




Phase Space Projection of Dynamical Systems

Nemanja Bartolovic , Markus Gross  and Tobias Günther 

Department of Computer Science, ETH Zurich, Switzerland

Abstract

Dynamical systems are commonly used to describe the state of time-dependent systems. In many engineering and control problems, the state space is high-dimensional making it difficult to analyze and visualize the behavior of the system for varying input conditions. We present a novel dimensionality reduction technique that is tailored to high-dimensional dynamical systems. In contrast to standard general purpose dimensionality reduction algorithms, we use energy minimization to preserve properties of the flow in the high-dimensional space. Once the projection operator is optimized, further high-dimensional trajectories are projected easily. Our 3D projection maintains a number of useful flow properties, such as critical points and flow maps, and is optimized to match geometric characteristics of the high-dimensional input, as well as optional user constraints. We apply our method to trajectories traced in the phase spaces of second-order dynamical systems, including finite-sized objects in fluids, the circular restricted three-body problem and a damped double pendulum. We compare the projections with standard visualization techniques, such as PCA, t-SNE and UMAP, and visualize the dynamical systems with multiple coordinated views interactively, featuring a spatial embedding, projection to subspaces, our dimensionality reduction and a seed point exploration tool.

CCS Concepts

• **Human-centered computing** → **Visualization techniques; Scientific visualization;**

1. Introduction

The interpretation of high-dimensional data is challenging, since humans are used to three dimensions. As computational resources become more powerful, numerical simulations of dynamical processes model more and more variables, which effectively increases the dimensionality. Dynamical systems are time-evolving processes that can be interpreted as high-dimensional flows (so-called phase flows) which are embedded in a high-dimensional space (so-called phase space). Examples are the trajectories of spacecraft [MQ14], the evolution of a pendulum [LL00], and the motion of finite-sized objects in fluids [BRGG18]. In general, it is not meaningful to simply project a high-dimensional vector field to lower dimensions and to trace particles in the low-dimensional projection. Depending on the projection, this can lead to different conclusions. Our goal is to find a low-dimensional projection of a given set of high-dimensional trajectories such that certain characteristics of the high-dimensional flow are maintained. To this end, we introduce a projection operator that maintains useful flow properties by construction, e.g., critical points in the high-dimensional space remain critical points after projection. Since flow maps are preserved, our approach can be applied to study stable sets, for instance to determine spacecraft trajectories that lead to the same celestial object. The projection is optimized to preserve tangent curve properties, such as curvature, tangent magnitude, as well as density. To parameterize our projection, we use tensor products to represent an

explicit multivariate polynomial map. We compare our projection method to existing dimensionality reduction techniques, including t-SNE, UMAP and PCA, and apply our algorithm to a number of different dynamical systems. Among others, our method has the immediate advantage that further trajectories can be projected easily, once the projection operator was optimized. We explore the projection result for varying initial conditions using multiple coordinated views, featuring a spatial 2D/3D embedding, subspace views, our dimensionality reduction and a seed point exploration tool. In summary, the contributions of our paper are the following:

- We formalize the projection of a high-dimensional vector field and describe its local and integrated flow properties.
- We formulate the projection as tensor product and optimize the projection for varying polynomial degrees.

In order to demonstrate the method, we apply the projection to inertial particles in fluids, to the motion of small objects in the Earth-Moon gravitational system, and to a damped double pendulum.

Notation. Throughout the paper, we print low-dimensional vectors in bold letters \mathbf{x} and high-dimensional vectors with overlines $\overline{\mathbf{x}}$.

2. Related Work

First, we introduce the dynamical systems that we will use to evaluate our projection method. Afterwards, we describe recent work on dimensionality reduction and high-dimensional flow visualization.

2.1. Dynamical Systems

Dynamical systems describe the evolution of states, typically through ordinary differential equations (ODEs) [Per13]. Their modeling is the subject of research in mathematics and many engineering and control problems. We will use second-order ODEs, where we reserve $\mathbf{a}(\mathbf{x}, \mathbf{v})$ as a general acceleration field:

$$\bar{\mathbf{v}}(\bar{\mathbf{x}}) = \frac{d\bar{\mathbf{x}}}{dt} = \frac{d}{dt} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{x}, \mathbf{v}) \end{pmatrix} \quad (1)$$

Note that the acceleration field $\mathbf{a}(\mathbf{x}, \mathbf{v})$ may depend on both the *generalized* position \mathbf{x} and its *generalized* velocity \mathbf{v} , where *generalized* means that \mathbf{x} is not necessarily a physical spatial coordinate, but could also represent another variable that parameterizes the dynamical system, for instance the angle of a pendulum. The trajectory $\bar{\mathbf{x}}(t)$ of a state arises by numerical integration of a streamline in the above high-dimensional vector field, using $\frac{d\bar{\mathbf{x}}(t)}{dt} = \bar{\mathbf{v}}(\bar{\mathbf{x}}(t))$.

Throughout the paper, we work with three dynamical systems. The first system describes the motion of finite-sized objects in fluids [CST98], which are also known as inertial particles. The second system governs the motion of a small object in the gravitational field of two massive objects, which is the so-called circular restricted three-body problem [Lag72, MQ14]. Finally, we visualize the evolution of a damped double pendulum [LL00], in which the state is expressed by angles rather than positions. The acceleration fields of all three systems are given in Appendix A.

2.2. Multi-dimensional Data Visualization

In information visualization, many techniques for visualizing high-dimensional data have been developed over the past decades, as recently summarized by Liu et al. [LMW*16]. In this survey, three main categories were proposed based on their location in the visualization pipeline. The first category focuses on data transformation and involves a wide range of techniques focused on projection-based dimensionality reduction [PEP*11], subspace clustering [LT15] and topological data analysis [WSPVJ11]. The second category represents visual mappings, where data is encoded onto method-defined axes and specific plot types [LT13, CVW11], often enhanced with various glyphs [War08], animations [EDF08] and hierarchical views [OHJ*11]. The third category contains methods that focus on screen space and rendering, which includes techniques that extract additional shading features from simple data representations [SW09, MG13] or aim to find a view that reduces visual clutter [AdOL04]. Our work falls into the first category.

2.3. Dimensionality Reduction

Several approaches exist to reduce the complexity of ODEs [RT05]. Geometric reduction detects symmetries [BKMM96], whereas model reduction uses approximations [Ant05, TAJPO7]. We simulate solutions in the original ODE and project the solutions into a low-dimensional space. Next, we visit several standard dimensionality reduction techniques. Principal component analysis [Hot33] (PCA) forms a linear orthogonal basis, in which the coordinate axes are ordered with respect to data variance. For reduction, the axes with the highest variance are chosen. Many extensions have

been developed based on nonlinear techniques such as Kernel PCA [SSM98]. t-SNE [MH08] is a nonlinear dimensionality reduction technique where each observation in high-dimensional space is modeled by a point in a 2D or 3D space. The relationships between points in the high-dimensional space are modeled with Gaussian probability distributions which are then recreated for the corresponding projections in low-dimensional space with the help of a Student t-distribution. t-SNE is frequently used [GHS*19] and was optimized for performance [CRHC18, PTM*19]. Uniform manifold approximation and projection [MHM18] (UMAP) is a manifold learning technique used for nonlinear dimensionality reduction. This algorithm builds a weighted k nearest neighbor graph to establish the structure within the data and then applies a force-directed layout to optimize the low-dimensional graph representation. UMAP was applied for instance in biological data analysis [BMH*19]. We refer to van der Maaten et al. [VDMPVdH09] and Nonato and Aupetit [NA18] for a general entry to dimensionality reduction methods. In flow visualization, Rössl and Theisel [RT12] embedded streamlines by preserving Hausdorff distances. Han et al. [HTW18] used autoencoders to embed a binary voxelization of line and surface geometry in a latent space.

2.4. High-Dimensional Flow Visualization

In scientific visualization, several feature extraction algorithms have been extended to higher-dimensional flows [LMGP97], including the analysis of finite-sized particles in fluids. For instance, Günther et al. extracted critical points in the high-dimensional phase space [GG17] and extracted attracting manifolds via backward integration of inertial particles [GT17]. Due to the structure of the inertial phase space, a globally attracting manifold [MBZ06] exists that was visualized by Baeza Rojo et al. [BRGG18]. Tricoche et al. [TGS11] studied area-preserving maps in Hamiltonian systems. The Lagrangian coherent structure of n -body problems have been visualized by Sagrista et al. [SJJ*17]. Hofmann et al. [HRS18] classified and extracted critical points in 4D spaces and developed a camera projection for navigation. Recently, Amirkhanov et al. [AKS*19] developed a system that allows users to explore all combinations of subspace projections of 4D trajectories. While the topology of high-dimensional flows is increasingly better understood, we still lack embeddings in low-dimensional spaces that convey the high-dimensional flow patterns in an intuitive manner.

3. Phase Space Projection of Dynamical Systems

In this paper, we develop a new energy minimization approach to reduce the dimensionality of high-dimensional vector fields. First, we introduce the unknown of the minimization: a differentiable map that takes a high-dimensional flow to a low-dimensional domain. After describing the properties of the map, we formulate an energy function that rates any given map and present our optimization method in Section 4, which finds projections that follow certain geometric and user-defined constraints. The resulting projections are explored in Section 5 using multiple coordinated views.

3.1. Projection of a Vector Field

Our goal is to find an explicit, differentiable map $\mathbf{P}(\bar{\mathbf{x}})$ that takes an m -dimensional point $\bar{\mathbf{x}} \in \bar{\mathcal{D}} \subseteq \mathbb{R}^m$ to an n -dimensional point

$\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^n$, where $\overline{\mathcal{D}}$ and \mathcal{D} are the input and output domain, respectively. With $n < m$, our map is non-invertible:

$$\mathbf{x} = \mathbf{P}(\overline{\mathbf{x}}) \quad (2)$$

By considering the projection of a high-dimensional streamline $\overline{\mathbf{x}}(t)$ and by computing its material derivative, we see with Eq. (2) how a streamline tangent behaves under transformation with $\mathbf{P}(\overline{\mathbf{x}})$:

$$\frac{d\mathbf{x}(t)}{dt} = \frac{D\mathbf{P}(\overline{\mathbf{x}}(t))}{Dt} \quad (3)$$

$$= \frac{\partial \mathbf{P}(\overline{\mathbf{x}}(t))}{\partial \overline{\mathbf{x}}} \cdot \frac{d\overline{\mathbf{x}}(t)}{dt} \quad (4)$$

Since streamlines are tangent curves of a vector field $\overline{\mathbf{v}}(\overline{\mathbf{x}}(t)) = \frac{d\overline{\mathbf{x}}(t)}{dt}$, a high-dimensional flow $\overline{\mathbf{v}}(\overline{\mathbf{x}})$ is projected by Eq. (4) to:

$$\mathbf{v}(\mathbf{x}) = \nabla \mathbf{P}(\overline{\mathbf{x}}) \cdot \overline{\mathbf{v}}(\overline{\mathbf{x}}) \quad (5)$$

Likewise, the acceleration of the streamlines is transformed to:

$$\mathbf{a}(\mathbf{x}) = \frac{D\mathbf{v}(\mathbf{x})}{Dt} = \frac{D^2\mathbf{P}(\overline{\mathbf{x}})}{Dt^2} \quad (6)$$

$$= \frac{D\nabla \mathbf{P}(\overline{\mathbf{x}})}{Dt} \cdot \overline{\mathbf{v}}(\overline{\mathbf{x}}) + \nabla \mathbf{P}(\overline{\mathbf{x}}) \cdot \overline{\mathbf{a}}(\overline{\mathbf{x}}) \quad (7)$$

$$= (\nabla(\nabla \mathbf{P}(\overline{\mathbf{x}})) \cdot \overline{\mathbf{v}}(\overline{\mathbf{x}})) \cdot \overline{\mathbf{v}}(\overline{\mathbf{x}}) + \nabla \mathbf{P}(\overline{\mathbf{x}}) \cdot \overline{\mathbf{a}}(\overline{\mathbf{x}}) \quad (8)$$

3.2. Streamline Properties

Given a projection operator \mathbf{P} and knowing from Eqs. (5) and (6) how $\overline{\mathbf{v}}$ and $\overline{\mathbf{a}}$ are projected to \mathbf{v} and \mathbf{a} , we can measure geometric properties to later define an energy that rates a given projection \mathbf{P} .

Curvature. In any dimension, the streamline curvature $\kappa(\overline{\mathbf{x}})$ before projection and the curvature $\kappa(\mathbf{x})$ after projection are:

$$\kappa(\overline{\mathbf{x}}) = \frac{\sqrt{\|\overline{\mathbf{v}}\|^2 \|\overline{\mathbf{a}}\|^2 - (\overline{\mathbf{v}} \cdot \overline{\mathbf{a}})^2}}{\|\overline{\mathbf{v}}\|^3}, \quad \kappa(\mathbf{x}) = \frac{\sqrt{\|\mathbf{v}\|^2 \|\mathbf{a}\|^2 - (\mathbf{v} \cdot \mathbf{a})^2}}{\|\mathbf{v}\|^3} \quad (9)$$

Intuitively, we consider the high-dimensional streamlines and their projections to be curves embedded in a Euclidean space, in order to use their curvatures as similarity measure across dimensions.

Tangent Magnitude. In addition, we calculate the tangent magnitude before projection $\delta(\overline{\mathbf{x}})$ and after projection $\delta(\mathbf{x})$:

$$\delta(\overline{\mathbf{x}}) = \left\| \frac{d\overline{\mathbf{x}}(t)}{dt} \right\|_2, \quad \delta(\mathbf{x}) = \left\| \frac{d\mathbf{x}(t)}{dt} \right\|_2 = \left\| \frac{d\mathbf{P}(\overline{\mathbf{x}}(t))}{dt} \right\|_2 \quad (10)$$

Matching the tangent magnitude in the output domain with the tangent magnitude of the high-dimensional input domain preserves the differences between slow and fast flow regimes.

3.3. Properties of Projection

Our projection $\mathbf{P}(\overline{\mathbf{x}})$ has a number of useful properties, which we describe in the following.

Tangent Curves. Consider a high-dimensional tangent curve $\overline{\mathbf{x}}(\tau)$ that is integrated for duration τ , starting at $\overline{\mathbf{x}}(t_0) = \overline{\mathbf{x}}_0$.

$$\overline{\mathbf{x}}(\tau) = \overline{\mathbf{x}}_0 + \int_{t_0}^{t_0+\tau} \overline{\mathbf{v}}(\overline{\mathbf{x}}(t)) dt \quad (11)$$

Since $\mathbf{P}(\overline{\mathbf{x}})$ is differentiable, we obtain a continuous tangent curve

$\mathbf{x}(\tau)$ in the low-dimensional space. With Eq. (2), the seed point is projected to $\mathbf{x}_0 = \mathbf{P}(\overline{\mathbf{x}}_0)$. Furthermore, by applying Eq. (5) we obtain the curve by projection:

$$\mathbf{x}(\tau) = \mathbf{P}(\overline{\mathbf{x}}(\tau)) = \mathbf{P}(\overline{\mathbf{x}}_0) + \int_{t_0}^{t_0+\tau} \nabla \mathbf{P}(\overline{\mathbf{x}}(t)) \cdot \overline{\mathbf{v}}(\overline{\mathbf{x}}(t)) dt \quad (12)$$

Every point of a high-dimensional curve is projected onto a low-dimensional curve, i.e., flow maps are preserved by construction.

Critical Points. Another desirable property is that every critical point $\overline{\mathbf{c}}$ in the high-dimensional flow is always mapped to a critical point \mathbf{c} in the low-dimensional flow. With Eq. (5), the requirement is fulfilled by construction:

$$\overline{\mathbf{v}}(\overline{\mathbf{c}}) = \overline{\mathbf{0}} \Rightarrow \mathbf{v}(\mathbf{c}) = \nabla \mathbf{P}(\overline{\mathbf{c}}) \cdot \overline{\mathbf{0}} = \mathbf{0} \quad (13)$$

The projection generates critical points everywhere in the low-dimensional space when $\nabla \mathbf{P} = \mathbf{0}$. In this degenerate case, we would have $\mathbf{P} = \text{const}$, which never occurs in practice due to the regularizing effect of the energy terms.

Stable Sets. In dynamical systems, the stable set $\mathbf{S}(\mathbf{c})$ of a critical point \mathbf{c} is the union of all seed points in domain \mathcal{D} that lead asymptotically to \mathbf{c} [Per13]. Before and after projection, we have:

$$\overline{\mathbf{S}}(\overline{\mathbf{c}}) = \{\overline{\mathbf{c}}\} \cup \{\overline{\mathbf{x}} \in \overline{\mathcal{D}} : \exists \overline{\mathbf{x}}(t), t_0 : \overline{\mathbf{x}}(t_0) = \overline{\mathbf{x}} \wedge \lim_{t \rightarrow \infty} \overline{\mathbf{x}}(t) = \overline{\mathbf{c}}\} \quad (14)$$

$$\mathbf{S}(\mathbf{c}) = \{\mathbf{c}\} \cup \{\mathbf{x} \in \mathcal{D} : \exists \mathbf{x}(t), t_0 : \mathbf{x}(t_0) = \mathbf{x} \wedge \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{c}\} \quad (15)$$

Since flow maps are preserved, we directly retain stable sets, i.e., $\mathbf{P}(\overline{\mathbf{S}}(\overline{\mathbf{c}})) = \mathbf{S}(\mathbf{c})$. Later, we will use colors to visualize trajectories that originate from the same stable set. In the past, Günther and Theisel [GT16] visualized cross sections of the stable sets, which were decaying into pieces due to the arbitrary placement of the cross section. In a differentiable dynamical system, stable sets are connected components, which are preserved under our projection.

4. Optimization of Projection

The previous section introduced an abstract projection operator $\mathbf{P}(\overline{\mathbf{x}})$ and described its differential properties. In this section, we set out to find a desirable projection for a given set of high-dimensional trajectories $\mathcal{X} = \{\overline{\mathbf{x}}(t)\}$. For this, we define an energy E that rates a given projection \mathbf{P} by penalizing undesired behavior. Afterwards, we describe how projection \mathbf{P} is defined, which we formulate compactly as a tensor product.

4.1. Energy Minimization

The search for an optimal projection requires an energy function that computes a score for a given map. In the following, we introduce a number of energy terms that are eventually combined to form the full energy.

Geometric Costs. The first two terms specify geometric properties of the tangent curves. Since tangent curves are numerically integrated, let $\overline{\mathbf{x}}_i$ be the discrete points of tangent curve $\overline{\mathbf{x}}(t)$. Using the curvature $\kappa(\overline{\mathbf{x}})$ as defined in Eq. (9), we minimize the difference between the high-dimensional and the low-dimensional curvature,

summed up over all curves $\bar{\mathbf{x}}(\tau) \in \mathcal{X}$ in the user-defined set \mathcal{X} .

$$E_{\kappa} = \sum_{\bar{\mathbf{x}} \in \mathcal{X}} \int_{t_0}^{t_0+\tau} (\kappa(\bar{\mathbf{x}}) - \kappa(\mathbf{x}))^2 dt \quad (16)$$

$$\approx \sum_{\bar{\mathbf{x}} \in \mathcal{X}} \sum_{\bar{\mathbf{x}}_i \in \bar{\mathbf{x}}} (\kappa(\bar{\mathbf{x}}_i) - \kappa(\mathbf{x}_i))^2 \Delta t \quad (17)$$

Similarly, we use Eq. (10) to minimize the difference in tangent magnitude between low-dimensional and high-dimensional curves.

$$E_{\delta} = \sum_{\bar{\mathbf{x}} \in \mathcal{X}} \int_{t_0}^{t_0+\tau} (\delta(\bar{\mathbf{x}}) - \delta(\mathbf{x}))^2 dt \quad (18)$$

$$\approx \sum_{\bar{\mathbf{x}} \in \mathcal{X}} \sum_{\bar{\mathbf{x}}_i \in \bar{\mathbf{x}}} (\delta(\bar{\mathbf{x}}_i) - \delta(\mathbf{x}_i))^2 \Delta t \quad (19)$$

Neighborhood Cost. When multiple streamlines are part of the input set \mathcal{X} , we include neighborhood information into the optimization. For a discrete high-dimensional curve point $\bar{\mathbf{x}}_i \in \bar{\mathbf{x}}(t)$ we denote the set of the k closest discrete neighboring curve points as $\mathcal{N}_{\bar{\mathbf{x}}_i} \subset \mathcal{X} \setminus \bar{\mathbf{x}}(t)$ with $k = |\mathcal{N}|$. With this, we define the energy term:

$$E_N = \sum_{\bar{\mathbf{x}} \in \mathcal{X}} \sum_{\bar{\mathbf{x}}_i \in \bar{\mathbf{x}}} \sum_{\bar{\mathbf{y}}_j \in \mathcal{N}_{\bar{\mathbf{x}}_i}} \left(d^p(\bar{\mathbf{x}}_i, \bar{\mathbf{y}}_j) - d^p(\mathbf{x}_i, \mathbf{y}_j) \right)^2 \quad (20)$$

where $d^p(\mathbf{x}, \mathbf{y})$ denotes the Minkowski distance. In our experiments, we set $p = 2$, which corresponds to the Euclidean distance. This neighborhood term ensures that the distance to nearby trajectories is preserved after projection. In practice, we set $k = 10$ for all experiments. Other choices are discussed later in Section 7.5.

Position Constraints. Moreover, users can place q optional constraints to pin down the projection at user-defined points, i.e., $\mathbf{P}(\bar{\mathbf{q}}_i) = \mathbf{q}_i$ for $i = \{1, \dots, q\}$.

$$E_P = \sum_{i=1}^q (\mathbf{P}(\bar{\mathbf{q}}_i) - \mathbf{q}_i)^2 \quad (21)$$

We demonstrate later that these constraints can be used to place critical points at desired locations.

Total Energy. Finally, the terms above are weighted and summed up to form the total energy, which we wish to minimize:

$$E = \lambda_{\kappa} E_{\kappa} + \lambda_{\delta} E_{\delta} + \lambda_N E_N + \lambda_P E_P \rightarrow \min \quad (22)$$

with $\lambda_{\kappa}, \lambda_{\delta}, \lambda_N, \lambda_P$ being user-defined weights. The weights can either be normalized to sum up to one, or one of the weights is pinned down. We set $\lambda_{\kappa} = 1$ in all examples. If not mentioned otherwise, we set $\lambda_{\delta} = 1$. Similarly, we set $\lambda_N = 1$ when including neighborhood constraints, unless specified otherwise. The parameters are listed for all experiments and are studied later in Section 7.5.

4.2. Parameterization of Projection

In the previous sections, we introduced a differentiable projection $\mathbf{P}(\bar{\mathbf{x}})$, described its properties and defined an energy that rates the quality of a given projection. Next, we describe how the projection is modeled, i.e., what its degrees of freedoms are.

The projection $\mathbf{P}(\bar{\mathbf{x}})$ takes an m -dimensional input coordinate to an n -dimensional output. To obtain a differentiable map, we use

an n -dimensional component-wise multi-variate polynomial expansion of degree d , which we express compactly as tensor product:

$$\mathbf{P}(\bar{\mathbf{x}}) = \sum_{i=0}^d \frac{1}{i!} \nabla^i \mathbf{P}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{x}}^i \quad (23)$$

With this, derivatives are easy to calculate:

$$\nabla \mathbf{P}(\bar{\mathbf{x}}) = \sum_{i=1}^d \frac{1}{(i-1)!} \nabla^i \mathbf{P}(\bar{\mathbf{x}}) \cdot \bar{\mathbf{x}}^{i-1} \quad (24)$$

An example of the tensor product expansion into polynomials is shown in Appendix B for a mapping from 4D to 3D. The expansion is analog for mappings from higher dimensions. The total number of coefficients in this map is $n^{\binom{m+d}{d}}$. In practice, we used expansions up to polynomial degree $d = 2$ or $d = 3$. Other choices are discussed later in Section 7.5.

5. Visual Analysis of Second-Order Dynamical Systems

Next, we devise an interactive visualization system that allows us to explore the projections of high-dimensional trajectories.

Overview of the System. In order to visualize the behavior of high-dimensional trajectories, we combine a number of coordinated views. Each view sheds light onto different aspects of the data. The first two views display selected subspaces, which is necessary to obtain a spatial embedding and to get a glimpse of the structures in the spatial domain and the velocity domain. The next two views show the result of our projection and a visualization of the energy residuals, which allows us to assess the quality of a projection with respect to the energy costs. Finally, the last view shows us the projection for a range of different trajectories, i.e., we obtain a quick overview of the seed point dependence, supporting the user in the exploration. An overview of our views is given in Fig. 1. Colors used to label different trajectories were selected according to [GA10]. In the following, we explain each view in detail.

Spatial Embedding. Second-order ODEs of the form given in Eq. (1) describe the state of a system using generalized positions and velocities. If the dynamical system uses generalized coordinates that do not directly translate to spatial coordinates, a coordinate transformation is necessary to obtain meaningful spatial embeddings. For example, the double pendulum in Eq. (27) is expressed in terms of angles, from which we can calculate the exact joint and rod locations. Depending on the dimensionality of the spatial domain, we either use an interactive 2D or 3D view, as demonstrated in Fig. 1a. In all examples, trajectories are rendered with illuminated streamline shading [ZSH96] and shadow projections to provide visual depth cues. Curves are animated to give a sense of direction and to visually encode the tangent curve magnitude. Whenever available, critical points of the high-dimensional flow are visualized by colored spheres after projection.

Subspace View. For debugging purposes and to develop a general understanding of structures, symmetries and value ranges, we provide two subspace views onto the generalized positions and generalized velocities. Fig. 1b gives an example of symmetries visible in the subspaces. Across all views, selected curve points can be animated, showing the correspondence across views. If the generalized coordinates directly translate to a spatial domain, the spatial

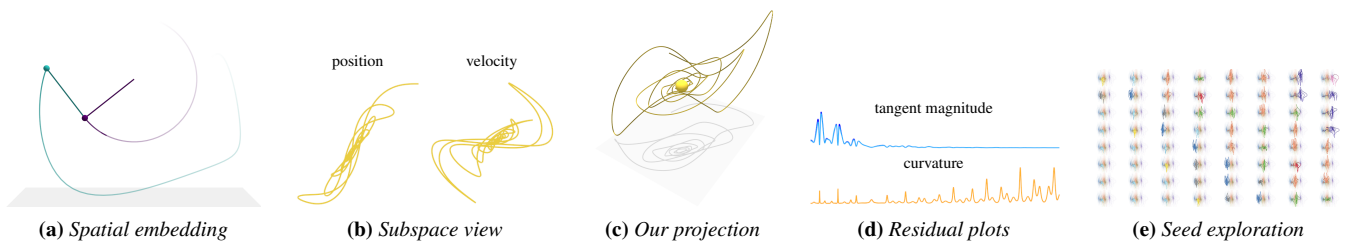


Figure 1: Overview of our visualization system for a single high-dimensional trajectory in a DAMPED DOUBLE PENDULUM. (a) The spatial embedding shows the spatial coordinates of the dynamical system in 2D or 3D. (b) The subspace view visualizes the generalized coordinates in which the system is simulated. (c) Our projection reduces the domain to 3D. (d) The residual plots show how well the individual energy terms are minimized. (e) The seed point exploration shows projections for different seed points. Color distinguishes curves or stable sets.

subspace and the aforementioned spatial embedding are the same. For more than three generalized coordinates for position or velocity, dicing methods could be used to explore subspaces [EDF08].

Projection View. The projection view in Fig. 1c shows our optimized 3D embedding. Since the energy minimization is carried out asynchronously, the user can observe how the output space is transforming as the optimization is running. We refer to the video for an example. Here, and also in all other views, we can color-code the energy residuals, color each trajectory differently, or encode the sink that is reached asymptotically in order to reveal stable sets. A user interface allows the adjustment of energy weights and parameters, and the placement of optional positional constraints.

Residual Plots. The residual plots in Fig. 1d are a useful aid in the adjustment of energy weights. For a selected optimized trajectory, each energy cost term is visualized in form of two lines in a line plot, where the horizontal direction encodes the integration duration. We plot the high-dimensional per-point tangent magnitudes as a dark blue line, with the matching low-dimensional values in light blue overlay. Similarly, we use a red line for the high-dimensional curvature and an orange overlay for its low-dimensional counterpart. Curvature is completely matched for the given example configuration. This plot allows us to identify, how well the individual constraints are met, which informs further weight adjustment.

Seed Point Exploration View. Due to the high dimensionality of the input domain, it is difficult to quickly see how the seed point influences the resulting trajectory. Our goal is to show the smoothness and stability of the projections, as we vary the seed points of the input trajectories by small amounts. Inspired by multi-dimensional stacking [GT16, SJJ*17, HG18], we tile the 2D display showing the result of a different seed point in each tile. If multiple trajectories are present in the scene, we employ a focus and context encoding. For this, the selected trajectory is highlighted in full saturation, whereas context trajectories are transparent in the background. Colors identify trajectories or stable sets, depending on the application task. In this view, similarities and differences can be quickly identified, which guides the exploration of the high-dimensional flow.

6. Implementation

The optimization of our projection requires the minimization of the energy defined in Eq. (22). In the tensor product formulation, our differentiable map is defined by polynomials and our unknowns are their monomial basis coefficients. For a polynomial

degree larger than one, the optimization problem becomes non-linear. We perform the minimization with Ceres [AMO], a library designed for large-scale nonlinear optimization. The gradients required for the solver are obtained using built-in auto-differentiation feature. For comparison, we use the public Python implementation of UMAP [MHM18]. For t-SNE, we used the reference Python implementation with Barnes-Hut optimization present in scikit-learn package [PVG*11]. We point out that recent GPU implementations of t-SNE obtain far superior performance [CRHC18, PTM*19].

7. Results

Next, we apply our method to a number of dynamical systems, study the properties of the projection, and compare with existing dimensionality reduction algorithms that were designed for projection of points rather than curves. Unless stated otherwise, we set the number of neighbors for UMAP to 10 and use a perplexity of 15 for t-SNE. We employ the Euclidean distance metric in both methods.

7.1. Undersampling Stability

Ideally, the projection of a manifold is independent of its sampling. To measure projection stability given the same initial conditions, we conduct two experiments, where we vary the integration duration, but fix the time step, and another one vice versa. Undersampling stability is important in situations where a dense sampling is computationally infeasible. Fig. 2 presents the first experiment, where we show results for PCA, t-SNE and UMAP next to our method. Note that PCA is dominated by the higher variance in the velocity subspace, giving it an almost disk-like appearance. Even when recomputing t-SNE and UMAP multiple times, we could not obtain projections that were similar across different integration durations. In comparison, our projection is stable for the same initial conditions, even when the duration of the simulation is changed. In the second experiment in Fig. 3, we vary the time step of the simulation. Neither t-SNE nor UMAP have a consistent shape across different subsampling scenarios. In contrast, our method keeps a similar shape throughout. We omitted PCA here, since it gave results very similar to those in Fig. 2.

7.2. Flow Map Preservation

Both t-SNE and UMAP work well for general purpose dimensionality reduction, where the main quality assessment lies in the algo-

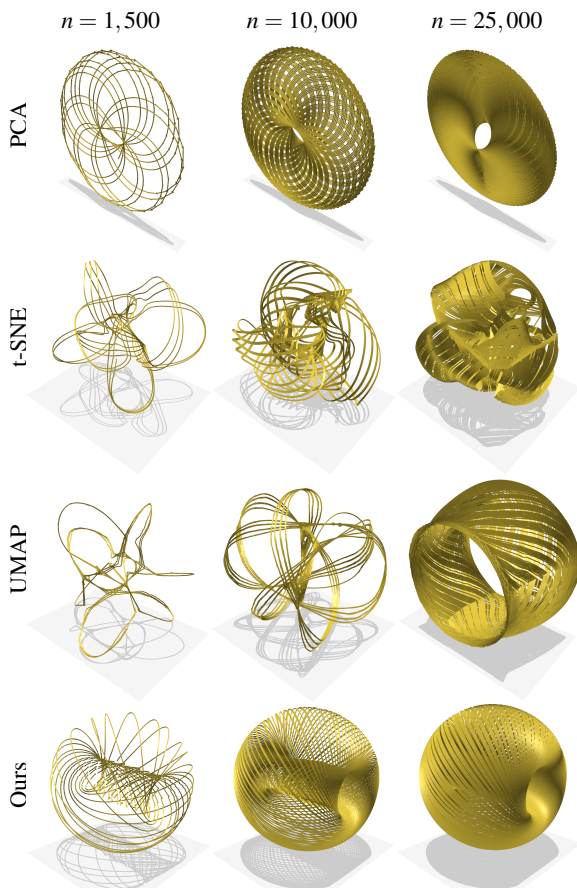


Figure 2: Comparison of dimensionality reduction methods for a single trajectory with varying number of integration steps n in the CRTBP problem. PCA is dominated by the variance in the velocity subspace. In contrast to t-SNE and UMAP, our projection already reaches stable results with fewer vertices. Initial conditions for position and velocity are $\mathbf{x}_0 = (0.42, 0)^T$ and $\mathbf{v}_0 = (0, 0.5)^T$, while the simulation timestep is $\Delta t = 10^{-2}$. We initialize the coefficients of the projection map using normal distribution with unit variance and use polynomial degree $d = 2$.

rithm's ability to cluster similar points and spread apart dissimilar ones. When applied to line geometry, this tends to result in disjoint sets of points, which disregards flow map characteristics. Note for instance in Fig. 4, how UMAP segments the trajectory into pieces. The broken links are encoded with red arrows. In our approach, flow maps are preserved by construction. As a result, there are no sudden discontinuities along the curves.

7.3. Stable Set Visualization

An important task in the analysis of dynamical systems is the identification of stable sets, i.e., sets of initial conditions that lead to the same outcome. In Fig. 5, we compare our approach with UMAP and t-SNE in a scene that contains four trajectories that approach two critical points in the limit. Even though UMAP and t-SNE manage to separate all trajectories, it would be more desirable to group them with respect to stable sets. Unlike the other methods, our ap-

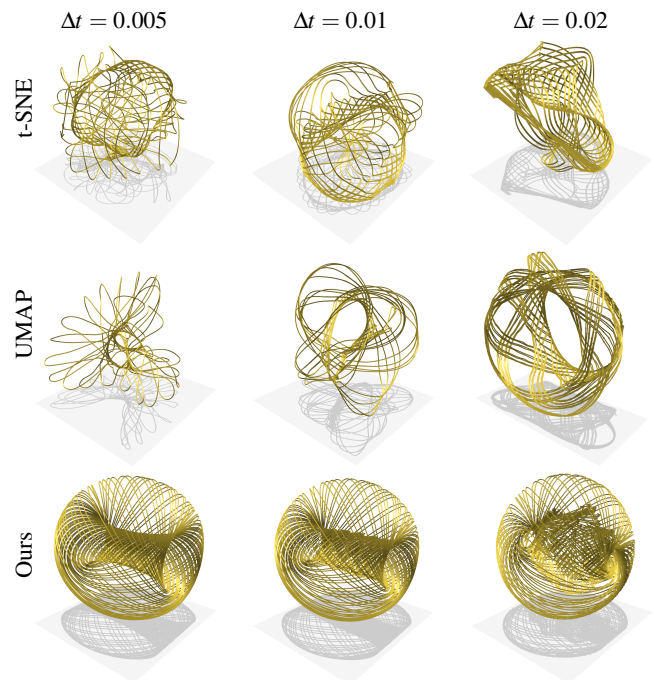


Figure 3: Comparison of dimensionality reductions for a single trajectory with varying integration step sizes in the CRTBP problem. While t-SNE and UMAP obtain different results, our embedding is not drastically changed when undersampling the curve by factor two or four. For t-SNE we used a perplexity of 15 and for UMAP we used a neighborhood size of 10, both with Euclidean distance metric. Initial conditions for position and velocity are $\mathbf{x}_0 = (0.42, 0)^T$ and $\mathbf{v}_0 = (0, 0.5)^T$ with $\tau = 50$, $\lambda_\kappa = \lambda_\delta = 1$, $d = 2$.

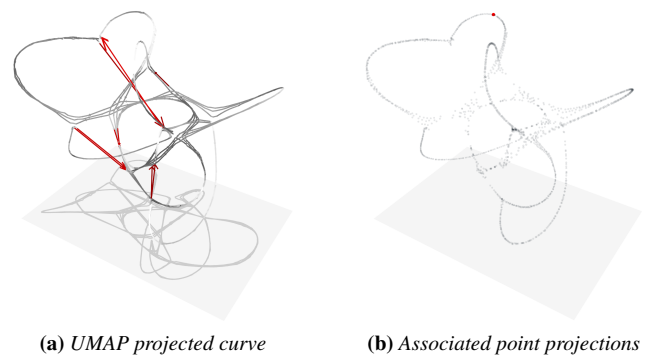


Figure 4: With UMAP, trajectories can decay into pieces, i.e., flow maps are not preserved. The red arrows in (a) indicate curve discontinuities. In (b), the underlying point set is visualized, which shows the gaps. Here, 1500 time steps in the CRTBP problem.

proach shows the two critical points, and the pairs of trajectories that enter the same critical point are mapped to similar locations.

By coloring the trajectories according to the sink that they reach asymptotically, we are able to reveal stable sets. By using the seed point exploration view, we show the projections for multiple seeding configurations at a glance. For instance in Fig. 6, we vary the initial velocities of the two bobs in the damped double pendulum.

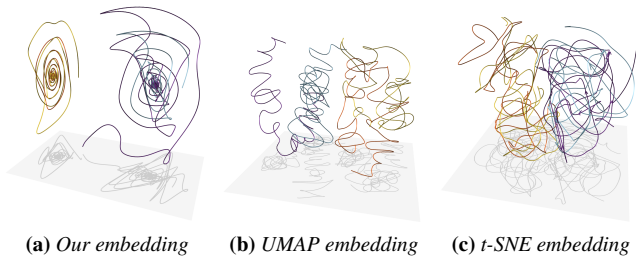


Figure 5: Visualization of four trajectories, asymptotically moving into two different sinks. Unlike UMAP and t-SNE, our method shows critical points and the asymptotic convergence. Initial conditions: $(\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)^T$, $(\frac{\pi}{2}, \frac{\pi}{2}, 6, 2.5)^T$, $(1, 1, 0, 0)^T$, $(7.7, 7.7, 0, 0)^T$. Polynomial degree $d = 3$.

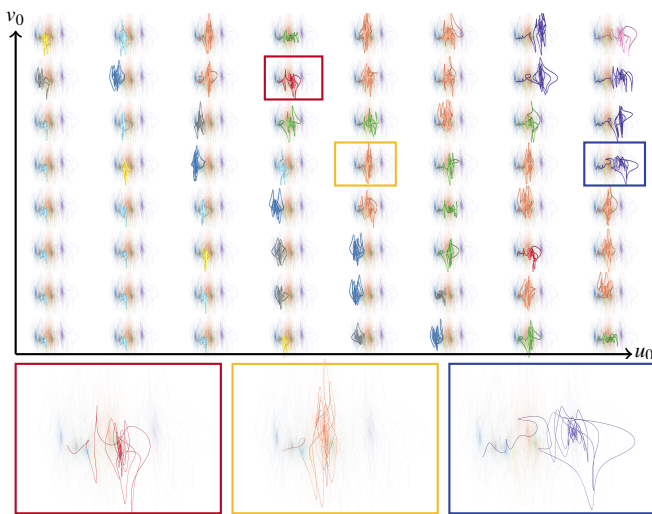


Figure 6: Projections of multiple trajectories in the double pendulum system, obtained from linearly varying seed velocity in the domain $[\pi, \pi] \times [2\pi, 2\pi]$. Thereby, (u_0, v_0) are the initial values for $(\hat{\theta}_1, \hat{\theta}_2)$. Colors indicate the stable set that the trajectory belongs to. Below, close-ups are shown for selected configurations. Polynomial degree $d = 2$.

The color mapping pre-attentively encodes which sink is reached, giving a quick overview of the areas that have coherent asymptotic behavior. As context, the projections of unselected trajectories are shown transparently in the background.

Another example of a stable set visualization using the seed point exploration is given in Fig. 7. Here, the trajectories of finite-sized particles are explored in the forced-damped Duffing vector field [GT15], cf. Eq. (25). This vector field contains two attracting focus critical points, around which hyperbolic LCS are winding. Constrained by the LCS, inertial particles reach either the left or right critical point, which is here encoded by color. With our visualization the stable sets become apparent, while also conveying an impression of the shape of the projected trajectories.

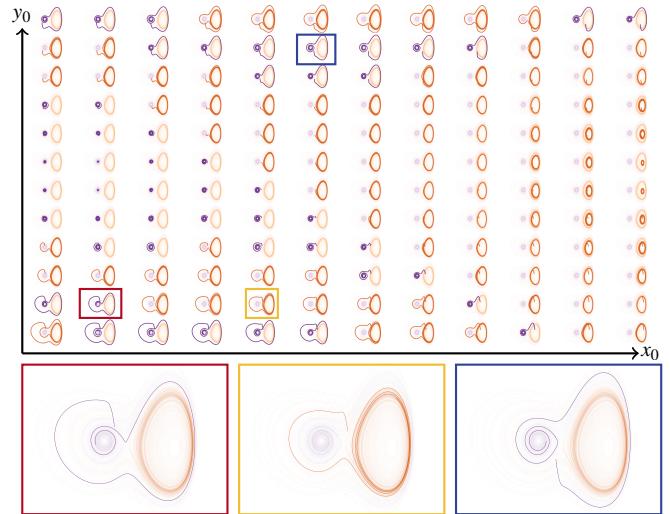


Figure 7: Projections of inertial particle trajectories for varying seed position (x_0, y_0) in the phase space of forced-damped Duffing oscillator. We vary the initial position of the particle in the domain $[-1, 1] \times [-1, 1]$. Colors indicate the critical point that the trajectory approaches asymptotically. Polynomial degree $d = 2$.

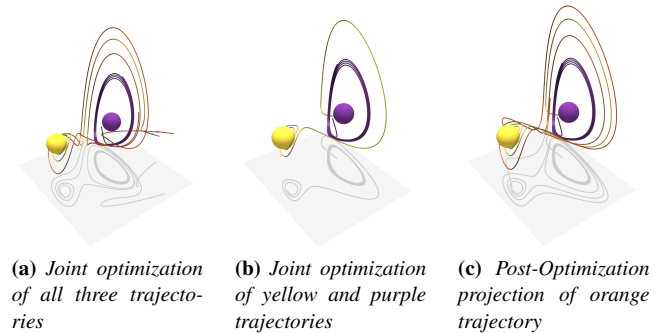


Figure 8: Since we obtain an explicit map, additional curves can be projected after optimization. Left: optimization of all three trajectories, middle: optimization of yellow and purple trajectories, right: optimization of yellow and purple trajectories with the orange being projected afterwards. The initial velocity was set to $\mathbf{v} = (0, 0)^T$, initial positions were set to $\mathbf{x}_1 = (1, 1)^T$ (yellow), $\mathbf{x}_2 = (0.6, 0)^T$ (purple) and $\mathbf{x}_3 = (-0.5, 1)^T$ (orange). Poly. deg. $d = 2$.

7.4. Post-Optimization Projection

Unlike t-SNE and UMAP, we compute an explicit projection which allows us to project further trajectories after the optimization. If the additional lines behave similar to the ones we optimized the projection for, this property is very useful to obtain quick previews and to explore seeding configurations. In Fig. 8, an example is shown for the forced-damped Duffing oscillator of the previous section. In this example, a third trajectory (orange) was projected after the joint optimization of the first two trajectories (yellow and purple). We see that the low-dimensional representation of the orange curve has a very similar shape compared to the result that is obtained when optimizing the projection for all three trajectories together.

Variable	Def. Value	Description
λ_κ	1	Curvature constraint weight
λ_δ	1	Tangent magnitude constraint weight
λ_N	1	Neighbor distance constraint weight
λ_P	1	Position constraint weight
d	2	Degree of the polynomial basis
k	10	Number of neighbors in neighborhood distance constraints

Table 1: Variables and default values. Neighborhood related variables, such as λ_N and k are only relevant when multiple trajectories are present in the input. Similarly, position constraint weight λ_P is only used when positional constraints are specified.

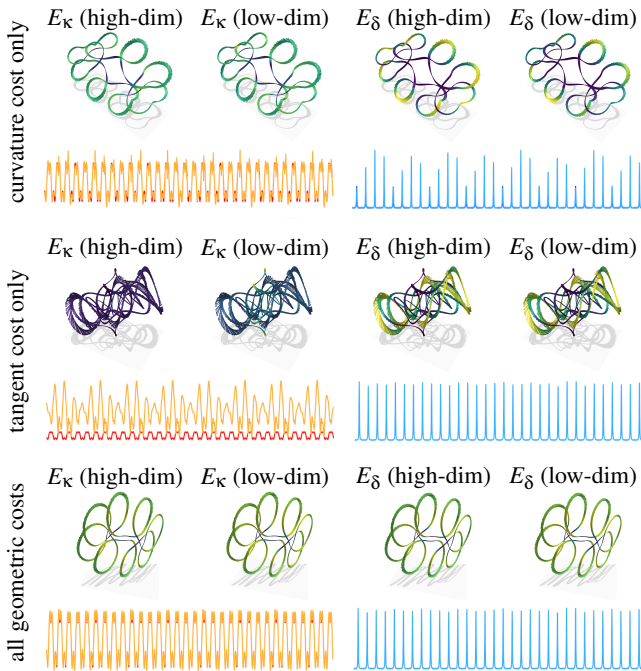


Figure 9: Parameter study of the weights for the geometric energy costs. The residual plots show for a selected trajectory how well the energy terms are minimized, which helps users to balance the energy weights. Here, for a trajectory in the CRTBP. Polynomial degree $d = 3$.

7.5. Parameter Study

Our method has a number of parameters that allow the user to control the embedding, see Table 1 for an overview. In Fig. 9, we demonstrate the influence of the geometric energy terms. The first two rows show the result for a projection in which either only the curvature cost or the tangent magnitude cost is enabled. In addition to the residual plot, we also color-code the geometric properties on the projected line geometry, encoding both the quantity in the high-dimensional space (left) and after projection (right). If the geometric properties are matched well, the two color-codings are identical. It comes at no surprise that geometric properties that are not included in the optimization are not matched well. However, by color-coding the residuals, we see where the energy is not mini-

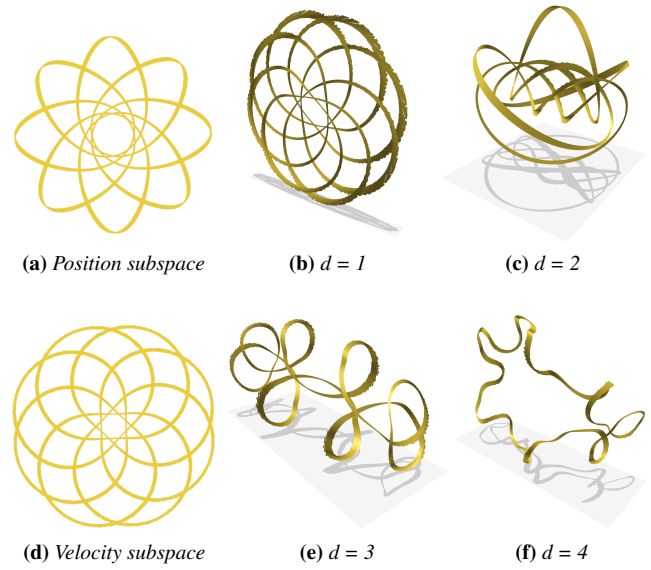


Figure 10: Parameter study of polynomial degree d in the CRTBP problem. The more degrees of freedoms are added, the more the curve can be untangled. Initial conditions for position and velocity are $\mathbf{x}_0 = (0.4, 0)^T$ and $\mathbf{v}_0 = (0, 0.5)^T$. The trajectory contains 10,000 points with a time step of 10^{-2} .

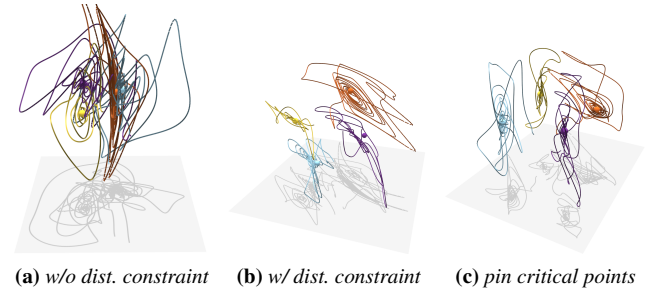


Figure 11: Users can control the preservation of distance relationships among multiple trajectories. Left: no explicit position constraints, middle: preserve neighborhoods of the high-dimensional domain, right: pin down critical points at user-defined locations. We use as initial conditions: $(\frac{\pi}{2}, \frac{\pi}{2}, 0, 0)^T$, $(\frac{\pi}{2}, \frac{\pi}{2}, 6, 2.5)^T$, $(\frac{\pi}{2}, \frac{\pi}{2}, 5.5, 0)^T$, $(\frac{\pi}{2}, \frac{\pi}{2}, 0, 5)^T$. Poly. deg. $d = 2$. In (c), $\lambda_P = 100$.

mized. In the last row, all geometric terms are enabled, which gave the best results.

In Fig. 10, we show projection results for varying polynomial degrees, along with the embedding of the position and velocity subspaces. For $d = 1$, the projection resembles the velocity subspace, as it has higher magnitude than the positions. Across all experiments, we found that the most organized results were achieved using degrees 2 and 3, which comparably represent the best trade off between quality and computation time. With degrees of 4 and higher, convergence is significantly slower, while the overall error did not reduce significantly compared to lower-degree solutions.

Next, we investigate the incorporation of distance-based constraints. Fig. 11 shows the simulation of a damped double pendu-

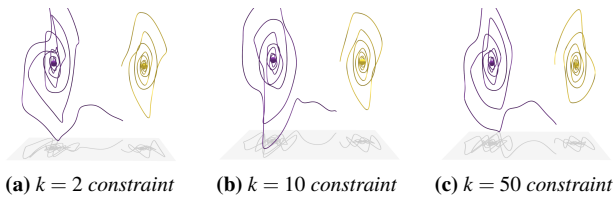


Figure 12: Embeddings for different choices of the number of neighbors k in our neighborhood costs. Different numbers of neighbors do not change the projection drastically, but still help to spread apart dissimilar trajectories. Polynomial degree $d = 2$.

lum. Here, we set up the simulations such that each trajectory ends up in a different critical point of the phase space. In the first image, we show the result of the optimization using only geometric terms. In the second picture, we include the neighborhood cost, which unclutters the view. In the last image, we also enable fixed point costs, which allows us to specify the position of critical points after projection. This optional term allows us to control the position and spacing between different trajectories.

Finally, Fig. 12 shows results for different choices of the number of neighbors k in the neighborhood cost computation. A larger neighborhood places more restrictions to the final spacing when a high amount of trajectories are present in the input data set. Otherwise, the projection remains stable.

7.6. Higher Dimensional Trajectories

While most of the experiments and examples that we have previously presented contained 4D phase spaces, our method is not limited to this particular dimension. Fig. 13 contains two examples of dynamical systems in 5D and 6D, respectively. The 5D problem in Fig. 13a describes the motion of a finite-sized object in a time-dependent 2D cylinder flow, simulated with Gerris [Pop04, GGT17]. Here, the phase space contains position (2D), velocity (2D) and time (1D). In such time-dependent systems, critical points do not exist in the phase flow, since the change in the time dimension is always one, and therefore never equal to zero. Thus, stable sets cannot be observed. Nevertheless, our phase space projection can be applied to embed 5D trajectories in a 3D space. Here, the rotating motion of inertial particles around two vortices is shown. The bend on the left is due to the initial acceleration.

The dynamical system in Fig. 13b visualizes the trajectories in the full 3D form of the circular restricted three-body problem CRTBP. Here, objects can be released in the z -dimension, resulting in an oscillating motion in this spatial dimension. In the high-dimensional space, this oscillation is manifested as a rotation, which becomes apparent in our projection.

7.7. Performance

In Table 2, we report the runtime in seconds for the data sets used throughout the experiments. The hardware setup consists of an Intel Core i7-8700k running at 4.7 GHz and 16 GB of DDR4 RAM. On average, the nonlinear optimization needs between 10 and 100 iterations to converge depending on the initial conditions. Higher-order

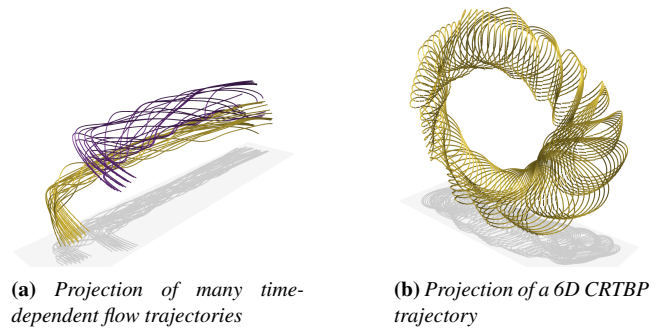


Figure 13: Examples of higher dimensional projections. Left: 5D time-dependent inertial particle flow with response time $r = 0.05$, cf. Eq. (25). Right: 6D variation of the CRTBP, with starting conditions $\mathbf{x}_0 = [0.4, 0, 0.2]^T$, $\mathbf{v}_0 = [0, 0.5, -0.2]^T$. Poly. deg. $d = 3$.

Dataset-points (dim)	UMAP	t-SNE	Our $d = 2$	Our $d = 3$
CRTBP-1500 (4D)	1.7	5.5	2.1	8.6
CRTBP-10K (4D)	5.7	56.5	7.5	34.5
CRTBP-25K (4D)	14.5	157.9	20.6	76.3
Dbl Pend-4K (4D)	7.1	52.4	12.7	51.2
Dbl Pend-8K (4D)	14.6	121.4	19.6	66.4
InertialDuff-6K (4D)	10.8	27.8	7.2	26.7
InertialCyl-12K (5D)	11.7	69.1	41.3	147.5
CRTBP6-10K (6D)	7.0	53.1	28.3	115.1

Table 2: Computation time (in seconds) of our dimensionality reduction for varying total numbers of vertices. Here, listed for the data sets in the paper. For the double pendulum and inertial particles, we used multiple trajectories and enabled the neighborhood cost with $\lambda_N = 1$. The polynomial degree is denoted with d .

polynomials typically require more iterations, in addition to being more computationally expensive. About 75 percent of the computation time is spent inside the solver, while the evaluation of Jacobian matrices and residuals takes up the remaining computational cost.

7.8. Discussion

Uniqueness. Similar to other optimization methods with a non-convex cost function, our minimization can reach a local minimum. Moreover, it can happen that the local minima differ across multiple optimization runs. This is illustrated in Fig 14, where the most common solutions are shown.

Intersecting Projections. Two points could be mapped onto the same location. With neighboring constraints enabled, this is prevented by the optimization. However, when projecting further trajectories after optimization, no guarantees can be given due to the non-invertibility of the map.

Scalability. The computational cost of a single optimization step is on the order of $O(N \cdot n \cdot m^d)$ with total number of points N , input dimension m , output dimension n and polynomial degree d . We limit d to 2 or 3, which provides an adequate model complexity for a moderate amount of trajectories (less than 100) and input dimensions m less than 10. An increase in dimensions might require a dif-

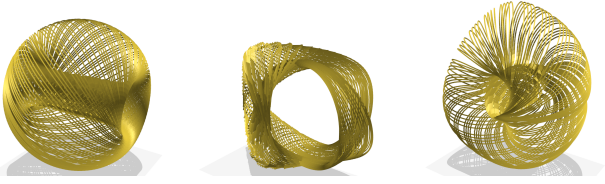


Figure 14: Different minima for random initializations of projection \mathbf{P} . The initial energy starts at the order of 10^4 and ends at 1.4×10^{-1} , 7.5×10^{-1} and 2.5×10^{-1} , respectively. Trajectories contains 10,000 points in the CRTBP with a time step of 10^{-2} , optimized for a polynomial degree $d = 2$.

ferent optimization approach due to the computational cost. Larger input data sets require more degrees of freedom for the projection.

Constraint Balancing. Increasing the energy weight of a cost term too much hinders the minimization of other cost terms, for instance in case of neighboring and positional constraints. To guide the user, we visualize the residuals and corresponding properties in color-coded plots at every intermediate optimization step, see Fig. 9. In general, energy weights of different units need balancing.

Positional Constraints. Most of our constraints operate on tangents and derived properties. If the projection is optimized without a position constraint, the problem will have a null space. We pin the solution by translating the projections to the origin of the scene.

8. Conclusions

Dynamical systems are commonly used to describe the motion of objects. The governing equations of motion, however, are often high-dimensional and therefore difficult to visualize. In this paper, we presented a novel approach to perform dimensionality reduction of high-dimensional vector fields. By describing the projection as differentiable map, we formulated an energy that preserves geometric properties and user constraints, in addition to the preservation of critical points and flow maps, which are inherent by construction. We applied the method to the trajectories in a number of second-order dynamical systems, including finite-sized objects in fluids, the circular restricted three-body problem and a damped double pendulum. We analyzed the motion with multiple coordinated views, including spatial embeddings, subspace views, our projection, residual plots and seed point exploration. The latter enabled an analysis of stable sets, which are significantly better preserved with our method compared to existing dimensionality reduction techniques, such as t-SNE, UMAP and PCA.

To improve scalability in the future, other orthogonal basis formulations with a computational framework to choose these basis functions adaptively could be explored. It remains to be shown how well the method scales to much higher dimensional systems, and how useful the method is to domain scientists in more complicated systems. Further, GPU acceleration will improve the performance. An additional direction of improvement could be the quality of the projection, by identifying additional constraints and their relation to the underlying phase-space manifolds. Finally, we would like to explore the idea of selecting optimal seed points in the high-dimensional space such that the projection yields lowest residuals at lowest computational cost.

Acknowledgements. This work was supported by ETH Research Grant ETH-07 18-1.

Appendix A: Dynamical Systems

Inertial Particles. Given a fluid flow $\mathbf{u}(\mathbf{x})$, the motion of a finite-sized object that is floating in the fluid is given by [CST98]:

$$\mathbf{a}(\mathbf{x}, \mathbf{v}) = \frac{\mathbf{u}(\mathbf{x}) - \mathbf{v}}{r} \quad (25)$$

where r is a parameter that determines the inertia. In our experiments, we use $r = 0.05$. We refer to Baeza Rojo et al. [BRGG18] for more details and further generalizations.

CRTBP. In celestial mechanics, the circular restricted three-body problem (CRTBP) [Lag72, MQ14] describes the motion of a small object in the gravity field of two massive objects placed at $\mathbf{x}_1 = (-\mu, 0)$ and $\mathbf{x}_2 = (1 - \mu, 0)$. In a reference frame rotating uniformly with the two massive bodies, the acceleration of the third body is:

$$\mathbf{a}(\mathbf{x}, \mathbf{v}) = \mathbf{x} + 2 \times \mathbf{v} + \frac{1 - \mu}{\|\mathbf{x}_1 - \mathbf{x}\|^3} (\mathbf{x}_1 - \mathbf{x}) + \frac{\mu}{\|\mathbf{x}_2 - \mathbf{x}\|^3} (\mathbf{x}_2 - \mathbf{x}) \quad (26)$$

Thereby, μ is the mass ratio between the two bodies. For instance, for the Earth-Moon problem, we have $\mu = 0.012150585609624$.

Double Pendulum. As example for a mechanical system, we use a double pendulum. Let l_1 and l_2 be the lengths of the rods with masses m_1 and m_2 at their ends. The system is described by the angles θ_1, θ_2 and their changes $\dot{\theta}_1, \dot{\theta}_2$, i.e., $\mathbf{a} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ with:

$$\ddot{\theta}_1 = \frac{f_1 - \alpha_1 f_2}{1 - \alpha_1 \alpha_2} - \varepsilon \dot{\theta}_1, \quad \ddot{\theta}_2 = \frac{f_2 - \alpha_2 f_1}{1 - \alpha_1 \alpha_2} - \varepsilon \dot{\theta}_2 \quad (27)$$

where ε is the damping rate and with the terms $\alpha_1, \alpha_2, f_1, f_2$ being:

$$\alpha_1 = \frac{l_2}{l_1} \frac{m_2}{m_1 + m_2} \cos(\theta_1 - \theta_2), \quad \alpha_2 = \frac{l_1}{l_2} \cos(\theta_1 - \theta_2) \quad (28)$$

$$f_1 = -\frac{l_2}{l_1} \frac{m_2}{m_1 + m_2} \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - \frac{g}{l_1} \sin \theta_1 \quad (29)$$

$$f_2 = \frac{l_1}{l_2} \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) - \frac{g}{l_2} \sin \theta_2 \quad (30)$$

This system is derived using Lagrangian mechanics [LL00].

Appendix B: Tensor Product Mapping

In Section 4.2, we used a tensor product notation to describe our map. In the following, we give an example for its translation to a polynomial expansion of degree k . A projection from 4D to 3D is:

$$\mathbf{P}(\bar{\mathbf{x}}) = \sum_{i=0}^k \sum_{\substack{a+b+c+d=i \\ 0 \leq a,b,c,d \leq k}} \frac{x^a y^b z^c w^d}{a!b!c!d!} \begin{bmatrix} p_{a,b,c,d} \\ q_{a,b,c,d} \\ r_{a,b,c,d} \end{bmatrix} \quad (31)$$

where $\bar{\mathbf{x}} = (x, y, z, w)$ and $p_{a,b,c,d}, q_{a,b,c,d}, r_{a,b,c,d}$ are the coefficients that we optimize for. The partial derivatives in the gradient $\nabla \mathbf{P}(\bar{\mathbf{x}})$ are straightforward to calculate, for example:

$$\frac{\partial \mathbf{P}(\bar{\mathbf{x}})}{\partial x} = \sum_{i=0}^k \sum_{\substack{a+b+c+d=i \\ 1 \leq a \leq k, 0 \leq b,c,d \leq k}} \frac{x^{a-1} y^b z^c w^d}{(a-1)!b!c!d!} \begin{bmatrix} p_{a,b,c,d} \\ q_{a,b,c,d} \\ r_{a,b,c,d} \end{bmatrix} \quad (32)$$

The other partial derivatives are analog.

References

- [AdOL04] ARTERO A. O., DE OLIVEIRA M. C. F., LEVKOWITZ H.: Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization* (2004), IEEE, pp. 81–88. [2](#)
- [AKS*19] AMIRKHANOV A., KOSIUK I., SZMOLYAN P., AMIRKHANOV A., MISTELBAUER G., GRÖLLER M. E., RAIDOU R. G.: Manylands: A journey across 4D phase space of trajectories. *Computer Graphics Forum (Proc. Pacific Graphics)* 38, 7 (2019), 191–202. [2](#)
- [AMO] AGARWAL S., MIERLE K., OTHERS: Ceres solver. <http://ceres-solver.org>. [5](#)
- [Ant05] ANTOULAS A.: An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control* 29, 2 (2005), 181–190. [2](#)
- [BKMM96] BLOCH A. M., KRISHNAPRASAD P., MARSDEN J. E., MURRAY R. M.: Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis* 136, 1 (1996), 21–99. [2](#)
- [BMH*19] BECHT E., MCINNES L., HEALY J., DUTERTRE C.-A., KWOK I. W., NG L. G., GINHOUX F., NEWELL E. W.: Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology* 37, 1 (2019), 38. [2](#)
- [BRGG18] BAEZA ROJO I., GROSS M., GÜNTHER T.: Visualizing the phase space of heterogeneous inertial particles in 2D flows. *Computer Graphics Forum (Proc. EuroVis)* 37, 3 (2018), 289–300. [1, 2, 10](#)
- [CRHC18] CHAN D. M., RAO R., HUANG F., CANNY J. F.: t-SNE-CUDA: GPU-accelerated t-SNE and its applications to modern data. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (2018), IEEE, pp. 330–338. [2, 5](#)
- [CST98] CROWE C., SOMMERFIELD M., TSUJI Y.: *Multiphase Flows with Droplets and Particles*. CRC Press, 1998. [2, 10](#)
- [CVW11] CLAESSEN J. H., VAN WIJK J. J.: Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2310–2316. [2](#)
- [EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE transactions on visualization and computer graphics* 14 (11 2008), 1141–8. [2, 5](#)
- [GA10] GREEN-ARMYTAGE P.: A colour alphabet and the limits of colour coding. *JAIC-Journal of the International Colour Association* 5 (2010). [4](#)
- [GG17] GÜNTHER T., GROSS M.: Flow-induced inertial steady vector field topology. *Computer Graphics Forum (Proc. Eurographics)* 36, 2 (2017), 143–152. [2](#)
- [GGT17] GÜNTHER T., GROSS M., THEISEL H.: Generic objective vortices for flow visualization. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 36, 4 (2017), 141:1–141:11. [9](#)
- [GHS*19] GALLI E., HARTMANN F. J., SCHREINER B., INGELFINGER F., ARVANITI E., DIEBOLD M., MRDJEN D., VAN DER MEER F., KRIEG C., AL NIMER F., ET AL.: GM-CSF and CXCR4 define a T helper cell signature in multiple sclerosis. *Nature medicine* 25, 8 (2019), 1290–1300. [2](#)
- [GT15] GÜNTHER T., THEISEL H.: Finite-time mass separation for comparative visualizations of inertial particles. *Computer Graphics Forum* 34, 3 (2015), 471–480. [7](#)
- [GT16] GÜNTHER T., THEISEL H.: Inertial steady 2D vector field topology. *Computer Graphics Forum (Proc. Eurographics)* 35, 2 (2016), 455–466. [3, 5](#)
- [GT17] GÜNTHER T., THEISEL H.: Backward finite-time Lyapunov exponents in inertial flows. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)* 23, 1 (2017), 970–979. [2](#)
- [HG18] HOLBEIN J., GÜNTHER T.: Parameter space comparison of inertial particle models. In *Vision, Modeling and Visualization* (2018), pp. 63–70. [5](#)
- [Hot33] HOTELLING H.: Analysis of a complex of statistical variables into principal components. *Journal of educational psychology* 24, 6 (1933), 417. [2](#)
- [HRS18] HOFMANN L., RIECK B., SADLO F.: Visualization of 4D vector field topology. *Computer Graphics Forum* 37, 3 (2018), 301–313. [2](#)
- [HTW18] HAN J., TAO J., WANG C.: Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE Transactions on Visualization and Computer Graphics* (2018). [2](#)
- [Lag72] LAGRANGE J.-L.: Essai sur le probleme des trois corps. *Prix de l'académie royale des Sciences de paris* 9 (1772), 292. [2, 10](#)
- [LL00] LANDAU L., LIFSHITZ E.: *Mechanics: Volume 1 (course of theoretical physics series)*, 2000. [1, 2, 10](#)
- [LMGP97] LÖFFELMANN H., MROZ L., GRÖLLER E., PURGATHOFER W.: Stream arrows: enhancing the use of stream surfaces for the visualization of dynamical systems. *The Visual Computer* 13, 8 (Nov 1997), 359–369. [2](#)
- [LMW*16] LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics* 23, 3 (2016), 1249–1268. [2](#)
- [LT13] LEHMANN D. J., THEISEL H.: Orthographic star coordinates. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2615–2624. [2](#)
- [LT15] LEHMANN D. J., THEISEL H.: Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2015), 609–618. [2](#)
- [MBZ06] MOGRABI E., BAR-ZIV E.: On the asymptotic solution of the Maxey-Riley equation. *Physics of Fluids* 18, 5 (2006). [2](#)
- [MG13] MAYORGA A., GLEICHER M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE transactions on visualization and computer graphics* 19, 9 (2013), 1526–1538. [2](#)
- [MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research* 9, Nov (2008), 2579–2605. [2](#)
- [MHM18] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). [2, 5](#)
- [MQ14] MUSIELAK Z. E., QUARLES B.: The three-body problem. *Reports on Progress in Physics* 77, 6 (jun 2014), 065901. [1, 2, 10](#)
- [NA18] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2018), 2650–2673. [2](#)
- [OHJ*11] OESTERLING P., HEINE C., JANICKE H., SCHEUERMANN G., HEYER G.: Visualization of high-dimensional point clouds using their density distribution's topology. *IEEE Transactions on Visualization and Computer Graphics* 17, 11 (2011), 1547–1559. [2](#)
- [PEP*11] PAULOVICH F. V., ELER D. M., POCO J., BOTHA C. P., MINGHIM R., NONATO L. G.: Piece wise laplacian-based projection for interactive data exploration and organization. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1091–1100. [2](#)
- [Per13] PERKO L.: *Differential equations and dynamical systems*, vol. 7. Springer Science & Business Media, 2013. [2, 3](#)
- [Pop04] POPINET S.: Free computational fluid dynamics. *ClusterWorld* 2, 6 (2004). [9](#)
- [PTM*19] PEZZOTTI N., THIJSSSEN J., MORDVINTSEV A., HÖLLT T., VAN LEW B., LELIEVELDT B. P. F., EISEMANN E., VILANOVA A.: GPGPU linear complexity t-SNE optimization. *IEEE Transactions on Visualization and Computer Graphics* (2019), 1–1. [2, 5](#)

- [PVG*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPEAU D., BRUCHER M., PERROT M., DUCHESNAY E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. 5
- [RT05] REGA G., TROGER H.: Dimension reduction of dynamical systems: Methods, models, applications. *Nonlinear Dynamics* 41, 1 (Aug 2005), 1–15. 2
- [RT12] RÖSSL C., THEISEL H.: Streamline embedding for 3D vector field exploration. *IEEE Transactions on Visualization and Computer Graphics* 18, 3 (March 2012), 407–420. 2
- [SJJ*17] SAGRISTA A., JORDAN S., JUST A., DIAS F., NONATO L. G., SADLO F.: Topological analysis of inertial dynamics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 950–959. 2, 5
- [SSM98] SCHÖLKOPF B., SMOLA A., MÜLLER K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation* 10, 5 (1998), 1299–1319. 2
- [SW09] SANFTMANN H., WEISKOPF D.: Illuminated 3D scatterplots. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 751–758. 2
- [TAJP07] TABUADA P., AMES A. D., JULIUS A., PAPPAS G. J.: Approximate reduction of dynamical systems, 2007. 2
- [TGS11] TRICOCHÉ X., GARTH C., SANDERSON A.: Visualization of topological structures in area-preserving maps. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1765–1774. 2
- [VDMPVdH09] VAN DER MAATEN L., POSTMA E., VAN DEN HERIK J.: Dimensionality reduction: a comparative. *J Mach Learn Res* 10, 66–71 (2009), 13. 2
- [War08] WARD M. O.: Multivariate data glyphs: Principles and practice. In *Handbook of data visualization*. Springer, 2008, pp. 179–198. 2
- [WSPVJ11] WANG B., SUMMA B., PASCUCCI V., VEJDEMO-JOHANSSON M.: Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 1902–1911. 2
- [ZSH96] ZOCKLER M., STALLING D., HEGE H.-C.: Interactive visualization of 3D-vector fields using illuminated stream lines. In *Proceedings of Seventh Annual IEEE Visualization'96* (1996), IEEE, pp. 107–113. 4