf{d}_j$
- 4: **until** $\boldsymbol{\mu}_i \forall i = 1, \dots, k$ remains unchanged

One issue in k -means is that using the euclidean distance as similarity measure assumes a convex shape to the underlying clusters [12].

Spectral clustering can be effective even if the clusters shape are more complex. There are several versions of the spectral clustering algorithms, the main difference is in which graph Laplacian is used [6]. Herein, the symmetric normalized Laplacian $\mathbf{A}_{\text{sym}} = \mathbf{I} - \mathbf{B}^{-\frac{1}{2}} \mathbf{W} \mathbf{B}^{-\frac{1}{2}}$ [11] is used and the corresponding algorithm is summarized in Alg. 2 [6].

After the data are partitioned in k clusters, a PCA is performed on them solving k PCA eigenproblem

$$\mathbf{C}_i \mathbf{z}_i = \lambda_i \mathbf{z}_i \quad \forall i = 1, \dots, k \quad (15)$$

LPCA results are highly dependent by the clustering procedure and specially by the number of clusters used. Moreover, the number of clusters k should be set carefully to avoid extensive computation.

Algorithm 2 Normalized Spectral Clustering

Require: Let k the number of clusters to identify, build a similarity graph as:

- K -nearest neighbor graphs: fix K , \mathbf{d}_i is connected to a point \mathbf{d}_j if it is among the K -nearest neighbor of \mathbf{d}_i or viceversa.

- 1: Compute the adjacency matrix \mathbf{W} of the graph and the diagonal degree matrix \mathbf{B} , where each element is equal to $b_{ii} = \sum_{j=1}^S w_{ij}$.
 - 2: Compute the symmetric normalized Laplacian \mathbf{A}_{sym} .
 - 3: Find the first k eigenvector $\mathbf{v}_1, \dots, \mathbf{v}_k$ corresponding to the k smallest eigenvalues of \mathbf{A}_{sym} .
 - 4: Construct a $[S \times k]$ matrix \mathbf{V} with the eigenvectors as columns.
 - 5: Normalize the rows of matrix \mathbf{V} by $\hat{v}_{ij} = v_{ij} / (\sum_k v_{ik}^2)^{\frac{1}{2}}$
 - 6: Run k -means on matrix \mathbf{V} .
-

2.5 Deep Autoencoders

An autoencoder (AE) is an ANN that performs two main tasks [13]: (1) an encoder function \mathcal{E} maps the data \mathbf{d} to compress data $\boldsymbol{\alpha}$; (2) a decoder function \mathcal{D} maps from the compressed data $\boldsymbol{\alpha}$ back to $\hat{\mathbf{d}}$. This operation is performed setting the same number of neurons L in the input and output layer and constraining the hidden layer to have $N < M$ neurons.

Consider a single hidden layer AE, if the new design variable $\boldsymbol{\alpha}$ can be written as

$$\boldsymbol{\alpha} = \mathcal{E}(\mathbf{H}^{(1)}\mathbf{d} + \mathbf{b}^{(1)}) \quad (16)$$

where \mathbf{H} is a relative weight matrix, \mathbf{b} the bias vector, and the apex “(1)” represent the hidden layer, then the reconstruction vector $\hat{\mathbf{d}}$ from $\boldsymbol{\alpha}$ can be expressed as

$$\hat{\mathbf{d}} = \mathcal{D}(\mathbf{H}^{(2)}\boldsymbol{\alpha} + \mathbf{b}^{(2)}) \quad (17)$$

where the apex “(2)” represent the output layer. The network parameters \mathbf{H} and \mathbf{b} , are evaluated minimizing the reconstruction error

$$\begin{aligned} E(\mathbf{H}^{(1)}, \mathbf{b}^{(1)}, \mathbf{H}^{(2)}, \mathbf{b}^{(2)}) &= \frac{1}{2} \sum_{k=1}^S \|\mathbf{d}_k - \hat{\mathbf{d}}_k\|^2 \\ &= \frac{1}{2} \sum_{k=1}^S \|\mathbf{d}_k - \mathcal{D}(\mathbf{H}^{(2)}\mathcal{E}(\mathbf{H}^{(1)}\mathbf{d}_k + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})\|^2 \end{aligned} \quad (18)$$

If \mathcal{E} and \mathcal{D} are linear then the Eq. 18 has a unique global minimum, in which the weights in the hidden layer span the same subspace as the first N -principal components of the data [14, 15]. AE with nonlinear activation functions and more hidden layers (called deep autoencoder, DAE) provides a nonlinear generalization of the PCA [16], but in this case the error function (Eq. 18) becomes non convex and the optimization algorithm may get stuck in poor local minima. Moreover, the intrinsic dimensionality of the data (the number of neurons N in the hidden layer) cannot be known a priori and have to be fixed respect to the reconstruction error.

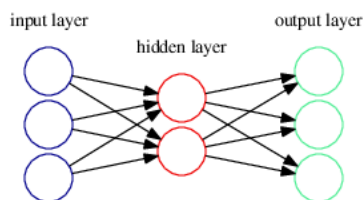


Fig. 2: Example of AE with one hidden layer with $L = 3$ and $N = 2$

3 Shape Modification of a Destroyer Hull

The DTMB 5415 model is an open-to-public early concept of the DDG-51, a USS Arleigh Burke-class destroyer, widely used for both towing tank experiments [17] and hull-form SBDO [18]. Figure 3 shows its geometry and body surface grid used to discretize the shape modification domain.

The offline design-space assessment and dimensionality reduction of the DTMB 5415 hull form (assuming full-scale with a length between perpendiculars $L_{pp} = 142$ m) is presented as a pre-optimization study of the following problem

$$\begin{aligned} & \text{Minimize } f(\mathbf{u}) \\ & \text{subject to } g_a(\mathbf{u}) = 0, \quad \text{with } a = 1, \dots, A \\ & \quad \text{and to } h_e(\mathbf{u}) \leq 0, \quad \text{with } e = 1, \dots, E \end{aligned} \quad (19)$$

where f is the objective function related to the ship performance (*i.e.* resistance, seakeeping, etc.) and \mathbf{u} are the (original) design variables. Geometrical equality constraints, g_a , include fixed length between perpendicular (L_{pp}) and displacement (∇), whereas geometrical inequality constraints, h_e , include 5% maximum variation of beam and draught and reserved volume for the sonar in the bow dome, corresponding to 4.9 m diameter and 1.7 m length (cylinder).

Shape modifications $\delta(\mathbf{x}, \mathbf{u})$ are applied directly on the Cartesian coordinates \mathbf{g} of the computational body surface grid, as per

$$\mathbf{g}(\mathbf{u}) = \mathbf{g}_0 + \delta(\mathbf{x}, \mathbf{u}) \quad (20)$$

where \mathbf{g}_0 represents the original grid.

The shape modification is defined using a linear combination of $M = 27$ vector-valued functions of the Cartesian coordinates \mathbf{x} over a hyper-rectangle embedding the demi hull [18]

$$\psi_i(\mathbf{x}) : \mathcal{V} = [0, L_{x_1}] \times [0, L_{x_2}] \times [0, L_{x_3}] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3 \quad (21)$$

with $i = 1, \dots, M$, as

$$\delta(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^M u_i \psi_i(\mathbf{x}) \quad (22)$$

where the coefficients $u_i \in \mathbb{R}$ ($i = 1, \dots, M$) are the (original) design variables,

$$\psi_i(\mathbf{x}) := \prod_{j=1}^3 \sin\left(\frac{a_{ij}\pi x_j}{L_{x_j}} + r_{ij}\right) \mathbf{e}_{q(i)} \quad (23)$$

and the following orthogonality property is imposed:

$$\int_{\mathcal{V}} \psi_i(\mathbf{x}) \cdot \psi_k(\mathbf{x}) d\mathbf{x} = \delta_{ik} \quad (24)$$

In Eq. 23, $\{a_{ij}\}_{j=1}^3 \in \mathbb{R}$ define the order of the function along j -th axis; $\{r_{ij}\}_{j=1}^3 \in \mathbb{R}$ are the corresponding spatial phases; $\{L_{x_j}\}_{j=1}^3$ are the hyper-rectangle edge lengths; $\mathbf{e}_{q(i)}$ is a unit vector. Modifications are applied along x_1 , x_2 , or x_3 , with $q(i) = 1, 2$, or 3 respectively. The parameter values used here are taken from [18].

Fixed L_{pp} and ∇ are satisfied by automatic geometric scaling, while geometries exceeding the constraints are not considered.

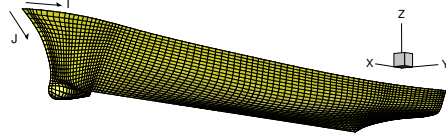


Fig. 3: DTMB 5415 geometry and body surface discretization

4 Numerical Results

The results obtained by linear PCA and the nonlinear methods (KPCA, LPCA-KM, LPCA-SC, and DAE) are presented in the following subsections. Two evaluation metrics are used to assess the methods' performance and compare them.

4.1 Evaluation Metrics

The methods are assessed by the portion of original geometric variance resolved ($\hat{\beta}$) and the root mean square error (RMSE) of matrix reconstruction $\hat{\mathbf{D}}$, defined as

$$\hat{\beta} = \frac{\frac{1}{S} \sum_{j=1}^L \sum_{k=1}^S (\hat{d}_{jk} - \hat{\mu}_j)^2}{\frac{1}{S} \sum_{j=1}^L \sum_{k=1}^S (d_{jk} - \mu_j)^2} \quad \text{and} \quad \text{RMSE} = \sqrt{\frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k - \hat{\mathbf{d}}_k\|^2} \quad (25)$$

where $\hat{\mu}_j$ is the mean value of $\hat{\mathbf{D}}$ j -th column.

4.2 Evaluation of Design-Space Dimensionality Reduction Capabilities

In assessing the methods' performance, a cubic polynomial kernel is used for the KPCA, a number of cluster $k = 32$ and 24 is used for LPCA-KM and LPCA-SC respectively, a seven hidden layer DAE (composed by 300-150-50- N -50-150-300 neurons) with hyperbolic tangent (as activation function) is used and trained with Adam optimization algorithm [19].

The design space ($M = 27$) is sampled using a uniform random distribution of $S = 1,000$ hull-form designs. For each dimensionality-reduction method, Fig. 4a shows the geometric variance ($\hat{\beta}\%$) resolved by a N -dimensional design space, whereas Fig. 4b shows the corresponding reconstruction error (RMSE). The nonlinear methods

Table 1: Numerical results

Method	N [-]	$\hat{\beta}\%$	RMSE/ L_{pp}
PCA	24	95.0	1.12E-1
KPCA	18	100.	0.00E+0
LPCA-KM	12	95.0	1.12E-1
LPCA-SC	15	95.4	1.08E-1
DAE	5	97.8	9.60E-2

result to be more effective than the linear PCA in terms of both $\hat{\beta}\%$ and RMSE.

Specifically, in order to reduce the design-space dimensionality while resolving at least the 95% of the original geometric variance, $N = 24$ is required by PCA, whereas $N = 18, 12, 15,$ and 5 are needed by KPCA, LPCA-KM, LPCA-SC, and DAE, respectively. The results are summarized in Tab. 1. It is worth noting that KPCA requires $N = 18$, but resolves the 100% of the original variance and shows a reconstruction error equal to zero. In the current study, it was not possible to reduce N further, due to numerical issue associated to the computation of pre-images.

Finally, Fig. 5 shows the shape modification (δ_y) and the reconstruction error ($\Delta\delta_y$) versus grid-node index (I, J), and the corresponding hull stations for a design originally included in the data matrix \mathbf{D} . For this design, LPCA shows

the largest reconstruction error. PCA and DAE produce a close reconstruction to the target, whereas KPCA reproduce the target exactly. With only $N = 5$, DAE is the most efficient overall.

5 Conclusions and Future Work

Four nonlinear methods for design-space dimensionality-reduction in shape optimization have been presented and compared. Specifically, kernel PCA (KPCA), local PCA with k -means and spectral clustering (respectively LPCA-KM and LPCA-SC), and deep autoencoder (DAE) have been used for an offline pre-optimization dimensionality-reduction of the hull-form parametrization of the DTMB 5415 model hull. A linear PCA method from earlier studies has been also included in the analysis, for comparison.

The original shape parametrization was defined by $M = 27$ design variables. The reduced-dimensionality space is required to resolve at least the 95% of the original design variability, based on the concept of geometric variance. The linear PCA achieved a reduction of 11.2% of the original design dimensionality (requiring a number of design variables $N = 24$). All nonlinear methods outperform the linear PCA. Specifically, a 33.4% dimensionality reduction is achieved by KPCA ($N = 18$), 55.5% by LPCA-KM ($N = 12$), 44.4% by LPCA-SC ($N = 15$), and finally a remarkable 81.5% by DAE ($N = 5$). Nonlinear methods have shown their superior effectiveness in terms of both variance resolved and reconstruction error, compared to linear PCA. DAE have shown the best performance overall.

The analysis of some specific behavior of the methods presented, such as the assessment of the clusters used by the LPCA, will be addressed in future work. Moreover, in order to investigate further on the methods' effectiveness, future work will include the optimization of the DTMB 5415 using the reduced-dimensionality space produced by linear and nonlinear methods, with comparison of objective function improvement and convergence to the optimum. Also, combined geometry and physics based design variability studies [20, 21] will be addressed using current nonlinear methods.

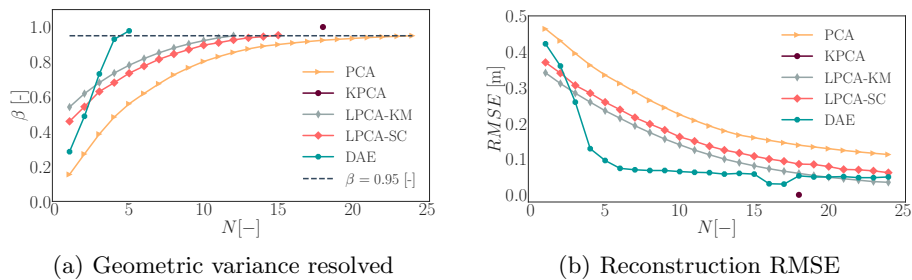


Fig. 4: Convergence of dimensionality-reduction methods in terms of $\hat{\beta}\%$ (a) and RMSE (b) versus the reduced-dimensionality N

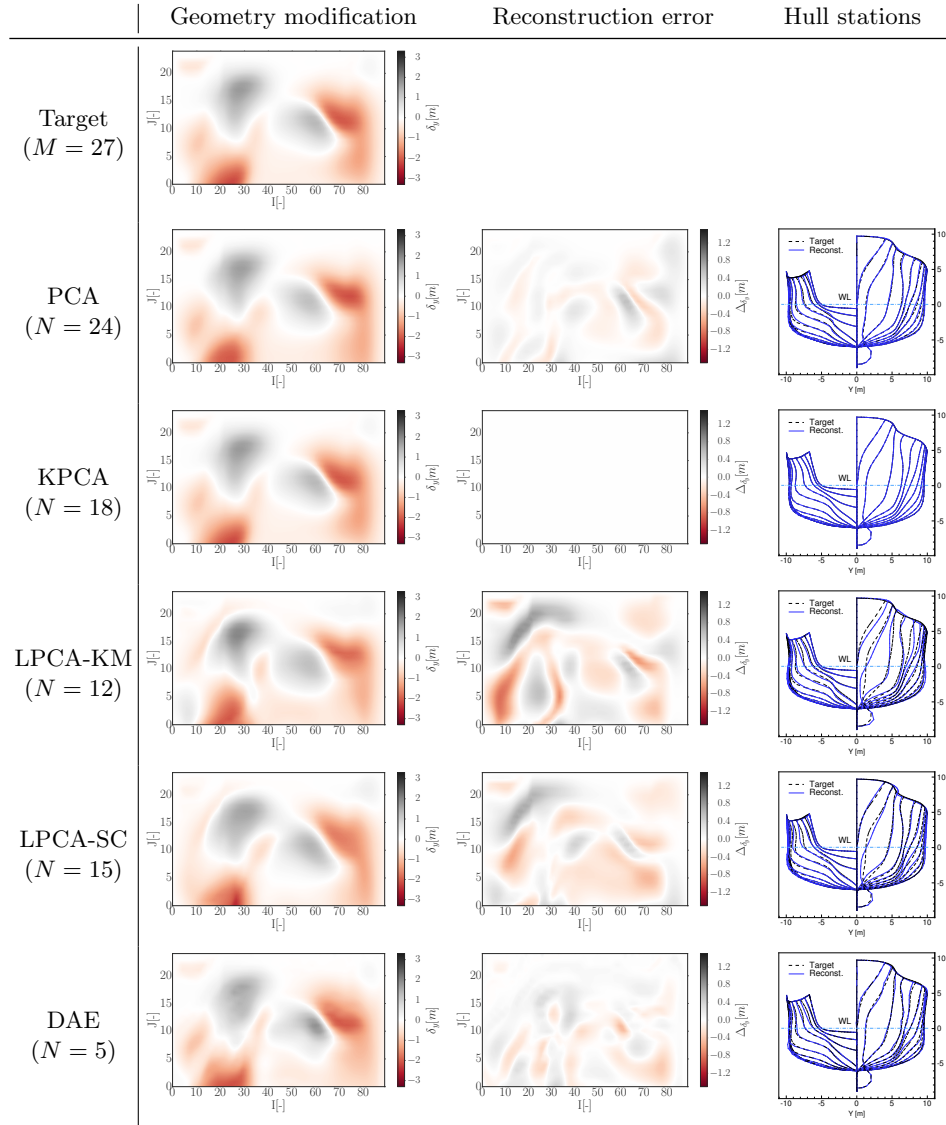


Fig. 5: Reconstruction of the geometry modification vector δ_y , reconstruction error, and corresponding hull stations of target geometry (original input)

Acknowledgments. The work is supported by the US Office of Naval Research Global, NICOP grant N62909-15-1-2016, under the administration of Dr. Woei-Min Lin, Dr. Salahuddin Ahmed, and Dr. Ki-Han Kim, and by the Italian Flagship Project RITMARE. The research is performed within NATO STO Task Group AVT-252 Stochastic Design Optimization for Naval and Aero Military Vehicles. The authors wish to thank Prof. Frederick Stern and Dr. Manivannan Kandasamy of The University of Iowa for inspiring the current research on

nonlinear dimensionality reduction methods.

References

1. Diez, M., Campana, E.F., Stern, F.: Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion. *Computer Methods in Applied Mechanics and Engineering* **283** (2015) 1525–1544
2. Raghavan, B., Breilkopf, P., Tourbier, Y., Villon, P.: Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization* **48** (2013) 9871000
3. Lukaczyk, T., Palacios, F., Alonso, J.J., Constantine, P.: Active subspaces for shape optimization. In: *Proceedings of the 10th AIAA Multidisciplinary Design Optimization Specialist Conference, National Harbor, Maryland, USA, 13-17 January*. (2014)
4. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* **10**(5) (1998) 1299–1319
5. Kambhatla, N., Leen, T.K.: Dimension reduction by local principal component analysis. *Neural computation* **9**(7) (1997) 1493–1516
6. Von Luxburg, U.: A tutorial on spectral clustering. *Statistics and computing* **17**(4) (2007) 395–416
7. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
8. Smola, A.J., Schölkopf, B.: *Learning with kernels*. Citeseer (1998)
9. Bakır, G.H., Weston, J., Schölkopf, B.: Learning to find pre-images. *Advances in neural information processing systems* **16** (2004) 449–456
10. Lloyd, S.: Least squares quantization in PCM. *IEEE transactions on information theory* **28**(2) (1982) 129–137
11. Ng, A.Y., Jordan, M.L., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. In: *NIPS*. Volume 14. (2001) 849–856
12. Aggarwal, C.C., Reddy, C.K.: *Data clustering: algorithms and applications*. Chapman and Hall/CRC (2013)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786) (July 2006) 504–507
14. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics* **59**(4) (1988) 291–294
15. Baldi, P., Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks* **2**(1) (1989) 53–58
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553) (2015) 436–444
17. Stern, F., Longo, J., Penna, R., Olivieri, A., Ratcliffe, T., Coleman, H.: International collaboration on benchmark CFD validation data for surface combatant DTMB model 5415. In: *Proceedings of the Twenty-Third Symposium on Naval Hydrodynamics, Val de Reuil, France, September 17-22* (2000)
18. Serani, A., Fasano, G., Liuzzi, G., Lucidi, S., Iemma, U., Campana, E.F., Stern, F., Diez, M.: Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms. *Applied Ocean Research* **59** (2016) 115 – 128
19. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)

20. Diez, M., Serani, A., Stern, F., Campana, E.F.: Combined geometry and physics based method for design-space dimensionality reduction in hydrodynamic shape optimization. In: Proceedings of the 31st Symposium on Naval Hydrodynamics, Monterey, CA, USA. (2016)
21. Serani, A., Campana, E.F., Diez, M., Stern, F.: Towards augmented design-space exploration via combined geometry and physics based Karhunen-Loève expansion. In: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MA&O), AVIATION 2017, Denver, USA, June 5-9 (2017)