Chapter 16
Adaptive Sampling for Nonlinear Dimensionality Reduction Based on Manifold Learning

Thomas Franz, Ralf Zimmermann, and Stefan Görtz

Abstract We make use of the non-intrusive dimensionality reduction method *Isomap* in order to emulate nonlinear parametric flow problems that are governed by the Reynolds-averaged Navier-Stokes equations. Isomap is a manifold learning approach that provides a low-dimensional embedding space that is approximately isometric to the manifold that is assumed to be formed by the high-fidelity Navier-Stokes flow solutions under smooth variations of the inflow conditions. The focus of the work at hand is the adaptive construction and refinement of the Isomap emulator: We exploit the non-Euclidean Isomap metric to detect and fill up gaps in the sampling in the embedding space. The performance of the proposed manifold filling method will be illustrated by numerical experiments, where we consider nonlinear parameter-dependent steady-state Navier-Stokes flows in the transonic regime.

16.1 Introduction

In [8], the authors proposed a non-intrusive low-order emulator model for nonlinear parametric flow problems governed by the Navier-Stokes equations. The approach is based on the manifold learning method Isomap [17] combined with an interpolation scheme and will be referred to hereafter as Isomap+I. Via this method, a low-dimensional embedding space is constructed that is approximately isometric to the manifold that is assumed to be formed by the high-fidelity Navier-Stokes flow

T. Franz (✉) • S. Görtz
Institute for Aerodynamics and Flow Technology, German Aerospace Center (DLR), Braunschweig, Germany
e-mail: thomas.franz@dlr.de; stefan.goertz@dlr.de

R. Zimmermann
Institute 'Computational Mathematics', TU Braunschweig, Braunschweig, Germany

Department of Mathematics and Computer Science, University of Southern Denmark, Odense M, Denmark
e-mail: ralf.zimmermann@tu-bs.de; zimmermann@imada.sdu.dk

P. Benner et al. (eds.), *Model Reduction of Parametrized Systems*,
MS&A 17, DOI 10.1007/978-3-319-58786-8_16

solutions under smooth variations of the inflow conditions. As with almost all model reduction methods, the offline stage for the Isomap+I approach requires a suitable design of experiment, i.e., a well-chosen sampling of high-fidelity flow solutions, the so-called snapshots. The online stage, however, might be considered as an adaptive way for choosing for each low-order prediction the most suitable local snapshot neighborhood rather than using all available snapshot information in a brute-force way. The notion of locality is based on the Isomap metric. The focus of this article is on an adaptive construction and refinement of the underlying design of experiment. Since Isomap comes with a natural non-Euclidean metric for measuring snapshot distances, we make use of this metric to detect gaps in the embedding space. By the (approximate) isometry between the embedding space and the manifold of flow solutions, we obtain in this way a *manifold filling* design of experiment. In contrast, standard approaches like the Latin Hypercube method [6] aim at a *parameter-space filling* design of experiment. The performance of the proposed manifold filling method is illustrated by numerical experiment, where we consider nonlinear parameter-dependent steady-state Navier-Stokes flows in the transonic regime.

Organization In Sect. 16.2, the Isomap-based emulator model is briefly introduced. The adaptive sampling strategy based on the manifold characterization is developed in Sect. 16.3.1, followed by a proof of concept in Sect. 16.3.2. Afterwards, the methods are demonstrated for an engineering application in Sect. 16.4. Finally, conclusions are drawn in Sect. 16.5.

16.2 The Isomap-Based Emulator Model

In this section, we briefly review the manifold learning based approach to emulate steady-state flows governed by the Reynolds-averaged Navier Stokes (RANS) equations that was introduced in [7, 8]. For background information on computational fluid dynamics see, e.g., [3], for an introduction to differentiable manifolds see, e.g., [16].

Let $\mathscr{M} \subset \mathbb{R}^n$ be an embedded submanifold in the Euclidean space with intrinsic dimension $\dim(\mathscr{M}) = d < n$. Let $\mathscr{W} \subset \mathscr{M}$ be an open domain in $\mathscr{M}$ such that there exists a coordinate chart[1] $h : \mathscr{W} \to \mathscr{Y}$ onto an open domain $\mathscr{Y} \subset \mathbb{R}^d$. The fundamental objective of manifold learning (ML) [5, 18] is to solve the *isometric embedding problem* [2, 18], which we reformulate as follows:

> For a given finite set of sampled data points $W = \{\mathbf{W}^1, \ldots, \mathbf{W}^m\} \subset \mathscr{W} \subset \mathbb{R}^n$ compute an approximation of the coordinate chart h such that the restriction to the discrete sample points
>
> $$h|_W : \mathscr{W} \supset W = \{\mathbf{W}^1, \ldots, \mathbf{W}^m\} \to Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^m\} \subset \mathscr{Y}, \quad h(\mathbf{W}^i) = \mathbf{y}^i,$$

[1] i.e., a bijective both-ways differentiable mapping.

is such that the image point set Y features (approximately) the same inter-point distances as the high dimensional data set W.

One of the most popular ML methods is *Isomap* [17]. Isomap works by approximating the *geodesic distance* between data vectors $\mathbf{W}^i$ and $\mathbf{W}^j$ via computing the length of an Euclidean polygon course that connects $\mathbf{W}^i$ and $\mathbf{W}^j$. The polygon course is determined based on a graph-theoretical shortest path problem, which is detailed in [8] and [17]. The basic idea is illustrated in Fig. 16.1a. Once the geodesic distances are estimated, a distance matrix $D \in \mathbb{R}^{m\times m}$ is formed, where the entry d_{ij}, $i,j = 1,\ldots,m$, is the approximated geodesic distance between $\mathbf{W}^i$ and $\mathbf{W}^j$. The next step is to employ classical multidimensional scaling [11, Sect. 14] with the distance matrix D as an input. This results in a data set $Y = \{\mathbf{y}^1,\ldots,\mathbf{y}^m\}$ with $\left\|\mathbf{y}^i - \mathbf{y}^j\right\| \approx d_{ij}$ for $i,j = 1,\ldots,m$. Moreover, the data set Y is tuned for the envisioned application by minimizing an additional *loss function* afterwards, see [7, Sect. 4.3.1]. The resulting embedding space when applying Isomap to the 'swiss roll' standard example in manifold learning (see Fig. 16.1b) is displayed in Fig. 16.1c.

So far, we have constructed a low-dimensional representation of the high-dimensional input data. In order to obtain a valid emulator, a mapping from the low-dimensional space to the high-dimensional manifold is required. As it is common in many model reduction methods, including proper orthogonal decom-

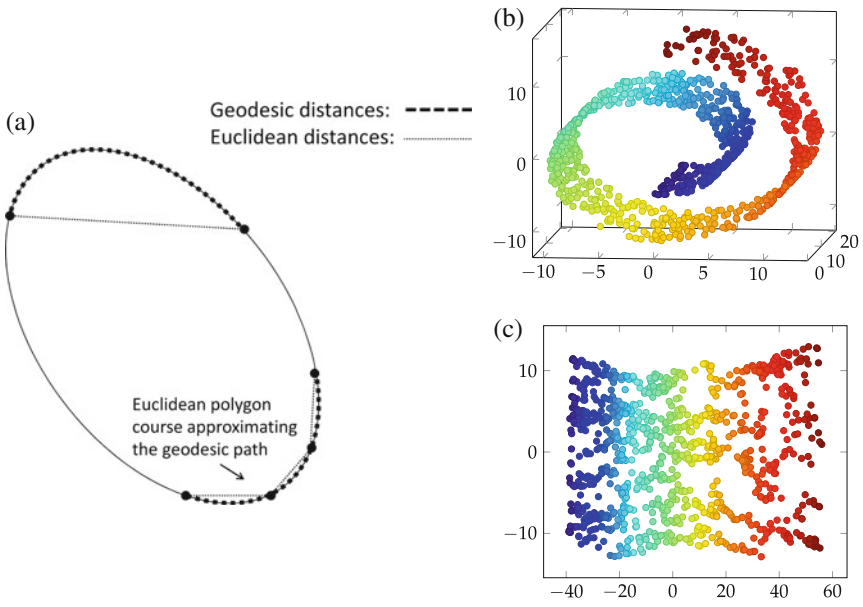

Fig. 16.1 *Left*: Geodesic distances vs. Euclidean distances. *Right*: The 'swiss roll' standard example. (**a**) Approximation of the geodesic distance. (**b**) Swiss roll: Original data set in $\mathbb{R}^3$. (**c**) Swiss roll: Isometric embedding in $\mathbb{R}^2$

position [12] and the reduced basis method [15], we assume that the output of the emulator is a linear combination of the input snapshots. In our setting, the input data vectors stem from solutions to the RANS equations under parametric variations, i.e., $\mathbf{W}^j = \mathbf{W}(\mathbf{p}^j)$, where $\mathbf{p}^j$ is the parameter vector specifying the inflow conditions. The ansatz at an untried flow condition $\mathbf{p}^*$ is $\mathbf{W}(\mathbf{p}^*) = \sum_{j=1}^m a_j(\mathbf{p}^*)\mathbf{W}^j$. Hence, the nonlinear parametric dependency is in the coefficients $a_j = a_j(\mathbf{p})$ while the snapshots $\mathbf{W}^j$ are fixed. The essential idea of manifold learning is to localize the information in the sense that only the N nearest neighbors $\{\mathbf{W}^j = \mathbf{W}(\mathbf{p}^j) | j \in \mathscr{I}, |\mathscr{I}| = N\}$ contribute to $\mathbf{W}(\mathbf{p}^*)$, where the notion of proximity depends on the Isomap metric.

The exact procedure is as follows: If the flow at $\mathbf{p}^*$ is to be emulated, we then first determine the corresponding location in the embedding space $\mathbf{y}^* = \mathbf{y}(\mathbf{p}^*) \in \mathbb{R}^d$ via multivariate interpolation based on the embedded data set $\{(\mathbf{p}^j, \mathbf{y}^j)\}_{j=1}^m$. Isomap provides us with the nearest neighbors $\{\mathbf{y}^j \mid j \in \mathscr{I}\}$ of $\mathbf{y}^*$. Next, we represent $\mathbf{y}^*$ approximatively as a weighted linear combination of the nearest neighbors as $\mathbf{y}^* \approx \sum_{j\in\mathscr{I}} a_j\mathbf{y}^j$, where we determine the weights a_j via the following optimization problem:

$$\min_{\mathbf{a}\in\mathbb{R}^N} \|\mathbf{y}^* - \sum_{j\in\mathscr{I}} a_j\mathbf{y}^j\|_2^2 + \|\mathbf{a}\|_c^2 \qquad \text{s. t.} \quad \sum_{j\in\mathscr{I}} a_j = 1, \tag{16.1}$$

with penalty term

$$\|\mathbf{a}\|_c^2 := \sum_{j\in\mathscr{I}} c_j a_j^2, \quad c_j = \varepsilon \left(\frac{\|\mathbf{y}^* - \mathbf{y}^j\|_2}{\max_i\{\|\mathbf{y}^* - \mathbf{y}^i\|_2\}} \right)^k, \quad 0 < \varepsilon \ll 1,\ 1 < k \in \mathbb{N}.$$

The penalty term weights the influence of the snapshots based on their distance to the prediction point $\mathbf{y}^*$. Let $\mathbf{a}^* \in \mathbb{R}^N$ be the solution to (16.1). Because of the inherent (approximate) isometry between the snapshots $\mathbf{W}^j$ and the locations $\mathbf{y}^j$ in the embedding space, we use the same weight vector to construct the high-dimensional flow state

$$\mathbf{W}^* = \sum_{j\in\mathscr{I}} a_j^*\mathbf{W}^j. \tag{16.2}$$

The extra condition in Eq. (16.1) is such that when the whole set of embedded snapshots y^j, $j \in \mathscr{I}$, is translated via $T : y \mapsto y + \mu$ to a new set $z^j = T(y_j)$, $j \in \mathscr{I}$, then

$$T(y^*) = y^* + \mu = (y^1, \ldots, y^{|\mathscr{I}|})a + \mu = (z^1, \ldots, z^{|\mathscr{I}|})a = z^*.$$

Best practice settings for the meta-parameters ε, k and further details are given in [7]. In addition, a heuristic choosing the size of the neighborhood $\mathscr{I}$ automatically

is developed in [7] and employed for all conducted predictions. We call the above process *Isomap+I*.

16.3 Adaptive Sampling

The algorithmic efficiency and the numerical accuracy of the Isomap-based emulator strongly depend on the selected input information. Computing the input snapshots is costly by nature, because high-fidelity solutions to the very system that is to be emulated are required. Moreover, spatial sampling methods suffer from the *curse of dimensionality* [6, Sect. 1.1] in the sense that the number of sample points that is required to achieve a certain sampling density grows exponentially with the spatial dimension. To keep the number of full system solves as small as possible, we present an incremental sampling method that attempts to create a homogeneously distributed data set of the manifold based on geometric information.

16.3.1 Manifold Filling Adaptive Sampling Strategies

As outlined in Sect. 16.2, Isomap preserves the interpoint distances of the underlying manifold domain $\mathscr{W}$. This property is what we exploit for detecting gaps in the input data set.

Let $\{\mathbf{y}^1, \ldots, \mathbf{y}^m\} = Y_m \subset \mathbb{R}^d$ be the low-dimensional representative of the large-scale input snapshot set $\{\mathbf{W}^1, \ldots, \mathbf{W}^m\} = W_m \subset \mathbb{R}^n$ and let $\mathbf{y} : \mathscr{P} \to \mathbb{R}^d$ with $\mathbf{y}(\mathbf{p}^j) = \mathbf{y}^j$, $\mathbf{p}^j \in \mathscr{P} \subset \mathbb{R}^d, j = 1, \ldots, m$. If there is a location $\mathbf{y}_g \in \{\mathbf{y}(\mathbf{p}) \mid \mathbf{p} \in \mathscr{P}\}$ and a radius $\gamma > 0$ such that the γ-ball $B_\gamma(\mathbf{y}_g) = \{\tilde{\mathbf{y}} \in \mathbb{R}^d \mid \|\mathbf{y}_g - \tilde{\mathbf{y}}\|_2 < \gamma\}$ does not contain any sampled representatives, i.e., $\mathbf{y}^j \notin B_\gamma(\mathbf{y}_g) \forall j = 1, \ldots, m$, then we say that there is a *gap* of size γ at $\mathbf{y}_g \in \{\mathbf{y}(\mathbf{p}) \mid \mathbf{p} \in \mathscr{P}\}$. The objective is to detect these gaps and fill them by adding suitable snapshots to the input data set.

We device an iterative adaptation process. Let $\mathscr{P} \subset \mathbb{R}^d$ be the parameter domain of interest and let $P_{\tilde{m}} = \{\mathbf{p}^1, \ldots, \mathbf{p}^{\tilde{m}}\} \subset \mathscr{P}$ be a set of $\tilde{m} \in \mathbb{N}$ preselected sample locations. Moreover, let $1 \leq i \leq m - \tilde{m}$ be the number of the current iteration of the adaptive sampling process, where $i, m \in \mathbb{N}$ and $m > \tilde{m}$ is the maximal number of affordable snapshots. Starting with the initial design of experiment (DoE) of $\tilde{m}$ snapshots $W_{\tilde{m}} = \{\mathbf{W}^1, \ldots, \mathbf{W}^{\tilde{m}}\} \subset \mathbb{R}^n$, where $\mathbf{W}^j = \mathbf{W}(\mathbf{p}^j)$, the associated initial embedding $Y_{\tilde{m}} = \{\mathbf{y}^1, \ldots, \mathbf{y}^{\tilde{m}}\} \subset \mathbb{R}^d$ is calculated via Isomap.

The procedure to detect gaps is as follows: For a given location $\mathbf{p} \in \mathscr{P}$ the corresponding location in the embedding space $\mathbf{y} : \mathscr{P} \to \mathbb{R}^d$ is determined via interpolation based on the data set of current sample locations $\{(\mathbf{p}^j, \mathbf{y}^j)\}_{j=1}^{\tilde{m}}$, cf. Sect. 16.2. Then, the *weighted sum of the distances of the* $N \in \mathbb{N}$ *nearest neighbors* $y^j, j \in \mathscr{I}$ to $\mathbf{y}(\mathbf{p})$ is calculated:

$$dist(\mathbf{y}(\mathbf{p})) := \frac{d_{min}(\mathbf{y}(\mathbf{p}))}{d_{max}(\mathbf{y}(\mathbf{p}))} \sum_{j \in \mathscr{I}} \|\mathbf{y}(\mathbf{p}) - \mathbf{y}^j\|_2, \tag{16.3}$$

Algorithm 1 Manifold filling adaptive sampling algorithm

Require: Desired number of snapshots m, number of initial snapshots $\tilde{m}$
1: Generate $\tilde{m} < m$ parameter values $\mathbf{p}^1, \ldots, \mathbf{p}^{\tilde{m}} \in \mathscr{P}$, e.g. via LHS
2: $P \leftarrow \{\mathbf{p}^1, \ldots, \mathbf{p}^{\tilde{m}}\}$
3: Compute snapshot solutions $\mathbf{W}(\mathbf{p})$ at each parameter value $\mathbf{p} \in P$
4: $W \leftarrow \{\mathbf{W}(\mathbf{p}^1), \ldots, \mathbf{W}(\mathbf{p}^{\tilde{m}})\}$ ▷ initial sampling
5: **for** $i = 1$ to $m - \tilde{m}$ **do**
6: Calculate embedding Y of the generated snapshot set W via Isomap
7: Compute interpolation model for $\mathbf{y}$ based on $\{(\mathbf{p}^j, \mathbf{y}^j)\}_{j=1}^{\tilde{m}+i-1}$
8: Determine $\mathbf{p}^* \in \mathscr{P}$ by maximizing E_{dist} or E_{rec}
9: Compute snapshot solution $\mathbf{W}^*$ at parameter configuration $\mathbf{p}^* \in \mathscr{P}$
10: $P \leftarrow P \cup \{\mathbf{p}^*\}$
11: $W \leftarrow W \cup \{\mathbf{W}^*\}$
12: **end for**
13: **return** Set W of m snapshots

where $d_{min}(\mathbf{y}(\mathbf{p})) = \min_{j \in \mathscr{I}} \|\mathbf{y}(\mathbf{p}) - \mathbf{y}^j\|_2$ and $d_{max}(\mathbf{y}(\mathbf{p})) = \max_{j \in \mathscr{I}} \|\mathbf{y}(\mathbf{p}) - \mathbf{y}^j\|_2$. The distance function (16.3) is multiplied by an indicator function ω:

$$E_{dist}(\mathbf{y}(\mathbf{p})) := dist(\mathbf{y}(\mathbf{p})) \cdot \omega(\mathbf{p}), \quad \omega(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} \in \mathscr{P}, \\ 0 & \text{else,} \end{cases} \tag{16.4}$$

which ensures that the adaptation process takes place only in the inside of the parameter domain of interest. The maximizer $\mathbf{p}^* = \arg\max E_{dist}(\mathbf{y}(\mathbf{p}))$ determines the next snapshot to be added to the model. The above method will be referred to as the maximum distance error (MDE) strategy. A pseudo code of this method is outlined in Algorithm 1.

On top of the distance based error criterion (16.4), we introduce a reconstruction error indicator that yields reliable results when the manifold is sufficiently homogeneously sampled, i.e. the sampling does not divide into disconnected clusters. Let $Y_{\tilde{m}+i-1}$ be the embedding data set at iteration $i-1$ of the adaptive sampling process. For each $\mathbf{y}^j \in Y_{\tilde{m}+i-1}$, we compute a prediction $\hat{\mathbf{W}}(\mathbf{y}^j) = \hat{\mathbf{W}}^j$ based on its N nearest neighbors and the relative error $E_{rel}(\mathbf{y}^j) = \frac{\|\hat{\mathbf{W}}^j - \mathbf{W}^j\|_2}{\|\mathbf{W}^j\|_2}$ to the corresponding snapshot $\mathbf{W}^j$. Note that $\mathbf{y}^j$ is not counted as a neighbor of itself and hence $\hat{\mathbf{W}}^j \neq \mathbf{W}^j$. Subsequently, interpolation is performed to approximate the relative error at an arbitrary location $\mathbf{y} \notin Y_{\tilde{m}+i-1}$ based on the data set $\{(\mathbf{y}^j, E_{rel}(\mathbf{y}^j))\}_{j=1}^{\tilde{m}+i-1}$. To ensure that the error is zero at the given sample points, the reconstruction error is defined as

$$E_{rec}(\mathbf{y}(\mathbf{p})) := E_{rel}(\mathbf{y}(\mathbf{p})) \cdot E_{dist}(\mathbf{y}(\mathbf{p})). \tag{16.5}$$

Since an almost homogeneously sampled manifold must be given, we employ the error function E_{rec} only every kth iteration in practice. For the remaining iterations

E_{dist} is utilized exclusively to ensure a homogeneously distributed manifold. The resulting hybrid error sampling strategy is referred to as HYE in the following.

Remark 1 It is not a necessity that we add only one snapshot per iteration. In each iteration, we may choose to determine several local maximizers to E_{dist} and E_{rec}, respectively, and add the corresponding snapshots to the information pool.

Choice of the Initial Sampling Plan and Starting Points for Optimization When starting from scratch, the initial sampling plan of $\tilde{m}$ points in the parameter domain of interest $\mathscr{P}$ is chosen randomly. More precisely, we employ either space filling random Latin Hypercube Sampling (LHS) [6] or Halton sequences [9] to construct the initial DoE. The selection of the starting points for the maximization of either (16.4) or (16.5) requires special consideration as the objective functions features many local maxima. We make the following differentiation:

1. If the initial DoE $P = \{\mathbf{p}^1, \ldots, \mathbf{p}^{\tilde{m}}\}$ is such that its convex hull coincides with the parameter space $\mathscr{P}$ of interest, then we treat the convex hull of the corresponding embedding points $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^{\tilde{m}}\}$ as the domain of interest in the embedding space, even though the mapping is not convex in general. In this case, we perform a *Delaunay triangulation* [14] of Y and determine the centers $\mathbf{y}(\mathbf{c}^i) \in \mathrm{conv}(Y)$, $i = 1, \ldots, l$ of the Delaunay simplices of largest volume. The corresponding locations $\mathbf{p}(\mathbf{y}(\mathbf{c}^i)) \in \mathscr{P}$ are selected as starting points for optimizing (16.4). (The $\mathbf{p}(\mathbf{y}(c^i))$ are found via interpolation.)
2. Otherwise, we perform another space filling LHS to create the starting points randomly in order to avoid clustering effects. This procedure is also followed for determining the starting points for optimizing (16.5) in order to increase the probability to locate the global maximum.

16.3.2 Proof of Concept

In this section, we illustrate the performance of Algorithm 1 on two academic examples.

Detection of Gaps Reconsider the swiss role, parameterized by two parameters t and h:

$$\mathbf{s} : \mathscr{P} \to \mathscr{W} \subset \mathbb{R}^3, \quad (t, h) \mapsto (t\cos(t), h, t\sin(t)), \quad \mathscr{P} = [\tfrac{3}{2}\pi, \tfrac{9}{2}\pi) \times [0, 21)$$

To artificially create a hole in the sample set, we exclude the rectangle $(9.5, 10.5) \times (8, 13)$ from the parameter domain and construct an initial random-based DoE P of $|P| = 748$ sample points in $\mathscr{P} \setminus (9.5, 10.5) \times (8, 13)$.

Now, we conduct a single step of Algorithm 1, where we perform step 8 with respect to (16.4) and consider only the single nearest neighbor in evaluating the distance function (16.3). This results in an optimal location $\mathbf{p}^* \in \mathscr{P}$ that is displayed

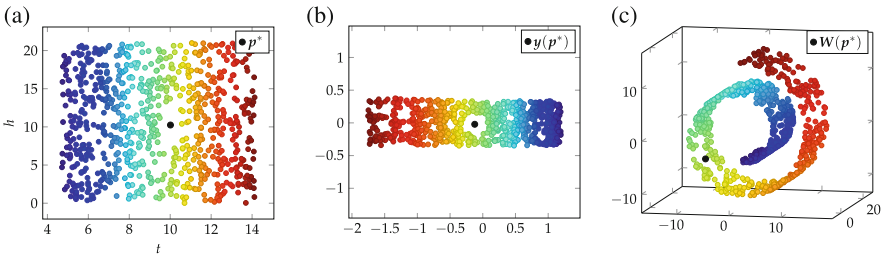

Fig. 16.2 Detection of a gap in the DoE illustrated for the swiss roll. (**a**) Point cloud and detected gap center in the parameter domain $\mathscr{P}$. (**b**) Point cloud and detected gap center in the embedding domain $\mathscr{Y}$. (**c**) Point cloud and detected gap center on the swiss roll manifold $\mathscr{W}$

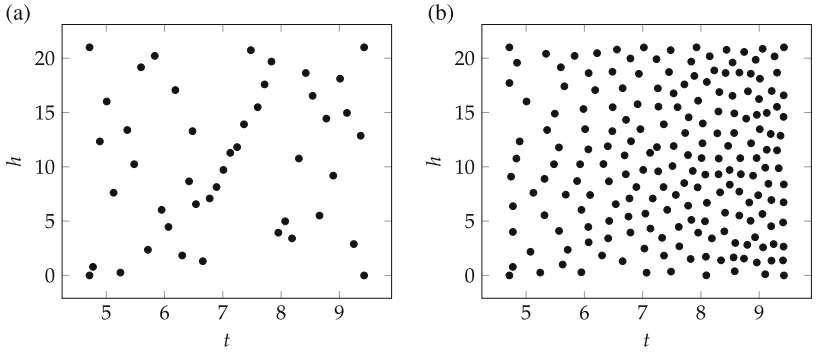

Fig. 16.3 Curved plate: Locations of the initial and refined parameter samples. (**a**) Initial parameter sampling. (**b**) Refined sample set

in Fig. 16.2a. Figure 16.2b, c depict the corresponding point $\mathbf{y}^* = \mathbf{y}(\mathbf{p}^*)$ in the embedding space and $\mathbf{s}(\mathbf{p}^*) \in \mathbb{R}^3$ on the swiss roll manifold, respectively.

Manifold Filling As a second academic example, we consider a curved plate parameterized by

$$\mathbf{c} : \mathscr{P} \to \mathscr{W} \subset \mathbb{R}^3, (t,h) \mapsto \left(\tfrac{t^2}{10}\cos(t), h, \tfrac{t^2}{10}\sin(t)\right), \quad \mathscr{P} = [\tfrac{3}{2}\pi, 3\pi] \times [0,21].$$

We start with a Latin hypercube sampling of 40 data points selected from the interior of $\mathscr{P}$ and add the four corner points of the rectangle $\mathscr{P}$, see Fig. 16.3a. The corresponding initial sample data set $W_{44} \subset \mathscr{W}$ and its discrete Isomap embedding Y_{44} are depicted in Fig. 16.4a, b, respectively.

We detect the regions of low sampling density via the MDE approach. The starting points for the optimization procedures are chosen by a LHS of size 30 in each iteration. In Fig. 16.4c, d, the generated snapshot set $W_{\tilde{m}+i}$ and its embeddings

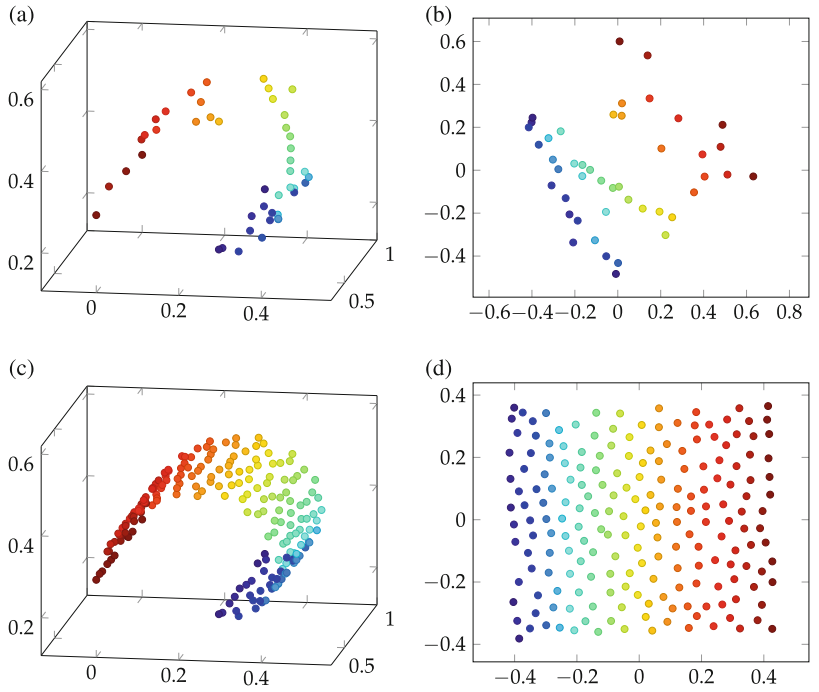

Fig. 16.4 Manifold filling adaptive sampling strategy illustrated for a curved plate. (**a**) $W_{44} \subset \mathscr{W}$. (**b**) $Y_{44} \subset \mathscr{Y}$. (**c**) $W_{194} \subset \mathscr{W}$. (**d**) $Y_{194} \subset \mathscr{Y}$

after $i = 150$ iterations is shown, respectively.[2] The $\tilde{m}+i = 194$ parameter locations in $\mathscr{P}$ associated with the final refined snapshot set are depicted in Fig. 16.3b. Note that the sampling plan is denser for larger t, which is in line with the fact that the function **c** exhibits a higher angular velocity for increasing t.

16.4 An Engineering Example

As an engineering application, we emulate the high-Reynolds number flow past the two-dimensional NACA 64A010 airfoil in the transonic flow regime. The geometry of the airfoil is shown in Fig. 16.5b. The hybrid unstructured grid features 21, 454 grid points, including 400 surface grid points, and is depicted in Fig. 16.5.

The objective is to emulate the distribution of the pressure coefficient C_p on the surface of the airfoil under varying angle of attack, α, and Mach number, Ma. To

[2]The number of nearest neighbors used for the embedding was chosen automatically in each iteration according to [7, Sect. 4.3.3].

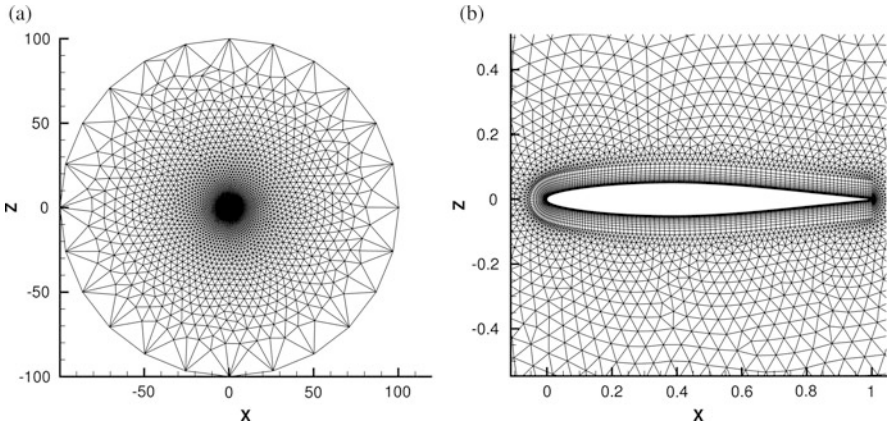

Fig. 16.5 Computational grid for the NACA 64A010 airfoil. (**a**) View of the entire flow field. (**b**) Detailed view close to the surface

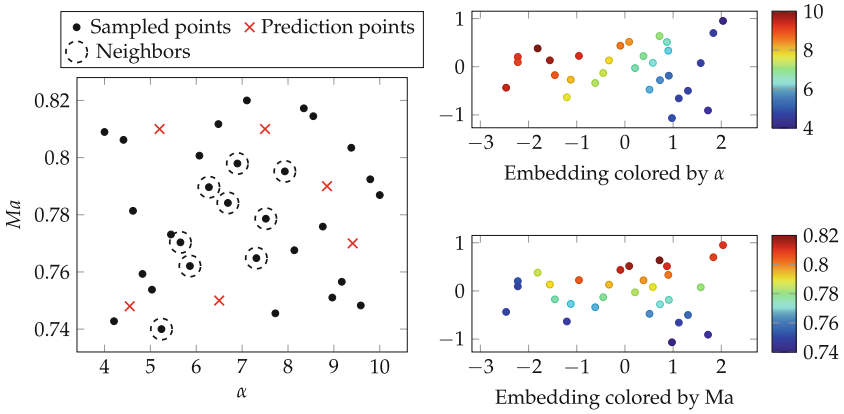

Fig. 16.6 *Left*: Locations of the snapshots and various prediction points in the α-Ma-space for the NACA 64A010 test case. Furthermore, the employed snapshots for the prediction at $(\alpha, \mathrm{Ma}) = (6.5°, 0.75)$ are encircled. *Right*: Representatives within the embedding space colored corresponding to the angle of attack α (*top*) and the Mach number Ma (*bottom*)

this end, we generate a snapshot set of flow solutions, where the initial parameter locations P are selected via a LHS of $m = 30$ samples from in the parameter space $\mathscr{P} = \{(\alpha, \mathrm{Ma}) \in [4°, 10°] \times [0.74, 0.82]\}$, see Fig. 16.6. The corresponding viscous flow solution snapshots $\mathbf{W}(\mathbf{p})$, $\mathbf{p} \in P$, are computed with DLR's RANS solver TAU [10] using the negative Spalart-Allmaras one-equation turbulence model [1]. Convergence is detected based on a reduction of the normalized density residual by seven orders of magnitude in each solver run. The Reynolds number is fixed

throughout at a value of Re = 7,500,000. Computing a full CFD solution under this conditions took 474 iterations or 63 CPU seconds on average.[3]

From the flow solution snapshots, we extract the vectors $\mathbf{W}(\mathbf{p})$, $\mathbf{p} \in P$ containing the discretized surface-C_p distributions, which form our initial point cloud W. Since in this test case two varying parameters are considered, the full-order solution manifold $\mathscr{W} = \{\mathbf{W}(\alpha, \mathrm{Ma}), \quad (\alpha, \mathrm{Ma}) \in \mathscr{P}\} \subset \mathbb{R}^{400}$ is of intrinsic dimension two.[4] The low intrinsic dimension is not a technical requirement, but an natural assumption in the context of model order reduction. We use the Isomap+I process of Sect. 16.2 to predict the C_p distributions at untried parameter locations and compare the results to the approximations computed via proper orthogonal decomposition combined with interpolation, which yields predictions at untried parameter combinations by interpolating the POD coefficients as done in [4]. This method will be referred to as POD+I in the following. Both interpolation based ROMs are coupled with the RBF interpolation using a TPS kernel augmented by a polynomial $\varphi \in \Pi_1$ [6, 13], $\varphi : \mathbb{R}^d \to \mathbb{R}$, where Π_1 is the space of polynomials of degree of at most one. Prior to each interpolation process, the sample locations in the parameter space are scaled to the unit hypercube, with the result that the input scaling is normalized and does not thwart the Isomap metric. The TPS kernel has been chosen for its good approximation quality and robustness based on best practice observations made in [19]. The first author's thesis features the results at all the prediction points indicated in Fig. 16.6. Here, we display only the worst result, which is obtained at $(\alpha, \mathrm{Ma}) = (6.5°, 0.75)$, since we aim at improving the prediction by adaptively refining the snapshot sampling according to the MDE and HYE strategy. The nine nearest neighbors on the manifold that are used to compute the prediction are encircled in Fig. 16.6 and the resulting C_p distribution is shown in Fig. 16.7.

We start with an initial DoE of 5 sample points generated by a Halton sequence, where none of the points is considered to lie on the boundary of $\mathscr{P}$. We perform 25 iterations of Algorithm 1 to arrive at a final sampling of 30 snapshots. In both sampling strategies, we consider only the nearest neighbor when evaluating the objective function (16.4). In the hybrid strategy HYE, we maximize (16.5) instead of (16.4) in every third iteration. In Table 16.1, we list the mean relative error, the standard deviation and the maximum relative error for the Isomap emulator associated with the adaptively refined data sets obtained via the MDE strategy and the HYE strategy, respectively.[5] For comparison, we include the errors corresponding to Isomap emulators based on the non-adaptive random DoEs of the same cardinality 30 that are obtained by a Halton sequence and a space filling

[3] All computations were conducted sequentially on the same standard desktop computer endowed with an Intel® Xeon® E3-1270 v3 Processor (8M Cache, 3.50 GHz) and 32 GB RAM.

[4] For applications where the dimension of the manifold is unknown, there exist various methods to estimate the intrinsic dimensionality of the data, e.g. by looking for the "elbow" [17].

[5] Error quantification is with respect to the surface C_p distributions and is based on 2500 uniformly distributed TAU reference CFD solutions.

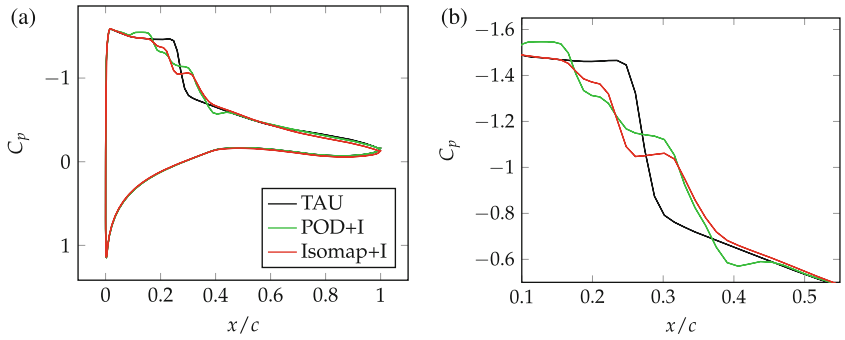

Fig. 16.7 Surface C_p distribution at $(\alpha, \text{Ma}) = (6.5°, 0.75)$. The upper and lower curves correspond to the suction and pressure side of the airfoil, respectively. Results obtained based on a non-adaptive random sampling of 30 input snapshots. (**a**) Complete surface C_p distribution. (**b**) Detailed view close to shock

Table 16.1 The mean relative error, its standard deviation and the maximum relative error after a full sampling process of various sampling strategies/designs for the NACA 64A010 test case

Method	Mean rel. error	STD. deviation	Max. rel. error
MDE	$2.3347 \cdot 10^{-2}$	$1.6616 \cdot 10^{-2}$	$9.2956 \cdot 10^{-2}$
HYE	$2.1903 \cdot 10^{-2}$	$1.0320 \cdot 10^{-2}$	$5.3337 \cdot 10^{-2}$
Halton	$2.6670 \cdot 10^{-2}$	$2.7398 \cdot 10^{-2}$	$2.3016 \cdot 10^{-1}$
LHS	$3.1262 \cdot 10^{-2}$	$2.6257 \cdot 10^{-2}$	$1.8009 \cdot 10^{-1}$

LHS. The adaptive sampling strategies developed here yield samplings with a smaller change of the relative errors than in both random samplings. Hence the maximum relative error is closer to the mean relative error, which leads to a more reliable global emulator with less outliers in prediction accuracy. Note, that the mean relative errors are also smaller for the adaptive strategies. The embeddings of the final samplings are shown in Fig. 16.8. As aspired by MDE, the embedding of the corresponding sampling is quite evenly distributed. This also holds for the embedding of the sampling obtained by HYE, even if E_{rec} is applied in every third iteration. In contrast, the embeddings of both random samplings feature close-by points, which may lead to redundant information.

We use the HYE-adaptively constructed emulator to predict the surface pressure at the flow condition of $(\alpha, \text{Ma}) = (6.5°, 0.75)$, where a poor approximation quality was observed in Fig. 16.7. Recall that those results were obtained with the same number of 30 input snapshots, but chosen randomly (LHS) rather than adaptively.

The C_p-distributions obtained from the emulators are shown in Fig. 16.9a, where we compare the CFD reference and the Isomap+I and the POD+I emulators. As can be seen, both the Isomap+I and the POD+I predictions greatly benefit from the adaptive sampling process. (Compare Figs. 16.7, 16.8, and 16.9a.) The Isomap+I prediction matches the reference solution with high accuracy throughout by using

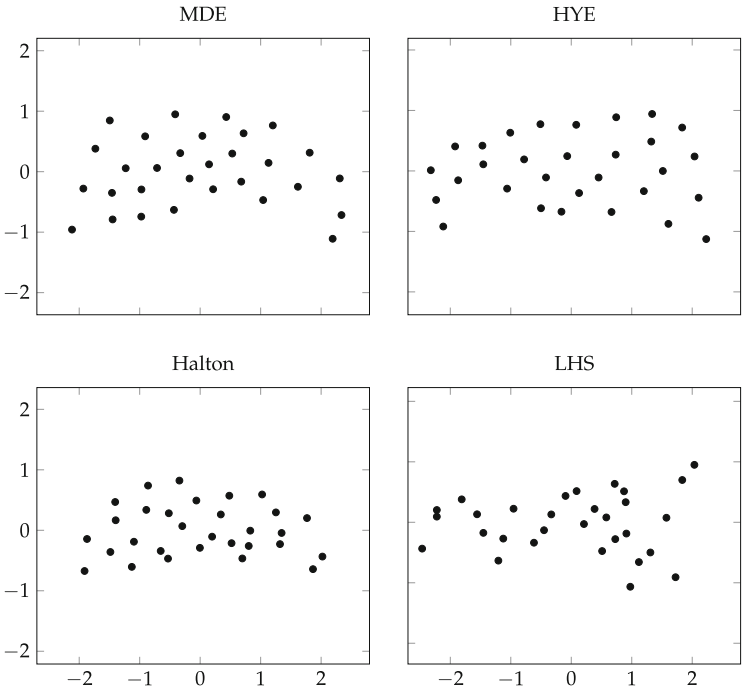

Fig. 16.8 Embeddings of the final samplings obtained by various sampling methods and DoEs for the NACA 64A010 test case

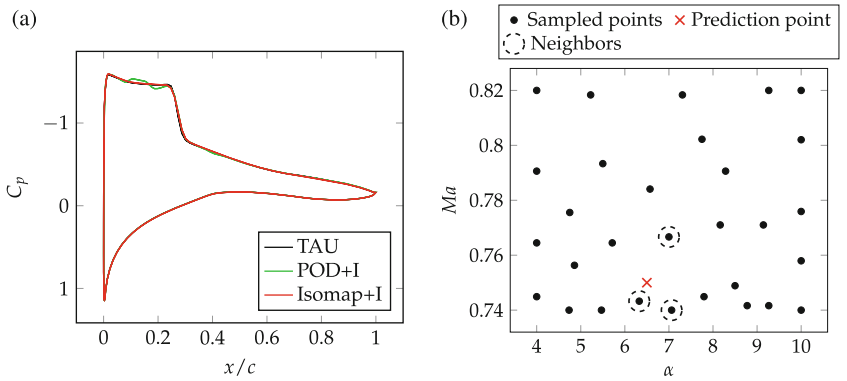

Fig. 16.9 Prediction of the surface C_p-distribution at $(\alpha, \mathrm{Ma}) = (6.5°, 0.75)$ based on 5 initial plus 25 adaptively sampled snapshots via HYE. (**a**) Prediction of the surface C_p-distribution at $(\alpha, \mathrm{Ma}) = (6.5°, 0.75)$. (**b**) HYE based final sampling

only three neighboring snapshots (see Fig. 16.9b). The POD+I based prediction only shows a small mismatch upstream of the shock.

16.5 Conclusions

We have developed two adaptive sampling strategies, referred to as the maximum distance error (MDE) and the hybrid error (HYE) strategy, respectively, that aim at determining sample locations in a given parameter domain of interest such that a well-distributed homogeneous design of experiment is achieved in the embedding space with as few high-fidelity sample computations as possible. The underlying assumption is that the sample data is contained in a submanifold of low intrinsic dimension that is embedded in a large-dimensional Euclidean vector space. Thus, the notions of 'well-distributed' and 'homogeneous' are to be understood with respect to the geometry of this submanifold.

Both adaptive sampling methods try to generate manifold filling sample data sets such that the essential geometric characteristics of the underlying submanifold are captured. The MDE strategy relies on the geodesic interpoint distances that are approximated using the Isomap manifold learning technique. The HYE strategy additionally considers the reconstruction error of an Isomap+I emulator during the sampling process, such that the sample density in the highly nonlinear regions of the manifold, where the error is expected to be larger, is augmented.

In the numerical experiments, we have shown that the adaptive sampling strategies eventually lead to more accurate emulators than when using space filling random samplings of the same cardinality. More precisely, the advantages over random samplings have been demonstrated for an Isomap-based emulator of the viscous flow around the 2D NACA 64A010 airfoil. Moreover, we observed that the standard POD-based flow emulator also benefits from the Isomap-induced adaptive sampling process.

References

1. Allmaras, S.R., Johnson, F.T.: Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7), pp. 1–11 (2012)
2. Bernstein, A.V., Kuleshov, A.P.: Tangent bundle manifold learning via Grassmann & Stiefel eigenmaps. CoRR, abs/1212.6031 (2012)
3. Blazek, J.: Computational Fluid Dynamics: Principles and Applications, 1st edn. Elsevier, Amsterdam/London/New York/Oxford/Paris/Shannon/Tokyo (2001)
4. Bui-Thanh, T., Damodaran, M., Willcox, K.: Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. AIAA J. **42**(8), 1505–1516 (2004)
5. Cayton, L.: Algorithms for manifold learning. University of California at San Diego Tech. Rep, pp. 1–17 (2005)

6. Forrester, A., Sóbester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, London (2008)
7. Franz, T.: Reduced-order modeling for steady transonic flows via manifold learning. PhD thesis, TU Braunschweig (2016)
8. Franz, T., Zimmermann, R., Görtz, S., Karcher, N.: Interpolation-based reduced-order modeling for steady transonic flows via manifold learning. Int. J. Comput. Fluid Mech. (Special Issue on Reduced Order Modeling) **228**, 106–121 (2014)
9. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. **2**(1), 84–90 (1960)
10. Langer, S., Schwöppe, A., Kroll, N.: The DLR flow solver TAU—status and recent algorithmic developments. In: 52nd AIAA Aerospace Sciences Meeting. AIAA Paper, vol. 80 (2014)
11. Mardia, K.V., Kent, J.T., Bibby, J.M.: Multivariate Analysis. Academic (1979)
12. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 95–109. Springer, Berlin/Heidelberg (2008)
13. Powell, M.J.D.: Radial basis function methods for interpolation to functions of many variables. In: Proceedings of Hellenic European Conference on Computer Mathematics and Its Applications (HERCMA), pp. 2–24 (2001)
14. Rajan, V.T.: Optimality of the Delaunay triangulation in $\mathbb{R}^d$. Discrete Comput. Geom. **12**(1), 189–202 (1994)
15. Rozza, G., Huynh, D.B.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. Arch. Comput. Methods Eng. **15**, 229–275 (2008)
16. Spivak, M.: A Comprehensive Introduction to Differential Geometry, vol. 1, 3rd edn. Publish or Perish (1999)
17. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**(5500), 2319–2323 (2000)
18. Van der Maaten, L.J.P., Postma, E.O., Van den Herik, H.J.: Dimensionality reduction: a comparative review. J. Mach. Learn. Res. **10**, 1–41 (2009)
19. Zimmermann, R., Görtz, S.: Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model. Aeronaut. J. **116**(1184), 1079–1100 (2012)