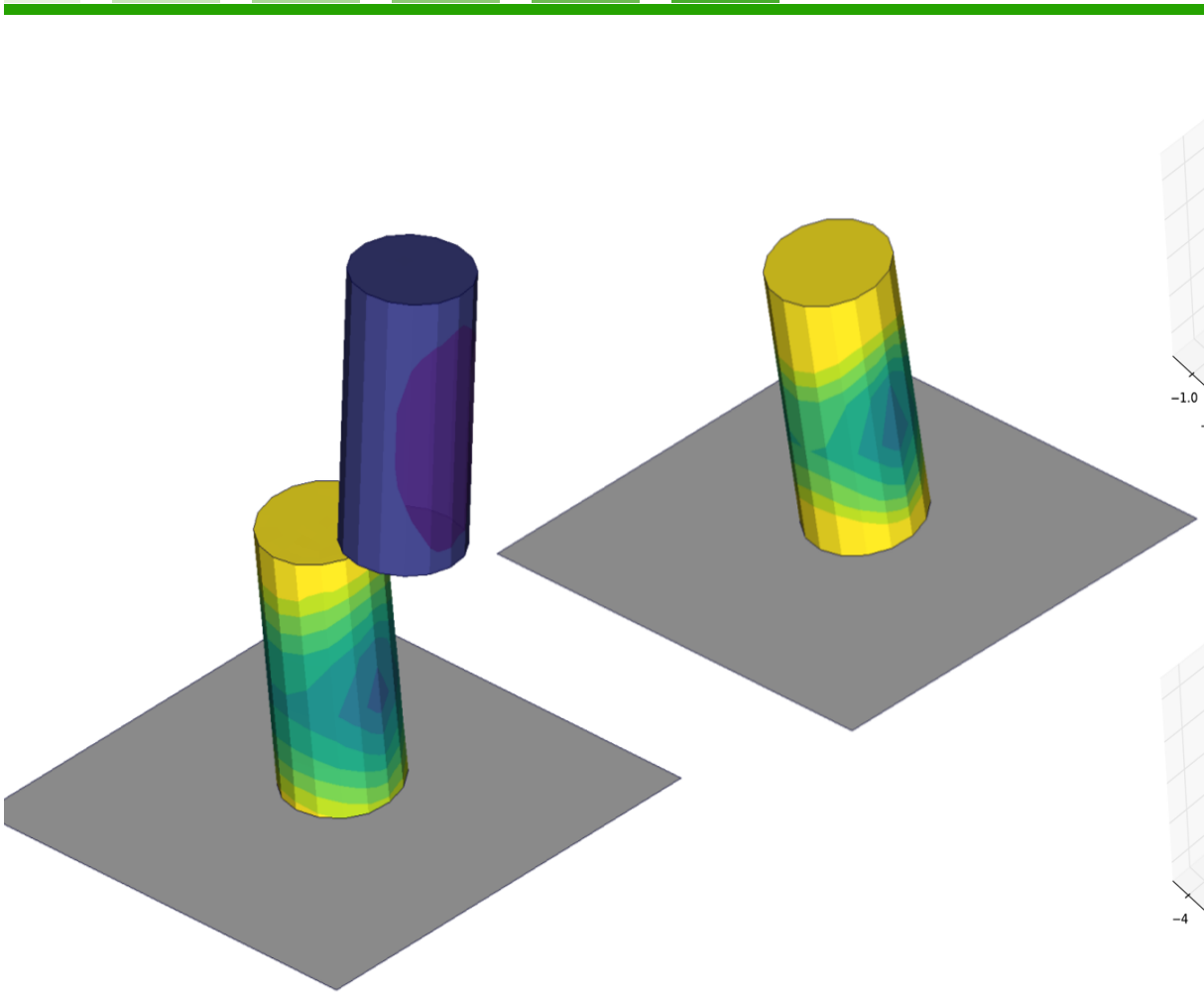
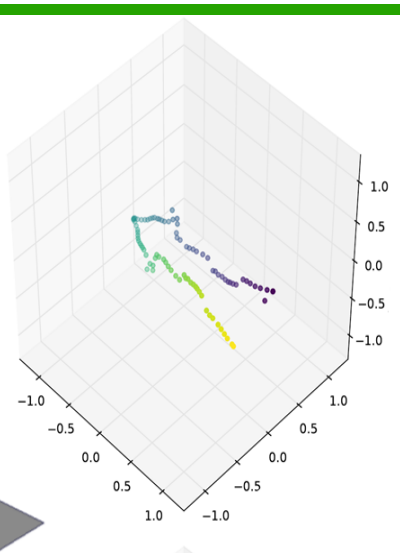
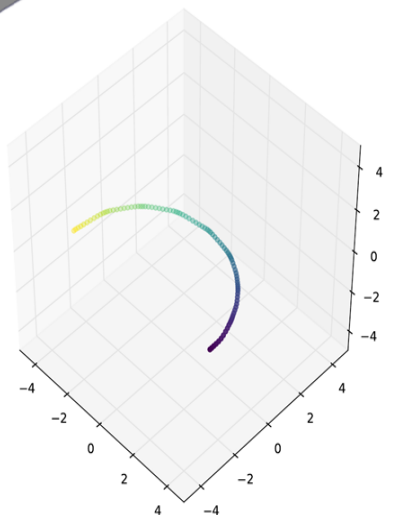
Comparative Analysis of Crash Simulation Results using Generative Nonlinear Dimensionality Reduction

Bergische Universität Wuppertal Lehrstuhl für Optimierung mechanischer Strukturen

Stefan Matthias Mertler







BERGISCHE
UNIVERSITÄT
WUPPERTAL

Comparative Analysis of Crash Simulation Results using Generative Nonlinear Dimensionality Reduction

Dissertation
to obtain a doctoral degree

in the
School of Mechanical Engineering and Safety Engineering

University of Wuppertal

Submitted by:
Stefan Matthias Mertler
from Radevormwald

Wuppertal 2022

Exam Date: 21st June 2022

Abstract

Numerical simulations are an integral part of today's product development process. Analysing and comparing multiple simulation results is a time consuming but necessary step in utilising several results. Thus, it is important to develop methods which speed up this Comparative Analysis by identifying differences and commonalities in the results and provide means to visualise possible variances. Furthermore, it is crucial to accurately determine how these variances or similarities are related to each other.
The so-called Dimensionality Reduction Methods (DRMs) have been used in extracting the underlying structure of variance in simulation results since several years. In recent years, the need for nonlinear reduction approaches has been shown in several applications. One widely used analysis method called Difference Principal Component Analysis (DPCA), which is specifically designed to compute the correlation between variation in different parts of the simulation results, is, however, based on a linear reduction approach. The aim of this dissertation is to extend the DPCA with nonlinear Dimensionality Reduction (DR), which has never been done until now.
To achieve this aim, the two underlying steps of the DPCA's analysis workflow were modified. For the first step of DR, several established methods of three classes of generative DRMs have been extended amongst others by importance factors to be used in this analysis. For the second so-called subtraction step, the new generalised concept of Difference Dimensionality Reduction was introduced.
Two specific implementations of this general concept were implemented in this work, which can be combined with the different reduction methods in the first step. The newly developed methods were thoroughly tested on multiple examples in this work: Firstly, on artificial examples to test the individual steps in an isolated environment and secondly on simulation results to evaluate the method's performance on realistic data sets. These new approaches were able to accurately determine the correlation between artificial data sets and provided better results for different parts of a set of simulation results compared to the state of the art. In the case of a nonlinear relation between these parts, the superiority over linear approaches was demonstrated in the evaluation, while underlying linear dependencies were also confirmed by the nonlinear methods.
With the successful modifications done in this work, the DPCA's workflow is now meaningfully applicable to data sets with nonlinear dependencies. While the linear variant was used in the context of crash simulations for several years, this application was questionable since it contains many nonlinearities.

The newly developed variants remove this uncertainty, and the evaluation examples suggest a broad range of possible applications for these new methods, as nonlinearities can occur in many data sets, resulting for example from topology optimisation or parameter variation.

Kurzfassung

Numerische Simulationen sind ein unverzichtbarer Bestandteil der heutigen Produktentwicklung. Mehrere Simulationsergebnisse zu analysieren und zu vergleichen ist ein zeitaufwändiger, aber notwendiger Schritt, um mehrere Simulationen in einem Entwicklungsprozess zu nutzen. Daher ist es wichtig, Methoden zu entwickeln, welche solch eine vergleichende Analyse beschleunigen, indem Unterschiede und Gemeinsamkeiten in den Simulationsergebnissen identifiziert, sowie Mittel bereitgestellt werden, mögliche Abweichungen zu visualisieren. Weiterhin ist es entscheidend, zuverlässig zu bestimmen, wie diese möglichen Abweichungen oder Ähnlichkeiten zusammenhängen.
Sogenannte Dimensionsreduktionsmethoden (DRM) werden seit mehreren Jahren dazu verwendet, die zugrundeliegenden Strukturen von Streuungen in Simulationsergebnissen zu ermitteln. In den letzten Jahren wurde die Notwendigkeit zur Verwendung nichtlinearer Reduktionsmethoden in mehreren Anwendungen gezeigt. Eine weitverbreitete Analysemethode genannt *Difference Principal Component Analysis* (DPCA), welche speziell dazu entwickelt wurde, Korrelationen zwischen Streuungen auf verschiedenen Bauteilen in Simulationsergebnissen zu ermitteln, basiert hingegen auf einem linearen Reduktionsansatz. Das Ziel dieser Dissertation ist diese DPCA durch nichtlineare Dimensionsreduktion (DR) zu erweitern, was bis jetzt noch nie getan wurde.
Um dieses Ziel zu erreichen, wurden die zwei zugrundeliegenden Schritte des Analyseprozesses der DPCA modifiziert. Für den ersten Schritt der DR wurden mehrere etablierte Methoden aus drei Klassen von DRM unter anderem um bestimmte *Importance Factors* erweitert, um in dieser Art von Analyse verwendet zu werden. Für den zweiten, den sogenannten Subtraktionsschritt, wurde das neue generalisierte Konzept der Differenz-Dimensionsreduktion eingeführt. Zwei spezifische Umsetzungen dieses allgemeinen Konzepts wurden in dieser Arbeit realisiert, welche mit den verschiedenen Reduktionsmethoden aus dem ersten Schritt kombiniert werden können.
Die neu entwickelten Methoden wurden in dieser Arbeit ausgiebig auf mehreren Beispielen getestet. Zuerst auf künstlichen Beispielen, um die einzelnen Schritte in einer isolierten Umgebung zu testen, und anschließend auf Simulationsergebnissen, um die Leistungsfähigkeit der Methoden auf realistischen Datensätzen zu evaluieren. Diese neuen Ansätze waren in der Lage, Korrelationen in künstlichen Datensätzen exakt zu bestimmen und ergaben bessere Ergebnisse auf verschiedenen Bauteilen einer Schar an Simulationsergebnissen als der aktuelle Stand der Technik.
An dem Fall eines nichtlinearen Zusammenhangs zwischen Bauteilen wurde die Überlegenheit gegenüber linearen Ansätzen demonstriert, während bestehende lineare Zusammenhänge durch die nichtlinearen Methoden bestätigt wurden.

Mit den erfolgreichen Modifikationen, welche in dieser Arbeit durchgeführt wurden, ist der Analyseprozess der DPCA nun sinnvoll auf Datensätze mit nichtlinearen Abhängigkeiten anwendbar. Obwohl die lineare Variante seit vielen Jahren im Kontext von Crash-Simulationen verwendet wurde, war diese Anwendung fragwürdig, da diese Simulationen viele Nichtlinearitäten enthalten.
Die neu entwickelten Varianten helfen diese Ungewissheit zu entfernen und die Evaluationsbeispiele legen ein breites Spektrum an möglichen Anwendungen für diese neuen Methoden nahe, da Nichtlinearitäten in vielen Datensätzen auftreten können, zum Beispiel in Ergebnissen aus Topologieoptimierungen oder Parametervariationen.

Acknowledgements

The content of this dissertation was created as part of my position at the SIDACT GmbH in cooperation with the Chair for the Optimization of Mechanical Structures at the University of Wuppertal. I would like to thank all friends and colleagues, who supported me and helped to make this thesis possible.

First of all, I would like to thank Prof. Dr.-Ing. Axel Schumacher for the continuous, valuable support of this dissertation project and for always finding time to discuss emerging topics. I am also very glad and thankful that Prof. Dr. rer. nat. Jochen Garcke agreed to be the second supervisor to this thesis and that he provided interesting further mathematical incentives.

Special thanks are also due to Clemens-August Thole, who introduced me to the theoretical background of the DPCA and also enabled the practical execution of this dissertation project at the SIDACT GmbH. Though all SIDACT employees were always very supportive of my research work, I would like to explicitly thank Dr. rer. nat. Lennart Jansen, Dr. rer. nat. Stefan Müller and Dominik Borsotto for their ongoing feedback, helpful advice and the interesting discussions. I also thank Yvonne Havertz for proofreading this thesis and for her helpful remarks.

I was always enjoying the regular meetings with the doctoral candidates in Wuppertal. The fruitful discussions and interesting research topics were a great motivation for my own research. While I am thankful that many fellow students were supporting me during this project, I would like to point out Dr.-Ing. Constantin Diez, who integrated me at the start, and Dr.-Ing. Jana Büttner, who was a great help during the final steps of the project.

A very special thanks to all members of my family for their constant support. Especially to my parents Ragnhild and Matthias, whose advice and support helped me through school, university, and life so far. And finally to my lovely wife Sabrina, who was most affected by this project: Without your patience, your support, and your help, I could not have finished this project.

Bonn, June 2022 Stefan Mertler

Contents

List of Abbreviations

List of Symbols

1 Introduction

Numerical simulations are nowadays taking a key role in product development, where they are utilised to speed up development cycles and improve overall safety. In the automotive industry for example, this is achieved by replacing time and resource consuming physical prototypes and tests by simulating virtual experiments, which are not only cheaper in both ways, but can provide unique insights as well. That is especially the case for the application of crash applications, where prototypes cannot be re-used and measurements cannot be repeated easily. The more physical tests are replaced by virtual ones, the more important the understanding of these simulation and their results becomes. To get a better understanding, engineers need to analyse the simulations and compare them with each other, in other words, perform what is known as a Comparative Analysis.

Technical advances in information technologies over the last decade have been an enabler for both: An inclining number of calculated simulations and an increased level of detail in the individual simulation result as well. While the ever-faster accelerating development cycles lead to an increasing need to analyse and understand the generated simulations at an even faster pace, the greater number of simulation results and the more detailed results make this task even more difficult.

As a consequence, a number of methods, tools and software solutions have been developed in recent years to assist with this crucial task. Some of them focus on certain aspects of a single simulation, such as individual solving time or specific performance thresholds. Others focus on multiple simulations, which are related to each other, such as different variants of the same car under similar load cases. For the analysis of such sets of simulation results, the so-called Dimensionality Reduction Methods are becoming increasingly popular, in part because they calculate a low dimensional representation of the results, that allow analysts to get an overview over many simulations at once.

One of these reduction methods is the so-called Difference Principal Component Analysis (DPCA), which is implemented in the commercial software DIFFCRASH and used by a broad customer base all over the world. The basic idea is to apply a Principal Component Analysis on different parts of the simulation, in order to determine two things: First the number of important effects acting on these parts and second the peculiarity of the individual effects. In a unique additional step these effects are then correlated to statistically identify or validate dependencies between different parts.

Though the procedure can be applied to the results of many simulations, it is most commonly used to help engineers in the discipline of analysing crash load cases, which show a lot of nonlinearities. These nonlinearities can for example originate from large deformations, velocity dependent behaviour, nonlinear material properties, failure

conditions and contact phenomenons. While this approach is used successfully by several car manufacturers in automotive engineering worldwide for several years, the underlying concept is linear Dimensionality Reduction and therefore could be questioned in the context of nonlinear crash simulations. In recent years, many nonlinear Dimensionality Reduction Methods have been developed and some of them are already applied to crash test simulation results. This raises the central research questions of this dissertation:

> *How can the basic concept of the DPCA be extended to a certain category of nonlinear methods and which functional differences separate these new methods from the linear approach?*

The aim of this dissertation is to answer these questions, which is done in the following steps:
Initially, Chapter 2 begins with giving a more detailed introduction into the Comparative Analysis. The important aspects of this analysis are elaborated to show where the differences in the established approaches lie. This summarises the state of the art and highlights the gap in the existing research.
Next, the technical topic of Dimensionality Reduction is addressed in Chapter 3 and its subsections. A certain level of detail is needed, in order to introduce the underlying models of the various methods and their differences. Thus, after some common terms are introduced, the base models of several methods are explained in dedicated subsections followed by the modifications which were performed in this thesis. With these modified base models introduced, everything is in place to motivate and then describe the further steps.
Extending on these introductions, Chapter 4 firstly explains the concept of DPCA in detail. Secondly, it shows, how this concept was generalised to the new Difference Dimensionality Reduction and how this derived concept was extended to two nonlinear difference approaches.
The performances of these new approaches are evaluated and shown in Chapter 5, starting with results for artificial manifold examples and continuing with examples for simulation results. The artificial examples help to measure the exact performance in controlled environments while exploring various aspects in detail. The simulation result examples showcase the performance on real data and have increasing complexity resulting in the practical application.
Finally, the results for the given examples as well as other findings are summarised and critically discussed in Chapter 6, before concluding with an outlook on possible future work.

2 Comparative Analysis

This chapter explains important aspects of a Comparative Analysis in the context of simulation results. The underlying ideas of established approaches are introduced and their differences are explained. Based on these existing approaches, the need for the research conducted in this thesis is highlighted.

2.1 Wording and Usage

The term Comparative Analysis (CA) originates from the field of literature analysis, where it defines a written text to "compare and contrast two things: Two texts, two theories, two historical figures, two scientific processes, and so on. [It is] about two similar things that have crucial differences [...] or two [...] things that have crucial differences, yet turn out to have surprising commonalities", see [Wal98].
In the context of simulation results, a CA is not necessarily a written text, but any visual representation highlighting the differences and commonalities between at least two and possibly a large data set of simulation results. The wording Comparative Analysis is used in the context of simulation results amongst others in [GIT14], though the objective to find differences and commonalities in simulation results, has been around for several years, see for example [TM03] and [AGHH08].
In order to compare and contrast two data sets, analysts can compare each single value in one of these data sets with its best equivalent in the other. But the model size of the discretised geometry used in modern simulations has been growing since several years [BFG05]. Today's simulation results often contain millions of elements or nodes [MCR+18] and several physical quantities or post values, which are defined on these elements, such as nodal displacements or element strains [Sch20]. This huge dimension makes a manual value-by-value comparison infeasible. And it gets even more cumbersome for large sets with many simulation results, where this comparison would have to be done for all possible combinations. Since the number of conducted simulations in industry has been increasing [RBG+16], these sets may get even larger in the future. Thus, an alternative for this exhaustive approach is needed.
Automated evaluation processes in industry often rely on a subset of extracted key values or measurements, e.g. the Head Injury Criterion (HIC) [Hen98] or the Occupant Load Criterion (OLC) [KGE08]. The advantage of analysing such criteria is, that they are standardised and thus easily comparable and also transferable to other tests. The disadvantages include the possibility to miss certain behaviour or effects, if no measurements are defined for these effects. For example, an insufficient modelled contact might go unnoticed if only the HIC is evaluated. Furthermore, these simple time series or even scalar values often fail to capture the structure of the given data set.

An alternative to extracting a few values is to take all values in the region of interest and calculate a representation, which is understandable by the person conducting the analysis. One possibility is to compute the differences or variance for the values among the simulation results and visualise this range as a fringe plot on the geometry of the model, see e.g. [TM03] or [WB12]. While this approach can help to identify areas in the region of interest, where the values are similar or different, the structure of possible variation or scatter might be captured insufficiently.
Another possibility is to reduce the dimensionality of the problem, meaning to reduce the large number of quantities to fewer but still as meaningful values. This so-called Dimensionality Reduction approach [LV07] has already been used in the analysis of simulation results for several years [TM03], [Bel03],[MT05]. Recent advances [BGG16], [IT16], [Die19], [KGS20], [ITMHG20] in utilising this approach further underline the potential of these methods.
One particular approach is implemented in the DIFFCRASH software and is used in a variety of publications and applications [TNNC10], [EKM+13],[BBT13], [Oka15], [BST15],[Oka17], [MCR+18], [OOB19], [MSJ20].

2.2 Existing Approaches

The different existing approaches using Dimensionality Reduction in an analysis vary in two aspects: First, the underlying method, which is used to perform the reduction, and second, how the outcome of the reduction is utilised in the analysis. The differences in the reduction methods themselves will be discussed in detail later in Chapter 3, for this section it is only important to mention, that all methods are either linear or nonlinear, see Section 3.1 for more details. To explain the differences, in how the result is utilised in an analysis, it is important to first abstract the analysis process. The base principle of utilising Dimensionality Reduction in a Comparative Analysis can be roughly described in three steps: The data set is preprocessed, then the reduction is applied and finally some postprocessing is performed to extract knowledge from the given representation. For example, preprocessing could be normalisation of the data, and postprocessing could include clustering or rule mining. Though the workflow visualised in Fig. 2.1 is a strong simplification, the abstraction helps to compare different approaches.

Figure 2.1: Generic workflow of Dimensionality Reduction in a Comparative Analysis. This general concept is applicable to most analysis workflows and a similar representation can be found in [BGIT+13].

In [MT08] a linear technique was used for the Dimensionality Reduction step and afterwards the simulations were clustered according to the new low dimensional values in the postprocessing step. This concept of “spectral clustering” [BGIT+13] was

extended to several nonlinear reduction approaches in the second publication. The aim of these clustering investigations is to identify instabilities and bifurcations in the data by analysing the presence of clusters in the nodal displacements. The investigation in [BGIT+13] has shown that the nonlinear methods perform better in this task than the linear variants.
In [GIT15] and [IT16] the Dimensionality Reduction is used to enable analysts to explore large data sets of simulation results. This means that the low dimensional values are used to provide a visual representation for multiple results and to identify and select interesting candidates to be viewed in the original dimension. Furthermore, the methods help to traverse the different sample points in a meaningful order as well as understanding the so-called "deformation shapes" [IT16]. These deformation shapes refer to the different kinds of variation in the nodal displacements of the simulation results, e.g. buckling or torsion. Both publications provide convincing results for nonlinear approaches and [GIT15] explicitly shows the superiority compared to a linear approach. This comparison was extended to an additional nonlinear method in [KGS20].
In [DWHS16] and [Die19] the Dimensionality Reduction is used for knowledge generation. More precisely, the postprocessing step is the so-called "rule mining" [Die19], where decision trees are used in order to find relations between specified input variables of the simulations and the newly computed low dimensional values of the simulation results. Furthermore, this workflow performs a novel preprocessing in which the internal energy of the simulations' parts is mapped to simpler regression shapes such as curves or planes. This preprocessing has two effects: Firstly it enables a comparison of parts with different geometries and secondly it renders the reduction step nonlinear.

The focus of this work is on a certain type of analysis, which was first published in [TNNC10], where the correlation between scatter on different parts of a car are related to each other within a set of simulation results. This process called Difference Principal Component Analysis (DPCA) is in detail explained in Section 4.1. In this specific use case, the generic workflow is extended by a second input data set as well as an additional step of what is labelled as Difference Dimensionality Reduction in this thesis. In a first step, the Dimensionality Reduction is applied to the first data set "A" in order to get the low dimensional values. These low dimensional values are then "subtracted" [TNNC10] from the second data set "B" to determine the correlation of the two. This process is depicted in Fig. 2.2 and referred to as the Extended Workflow (EW) in this dissertation.

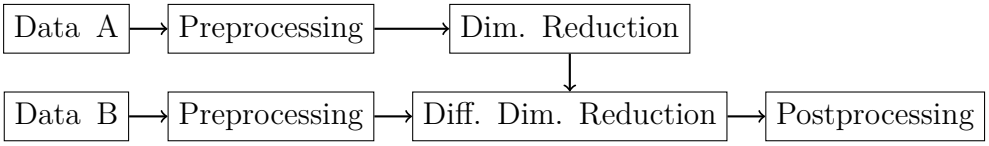


Figure 2.2: Extended Workflow with a second input data set and the additional step of Difference Dimensionality Reduction

This Extended Workflow is used by several car manufacturers and engineers around the world, but only a linear reduction method is utilised. All other examples of Dimensionality Reduction in CA of simulation results mentioned in this section utilise nonlinear approaches with very convincing findings and often explicit demonstrations of superiority over linear methods in their respective applications. This indicates that a nonlinear extension for the Extended Workflow should be investigated, as it did not exist until this thesis. Before investigating how the theoretical concept of this Extended Workflow can be expanded to nonlinear methods, the practical requirements should be revisited.

2.3 Practical Requirements

The practical application of methods for the CA must be considered when discussing or developing new approaches. These methods are utilised as tools to aid engineers in developing new designs or improving existing concepts. Hence, these tools should be tailored to this specific application and its requirements.
First of all, this means, that these methods are to be used primarily by experts in engineering and not necessarily by data scientists. Thus, the underlying methods should require only as few parameters as possible that are easy to calibrate, since they must also be applicable by users with a different field of experience than advanced data analytics.
Second, the methods must be able to handle simulation data, which is usually available in standard engineering development processes. On the one hand, this means, that several different quantities or post values are available in the simulation results, e.g. the nodal displacements or the element strains. Ideally, an analysis method should be capable of utilising all available post values. With the number of nodes or elements reaching several million in current simulation models and several computed post values for each of these elements at all states, this means, that the analysis approaches need to be able to process very high dimensional data.
On the other hand, the number of available samples is usually small compared to this high data dimension. While the number of elements can exceed several million values, the number of available simulation results ranges from only two for small investigations to a few thousand for the analysis of a complete development cycle of a car. Ideally, an analysis method should be capable of generating meaningful results for the full range of possible sample numbers. The linear DPCA approach is used for

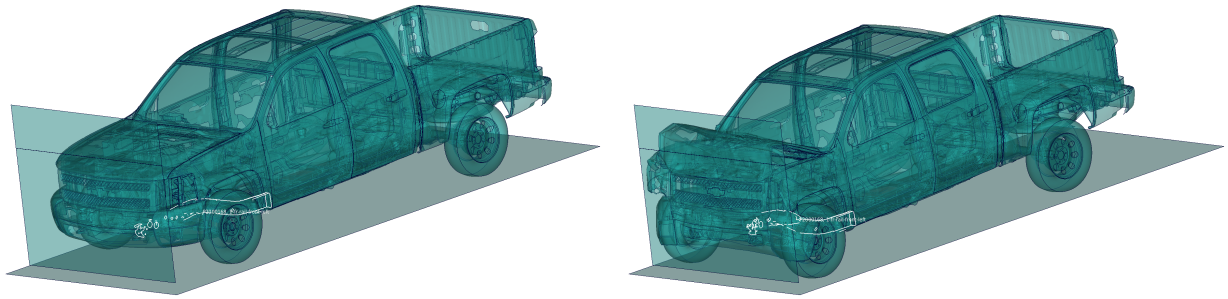
a number of years in part because it meets these requirements and it is important to keep in mind these requirements in any nonlinear modification.

The contents of this chapter can be summarised as following: The process of CA aims to find the differences and commonalities in a set of simulation results. Dimensionality Reduction has been successfully utilised in this application in the recent years. Lately, the advantages of nonlinear reduction methods over linear approaches have been shown for several applications such as spectral clustering or rule mining. For the specific case of investigating the correlation between different parts of the simulation, there is no nonlinear extension of the linear DPCA approach in literature yet. The DPCA's workflow is different from the other applications in that it does not only contain a Dimensionality Reduction step, but is extended by an additional subtraction step, which is referred to as Difference Dimensionality Reduction in this dissertation. Both steps must be adjusted if this Extended Workflow is enhanced by nonlinear methods.

3 Dimensionality Reduction

This chapter provides a detailed overview on the first step of the Extended Workflow for CA as introduced in the last chapter. The first step is the Dimensionality Reduction (DR) and its properties and capabilities for the analysis of simulation results are explained in the following sections.

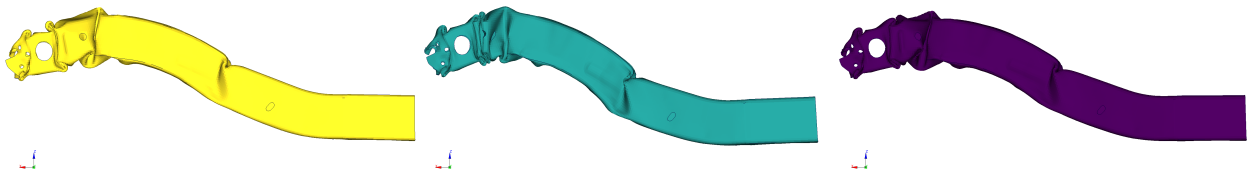
The analysis task can be described as the objective to process a certain input data set $Y \in \mathbb{R}^{D \times s}$ consisting of s samples of D-dimensional data. The quantities D and s depend on the analysis task and could for example be the number of nodal coordinates, i.e. three times the number of nodes in each simulation run and the total number of simulations runs respectively. More detailed examples will be examined later in the evaluation of Chapter 5; in this section, a simple example of three longitudinal rails is used first for illustrative purposes only. The example rails shown in Fig. 3.1 and Fig. 3.2 are extracted at the same state from different runs of the publicly available Silverado model, developed by the National Crash Analysis Center of the George Washington University [PRMB09]. The difference between the runs is a variation in certain material thicknesses, which will be covered in more detail later.



(a) Initial geometry (b) Geometry at 48 ms

Figure 3.1: Position of the left longitudinal rail PID 2000168 in the Silverado model.

In this case, the node positions for a single state are considered as input data, though any other value, e. g. plastic strain or internal energy, could be chosen as well as a different number of states. Since the rails consist of 16 030 nodes, analysing the three positions per point yields the dimensions $D = 48\,090$ and for $s = 3$ simulations.



(a) Thickness 3.11 mm (b) Thickness 2.68 mm (c) Thickness 2.91 mm

Figure 3.2: Deformed left longitudinal rail PID 2000168 of the three Silverado runs with different thicknesses at 48 ms.

The remainder of this chapter is dedicated to the explanation how this input data set is processed in DR, and this explanation is structured as following: First, some basic concepts common to the various approaches are defined. Then, these concepts are illustrated with a linear example of a Dimensionality Reduction Method (DRM), the so-called Principal Component Analysis (PCA), and the capabilities and limitations of this method are explained.
Subsequently, three classes of nonlinear methods with different approaches and capabilities are introduced. For each of these classes, several different methods are explained, which incorporate a different property in the respective class. Most of these methods have a paragraph describing the base approach and another one describing the adjustments and extensions made in this work specifically for the analysis of simulation results. The paragraphs describing the base methods simply provide an overview of previously published work and serve the purpose of providing a consistent notation while summarising the concepts needed in this work. A more detailed explanation is to be found in the references given. The only exception is the last method, which has not been published before. The different DRMs inside a class are introduced in chronological order by the date of their first publication. At the end of this section, a short recapitulation and an overview is given.
To explain the differences of the DR-classes and highlight their strengths and weaknesses, some basic terminology is required.

3.1 Basic Definitions

The following terms are needed to explain the basic concept of DR.

3.1.1 General Terminology

Starting point of a DR process is a given input data set, where each column is containing one sample.

Definition 3.1 (Original dimension, input data)
With the original dimension D, the high dimensional data set $Y \in \mathbb{R}^{D \times s}$

$$Y =: (y_1 \ \dots \ y_s) \ , \ \textit{with } y_i \in \mathbb{R}^D \ \forall \ i \in \{1, \dots, s\}$$

consisting of s samples is called the input data set.

The main assumption of DR is that the input data set Y is lying on a d-dimensional manifold, which means that it can be written as the image of a generating function f and a d-dimensional parametrisation.

Definition 3.2 (Intrinsic dimension, generating function, parameters)
With the so-called intrinsic dimension $d \in \mathbb{N}$, the function $f : \mathbb{R}^d \to \mathbb{R}^D$

$$y_i = f(x_i) \ , \ \forall \ i \in \{1, \ldots, s\}$$

is called the generating function and the values $x_i \in \mathbb{R}^d$ are called low dimensional parameters. For shorter representation, the following matrix notation F is introduced:

$$X := (x_1 \ \ldots \ x_s)$$
$$F(X) := ((f(x_1) \ \ldots \ f(x_s))) = Y$$

In practical applications, the exact generating function f and the true low dimensional parameters are usually unknown. Hence, the goal of DR is to find a low dimensional representation as an approximation for the true parameters.

Definition 3.3 (Low dimensional representation, DRM)
An approach to obtain a low dimensional representation $\widetilde{X}_d \in \mathbb{R}^{d \times s}$ with

$$\widetilde{X}_d \approx X$$

is called Dimensionality Reduction Method (DRM).

The results for low dimensional embedding of the rails introduced in the last section are displayed in Tab. 3.1. This embedding was generated using PCA and the maximum intrinsic dimension $d = 2$ for this approach with $s = 3$ samples. These values are obtained by a Singular Value Decomposition of the nodal coordinates of the three rails, though details are later explained.

Dimension	$\widetilde{x}_1$	$\widetilde{x}_2$	$\widetilde{x}_3$
1	6.71	-14.38	7.68
2	3.50	-0.15	-3.35

Table 3.1: Example of an embedding for the three rails, with $d = 2$ computed by PCA.

John A. Lee and Michel Verleysen defined “qualifications” [LV07] according to which these DRMs can be categorised and compared to other approaches. Three of their qualifications or categories are used in this dissertation.
The first qualification is that of linear and nonlinear DRMs.

Definition 3.4 (Linear or nonlinear DRM [LV07])
A DRM is categorised as linear or nonlinear if the assumed underlying generating function $f(\cdot)$ is linear or nonlinear respectively.

As foreshadowed in Chapter 1, this dissertation compares several nonlinear methods with the most popular linear approach of Principal Component Analysis.
The second qualification is that of a generative DRM.

Definition 3.5 (Generative DRM [LV07])
A DRM is classified as generative, if an approximation $\widetilde{f} : \mathbb{R}^d \to \mathbb{R}^D$ with $\widetilde{f}(\cdot) \approx f(\cdot)$ can also be obtained.

In this work, this category is relaxed in that such an approximation is only required to be obtained locally, e.g., in the vicinity of certain points and not for the entire parameter space.

Definition 3.6 (Locally generative DRM)
Given a fixed set of low dimensional points $x_j \in \mathbb{R}^d$ and their vicinities in the parameter space $B(x_j) \subset \mathbb{R}^d$, a DRM is classified as locally generative, if an approximation $\widetilde{f} : \bigcup_j B(x_j) \to \mathbb{R}^D$ with $\widetilde{f}(\cdot) \approx f(\cdot)$ can be obtained.

All approaches covered in the following sections are at least locally generative, since this property is important for the steps described in Section 3.1.3 and Chapter 4.
The third and last qualification is the one of incremental DRMs.

Definition 3.7 (Incremental DRM [LV07])
A DRM is classified as incremental or layered if the best d-dimensional embedding is an extension of the best $(d-1)$-dimensional embedding. This means $\exists\ v_d \in \mathbb{R}^s$ such that

$$\widetilde{X}_d = \begin{pmatrix} \widetilde{X}_{d-1} \\ v_d^{\mathsf{T}} \end{pmatrix} \quad \forall\ d \in \{1, \ldots, D\} \tag{3.1}$$

This is in general not the case for DRMs and raises the question on how to choose an appropriate d.

3.1.2 Importance Factors

In some DR applications the intrinsic dimension d is known before applying the actual method. Though this may be the case, for example, in signal processing [CA02], the dimension is commonly unknown beforehand in the application of crash simulations. In fact, determining or just getting a satisfactory estimate for d is one of the CA's objectives.
To solve this problem, an importance factor is calculated for each dimension of an embedding. This approach is used with linear DR in the context of the analysis of crash simulations since several years [TNNC10] and helps to estimate number of dimensions and their impact on the data: The higher an importance factor, the more

relevant the associated dimension is. If an importance factor is small compared to the others, this dimension can be neglected. This way, the intrinsic dimension can be over-estimated, then an embedding into a slightly higher dimension calculated, and then a satisfactory estimate for d deducted by discarding unimportant dimensions. Truncating an embedding in this way is only valid, if the underlying method is incremental, as explained in the last section, see Eq. (3.1).
The resulting importance factors for the low dimensional representation of Tab. 3.1 is listed in Tab. 3.2. These factors were also generated using PCA and the detail are explained in Section 3.2.3.

Dimension	Importance
1	17.63
2	4.85

Table 3.2: Importance factors for the embedding of the three rails computed by PCA, here given by the Euclidean norm of the corresponding row.

In this work, the concept of importance factors has been extended to nonlinear DRMs, although it must be tailored individually to the underlying model. Therefore, the details are explained in the respective sections. At this point, it should only be noted that they are calculated, and the explanation of how this is done will be given later.

3.1.3 Visual Representation of Effects

Each direction of the low dimensional embedding can be considered or interpreted as an underlying effect manifesting in the data. These directions are often also referred to as "modes" [TNNC10]. The coordinate of one data point for this mode then describes the magnitude or impact of this effect on a given data point.
These low dimensional coordinates can be used to get an overview of the simulation result set, for example, to identify outliers or which simulations are close to each other. For up to three dimensions this can easily be done by visualising scatter plots of these coordinates, e.g. Fig. 3.3. From this simple plot, it can be seen that the first and the third simulation are relatively similar, while the second simulation is different, which is not as easily visible from Fig. 3.2. But since only the d-dimensional embedding is known, the underlying coordinate system might be obscure and the visualisation of the underlying effects in the original high dimension D is not trivial.
In the context of DIFFCRASH so-called "virtual simulations" [BST15] are used in an attempt to visualise these effects. The term refers to files in the same format as the simulation results, which are not the result of a solving process but are artificially created for visualisation purposes.
One approach to visualise the underlying effect to which the coordinates are calculated is to generate a so-called "flipbook" [JBT17]. The data set is sorted according to their coordinate and then traversed in this order. For example, if the rails are traversed

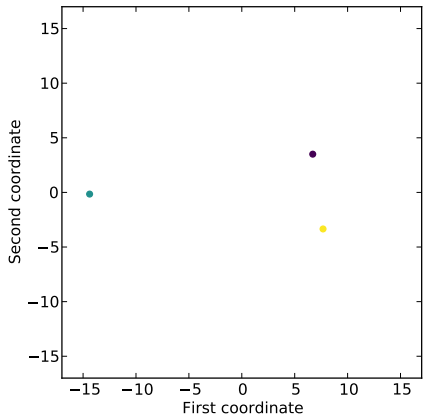


Figure 3.3: Scatter plot for the three rails of Fig. 3.2: The two coordinates of the low dimensional embedding are displayed as x- and y-axis. The points are coloured like the corresponding rails.

according to their second coordinate, the resulting flipbook is Fig. 3.2. An advantage of this approach is that it can be done for all DRMs. A disadvantage is that this becomes cumbersome for large numbers of samples. An even greater disadvantage is, that this representation mixes several effects, making it useful only for inherently one-dimensional data sets where mixing does not matter.
Another approach is the visualisation of certain points of the low dimensional data space in order to illustrate single effects.

Definition 3.8 (Evaluation points)
Given d-dimensional parameters x_i with $i \in \{1, \dots, s\}$ and a specific direction $1 \leq e \leq d$, a point $x_e^* \in \{x_e^+, x_e^-\} \subset \mathbb{R}^d$ with

$$x_e^+ := \begin{pmatrix} \mathbb{0}_{e-1} \\ \max_s x_{e,s} \\ \mathbb{0}_{d-e} \end{pmatrix} \qquad x_e^- := \begin{pmatrix} \mathbb{0}_{e-1} \\ \min_s x_{e,s} \\ \mathbb{0}_{d-e} \end{pmatrix}$$

is called evaluation point. Here, $\mathbb{0}_{e-1} \in \mathbb{R}^{e-1}$ and $\mathbb{0}_{d-e} \in \mathbb{R}^{d-e}$ are the vectors of the corresponding size, with all entries equal to 0.

An example of low dimensional coordinates with evaluation points is displayed in Fig. 3.4. If the e-th effect is to be visualised, the minimum and maximum of the low dimensional parameters belonging to this effect are determined by computing the points x_e^+ and x_e^- and afterwards, these evaluation points are projected to the original dimension:

$$y_e^+ := f\left(x_e^+\right) \qquad y_e^- := f\left(x_e^-\right)$$

A visual example of such evaluation points is given in Fig. 3.4. The evaluation points are highlighted as triangles and are the maximum or minimum coordinate of the respective axis.

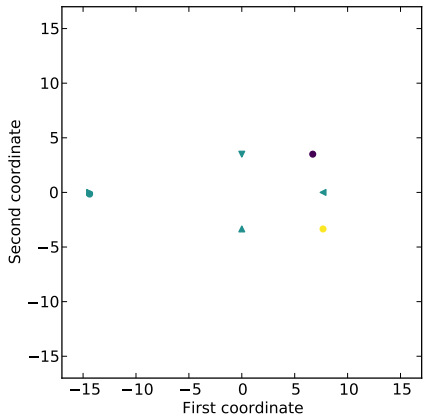
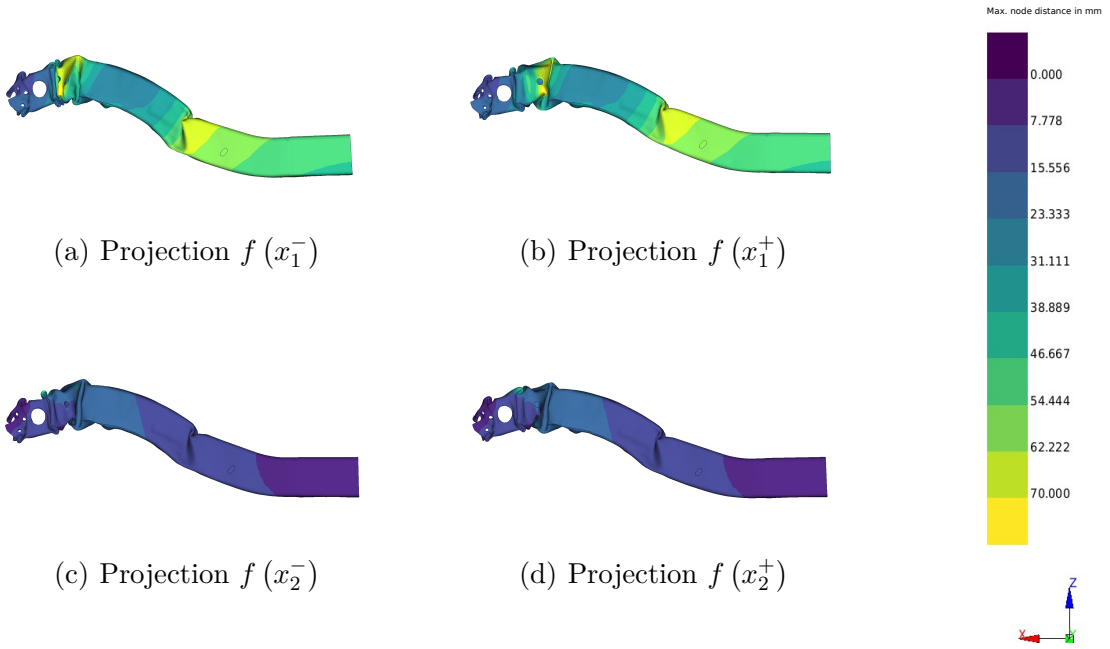


Figure 3.4: Scatter plot with evaluation points: Additional to Fig. 3.3, x_1^- and x_1^+ are the right- and left-pointing triangles, x_2^- and x_2^+ are the up- and downward-pointing ones respectively.

The first advantage of this method is that such a representation is possible even for large data sets containing many samples. The second advantage of this representation is that the effects can be observed relatively isolated. A disadvantage is that the evaluation points rarely coincide with real data points, i.e. the low dimensional representation of the given samples. This means first, that it is only applicable to generative DRMs because only they can generate high dimensional representations of new low dimensional data points, and second, that these high dimensional representations are only approximated.
In Fig. 3.5 such virtual simulation results for the rails are visualised. Virtual is referring to the fact that these results were not obtained by a simulation code, but by application of the generating function f. The upper row shows the visualisations of the first effect and the difference is clearly visible in two areas. In the left part of the rails, where the effect shows the amount of buckling in the simulation, and in the middle area, where the folding varies. The lower row shows the second effect, which is more subtle. In one case, the buckling tends to lean to the left, in the other to the right.



(a) Projection $f\left(x_1^-\right)$ (b) Projection $f\left(x_1^+\right)$

(c) Projection $f\left(x_2^-\right)$ (d) Projection $f\left(x_2^+\right)$

Figure 3.5: Example of virtual simulation results for the evaluation points in Fig. 3.4: The fringe shows the maximum difference in mm between the same node in the left and the right subfigures.

With these visual representations of the determined low dimensional effects, the DR can help engineers in analysing Crash simulations.

3.1.4 Assumptions

In this work, there are a few assumptions made to the input data of the DR, which are needed by or at least helpful for several methods.
The first assumption is, that the input is assumed to be centralised, which means that for $\mathbb{1}_s \in \mathbb{R}^s$ the vector with all entries equal to 1 and $\mathbb{0}_D \in \mathbb{R}^D$ the vector with all entries equal to 0 it holds that

$$Y\mathbb{1}_s = \mathbb{0}_D \tag{3.2}$$

If the data would not be centralised, a centralised version $\overline{Y}$ could be easily computed by a simple matrix multiplication.

Definition 3.9 (Centring matrix)
With I_s being $\in \mathbb{R}^{s\times s}$ the identity matrix, the matrix $Z_s \in \mathbb{R}^{s\times s}$

$$Z_s := I_s - \frac{1}{s}\mathbb{1}_s\mathbb{1}_s{}^\mathsf{T} \tag{3.3}$$

is called the centring matrix.

The centralised data set $\overline{Y} \in \mathbb{R}^{D \times s}$ can then be obtained by:

$$\overline{Y} := Y Z_s \tag{3.4}$$

For easier notation the overline is skipped in the following by making this assumption. Secondly, all data sets are assumed to be non-trivial $Y \neq \mathbb{0}_{D \times s} \in \mathbb{R}^{D \times s}$ meaning that at least one entry is non-zero. From an analysis point of view, the data set consisting of only zeros would be of little interest for any further investigation and is hence not considered here.
A third restriction is that the data is assumed to be given in the same appropriate metric units and that no further normalisation is needed. For example, if the data is comprised of node coordinates, all values are expected to be for example in mm and not in mixed units. If one coordinate is given for instance in mm and one in km, the values should be converted prior to the analysis to meet this assumption. There are some applications [LV07] where the data should be standardised by dividing each row by its absolute maximum, so that the values are unitless $-1 \leq y_{ij} \leq 1 \ \ \forall i, j$, but in this thesis the data is not normalised beyond unit alignment or global scaling of the complete matrix.
The fourth restriction is that the sampled data is assumed to lie on a single connected manifold of fixed dimension d. In many practical applications, the intrinsic dimension of the data may change through the given data set or the single points are separable into disjunct clusters. In both cases, the sampled data would be located on two or even more separate manifolds. If such a separation were found, this would be an important first conclusion from the CA point of view, and the analyst could proceed by investigating each separable subset independently, thus satisfying the aforementioned assumption. This last condition is later revisited in Section 5.1.5.2 but should hold during explanations of the different approaches.

3.2 Principal Component Analysis

The general concepts mentioned in the last paragraphs are substantiated in this section by introducing a first concrete approach.

3.2.1 Base Method

The so-called Principal Component Analysis (PCA) is a linear DRM and was introduced in 1933 by Harold Hotelling [Hot33], though the concept was already known in 1901 [Pea01]. This approach is well established in various fields [WRR03] and it achieves the reduction by first calculating a Singular Value Decomposition (SVD) of the input data and then truncating said decomposition [LV07]. An SVD is a representation of the form

$$Y = U \Sigma V^{\mathsf{T}} \tag{3.5}$$

which exists for each Matrix [LM12] and where with $r := \operatorname{rank}(Y)$ the following holds:

$$\begin{aligned}\Sigma &= \begin{pmatrix} \Sigma_r & \mathbb{0}_{r\times s-r} \\ \mathbb{0}_{D-r\times r} & \mathbb{0}_{D-r\times s-r} \end{pmatrix} \in \mathbb{R}^{D\times s} \\ \Sigma_r &= \operatorname{diag}(\sigma_1, ..., \sigma_r) \in \mathbb{R}^{r\times r} \\ U &\in \mathbb{R}^{D\times D} \text{ orthogonal} \\ V &\in \mathbb{R}^{s\times s} \text{ orthogonal}\end{aligned}$$

Note that $r < s$ because the data is centralised see Eq. (3.2). Without loss of generality, it is assumed that the singular values σ_i of Σ are sorted in descending order, meaning that $\sigma_1 \geq ... \geq \sigma_r$ holds. To perform a PCA a suitable $d \in \{1, \ldots, r\}$ is then determined so that the truncated or cut decomposition, which consists of the first d columns of U and the corresponding rows of V^T, provides a good approximation.

$$Y \approx Y_d := U|_d \Sigma_r|_d (V|_d)^\mathsf{T} \tag{3.6}$$

An important property of Y_d is that it is the optimal rank-d approximation of the data set Y according to the Euclidean norm $\|\cdot\|_2$, see [LM12]. If the intrinsic dimension is known and dependency is linear, the user may specify d for the SVD cut-off directly. A possibility to choose an approximation $\widetilde{d} \approx d$ for an unknown dimension is to determine the point, where the singular values σ_i are small compared to the largest one σ_1. For a given $0 < \theta < 1$, $\widetilde{d}$ is chosen as

$$\begin{aligned}\widetilde{d} &:= \arg\min_i \sigma_i \\ \text{subject to: } &\sigma_i > \theta\sigma_1\end{aligned}$$

though choosing the right θ depends on the use case and the underlying data, which can be difficult for some applications. Thus, another possibility to choose $\widetilde{d}$ for an unknown intrinsic dimension d will be discussed in Section 3.2.2.
In terms of a DRM the PCA approach yields the following approximations for the low dimensional coordinates and the generating function:

$$\widetilde{X} := \Sigma|_d V|_d^\mathsf{T} \tag{3.7}$$

$$\widetilde{F}(X) := U|_d \, X \tag{3.8}$$

Since the function $\widetilde{F}(\cdot)$ can be stated explicitly, the method is generative and from the function itself it is obvious, that the method is linear. With the explicit function, visualising the evaluation points $x_e^* \in \{x_e^+, x_e^-\}$ can be trivially done by applying the matrix to vector operation:

$$\begin{aligned}F(x_e^*) &\approx \widetilde{F}(x_e^*) \\ &= U|_d x_e^*\end{aligned}$$

PCA is closely related to Multidimensional Scaling as is shown later in Section 3.4.2.

3.2.2 Stochastic Interpretation

The PCA is also strongly connected to the stochastic theory, which is explained to motivate the further steps. The input data Y can be interpreted as s samples of a D-dimensional random variable. With $\mathrm{E}\{Y\}$ being the expectation of random variable Y the Covariance matrix for random variable Y [FHT15] is defined as

$$\begin{aligned} C_Y :&= \operatorname{cov}(Y) \\ &= \mathrm{E}\left\{(Y - \mathrm{E}\{Y\})(Y - \mathrm{E}\{Y\})^\mathsf{T}\right\} \\ &\overset{3.2}{=} \mathrm{E}\left\{YY^\mathsf{T}\right\} \\ &= \frac{1}{s} YY^\mathsf{T} \\ &\overset{3.5}{=} \frac{1}{s} U\Sigma^2 U^\mathsf{T} \in \mathbb{R}^{D\times D} \end{aligned}$$

This first of all provides means to compute SVD by calculating the Covariance matrix and then an Eigenvalue Decomposition (EVD) of that matrix. This is especially useful since covariance matrices are always symmetric and positive semi-definite [FHT15] and hence easy to decompose. Afterwards, the product of the eigenvectors with the original data set are computed to determine the missing matrix V:

$$\begin{aligned} U^\mathsf{T} Y &= U^\mathsf{T} U \Sigma V^\mathsf{T} \\ &= \Sigma V^\mathsf{T} \end{aligned}$$

Second, this gives meaning to the order of the components and another solution to find an appropriate approximation $\widetilde{d}$, if the true intrinsic dimension d is unknown: The diagonal matrix Σ contains the complete information about the variance in descending order. This means that the first dimension best explains the variance in the data. The next is the dimension that explains the variance second best, and so on.
The total variance can be calculated as the sum of the entries of Σ^2 and for a given value $0 < \kappa < 1$ a suitable d can be found by a variance cut as:

$$\begin{aligned} g(l) &:= \frac{\sum_{i<l} \sigma_i^2}{\sum_j \sigma_j^2} \\ \widetilde{d} &:= \arg\min_l g(l) \\ \text{subject to:} \quad & g(l) > \kappa \end{aligned} \tag{3.9}$$

In this approach, κ is the fraction of the total variance that should be preserved in the low dimensional embedding.
For practical applications, the EVD can also be determined from the Gram matrix

for random variable Y,

$$\begin{aligned} G_Y :&= \operatorname{cov}(Y^\mathsf{T}) \\ &= \frac{1}{D} Y^\mathsf{T} Y \\ &= \frac{1}{D} V \Sigma^2 V^\mathsf{T} \in \mathbb{R}^{s \times s} \end{aligned}$$

since it is containing the same Σ and thus the same information about the variance. The decomposition is usually done depending on which dimension D or s is smaller. However, this is only advisable if only the largest eigenvalues are of interest, since it can be subject to numerical instabilities [LV07].

3.2.3 Application in the Analysis of Simulation Results

With the base concept of the PCA method introduced, the steps of the CA in Section 3.1 can now be explained for this method.
In the application of analysing simulation results, the input data set consists of post values extracted from a subset of elements at a certain number of states of the simulation results. One example could be all coordinates for the nodes belonging to a selected PID at one or more states, another all internal energy values for the elements in a certain region over all states. This subset of elements will be referred to as a part, though it could be a fragment or a union of several PIDs as defined in a simulation. The part data is extracted from all simulation results, if the requested elements are not present in one of the results, e.g. due to changing geometry, a best possible approximation is chosen, for example by mapping the changed geometry to a reference presentation.
Since these subsets can vary in size, the input data is scaled by the square root of the number of rows D to make different analysis results comparable. This is motivated by the computation of the Gram matrix:

$$\begin{aligned} G_Y &= \frac{1}{D} Y^\mathsf{T} Y \\ &= \left(\frac{1}{\sqrt{D}} Y \right)^\mathsf{T} \left(\frac{1}{\sqrt{D}} Y \right) \end{aligned}$$

Furthermore, in this application, the Gram matrix $G_Y \in \mathbb{R}^{s \times s}$ is usually smaller, than the covariance matrix $C_Y \in \mathbb{R}^{D \times D}$, since a part might contain several thousand elements multiplied by the number of states, but the set of simulation results rarely exceeds a few hundred samples, so it is more effective to do the EVD of this matrix, since it is smaller and directly provides the right singular vectors.
As stated before, the low dimensional coordinates are the entries of these right singular vectors multiplied by the singular values, see Eq. (3.7). This means that each coordinate direction $1 < i < d$, which is stored in the i-th row of $\widetilde{X}$, has a Euclidean

norm of the corresponding singular value σ_i, since $V|_d^\mathsf{T}$ is a truncated orthogonal matrix. Said singular value is assigned as the importance factor for this dimension, see Section 3.1.2, since this is the amount of variance explained by this component in the low dimensional representation.

It is important to note, that the normalised low dimensional coordinates $V|_d$ provide a basis for the effects as well as the original principal components $U|_d$. Since V is an orthogonal matrix, the columns of this matrix form an orthonormal basis of the $\mathbb{R}^s$ space by definition. This means, that each vector in this space can be defined as a linear combination of these base vectors, which is especially interesting for the unit vectors $e_i \in \mathbb{R}^s$, corresponding to the axes of the canonical base. The coefficients which reconstruct a vector as a linear combination of an orthonormal basis can be computed by calculating the scalar products of that vector with the base vectors [LM12]. In the case of the unit vector e_i and the base of V, the products yield the corresponding row and these scalar products do not change for the first d unit vectors, if the data is restricted to the first d entries of V, since all missing values would be multiplied with zeros anyway.
This can be interpreted as categorising an effect not by the affected elements, e.g. buckling in front of the rail, but in which samples it is occurring, e.g. the mean deformation seen in simulations two and three. Here, an effect is identified by a linear combination of the sample points and it can be visualised by applying the interpolation with the weights $v_i^\mathsf{T} \in \mathbb{R}^d$ of this linear combination, which can be verified by examining the product:

$$\begin{aligned} Y\, v_i &= U\Sigma V^\mathsf{T} v_i \\ &= U\Sigma e_i \\ &= \sigma_i u_i \end{aligned}$$

Normalising the result will eliminate the remaining σ_i and give the base vector. An advantage of this representation is that it is usually smaller than the corresponding base vector $u_i \in \mathbb{R}^D$.
Another benefit of interpreting effects as a linear combination of samples is that it can be evaluated on all parts, states, and post values for a fixed sample of simulation results. This means that even though the PCA was only conducted for a single part, e.g. the coordinates of one longitudinal rail at a certain state, the underlying effect can be extended to all post values on the entire car at all states by interpolating these values accordingly. This property is important for the process described in Section 4.1.

3.2.4 Assessment

The linear approach is straight forward and easy to use, since it does not require any parameters other than the desired target dimension d. The SVD exists for all matrices and can always be computed, so the truncation will always result in an embedding

with the desired low dimension. The connection to the stochastic theory provides a solid foundation and interpretable results as well as meaningful importance factors. But the simple model also has some undesired properties. As a perfectly linear dependency between the high dimensional coordinates and intrinsic ones is assumed, the method cannot yield the correct result, if the dependency is nonlinear, i.e. the data was generated by a nonlinear function F. This typically manifests in the linear method overestimating the intrinsic dimension and thus the number of underlying effects. Furthermore, the minimisation of the Euclidean norm can sometimes result in finding an axis, which is mixing several underlying effects. Additionally, the image of the generating function $F(\cdot)$ is always the complete subspace, spanned by the base vectors U. This may overestimate the manifold drastically, especially in the case of a nonlinear data set, where the manifold could just be a subset of this subspace.

To overcome these drawbacks nonlinear methods can be utilised. There is a vast range of nonlinear methods available in literature and it is a priori unknown, which approach will yield the best results for the analysis of simulation results. In fact, the best method may depend on the use case or even the individual data set.
At the same time, a comprehensive evaluation or even introduction of all methods would exceed the scope of this thesis. A reasonable overview of different methods can be found in [LV07], though many new approaches have been invented since this publication. Nonetheless, John A. Lee and Michel Verleysen started in this publication to categorise and contrast different approaches, increasing the comparability of the diverse methods and showing, that most algorithms belong to certain classes. Three of the most common classes with some example methods are visited in the following sections: The so-called Local Methods, the Multidimensional Scaling, and the Nonlinear Mapping approaches.

3.3 Local Methods

In this section the first of the investigated classes of nonlinear DRMs is introduced as an alternative to the linear method described in the last section.

3.3.1 Commonalities

Approaches which are categorised as a Local Method (LM) [ST02] determine local neighbourhoods and specific properties about these neighbourhoods. Then a global low dimensional embedding is calculated, which best preserves all these local properties.

There are several rules to define the neighbourhood $\mathcal{N}(y_i) \subset \{1, \ldots, s\}$ of a point y_i in a given data set [LV07]. Two of the most popular choices are the ε- and the k-rule. In the former, the j-th point y_j is considered to be in the neighbourhood $\mathcal{N}(y_i)$, if it is within a ball $B(\varepsilon, y_i) \subset \mathbb{R}^D$ with given radius $\varepsilon \in \mathbb{R}$ and centre y_i. In the latter, the j-th point y_j is considered to be in the neighbourhood $\mathcal{N}(y_i)$ of point y_i, if it is

amongst its k-nearest Neighbours (kNN) for a given k. The different LM-approaches may use varying rules, but all construct the neighbourhood $\mathcal{N}(y_i)$ for all points and usually it holds that $i \notin \mathcal{N}(y_i)$.
However the neighbourhoods are constructed, afterwards a property is determined for each of these $\mathcal{N}(y_i)$. Which specific property is used, depends on the individual method, but all methods aggregate the local properties in one global alignment matrix $\Phi_{\mathrm{LM}} \in \mathbb{R}^{s \times s}$. How the data is aggregated is also different for the various approaches and hence explained in detail in the specific subsections.
Still, this matrix is used for all LM-approaches to solve the problem of determining the low dimensional coordinates, which best preserve all local properties: The matrix is aggregated in such a way, that for the ideal coordinates X it holds that:

$$\Phi_{\mathrm{LM}} X^{\mathsf{T}} \stackrel{!}{=} \mathbb{0}_{s \times d}$$

To make the problem well posed and to prevent trivial solutions, two constraints are applied [SR00]. First, the resulting low dimensional coordinates should have zero mean. Second, the new coordinates should have unit covariance.

$$X \mathbb{1}_s = \mathbb{0}_d \tag{3.10}$$

$$X X^{\mathsf{T}} = I_d \tag{3.11}$$

With these two constraints in place, the problem can be solved in a least squares sense by computing an EVD [SR00].

$$\Phi_{\mathrm{LM}} =: V \Lambda V^{\mathsf{T}}$$

The solution are the eigenvectors corresponding to the second to $d+1$ smallest eigenvalues of

$$\widetilde{X} := (v_{s-d-1} \ \ldots \ v_{s-1})^{\mathsf{T}}$$

since these are the global vectors, which best comply to all local properties. The eigenvector corresponding to the lowest eigenvalue is discarded since it is assumed to be the vector $\mathbb{1}_s$, which is an eigenvector to the eigenvalue 0 because of the zero mean constraint Eq. (3.10). An alternative to the case, where this is assumption is not met is listed in Section A.1 of the appendix, but since it was valid for all investigated examples, it is not discussed in detail. This way, LMs construct global low dimensional embeddings, which best preserve all previously determined local properties.

3.3.2 Locally Linear Embedding

The concept of LM can be explained in more detail with an example. The method described in this subsection was the first of this class and one of the earliest nonlinear DRMs in this work. It is the first approach that tries to fit a local description patch and align everything in one global context.

3.3.2.1 Base Method

The Locally Linear Embedding (LLE) was first published in 2000 by Sam T. Roweis and Lawrence K. Saul [RS00]. The idea is to reconstruct each point from its neighbours as best as possible and thus define the data in terms of local neighbourhood weights. Afterwards a global low dimensional embedding is calculated, which best preserves these local weights.
An example for preserved local weights is visualised in Fig. 3.6. The intersection of the dashed and the dotted lines in two dimensions is the best approximation of the middle point by its two nearest neighbours. The ratio of the two parts of the dashed line corresponds to the ratio of the weights for the best reconstruction. Said ratio is preserved in the one dimensional representation.

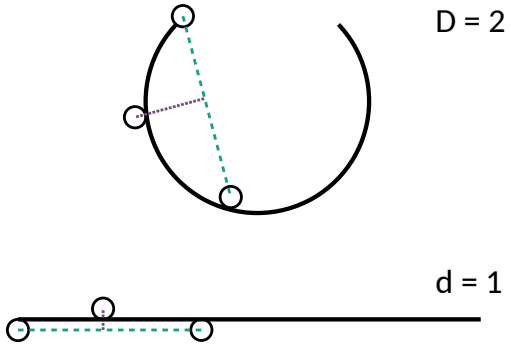


Figure 3.6: Graphical example of preserving the weights in a low dimensional embedding. The continuous line represents the manifold with three sample points highlighted in the high dimension of $D = 2$ as well as in the low dimension $d = 1$. This schematic was motivated by [LV07].

In LLE the user may specify any rule to construct the neighbourhoods. Once these neighbourhoods are constructed, the next step of LLE is to compute the weights w_{ij}, which best reconstruct a point y_i from its neighbours. This can be formulated as the following optimisation problem for $W \in \mathbb{R}^{s \times s}$:

$$\begin{aligned} \min_{W} \quad & \varepsilon(W) := \sum_{i=1}^{s} \left\| y_i - \sum_{j=1}^{s} w_{ij} y_j \right\|_2^2 \\ \text{subject to:} \quad & w_{ij} = 0 \quad \forall j \notin \mathcal{N}(y_i) \\ & \sum_{j=1}^{s} w_{ij} = 1 \quad \forall i \in 1, ..., s \end{aligned} \tag{3.12}$$

This problem can be solved in closed form for each point separately as stated in [SR00].

Since the weights sum to one, the problem for one point y_i can be reformulated as:

$$\begin{aligned}\varepsilon_i &= \left\| y_i - \sum_{j \in \mathcal{N}(y_i)} w_{ij} y_j \right\|_2^2 \\ &= \left\| \sum_{j \in \mathcal{N}(y_i)} w_{ij} \left(y_i - y_j \right) \right\|_2^2 \\ &= \sum_{j \in \mathcal{N}(y_i)} \sum_{k \in \mathcal{N}(y_i)} w_{ij} w_{ik} G_{ijk}\end{aligned}$$

The last term is an entry of the local Gram matrix for the neighbours of point i, centred onto said point:

$$G_{ijk} := \left(y_i - y_j \right)^\mathsf{T} \left(y_i - y_k \right)$$

With this matrix, the problem of Eq. (3.12) becomes

$$\sum_{j \in \mathcal{N}(y_i)} G_{ijk} w_{ik} = 0 \quad , \forall k \in \mathcal{N}(y_i) \tag{3.13}$$

$$\text{subject to: } \sum_{k \in \mathcal{N}(y_i)} w_{ik} = 1 \tag{3.14}$$

In [SR00] the authors propose to utilise this formulation by first solving Eq. (3.13) without the unit sum constraint and normalising the weights afterwards.
As they state, this poses a problem, if the local Gram matrix is singular or close to singular, which may be the case, for example, if the number of neighbours is larger than the intrinsic dimension or the data is not well sampled locally. In these cases, the authors advice to regularise the equation by adding a small multiple of the identity matrix prior to solving, with δ_{jk} being the Kronecker delta and ϱ_i a regularisation parameter chosen by the user.

$$G_{ijk} \leftarrow G_{ijk} + \varrho_i \delta_{jk} \tag{3.15}$$

After solving the optimisation problem for all points, in the next step of LLE the calculated weights are fixed and the objective is to calculate low dimensional coordinates, which best preserve these weights. This can be done by minimising the following cost function:

$$\begin{aligned}\min_X \quad & E\left(X\right) = \sum_{i=1}^{s} \left\| x_i - \sum_{j=1}^{s} w_{ij} x_j \right\|_2^2 \\ \text{subject to:} \quad & X \mathbb{1}_s = \mathbb{0}_d \\ & X X^\mathsf{T} = I_d\end{aligned} \tag{3.16}$$

As stated before, this is achieved by an EVD of a global alignment matrix. Here, the specific alignment matrix Φ_{LLE} is aggregated as

$$\begin{aligned}\Phi_{\text{LLE}} &:= \left(I_s - W \right)^\mathsf{T} \left(I_s - W \right) \\ &=: V \Lambda V^\mathsf{T}\end{aligned}$$

As for all LMs, the solution for the DR-problem are the eigenvectors corresponding to the second to $d+1$ smallest eigenvalues of

$$\widetilde{X} := (v_{s-d-1} \ \dots \ v_{s-1})^{\mathsf{T}}$$

since these are the global vectors, which best comply to all local properties. The vector corresponding to the smallest eigenvalue is discarded, since it is assumed to be $\mathbb{1}_s$, which has an eigenvalue of 0 by construction, which is enforced by constraint, that the weights sum to one, see Eq. (3.13).

1: Construct the neighbourhoods $\mathcal{N}(y_i)$ of each data point y_i
2: Determine the weights that best reconstruct each data point from its neighbours, minimising the cost in Eq. (3.12) by constrained linear fits.
3: Compute the vectors best reconstructed by the weights, minimising the quadratic form in Eq. (3.16) by the bottom non-zero eigenvectors of Φ_{LLE}.

Algorithm 3.1: LLE Algorithm. Modified from [SR00].

LLE can also be performed starting from pairwise Euclidean distances. At least for the two given ε- and the k-rules, the building of the neighbourhoods can be trivially done from distances as well as from values. In [SR00] the authors showed that the local Gram matrix can be assembled from the distances and so the weights can be derived from these values. The further steps do not change afterwards, but the computation of the pairwise distances can be more efficient than aggregating the full data matrix. Though the straightforward approach of the LLE makes it quite easy to use, it has some known drawbacks. The first one being the calibration of the regularisation parameter. Choosing a suitable ϱ_i can be challenging in some applications [KDM10], since it depends on the magnitude of the values, their distance to each other and the noise disturbance.
A second drawback was shown in [ZW07], that even if the weights are determined correctly, with or without regularisation, the embedding can sometimes still yield wrong embeddings. This is partially because the simple description by a single set of local weights can sometimes fail to capture the global structure of the data. An additional drawback can be, that the weights, which best reconstruct a point from its neighbours are preserved, but the information, whether or not this was a good approximation, is not carried on to the low dimensional embedding directly, see Fig. 3.7. In this figure, the intersection of the dashed and the dotted lines is the best approximation of the triangle and the square by the two circles. The triangle can be approximated perfectly, but the square rather poorly, though the optimal weights to interpolate them from the circles are the same. The problem is that this is sometimes beneficial to some extent, but in other cases undesirable.

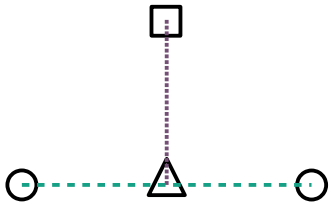


Figure 3.7: Graphical example of similar weights, but different quality., when approximating the square and the triangle with the circles.

Although these known issues exist and affect the performance of the method, some fundamental concepts of the CA can be demonstrated very well with this first approach.

3.3.2.2 Extension for Comparative Analysis

The base approach of LLE has a few options, to influence the behaviour of the method and needed to be addressed, if it is used in an automated analysis of simulation results.
A first option is the construction of the local neighbourhoods. The k-rule is often used in literature for simplicity reasons and because it is easier to calibrate for users than for example the ε-rule, especially for data sets with varying sample density: A ball with fixed radius ε could enclose too many points in one region and too few in another, while the kNN can handle these different densities. Such varying sample density can occur in the analysis of simulation results, as can be seen, for example, later in Section 5.2.2. Furthermore, the value of k can often be calibrated automatically which is crucial for an automated usage. Several methods are proposed in [LV07], though the user can always select a specific value. A good estimate for k and an expected intrinsic dimension of d is usually $k = 4d + 2$. Thus, in this work a slightly modified approach close to the k-rule was used. An undirected graph [BM$^+$76] was constructed in two steps: First, the kNN for each point were determined and undirected edges between these points inserted into the graph. Second, if the resulting graph was disconnected, a warning was issued and the shortest edges between connected components were inserted until the whole graph was connected. This is closely related to the assumption of the data lying on a single connected manifold: If the graph is not connected, this should be reported to the analyst, who must decide whether to analyse each connected component separately or to proceed with the single graph with fixed connectivity. The 1-ring neighbours in this amended graph were considered as the neighbourhood for LLE. Constructing an undirected graph imbues some degree of symmetry into the method, such that if $j \in \mathcal{N}(y_i)$ than also $i \in \mathcal{N}(y_j)$. Furthermore, enforcing a connected graph ensures that the unit covariance constraint Eq. (3.11) does not only affect a subset of the data. This constraint is needed to make the problem well posed but cannot prevent degenerate solutions in the case in a disconnected graph.

The second option to influence the behaviour of the method is the regularisation introduced in Eq. (3.15). Several publications with different choices for the regularisation parameter ϱ_i are available, some of which are [SR00], [SR03] and [DSAC10]. The importance of this parameter and its huge impact on the outcome of the method is discussed, amongst others, in a dedicated paper in [KDM10]. In this thesis, a new slightly modified version of the approach described in [SR03] was used:

$$\varrho_{i,\mathrm{SR03}} = \frac{\nu^2}{|\mathcal{N}(y_i)|} \operatorname{tr}(G_{ijk}) \tag{3.17}$$

$$\varrho_{i,\mathrm{NEW}} = \omega \, |\mathcal{N}(y_i)| \operatorname{tr}(G_{ijk}) \tag{3.18}$$

Here, $\operatorname{tr}(\cdot)$ denotes the trace [LM12] of a matrix, $|\mathcal{N}(y_i)|$ the number of neighbours in the respective neighbourhood and $0 < \nu, \omega \ll 1$ are hyper parameters which can be calibrated for the specific application. Though there are more advanced approaches available, this one has the benefit of being relatively easy to calibrate: The trace and number of neighbours already depend on the local neighbourhood, so the hyper parameter ω can be chosen globally, while the regularisation adapts for each point. In contrast, choosing one single appropriate $\varrho = \varrho_i \ \forall \ i$ directly is not feasible in practical applications. All examples shown in this work were computed with the regularisation in Eq. (3.18) and $\omega = 10^{-4}$ for simplicity reasons and because this value did perform reasonably well on artificial examples. For real applications, the correct choice of regularisation type and hyper parameters is still a challenge. The topic of regularisation is re-visited again in Section 3.3.4, where a superior alternative to this regularisation approach is introduced, so it will not be discussed further here.

Beyond the existing options of the base method, some further extensions are needed for the successful application in the analysis of simulation results. The importance factors are a critical component in the CA, as described in Section 3.1.2, but the base version of LLE has no equivalent to the importance factors of the linear PCA approach. In fact, the unit covariance constraint Eq. (3.11) enforces all coordinates to have a similar range, resulting in seemingly equally important dimensions. Therefore, a new importance measure had to be derived. The base method of LLE was extended by an additional step to compute such a measure and sort the computed coordinates accordingly. Both methods, the PCA and LLE, rely on computing an EVD to obtain the low dimensional coordinates, with one major difference: In contrast to PCA, where the eigenvectors associated with the largest eigenvalues are of interest since they are used as importance factors, in LLE the ones for the lowest eigenvalues are most important. The intuitive idea of utilising the eigenvalues Λ also for the computation of importance factors is discouraged, because their magnitude and relative difference may be related to the intrinsic dimension in some cases, but is generally independent of it, as demonstrated in [SR03].
Instead, a new approach depending on local differences has been developed. The goal is to make the embedding locally distance-preserving by scaling the individual directions of the low dimensional coordinates. The goal of LLE is to compute an

embedding, which best preserves the local weights in a least squares sense. This means, the residual $r_i \in \mathbb{R}^d$ of the nearest neighbour reconstruction is supposed to be small, if the LLE succeeded in computing a good embedding.

$$r_i := x_i - \sum_{j=1}^{s} w_{ij} x_j = x_i - \sum_{j \in \mathcal{N}(y_i)} w_{ij} x_j$$

Thus, the following relation holds $\forall\ 1 \leq i \leq s$ and any $\alpha_1, ..., \alpha_d \in \mathbb{R}$:

$$\begin{aligned} x_i &= r_i + \sum_{j \in \mathcal{N}(y_i)} w_{ij} x_j \\ \Rightarrow \operatorname{diag}(\alpha_1, ..., \alpha_d)\, x_i &= \operatorname{diag}(\alpha_1, ..., \alpha_d)\, r_j + \sum_j w_{ij} \operatorname{diag}(\alpha_1, ..., \alpha_d)\, x_j \end{aligned} \tag{3.19}$$

Therefore, the scaling in Eq. (3.19) will not affect the preservation of weights in this case. However, if the method produces large errors, these errors can be amplified by the scaling. The aim is to determine $\alpha_1, ..., \alpha_d$ such that:

$$\|\operatorname{diag}(\alpha_1, ..., \alpha_d)\,(x_i - x_j)\|_2 = \|y_i - y_j\|_2 \quad \forall\ 1 \leq i \leq s,\ j \in \mathcal{N}(y_i) \tag{3.20}$$

Squaring these $m := \sum_i |\mathcal{N}(y_i)|$ terms yields the following system of linear equations:

$$\begin{pmatrix} (x_{1,1} - x_{\gamma,1})^2 & \dots & (x_{1,d} - x_{\gamma,d})^2 \\ \vdots & \dots & \vdots \\ (x_{m,1} - x_{\phi,1})^2 & \dots & (x_{m,d} - x_{\phi,d})^2 \end{pmatrix} \begin{pmatrix} \alpha_1^2 \\ \vdots \\ \alpha_d^2 \end{pmatrix} = \begin{pmatrix} \|y_1 - y_\gamma\|_2^2 \\ \vdots \\ \|y_m - y_\phi\|_2^2 \end{pmatrix}$$

$$A \begin{pmatrix} \alpha_1^2 \\ \vdots \\ \alpha_d^2 \end{pmatrix} = b$$

with $A \in \mathbb{R}^{m \times d},\ b \in \mathbb{R}^m$. Using the k-rule, the number of rows m is equal to ks, in the case of the modified k-rule it is usually slightly greater. This system is solved by substituting and constraining the variables:

$$A \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_d \end{pmatrix} = b \text{ s.t. } \mu_i \geq 0 \quad \forall\ 1 \leq i \leq d \tag{3.21}$$

In a last step, the values are sorted, such that $\mu_{\pi(1)} \geq ... \geq \mu_{\pi(d)}$ and the same permutation π is applied to the result $\widetilde{X}$ of the base method. This yields the scaled embedding

$$\widehat{X} := \operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)^{\frac{1}{2}} \begin{pmatrix} \widetilde{x}_{\pi(1),1} & \dots & \widetilde{x}_{\pi(1),s} \\ \vdots & \dots & \vdots \\ \widetilde{x}_{\pi(d),1} & \dots & \widetilde{x}_{\pi(d),s} \end{pmatrix}$$

where the entries of the diagonal matrix are the new importance factors correctly in descending order.
The sorting and the resulting permutation could be skipped, by additionally enforcing $\mu_1 \geq ... \geq \mu_d \geq 0$ directly in the system of Eq. (3.21), but the following reasons discourage this approach:
One reason is that the sorting approach can be used to mitigate instabilities in the solving process of the EVD. The order of these eigenvalues can be shuffled due to their magnitude being very close to each other and to zero. Furthermore, [SR03] has shown, that there can be more than $d+1$ small eigenvalues which can also be important. The sorting approach could first overestimate d, then compute the importance factors and discard unimportant dimensions. This way, potential shuffling can be corrected afterwards possibly improving the performance of the DRM.
Furthermore, the usage of these scaling factors as importance factors is justified because they share an important property with the linear approach in that they are proportional to the amount of variance, they are inducing in the low dimensional embedding. This can be seen from the low dimensional Covariance matrix of the scaled coordinates:

$$\begin{aligned}\widehat{X}\widehat{X}^{\mathsf{T}} &= \left(\operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)^{\frac{1}{2}} \widetilde{X}\right)\left(\operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)^{\frac{1}{2}} \widetilde{X}\right)^{\mathsf{T}} \\ &= \operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)^{\frac{1}{2}} \widetilde{X}\widetilde{X}^{\mathsf{T}} \operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)^{\frac{1}{2}} \\ &= \operatorname{diag}\left(\mu_{\pi(1)}, ..., \mu_{\pi(d)}\right)\end{aligned}$$

Since $\widetilde{X}$ is constructed using eigenvectors of a symmetric matrix, it has a unit covariance, see Eq. (3.11), which is scaled according to the importance factors.

Finally, a last extension is needed for the visualisation of effects. As introduced in Section 3.1.3 this visualisation is based on the projection of evaluations points. The determination of the evaluation points $x_e^* \in \{x_e^+, x_e^-\}$ does not change since they are defined in the low dimensional representation. An extension is needed for approximating the projection of these points. LLE is calculating an embedding, preserving local weights, which means that for the same w_{ij} the following holds:

$$\begin{aligned}F(x_i) = y_i &\approx \sum_{j \in \mathcal{N}(y_i)} w_{ij} y_j = \sum_{j \in \mathcal{N}(y_i)} w_{ij} F(x_j) \\ x_i &\approx \sum_{j \in \mathcal{N}(y_i)} w_{ij} x_j\end{aligned}$$

This property was now used to approximate the unknown projection of the evaluation point using a method introduced in [Fra16] and fittingly referred to as Local Linear Interpolation (LLI) in [Hah16]. The LLI was motivated by LLE, but first introduced in [FZGK14] in the context of Isomap, which is introduced later. First the neighbourhood $\mathcal{N}(x_e^*)$ is determined for the evaluation points, e.g. by computing the kNN.

Afterwards the weights w_{ej}, which best reconstruct the point from its neighbours, are determined in the low dimensional space.

$$\begin{aligned} \min_{w_e} \quad & \varepsilon_e = \left\| x_e^* - \sum_{j \in \mathcal{N}(x_e^*)} w_{ej} x_j \right\|_2^2 \\ \text{subject to:} \quad & \sum_{j \in \mathcal{N}(x_e^*)} w_{ej} = 1 \end{aligned}$$

Subsequently, this linear combination is applied in the high dimensional space and therefore called local linear interpolation.

$$F\left(x_e^*\right) \approx \sum_{j \in \mathcal{N}(x_e^*)} w_{ej} y_j \tag{3.22}$$

One possible variation is to calculate the single nearest neighbour x_j of the evaluation point x_e^* and replace the neighbourhood $\mathcal{N}(x_e^*)$ by $\mathcal{N}(x_j)$ for the interpolation and weight determination. This can be motivated by the fact that the preservation of the local weights is only guaranteed for the given neighbourhoods. However, this raises problems if, for example, the single nearest neighbour is not uniquely determined. Additionally, points that are relatively far away in the high dimensional space can be closer in the low dimensional embedding. These newly discovered similarities, reflected in the low dimensional proximity, would be ignored in this case. Thus, the kNN of the evaluation point x_e^* are used in this work, assuming, that the local weights are approximately preserved for these new neighbourhoods as well. Since the analyst is only interested in a visual approximation, the projection does not need to be exact.

3.3.3 Local Tangent Space Alignment

In this section a second LM is presented to overcome some of the drawbacks of the first method.

3.3.3.1 Base Method

The Local Tangent Space Alignment (LTSA) approach was first published in 2004 by Zhen-yue Zhang and Hong-yuan Zha [ZZ04]. LTSA extends the basic idea of LLE to fit the data locally and then align globally. But instead of preserving local weights, LTSA preserves the local tangent spaces in a global embedding.
A visual example for preservation of approximated tangent spaces is given in Fig. 3.8. In this figure, the dashed line is the approximation of the one dimensional tangent space, spanned by the middle point and its two nearest neighbours. This tangent space is defined by the relative position of the points in this subspace and preserved in the one dimensional embedding. In contrast to Fig. 3.6, the middle point is included in the calculation of the tangent space and not only approximated by its neighbours.

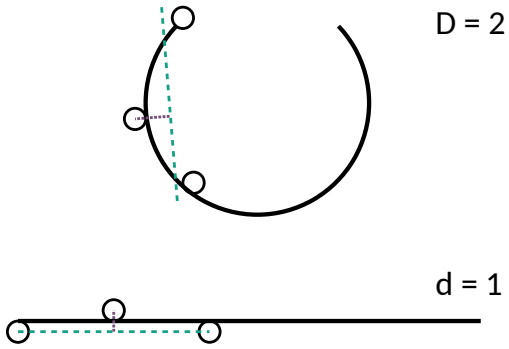


Figure 3.8: Graphical example of preserving the tangent space in a low dimensional embedding. The continuous line represents the manifold with three sample points highlighted. This schematic was motivated by [LV07].

In the first step of LTSA, the complete neighbourhood $\overline{\mathcal{N}}(y_i)$ is determined for each point y_i, by calculating the neighbourhood $\mathcal{N}(y_i) = \{i_1, ..., i_k\}$ based on the kNN and including the index i of the point itself. Then, the local data matrix Y_i for the complete neighbourhood is aggregated:

$$\begin{aligned} \overline{\mathcal{N}}(y_i) &:= \mathcal{N}(y_i) \cup \{i\} \\ Y_i &:= (y_{i_1} \ ... \ y_{i_k} \ y_i) \\ \overline{y}_i &:= \frac{1}{|\overline{\mathcal{N}}(y_i)|} Y_i \mathbb{1}_{|\overline{\mathcal{N}}(y_i)|} \end{aligned} \tag{3.23}$$

Afterwards, a PCA is performed for this local data matrix Y_i, by centralising it according to the mean value $\overline{y}_i$ of the columns and computing the eigenvectors corresponding to the d largest eigenvalues of the local Gram matrix $\overline{G}_i \in \mathbb{R}^{|\overline{\mathcal{N}}(y_i)| \times |\overline{\mathcal{N}}(y_i)|}$:

$$\begin{aligned} \overline{G}_i &:= \left(Y_i - \overline{y}_i \mathbb{1}^\mathsf{T}_{|\overline{\mathcal{N}}(y_i)|}\right)^\mathsf{T} \left(Y_i - \overline{y}_i \mathbb{1}^\mathsf{T}_{|\overline{\mathcal{N}}(y_i)|}\right) \\ &=: W_i \Lambda_i W_i^\mathsf{T} \end{aligned} \tag{3.24}$$

The eigenvectors of this local Gram matrix define the directions of the tangent space. As for all LM-approaches, the next step of LTSA is to aggregate these local properties into one global data matrix Φ_{LTSA}. First, a modified local Covariance matrix is reconstructed from the local eigenvectors. Then the global matrix Φ_{LTSA} is initialised to $\mathbb{0}_{s \times s}$ and iteratively updated with the modified reconstructions of all points. With $\Phi_{\text{LTSA}}\left[\overline{\mathcal{N}}(y_i), \overline{\mathcal{N}}(y_i)\right]$ being the square submatrix for the indices in $\overline{\mathcal{N}}(y_i)$ the update is performed $\forall i = 1, ..., s$ by:

$$\Gamma_i := \left(\frac{\mathbb{1}_{|\overline{\mathcal{N}}(y_i)|}}{\sqrt{|\overline{\mathcal{N}}(y_i)|}} \ w_1 \ ... \ w_d \right) \in \mathbb{R}^{|\overline{\mathcal{N}}(y_i)| \times d+1} \tag{3.25}$$

$$\Phi_{\text{LTSA}}\left[\overline{\mathcal{N}}(y_i), \overline{\mathcal{N}}(y_i)\right] \leftarrow \Phi_{\text{LTSA}}\left[\overline{\mathcal{N}}(y_i), \overline{\mathcal{N}}(y_i)\right] + I_{|\overline{\mathcal{N}}(y_i)|} - \Gamma_i \Gamma_i^\mathsf{T} \tag{3.26}$$

Finally, the low dimensional coordinates can be obtained by computing an EVD of this global data matrix and selecting the eigenvectors corresponding to the second to $d+1$-smallest eigenvalues:

$$\begin{aligned} \Phi_{\text{LTSA}} &=: V \Lambda V^{\mathsf{T}} \\ \widetilde{X} &:= (v_{s-d-1} \ \dots \ v_{s-1})^{\mathsf{T}} \end{aligned} \tag{3.27}$$

As for all LM-approaches, the vector corresponding to the smallest eigenvalue is discarded, since it is assumed to be $\mathbb{1}_s$, which has an eigenvalue of 0 by construction, which is achieved by adding the normalised vector in Eq. (3.25).

1: Construct the complete neighbourhoods $\overline{\mathcal{N}}(y_i)$ of each data point y_i.
2: Centralise data and compute d largest eigenvectors W_i according to Eq. (3.24).
3: Aggregate global alignment matrix Φ_{LTSA}, see Eq. (3.26).
4: Compute second to $d+1$-smallest eigenvalues of matrix Φ_{LTSA} and construct coordinates according to Eq. (3.27).

Algorithm 3.2: LTSA Algorithm. Modified from [ZZ04].

The local PCA described in Eq. (3.24) can also be computed from pairwise Euclidean distances, as explained in Section 3.4.2, and the calculation of the kNN is also trivially possible. Thus, since the subsequent steps do not depend on the actual values, the LTSA can be performed from pairwise Euclidean distances as well as from the actual values.
Like LLE, LTSA is still a LM but has two improvements over the first approach.
Rather than relying on a single property such as one set of reconstruction weights, LTSA preserves multiple properties as the tangent space is described in terms of multiple base vectors. These multiple properties leave less degrees of freedom for the low dimensional embedding, which alleviates the risk of not capturing the global structure correctly in the low dimensional embedding.
Furthermore, the actual position of the active point in the local neighbourhood is reflected: The LLE discards the information, whether the approximation by the nearest neighbours was good or bad, see Fig. 3.7. In LTSA, however, this information is encoded in the base vectors of the tangent space. If the active point is different from its neighbours, there is a dedicated base vector separating it from the other points. Since these base vectors are used to build the global alignment matrix, this information is incorporated in the process.
Unfortunately, LTSA also inherits some of LLE's problems. The overall performance heavily depends on the ability to describe the global manifold by local properties only. This is furthermore complicated by the fact that the local PCA of the neighbourhoods $\overline{\mathcal{N}}(y_i)$ can sometimes fail to determine the tangent spaces correctly, e.g. for data with heavy noise or bad choices of k.

In [ZQZ11], the authors introduced an improved version of LTSA by adjusting the determination of the tangent space. In this variant, the points in the neighbourhood $\mathcal{N}(y_i)$ are centralised onto the point y_i instead of the local mean value $\overline{y}_i$. This way, the origin of the approximation of the local tangent space is at the given point and not in the local mean. Furthermore, in the following local PCA, the points are weighted according to their distance to the point i in relation to a kernel width $t \in \mathbb{R}$:

$$\omega_{ij} = \exp\left(-\frac{\|y_j - y_i\|_2^2}{t}\right) \quad , \forall j \in \overline{\mathcal{N}}(y_i) \tag{3.28}$$

A benefit of this weighting is that points in the local neighbourhood, which are far away in relation to the kernel width, are not "polluting" [ZQZ11] the tangent space estimation. A drawback of this method is the additional parameter that needs to be calibrated to the underlying data, which faces the same challenges as the ε-rule mentioned in Section 3.3.2.1.

3.3.3.2 Extension for Comparative Analysis

For an automated use in a Comparative Analysis, the base version described in the last paragraph is easier to use than the improved method of [ZQZ11], since it is not concerned with calibrating the additional parameter, but only the neighbourhood size k. Therefore, for simplicity and better comparability, the base version was used, and the same modified k-rule was applied, as in the other previously mentioned approaches, see Section 3.3.2.2.

For use in a CA, the original method had to be adjusted. The initial design requires that the target dimension d is known before computing the embedding to determine the correct number of base vectors of the tangent space to be extracted from the local PCA. Since the intrinsic dimension is rarely known in the application of analysing simulation results, an alternative was developed: The local PCA was conducted for the full number $|\mathcal{N}(y_i)|$ of possible base vectors and then a variance cut according to Eq. (3.9) was applied with $\kappa = 0.96$. This way, the local tangent space dimension d_i is determined independently for all neighbourhoods. This needs to be reflected in the reconstruction of the local covariance matrices $\Gamma_i \Gamma_i^\mathsf{T}$. In the original approach, all base vectors were equally important and treated equally in the reconstruction. With the new variance cut, it is important to introduce some order or ranking to the tangent space base vectors and to transfer this ranking to the global alignment matrix Φ_{LTSA}. This was achieved by the following scaling for a given $0 < \eta \ll 1 \in \mathbb{R}$:

$$\widetilde{\Gamma}_i := \left(\frac{\mathbb{1}_{|\overline{\mathcal{N}}(y_i)|}}{\sqrt{|\overline{\mathcal{N}}(y_i)|}} \;\; (1-\eta)w_1 \;\; \ldots \; (1-j\eta)w_j \;\; \ldots \;\; (1-d_i\eta)w_{d_i}\right) \tag{3.29}$$

Trivially, setting η to zero would yield the original method. A value of $\eta = 10^{-7}$ was used in this work, but this could be adjusted for other applications. This way, the

influence of the base vectors on the reconstructed Covariance matrix $\widetilde{\Gamma}_i\widetilde{\Gamma}_i^\mathsf{T}$ is slowly declining for eigenvectors associated with smaller eigenvalues, but not so strong that they could be neglected. Furthermore, the scaling with η has a subtle but beneficial effect on the obtained embedding: If all tangent vectors are equally weighted, the orientation of the resulting low dimensional coordinates is arbitrary. If the scaling is applied, the first coordinate is the one that best aligns with the locally most important components, the second the second best and so on, which can contribute to orientate the result. Apart from using the modified matrices $\widetilde{\Gamma}_i$ the aggregation of the global alignment matrix Φ_{LTSA} can be done without any further modifications afterwards. Another benefit of this modification, apart from not needing to know the intrinsic dimension beforehand, is that this version of the LTSA method is incremental: Since the local tangent space estimation does not depend on the target dimension d, it does not affect the computation of the global alignment matrix Φ_{LTSA}. Hence, the resulting eigenvectors are always the same and the only difference is the number of vectors used for the low dimensional coordinates.
The base LTSA approach does not provide any equivalent to the importance factors needed for the application in an CA. A method which tries to compute global importance factors from the local eigenvalues of each neighbourhood was investigated, but the results were not encouraging. Instead, this modified method was extended by importance factors in the same manner as LLE in Section 3.3.2.2: After calculating low dimensional coordinates, these are scaled to make the embedding locally distance preserving, see Eq. (3.21). This is motivated by the fact that the tangent spaces are defined in terms of linear combinations of the nearest neighbours, which can be interpreted as weights for local interpolations as well, see Section 3.2.3.
In a similar way, the visualisation of the underlying effects can be achieved through a variant of LLI, here referred to Local Affine Interpolation (LAI). Analogous to the two methods described before, the neighbourhoods $\mathcal{N}(x_e^*) = \{j_1, ..., j_{k+1}\}$ for the evaluation points are first determined, but this time using the $(k+1)$-nearest neighbours. The next step of the LAI is to compute reconstruction weights in the low dimensional space, which can be done utilising the preservation of tangent spaces. The idea is the same as for PCA described in Section 3.2.3, but the operation is limited to the given neighbourhood, which is in general not centralised like the PCA result. Thus, the mean $\overline{x}_e^*$ of the neighbourhood without the evaluation point is computed first and the data centralised according to this mean value. Then, an SVD of the centralised neighbourhood is performed.

$$\left(x_{j_1} - \overline{x}_e^* \;\; ... \;\; x_{j_{k+1}} - \overline{x}_e^*\right) =: U\Sigma V^\mathsf{T} \tag{3.30}$$

Since U is an orthonormal basis of the tangent space, the weights to construct the projection of x_e^* onto the tangent space from its neighbours can be computed by subtracting the mean value, then multiplying with the pseudo inverse and adding the mean value in terms of weights afterwards:

$$\omega := V\Sigma^{-1}U^\mathsf{T}\left(x_e^* - \overline{x}_e^*\right) + \frac{1}{k+1}\mathbb{1}_{k+1} \tag{3.31}$$

Finally, the approximation of the high dimensional point can be computed as a linear combination of the nearest neighbours in the original dimension using these weights, see Eq. (3.22).

3.3.4 Modified Locally Linear Embedding

This subsection introduces a third LM, which is an extension to the LLE approach, dealing with the aforementioned regularisation issues of the original approach.

3.3.4.1 Base Method

Modified Locally Linear Embedding (MLLE) [ZW07] is sometimes also aptly called Multiple-weight Locally Linear Embedding [BLTD17] and was introduced 2007 by Zhen-yue Zhang and Jing Wang to overcome a core problem of LLE.
In LLE, the calculation of the weights that best reconstruct a point from its neighbours is achieved by aggregating a local Gram matrix and then solving the resulting system of linear equations, see Eq. (3.13). As already mentioned, this poses a problem, if the resulting matrix is singular or close to singular. This can be the case, for example, if the number of neighbours is bigger than the intrinsic dimension d or if the samples are not well distributed. The original LLE approach and several derived variants try solving this issue by various regularisations, but calibrating the underlying hyper parameters can be challenging for some applications.
In contrast, the base idea of MLLE is to treat the singularity of this local Gram matrix not as a problem to overcome but as a potential to use: The small eigenvalues of the matrix indicate not only that the system can be solved, but also that there are several close to optimal solutions existing simultaneously. In MLLE a low dimensional embedding is calculated, which best preserves all these close to optimal combinations of weights.
An example for the preservation of multiple weights is given in Fig. 3.9. Here, the intersection of the dashed and the dotted lines in two dimensions is the best approximation of the leftmost point by its nearest neighbours. This intersection can be defined by a weighted pair of any two out of the three points on the right. In one dimension, the position of the approximation relatively to all other points is preserved. This is similar to Fig. 3.6 in that the ratio of the line segments to the intersection is preserved, but this time all segments between the three neighbours are considered.

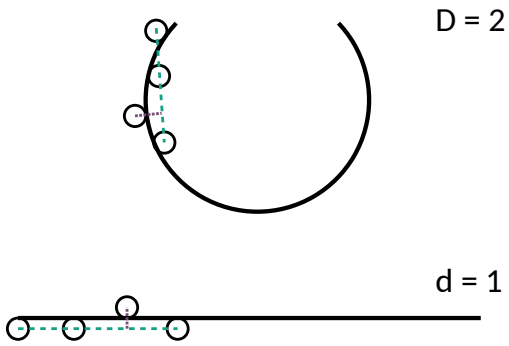


Figure 3.9: Graphical example of preserving multiple weights in a low dimensional embedding. The continuous line represents the manifold with four sample points highlighted. This schematic was motivated by [LV07].

The first step of MLLE is the same as in LLE. Initially, the local Gram matrix $G_i \in \mathbb{R}^{|\mathcal{N}(y_i)|\times|\mathcal{N}(y_i)|}$ is assembled for each point i and the solution $w_i(\varrho) \in \mathbb{R}^{|\mathcal{N}(y_i)|}$ to Eq. (3.12) with the $\varrho \in \mathbb{R}$ regularisation of Eq. (3.15) is computed. Additionally, an EVD of the local Gram matrix is computed, where it is assumed with

$$G_i =: V_i \operatorname{diag}\left(\lambda_1, ..., \lambda_{|\mathcal{N}(y_i)|}\right) V_i^{\mathsf{T}}$$

that $\lambda_1 \geq \lambda_{r_i} \gg \lambda_{r_i+1} \geq \lambda_{|\mathcal{N}(y_i)|}$, meaning that the first r_i eigenvalues of G_i are relatively large compared to the remaining $s_i := |\mathcal{N}(y_i)| - r_i$ values. The eigenvectors can also be separated in two groups, with:

$$V_i =: (V_{r_i} V_{s_i})$$

Starting from the same regularised solution as in LLE, Zhang and Wang construct s_i linear independent weight vectors $w_i^{(1)} ... \ w_i^{(s_i)}$ utilising the eigenvectors V_{s_i} of G_i which correspond to the aforementioned small eigenvalues. To construct these weight vectors, two instances are needed. The first is the fraction of the local mean value lying in the span of these eigenvectors. The mean value can be expressed in terms of weights and then projected onto these vectors:

$$\alpha_i := \frac{1}{|\mathcal{N}(y_i)|} \| V_{s_i}^{\mathsf{T}} \mathbb{1}_{|\mathcal{N}(y_i)|} \|_2$$

The second is $H_i \in \mathbb{R}^{s_i \times s_i}$, a Householder matrix [LM12] defined by:

$$\begin{aligned} h_i^0 &:= \alpha_i \mathbb{1}_{s_i} - V_i^{\mathsf{T}} \mathbb{1}_{|\mathcal{N}(y_i)|} \\ h_i &:= \begin{cases} \frac{h_i^0}{\|h_i^0\|_2} & , \text{if } \ h_i^0 \neq \mathbb{0}_{s_i} \\ \mathbb{0}_{s_i} & , \text{else} \end{cases} \\ H_i &:= I_{s_i} - 2 h_i h_i^{\mathsf{T}} \end{aligned}$$

With these two instances and $H_i(:,l)$ being the l-th column of H_i, several weight vectors can be constructed as

$$w_i^{(l)} := (1-\alpha_i)w_i(\varrho) + V_i H_i(:,l) \quad ,\forall l = 1,...,s_i \tag{3.32}$$

This means that the initial solution is varied along the axis of the kernel of matrix G_i to obtain multiple weight vectors. These multiple weight vectors should be preserved for all neighbourhoods in a global embedding. As for all LM approaches a global alignment matrix Φ_{MLLE} is aggregated. The authors use the same mechanism they previously utilised in LTSA. They initialise $\Phi_{\text{MLLE}} = \mathbb{0}_{s\times s}$ and then update it with local Gram matrices computed from the weight vector matrices for all points. Again, $\Phi_{\text{MLLE}}[\mathcal{N}(y_i),\mathcal{N}(y_i)]$ is the submatrix for the indices in $\mathcal{N}(y_i)$ and the update can be written as:

$$\begin{aligned} W_i &:= \left(w_i^{(1)}... \; w_i^{(s_i)}\right) \in \mathbb{R}^{|\mathcal{N}(y_i)|\times s_i} \\ \Phi_{\text{MLLE}}[\mathcal{N}(y_i),\mathcal{N}(y_i)] &\leftarrow \Phi_{\text{MLLE}}[\mathcal{N}(y_i),\mathcal{N}(y_i)] + W_i W_i^{\mathsf{T}} \quad \forall\ i = 1,...,s \end{aligned} \tag{3.33}$$

Analogous to LLE and LTSA, an EVD of the global alignment matrix Φ_{MLLE} is computed and the low dimensional coordinates retrieved from the eigenvectors by:

$$\begin{aligned} \Phi_{\text{MLLE}} &:= V\Lambda V^{\mathsf{T}} \\ \widetilde{X} &:= (v_{s-d-1} \;\; \ldots \;\; v_{s-1})^{\mathsf{T}} \end{aligned} \tag{3.34}$$

As with the other two approaches before, the vector corresponding to the smallest eigenvalue is discarded because it is assumed to be $\mathbb{1}_s$, which by construction has an eigenvalue of 0, because its contribution to the weight vectors was as α_i and then subtracted in Eq. (3.32).

1: Determine neighbourhood $\mathcal{N}(y_i)$ of each data point y_i.
2: Compute regularised solution $w_i(\varrho)$ according to Eq. (3.15).
3: Select the number of weights s_i to be preserved for each neighbourhood.
4: Assemble the global alignment matrix Φ_{MLLE} by the update of Eq. (3.33).
5: Compute second to $d+1$-smallest eigenvalues of matrix Φ_{MLLE} and construct coordinates according to Eq. (3.34).

Algorithm 3.3: MLLE Algorithm. Modified from [ZW07].

In Section 3.3.2 it was already mentioned that the aggregation of the local Gram matrix and computation of the regularised solution can be done from pairwise distances as well as from the high dimensional values. Since the first step of MLLE is to construct said local Gram matrix and all further steps are independent of the original values, MLLE can be performed from pairwise differences in the same way as LLE,

which can be interesting for data sets with a large original dimension.
The preservation of multiple weights in MLLE mitigates two problems of the LLE approach: Using multiple weights instead of a single set helps with the insufficient capture of the underlying structure of the manifold similar to the multiple base vectors of the tangent space in LTSA. Furthermore, the authors claim in [ZW07] that the solution is not as sensitive to the regularisation parameter ϱ as the original approach, which makes calibration much easier. This behaviour was also confirmed during the research for this work. The property of losing information regardless of whether the approximation is good or poor does still persist in MLLE because no further residual information is transferred to the global alignment matrix. This leaves more freedom for the global embedding than the LTSA method, which may or may not be desirable, depending on the applications and the level of noise present in the data.

3.3.4.2 Extension for Comparative Analysis

Similar to the basic LLE, the modified approach has a few options, to modify the outcome of the method and those need to be chosen appropriately in the context of the analysis of simulation results.
The first options are the construction of the local neighbourhoods $\mathcal{N}(y_i)$ and the choice of the regularisation parameter ϱ_i. Since they are the same as in the basic approach, these options were also treated in the same manner as explained in Section 3.3.2.2 for LLE: With the modified k-rule and the new regularisation $\varrho_{i,\mathrm{NEW}}$ introduced in Eq. (3.18), for the same reasons explained earlier, as well as to increase the comparability with all previous approaches. The weight factor ω was chosen identical in both LLE and MLLE. This is possible, since the claim of the authors in [ZW07], that MLLE is not so sensitive to the regularisation, was confirmed in most investigated examples.
A new option in MLLE is how to choose the number of different weight vectors s_i for each point. The original publication suggests choosing a threshold error value $\eta < 1$ and selecting the number of weights as:

$$\begin{aligned} & s_i := \max_l \; l \\ \text{subject to: } & l \leq |\mathcal{N}(y_i)| - d \\ & \frac{\sum_{j=|\mathcal{N}(y_i)|-l+1}^{|\mathcal{N}(y_i)|} \lambda_j^2}{\sum_{j=1}^{|\mathcal{N}(y_i)|-l} \lambda_j^2} < \eta \end{aligned} \tag{3.35}$$

with $\lambda_1 \geq ... \geq \lambda_{|\mathcal{N}(y_i)|}$ being the eigenvalues of the local Gram matrix G_i as introduced before. Unfortunately, this method is only applicable for applications where the target dimension d is known before applying the method, as it is explicitly used in Eq. (3.35). Furthermore, an appropriate threshold error value η is sometimes hard to calibrate, though the authors give a recommendation on how to calibrate this value in [ZW07]. In this work, a new approach was developed to determine the number of

different weights. The idea is to first compute the approximate rank $\tilde{r}_i$ of the problem by performing a variance cut

$$\begin{aligned} g(l) &:= \frac{\sum_{k<l} \lambda_k^2}{\sum_j \lambda_j^2} \\ \tilde{r}_i &:= \arg\min_l g(l) \\ \text{subject to:} \quad & g(l) > \kappa \end{aligned} \tag{3.36}$$

and then to calculate s_i by a simple subtraction to get an approximation of the kernel dimension:

$$s_i = |\mathcal{N}(y_i)| - \tilde{r}_i$$

If the number of weights is smaller than two, only the single regularised solution $w_i(\varrho)$ is used. This way, the target dimension does not need to be known before applying the reduction. Furthermore, for fixed neighbourhoods $\mathcal{N}(y_i)$ the extended method with this mechanism is modified to be incremental, since the calculated weights vectors are now the same and only the number of dimensions to be computed is depending on the target dimension d. This variance cut is a new hyper parameter, but it is fortunately relatively easy to calibrate, since $\kappa = 0.97$ was used for all tests in this work and showed viable results for all examples.

The last requirements for an CA are importance factors and the visualisation of underlying effects. Since the base concept is still the same, the MLLE method was extended by importance factors in the same manner as LLE in Section 3.3.2.2 and the creation of virtual simulation was done accordingly as well. Both operations require the preservation of local linear combinations, which is still valid for MLLE, thus no further modifications were needed. This way, all three introduced LMs are completed to be used in the analysis of simulation results.

3.4 Multidimensional Scaling

The next class of DRMs is the class of the Multidimensional Scaling (MDS) [KW78] methods.

3.4.1 Commonalities

The term MDS describes a family of methods that calculate low dimensional embeddings that best preserve given pairwise dissimilarities. These methods are divided into two subclasses of metric and non-metric MDS methods, depending on whether they are based on a continuous quantitative dissimilarities or discrete similarity ranks [LV07]. In this work, only metric MDS approaches are covered, and all dissimilarities are distance measures. Starting from the given distance values δ_{ij} from point number i to point number j, the first step of metric MDS is to square these values and aggregate them into a matrix $\Delta_Y \in \mathbb{R}^{s \times s}$. It is assumed that the distances are

symmetric, meaning that $\delta_{ij} = \delta_{ji}$. The next step is to perform a so-called "double centring" [LV07] using the centring matrix Z_s, see Eq. (3.3).

$$\begin{aligned} Z_s &:= I_s - \frac{1}{s}\mathbb{1}_s\mathbb{1}_s{}^\mathsf{T} \\ \Delta_Y &:= \left(\delta_{ij}^2\right)_{ij} \\ S_Y &:= -\frac{1}{2}\left(Z_s \Delta_Y Z_s\right) \end{aligned} \tag{3.37}$$

The matrix $S_Y \in \mathbb{R}^{s\times s}$ is decomposed into an EVD with:

$$S_Y =: V\Lambda V^\mathsf{T}$$

Then, a number of vectors $1 \leq d < s$ determined, such that the truncated EVD restricted to the eigenvectors associated with the largest eigenvalues yields a good approximation of the squared and centred distance matrix. These eigenvectors are transposed and multiplied by the square root of the eigenvalues to obtain the low dimensional representation.

$$\begin{aligned} S_Y \approx S_{Yd} &:= V|_d \Lambda|_d V|_d^\mathsf{T} \\ \widetilde{X} &:= \Lambda|_d^{\frac{1}{2}} V|_d^\mathsf{T} \end{aligned} \tag{3.38}$$

The Euclidean distance of these low dimensional points is then close to the high dimensional dissimilarity:

$$\|x_i - x_j\|_2 \approx \delta_{ij}$$

In more detail, the embedding computed by an MDS approach is minimising the following stress function globally in a least squares sense, see [LV07] for more details.

$$E_{\text{MDS}} = \frac{1}{2}\sum_{i,j=1}^{s}\left(\delta_{ij} - \|x_i - x_j\|_2\right)^2 \tag{3.39}$$

This means that the absolute error is minimised, resulting in longer distances being relatively more accurate than shorter distances.

1: Compute pairwise squared distances.
2: Perform double centring, see Eq. (3.37).
3: Compute EVD and low dimensional embedding according to Eq. (3.38).

Algorithm 3.4: MDS Algorithm. Modified from [LV07].

In contrast to the LM-approaches, which rely solely on local descriptions to capture the underlying manifold, MDS extracts features on a global scale as all pairwise dissimilarities are computed. There are several choices for different dissimilarity measures present in the literature. Three of them are further reviewed in the following subsections.

3.4.2 Classic Metric Multidimensional Scaling

In the case where the dissimilarities are the pairwise Euclidean distances $\delta_{ij} = \|y_i - y_j\|_2$, the method is called Classic Metric MDS. It is shown in [LV07] that the double centred matrix $\frac{1}{D}S_Y = Gy$, with G_Y is the Gram matrix of scalar products of the PCA-approach as introduced in Section 3.2. Since the matrices for the eigenvalue problem are the same PCA and classical metric MDS yield the same results [LV07]. Since the pairwise Euclidean distances are the starting point, it offers an alternative to the standard PCA approach: Sometimes the aggregation of the full data matrix can be challenging, especially for large result sets, while incremental computation of squared pairwise distances can be easier in these cases. Furthermore, the eigenvalues of the EVD directly yield the importance factors for the CA as described in Section 3.2. However, since Classic Metric MDS is equivalent to PCA, it also has the same aforementioned limitations resulting from the linear approach.

3.4.3 Isomap

In this subsection a second MDS approach is introduced, which is a nonlinear alternative to the linear method described in the last subsection.

3.4.3.1 Base Method

The Isomap approach, short for Isometric Feature Mapping, was first published by Joshua B. Tenebaum in 1998 [Ten98] and later together with Vin De Silva and John C. Langford [TDSL00]. The base idea is to compute a quasi-isometric embedding by preserving the geodesic distances of points in the data set [BST$^+$02]. The geodesic distance is the length of the path from one point to another on a manifold, which can largely vary from the Euclidean distance of the points, see Fig. 3.10.
Isomap is an MDS approach that aims to compute a low dimensional embedding in which the Euclidean distances are equal to these geodesic distances in the high dimensional space.

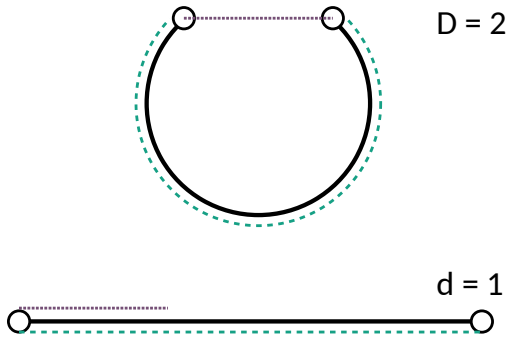


Figure 3.10: Graphical example of different distances. The continuous line represents the manifold with two sample points highlighted. The dotted line represents the Euclidean distance in two dimensions. The dashed line is the geodesic distance of the two points regarding this manifold, which is identical in one or two dimensions. This plot is a modified version of the original in [LV07].

Reconstructing geodesic paths in a manifold that is unknown and only given in the form of a finite number of possibly noisy samples is not trivial. In the Isomap method, the unknown true geodesic paths are approximated by graph distances. For this purpose, the neighbourhood $\mathcal{N}(y_i)$ for each point y_i is constructed, for example by the ε- or the k-rule, see Section 3.3.2.1, or by any rule the analyst sees fit, for example, the ones introduced in [LV07]. Then, an undirected graph is constructed by connecting a point with all points in its neighbourhood and assigning the Euclidean lengths of these edges as the edge weights. Next, the All-Pairs Shortest Paths (APSP) are computed in this graph and the lengths of these paths are considered as the geodesic distances of the points.
Finally, the MDS steps of assembling the dissimilarity matrix Δ_Y from these squared distances, double centring as in Eq. (3.37) and the EVD computation are performed.

1: Compute the neighbours $\mathcal{N}(y_i)$ of each data point y_i and build undirected neighbourhood graph
2: Compute APSP and the length of the paths in this graph as distances.
3: Perform regular MDS with these graph distances.

Algorithm 3.5: Isomap Algorithm. Modified from [TDSL00].

As the graph and the introduced neighbourhood rules only depend on the Euclidean distances, the graph distances of Isomap can either be computed from high dimensional coordinates or directly from Euclidean distances.
These graph distances can sometimes unravel global structures better than Euclidean distances or the LMs. Though this can help with the aforementioned issue of failing

to capture the global features in LMs, the graph distances introduce a new problem. Beyond one dimensional manifolds the paths in the neighbourhood graph tend to introduce "spurious geodesic curvature, i.e. zigzags" [BYF+19] into the data, since they cannot follow the manifold directly, but only the edges between nearest neighbours. These zigzags cause small detours in the geodesic paths increasing with the number of edges, since the graph paths are only an approximation of the true geodesics.
An example for such detour is given in Fig. 3.11. In this figure, the dashed line is the geodesic distance between the square and the triangle, while the dotted line is the significantly larger approximation yielded by the graph distances. The elongation effect of such detours highly depends on the number of neighbours and the structure of the data but can affect the embedding significantly. This effect can get worse the more edges a path is consisting of.

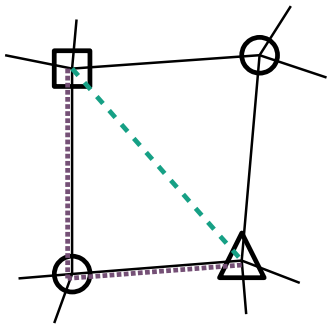


Figure 3.11: Graphical example of spurious geodesic curvature. Given four points sampled from a larger manifold, the continuous lines represent the edges of the neighbourhood graph.

Although the drawback of spurious geodesic curvature is known for Isomap, the approach is very popular in many applications [ST02], [BST+02], [Fra16], among others because of its stability [LV07]. The Isomap approach can handle noisy data as well as mixed dimension manifolds relatively well compared to other methods, as can be seen later in Section 5.1.

3.4.3.2 Extension for Comparative Analysis

Another reason for the popularity of the Isomap approach is that it has only a single parameter which needs to be calibrated to influence the outcome. This parameter is the construction of the neighbourhoods. In this work, the same approach of the modified k-rule, which constructs a connected graph as described in Section 3.3.2.2, was used. One reason is that the MDS steps require a connected graph, otherwise some of the dissimilarities could be infinite leading to problems with centralisation and EVD operations. Another reason is to make the different approaches and their performance better comparable: If the neighbourhood building is identical, the differences in the performance of the DRMs are purely due to the methods themselves.

Apart from the existing parameters, only a few adjustments are needed for the analysis of simulation results. The importance factors required for the CA can be easily obtained since Isomap is an MDS method: The eigenvalues of the EVD of the double-centred distance matrix, see Eq. (3.37) and Eq. (3.38), can explicitly be calculated, and used the same way as in the case of Classic Metric MDS.
The only addition is that the resulting matrix is not necessarily positive semi-definite since the graph distances are only an approximation of the true geodesic distances [LV07]. This should be considered when determining the number of dimensions used for an embedding. In contrast to classic metric MDS, not all computed eigenvectors are beneficial to preserving the pairwise distances. It can occur that extending the embedding to later modes actually worsens the similarity of the low dimensional coordinates and the high dimensional dissimilarities. Therefore, the incremental embedding should be stopped at that point and no further coordinate directions should be used.
The last part missing for a CA is the visualisation of the underlying effects utilising the points $x_e^* \in \{x_e^+, x_e^-\}$. As mentioned before, the LLI was initially introduced in the context of Isomap in [FZGK14] and can be used to get an approximation of the projection of the evaluation points in the same way as described in Section 3.3.2.2. First, the neighbourhood $\mathcal{N}(x_e^*)$ of the evaluation point is determined, for example by calculation of its kNN. Then the weights that best reconstruct the point from its neighbours are computed in the low dimensional space. The approach originally introduced in [FZGK14] is defining an additional norm

$$\|w_e\|_c := \sum_{j \in \mathcal{N}(x_e^*)} c_j w_{ej}^2, \qquad c_j := \upsilon \left(\frac{\|x_e^* - x_j\|_2}{\max_i \|x_e^* - x_j\|_2} \right)^p$$

with hyper parameters $0 < \upsilon \ll 1 \in \mathbb{R}$ and $1 < p \in \mathbb{N}$. This norm is used to modify the function for the weight calculation of LLI with an additional term:

$$\begin{aligned} \min_{w_e} \quad & \varepsilon_e = \left\| x_e^* - \sum_{j \in \mathcal{N}(x_e^*)} w_{ej} x_j \right\|_2^2 + \|w_e\|_c^2 \\ \text{subject to:} \quad & \sum_{j \in \mathcal{N}(x_e^*)} w_{ej} = 1 \end{aligned} \tag{3.40}$$

This way, large weights are penalised for points that are nearest neighbours but relatively far away. While this approach can improve the results for the interpolation by enforcing a local approximation, it has the drawback that two hyper parameters need to be calibrated. Regardless of whether the weights are calculated with or without the additional penalization of large weights for distant points, the approximation of the high dimensional points can afterwards be computed as a linear combination of the nearest neighbours in the original dimension, see Eq. (3.22).

3.4.4 Parallel Transport Unfolding

The method described in this subsection is the latest of all literature approaches mentioned in this work and can be viewed as an improvement of the Isomap method described in the previous subsection.

3.4.4.1 Base Method

The Parallel Transport Unfolding (PTU) approach was introduced in a preprint in 2018 but published in 2019 by Max Budninskiy and others [BYF+19]. Its goal is the same as in Isomap, which is to compute an embedding, where the low dimensional Euclidean distances match the high dimensional geodesic distances.
As explained in Section 3.4.3, the Isomap approach approximates the unknown true geodesic distances by path lengths in a graph defined by local neighbourhoods. This approach tends to overestimate the true geodesic distance for manifolds with an intrinsic dimension $d > 1$ due to paths having to zigzag through intermediate points, subsequently elongating the distance.
The PTU approach is Budninskiy's second method with the aim to correct these paths and to remove the "spurious geodesic curvature" [BYF+19]. The first method called Spectral Affine-Kernel Embedding (SAKE) [BLTD17] did not yield such a good correction as the superior PTU as is shown in the introductory paper of the latter [BYF+19]. PTU's better correction is achieved by projecting the polygon paths resulting from the neighbourhood graph into the tangent space at each point and then parallel transporting them recursively to the origin of said paths. These lengths of the projected and transported paths are called parallel transport distances. The low dimensional embedding is finally obtained by performing an MDS with these distances.
The preservation of such distances is visualised in Fig. 3.12. Here, the dotted line is the distance from the middle to the upper point in the tangent space of the second point. The dashed line is the parallel transport distance of the lower point to the upper point in the tangent space of the lower point. It consists of the distance from the lower point to the middle point in this tangent space plus the parallel transport of the dotted line into this tangent space. All distances parallel transported to the tangent spaces are preserved as Euclidean distances in the low dimensional embedding.

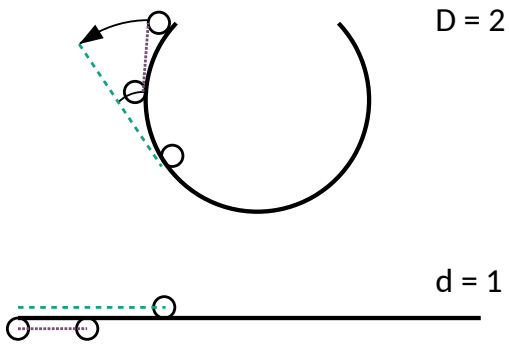


Figure 3.12: Graphical example of preserving parallel transport distances in a low dimensional embedding. The thick continuous line represents the manifold with three sample points highlighted and the dashed line is the parallel transport distance.

The first step of PTU is identical to Isomap, which is to construct the local neighbourhoods $\mathcal{N}(y_i)$ for each point y_i by a suitable rule, see Section 3.3.2.1. Additionally to Isomap, an orthonormal basis $T_i \in \mathbb{R}^{D\times d}$ for the d-dimensional tangent space at each of these points is computed, for example, by PCA or a robust alternative [ZL14]. With these so-called "tangent frames" [BYF+19] in place, the discrete parallel transport from data point j to data point i can be defined as

$$R_{j,i} := \underset{R\in\mathbb{O}(d)}{\arg\min} \|T_i - T_j R\|_F^2 \tag{3.41}$$

with $\mathbb{O}(d)$ being the group of orthogonal matrices in $\mathbb{R}^{d\times d}$ and $\|\cdot\|_F$ the Frobenius norm [LM12]. This means that the parallel transport is the orthogonal matrix, which best aligns the tangent frames to each other. As the authors state, $R_{j,i}$ as the optimal R of Eq. (3.41) can be obtained utilising an SVD of the product of said tangent frames:

$$\begin{aligned} T_i^\mathsf{T} T_j &=: U\Sigma V^\mathsf{T} \\ R_{j,i} &:= VU^\mathsf{T} \end{aligned} \tag{3.42}$$

From this follows that the transport in the opposite direction can be obtained as $R_{j,i}^\mathsf{T} = R_{i,j}$. In the next step of PTU, the shortest paths in the neighbourhood graph are computed and then corrected using these parallel transport matrices.
Let $(y_o, y_{i_1}, y_{i_2}, y_{i_3}, ..., y_{i_k}, y_t)$ be the shortest path through the high dimensional neighbourhood graph from an origin point y_o to a target point y_t. This path is incrementally projected into the tangent space of the origin, starting with the first edge y_o to y_{i_1}. The multiplication of the difference of the two points with the orthonormal basis

$$v_{o,i_1} := T_o^\mathsf{T} (y_{i_1} - y_o)$$

yields the coefficients of the linear combination, which expresses the first edge in terms of the tangent frame at the origin. This is considered as the first edge of the

corrected path in the representation of the orthonormal basis. The second edge is then also expressed in terms of the local tangent frame but of the first intermediate point y_{i_1} instead of the origin.

$$v_{i_1,i_2} := T_{i_1}^{\mathsf{T}} \left(y_{i_1} - y_{i_2}\right)$$

This projection into the intermediate tangent frame is then parallel- transported back to the origin point. The corrected path is then updated with this projected and parallel transported edge:

$$v_{o,i_1,i_2} := v_{o,i_1} + R_{o,i_1}\, v_{i_1,i_2}$$

Each point must be transported through all intermediate tangent spaces. Thus, the third edge is projected into the tangent space at the second intermediate point and then transported through two predecessor tangent spaces to get the update for the corrected path in the local frame at the origin.

$$v_{o,i_1,i_2,i_3} := v_{o,i_1,i_2} + R_{o,i_1} R_{i_1,i_2}\, T_{i_2}^{\mathsf{T}} \left(y_{i_3} - y_{i_2}\right)$$

This process is incrementally repeated until the target point is reached:

$$v_{o,i_1,\ldots,i_k,t} := v_{o,i_1,\ldots,i_k} + R_{o,i_1} \cdot \ldots \cdot R_{i_{k-1},i_k}\, T_{i_k}^{\mathsf{T}} \left(y_t - y_{i_k}\right) \tag{3.43}$$

The norm of the vector containing the corrected path is considered as the distance from origin to target, since it is composed of coefficients for a linear combination of orthonormal vectors. Parallel transports are path dependent, i.e. different paths can yield slightly different distances and the path from origin to target is not necessarily the same as from target to origin. Furthermore, even if the path is the same, the edges from one point to another are projected into a different tangent space, if traversed into the opposite direction. Thus, the parallel transport distance is the average of the two paths between the points where origin and source are switched:

$$\widetilde{\delta}_{ot} := \frac{1}{2}\|v_{o,i1,\ldots,ik,t}\|_2 + \frac{1}{2}\|v_{t,j1,\ldots,jl,o}\|_2 \tag{3.44}$$

As PTU is an MDS approach the low dimensional embedding can afterwards be obtained by calculating the first d eigenvectors of the squared and double centred distance matrix, see Section 3.4.

1: Determine neighbourhood $\mathcal{N}(y_i)$ of each data point y_i and build the neighbourhood graph.
2: Calculate the shortest paths in this neighbourhood graph and the tangent frames for all points y_i.
3: Compute all pairwise parallel transport distances incrementally according to Eq. (3.43) and Eq. (3.44).
4: Perform regular MDS with these parallel transport distances.

Algorithm 3.6: PTU Algorithm. Modified from [BYF+19].

The projection into the tangent spaces and transport to the origin of the paths can correct the spurious geodesic curvature introduced by the graph distances and thus yield a better approximation of the geodesic distance than the latter. This was demonstrated in [BYF+19], but it was also stated, that paths including many edges, may have a decreasing accuracy, since they involve many subsequent matrix-matrix multiplications which can accumulate errors.
Furthermore, for the PTU to yield meaningful results, certain requirements to the data and the intermediate structures must be met. One of these requirements is inherited from the Isomap approach: The neighbourhood graph needs to be connected, otherwise the distances cannot be computed reasonably.
A new additional requirement is that each of the neighbourhoods must span a tangent space of sufficient dimension at each node. The original method requires the computation of orthogonal transport matrices $R_{i,j} \in \mathbb{O}(d) \subset \mathbb{R}^{d \times d}$, which means, amongst others, that these matrices are square. This is only possible if the tangent frames are of the same dimension d. If a local neighbourhood does not yield enough information to span a d-dimensional tangent space at one of the points, this is not possible. This requirement will be revisited in Section 3.4.4.2.
Another somewhat hidden requirement is the smoothness of adjacent tangent spaces. The discrete formulation of a parallel transport by aligning the tangent frames connected in the nearest neighbour graph is only a valid approximation of the real continuous parallel transport, if the tangent spaces are relatively similar. If there are large differences in the orthonormal basis of neighbouring points, the alignment could yield jumps or reflections. This means that the transitions from one tangent frame to a neighbouring one must be relatively smooth, otherwise cracks and gaps can open in the embedded manifold. This topic is indirectly addressed in the original paper [BYF+19] by sometimes using different numbers k and K for the nearest neighbours. The original k is used to construct the neighbourhood graph and is selected rather small to prevent short cutting, which means the resulting edges in graph are leaving the manifold and are a poor approximation of geodesics. The other $K \geq k$ is used to compute the tangent spaces. In most of their examples, the authors chose $K = k$, but for some applications with rather noise data sets, they increase the value up to $K = 3k$, as they state, to improve the results. The larger neighbourhoods for the estimation of the tangent spaces mean, amongst others, that the differences from one point to its neighbours are smaller because more points are shared between connected samples than in cases where the neighbourhoods contain fewer points. A correct estimation of the tangent spaces for each point y_i is more important than for example in the LTSA approach, since it affects all paths that pass through this point and thus directly translates to all affected pairwise distances: If the estimation is bad, all paths could be shortened or elongated. In the LTSA approach, on the other hand, the global alignment matrix is assembled, and the global system solved in a least squares sense. This means that a poor approximation of a single tangent space will only affect a few rows and columns and can be corrected afterwards in the global embedding. In summary, this means that the PTU approach can improve the

Isomap method and yield better results, but it risks being more susceptible to noise or poorly sampled data.
Another benefit of this method is that it is "linearly precise" [BYF+19] under certain circumstances: If the underlying manifold is linear and the local neighbourhoods are sufficiently large to yield the correct tangent frame estimation, the authors have proven in their original paper that the linear manifold is precisely retrieved by the method. Such a proof does not yet exist for any of the other nonlinear methods.
The PTU algorithm is the first DRM covered in this work that cannot be computed directly from pairwise Euclidean distances, since the tangent frames need to be known to compute the parallel transport distances. This can be amended to some extended for data sets with only few samples by performing an initial classic metric MDS on the input data as a preprocessing for PTU. This step can always reduce the initial large dimension D to $\widetilde{D} = \min\{s-1, D\}$ yielding a dataset $\widetilde{X} \in \mathbb{R}^{\widetilde{D}\times s}$, which is at least as small as the pairwise distance matrix $\Delta_Y \in \mathbb{R}^{s\times s}$ and can be processed efficiently. This workaround is especially interesting for data sets with only a few samples s and a large initial high dimension D, which is usually the case in the CA of simulation results.

3.4.4.2 Extension for Comparative Analysis

Like all previous methods, the original PTU has several options to influence the outcome of the method. One of these options is the same as in the Isomap approach: It is the construction of the local neighbourhoods, and in this work the same approach of the modified k-rule was again used as in all previous methods.
But in contrast to Isomap, PTU has additional options that revolve around the estimation of the tangent frames. There are three factors influencing this estimation: The number of neighbours K, the centralisation, and the actual basis approximation itself.
First, the number of neighbours used in this thesis was set to $K = 2k$ to reduce the number of parameters to be calibrated by the user. Using a larger number of neighbours has proven beneficial, especially for noisy data sets, as was stated in the original paper and confirmed by the investigations conducted for this work.
For the second factor, the centralisation of the local neighbourhoods, the different methods, which were introduced in the previous sections, could also be applied. In the MDS approach the points involved in the EVD are centralised according to their mean value. This would translate to computing the mean of the K-nearest neighbours and subtracting it from the data, yielding an expectation of zero. In the LTSA approach, the point y_i was included in the neighbourhood for the mean computation, which would be equivalent to subtracting a different value form the data, yielding possibly not centralised values. The original PTU follows the idea introduced in the improved LTSA method introduced in [ZQZ11], where the neighbourhood is centralised by subtracting the value of y_i placing the origin of the tangent space in the point, where it should be approximated. In this work, the same centralisation as in

the LTSA approach was used in order to increase comparability and to smoothen the tangent spaces: Centralising the neighbourhood onto the point itself can produce artificially unbalanced tangent frames for outliers and points on the boundary of the manifold, while centralising onto the mean of the neighbours completely neglects the position of the active point. Thus, including the point into the centralisation of its neighbourhood is a compromise between the two approaches.
The third option is the actual approximation of the tangent space. In this work the simple PCA approach of a restricted SVD of the previously centralised neighbourhood was used, though it is important to note, that the alternative of a robust PCA mentioned in [BYF+19] might produce some better results for data sets containing few but strong outliers. With the larger neighbourhood size used for computation of the tangent frame and the modified centralisation, there was no need for a more robust approach on the investigated examples in this work, although this can be different in other application fields.

Apart from these options, the original method had to be further adjusted for the usage in the analysis of simulation results. First, the base approach requires the target dimension d to be known before computing the embedding in order to determine the correct number of base vectors for the tangent frames. Here the target dimension is in general unknown and overestimated, so the number of base vectors for each tangent frame needs to be determined automatically. Similar to the LTSA method a local PCA was conducted for the full number $|\mathcal{N}(y_i)|$ of possible base vectors for each neighbourhood and afterwards a variance cut according to to Eq. (3.9) was applied with $\kappa = 0.93$. The value of κ was chosen significantly smaller than in the LTSA approach since the neighbourhoods are much bigger with $K = 2k$ and the reduction should be stronger. This way, the local tangent space dimension d_i is determined independently for all neighbourhoods, which needs to be considered when computing the parallel transport. While the vectors in the resulting tangent frames still have the same size, the number of base vectors on the other hand could be different. This means that $T_i \in \mathbb{R}^{D\times d_i}$ and $T_j \in \mathbb{R}^{D\times d_j}$, which leads to the result that the discrete parallel transport matrix $R_{j,i} \in \mathbb{R}^{d_j\times d_i}$ is in general no longer square. Apart from the need to distinguish the different numbers of base vectors the algorithm itself does not change: The discrete parallel transport can still be computed by performing the SVD of Eq. (3.42) and all further steps remain unchanged, apart from utilising rectangular matrices. This modification relaxes the requirement for each neighbourhood to span a subspace of sufficient size, since this size is adapting for each neighbourhood separately.
The second requirement for the CA is importance factors. Fortunately, these can be easily obtained since PTU is an MDS method: The eigenvalues of the EVD of the double-centred distance matrix, see Eq. (3.37), directly yield the importance factors as for the other two MDS approaches before. Similar to Isomap, the resulting matrix is in general not necessarily positive semi-definite. Therefore, it should also be checked here whether an additional dimension really improves the approximation

of the pairwise distances in the incremental embedding, and the process should be stopped, if the approximation starts to worsen.
Finally, the last addition to the approach is the computation of virtual simulation results for the visualisation of underlying effects. Since the distances in the 1-ring of each node are the distances in the local tangent space at that node and all these pairwise distances are preserved, the PTU also preserves the local tangent spaces similarly to LTSA, except that this information is encoded in the distances rather than in a Gram matrix, see equivalence of classic metric MDS and PCA in Section 3.4.2. Hence, the same mechanism as for LTSA can be used to compute virtual simulations, where an LAI is performed with weights computed by a base representation of the evaluation point in the vectors of the local tangent frame, see Section 3.3.3.2.

3.4.5 Further Methods

After the last requirement for CA is completed, all three introduced approaches in the class of MDS methods can be used in the Extended Workflow. The scope of this work is too limited to cover all MDS approaches, but since this class of DRMs has been very popular in the analysis of simulation results in the recent years, as mentioned in Section 2.1, a few further methods should be mentioned. With the base concept of MDS and a few variations introduced, the difference to methods already used in literature and the reasons why they were not used in this work can be better explained, even though an in-depth introduction for these approaches is not given. Further references and explanations can be found in the respective cited literature for the methods.
In [BGIT+13] and [GIT15] so-called Diffusion Maps were used. Diffusion Maps are also a family of methods, where the used pairwise dissimilarity is the so-called diffusion distance. This distance is the expectation for the length of a random walk from one point to the other inside the manifold. To compute this expectation the transition probabilities need to be modelled, which is where the different versions inside the family of Diffusion Maps vary. The probabilities are computed using a density or kernel function. For these kernel functions, several different choices can be found in the literature, all of which have at least one parameter that needs to be calibrated. While this makes these approaches very powerful as they can adapt to many use cases, the customisation also poses a challenge: The appropriate choice of a kernel for an analysis task is not always clear and requires a certain degree of experience. Furthermore, these kernel functions also have parameters that need to be calibrated, which additionally makes this task more difficult. In [IT16] it was pointed out that in the analysis of simulation results there are usually not enough samples to calibrate these parameters automatically, so that the usage of these methods is only feasible for experts in the field of data analysis. Moreover, the property of PTU being linearly precise was not proven for Diffusion Maps and is in general not expected to hold [PHHV08], especially for non-uniformly sampled manifolds. Because of these complications, they were not used in this work.

In [DWHS16] and [Die19], a new regression method specifically designed for the analysis of simulation results was presented. Here, the post values which should be analysed are mapped to a regression shape for all simulation results individually. For example, the internal energy values for all shell elements of a longitudinal rail are mapped to curve running through the centre of the rail. The mapping result is then smoothened, and the similarity of these smoothened representation is computed by a scalar product and then converted to a dissimilarity by subtracting them from one. A strong advantage of this MDS method is that it is applicable to data with different geometric representations, as all data is mapped to a regression shape and only the regression shape needs to be topological unchanged. Unfortunately, this key feature prevents its usage in this work, since it cannot be reversed. It is thus rendering the method not generative. Thus, no equivalent for virtual simulations can be calculated. Finally, in [GIT16] and [IT16] a spectral transformation method was used that extracts certain features of the data by applying operators that are invariant under certain transformations. For example, the graph-based Laplace-Beltramy operator was applied to the nodes of the mesh and the respective values were used as new coordinates. These new coordinates are invariant under isometries, e.g. a rigid body motion of the complete part. Then, the pairwise distances in the new coordinates were computed and used for the final MDS step. These spectral methods can specifically extract these properties for which they are not invariant, such as strong local deformations or ruptures in the case of the Laplace-Beltramy operator and are very potent because of the invariances. But, if the desired property is not known beforehand, this benefit poses a problem, namely the uncertainty of which operator to choose. And more importantly, the invariance prevents it from being used in this work, since an approximation for low dimensional points is only possible for the operator coordinates, but not for the original data representation. For this representation the method is therefore not generative.
Summarising, these further methods from the class of MDS methods showed excellent performance for the analysis of simulation result, but unfortunately could not be used in this work, because they are not generative and require a level of interaction or expertise that was not feasible in the automated analysis of simulation results.

3.5 Nonlinear Mapping

The last class of DRMs featured in this work is the class of Nonlinear Mapping (NLM) methods.

3.5.1 Commonalities

This class of methods is named after Sammons Nonlinear Mapping, which was introduced in 1969 [Sam69] to find a mapping from a high dimensional to a low dimensional space utilising weighted Euclidean distances. The idea is to iteratively compute an embedding that minimises a stress function depending on the high and

low dimensional distances. In [LV07] it was pointed out, that in general any dissimilarity measure δ_{ij} in the high dimensional data space can be used, while still using Euclidean ones in the low dimensional space with the following stress function

$$E_{\text{NLM}} = \frac{1}{c} \sum_{i=1,j<i}^{s} \frac{(\delta_{ij} - \|x_i - x_j\|_2)^2}{\delta_{ij}} \tag{3.45}$$

where the normalisation constant c is defined as:

$$c := \sum_{i=1,j<i}^{s} \delta_{ij}$$

The division by the high dimensional dissimilarity provides a weighting to the approximations. It allows larger errors for longer distances while being relatively more precise on shorter distances in contrast to the MDS approaches, where all distances are weighted equally. Methods which compute low dimensional coordinates by minimizing this stress function are in this work referred to as NLM methods. Given a certain high dimensional distance the actual minimization is done as described in [LV07]: The low dimensional coordinates $x_i \in \mathbb{R}^d$ for all points could be initialised with any values, e.g. randomly or by performing MDS with the given distances. In this work, the latter one is done. The individual entries of the coordinates $1 \leq e \leq d$ are then iteratively updated for each point x_i according to the following "quasi-Newton optimisation" [LV07] of Eq. (3.46) with a hyper parameter $\alpha \in (0, 1] \subset \mathbb{R}$ to be chosen by the user [LV07].

$$x_{i,e} \leftarrow x_{i,e} - \alpha \frac{\frac{\partial E_{\text{NLM}}}{\partial x_{i,e}}}{\frac{\partial^2 E_{\text{NLM}}}{|\partial x_{i,e}^2|}} \tag{3.46}$$

As only the Euclidean norm is depending on the low dimensional coordinates, the derivatives can be analytically determined for each specific point x_i and its e-th entry, regardless of the type of high dimensional dissimilarity:

$$\begin{aligned}
\frac{\partial E_{\text{NLM}}}{\partial x_{i,e}} &= \frac{1}{c} \sum_{j \neq i} \frac{1}{\delta_{ij}} \frac{\partial \left(\delta_{ij} - \|x_i - x_j\|_2\right)^2}{\partial \|x_i - x_j\|_2} \frac{\partial \|x_i - x_j\|_2}{\partial x_{i,e}} \\
&= \frac{-2}{c} \sum_{j \neq i} \frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij}} \frac{\partial \|x_i - x_j\|_2}{\partial x_{i,e}} \\
&= \frac{-2}{c} \sum_{j \neq i} \frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij}} \frac{x_{i,e} - x_{j,e}}{\|x_i - x_j\|_2} \\
&= \frac{-2}{c} \sum_{j \neq i} \frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} (x_{i,e} - x_{j,e})
\end{aligned}$$

And for the second derivative the result is accordingly:

$$
\begin{aligned}
\frac{\partial^2 E_{\mathrm{NLM}}}{\partial x_{i,e}^2} &= \frac{\partial}{\partial x_{i,e}} \left(\frac{-2}{c} \sum_{j \neq i} \frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} \left(x_{i,e} - x_{j,e}\right) \right) \\
&= \frac{-2}{c} \sum_{j \neq i} \left(\frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} \frac{\partial \left(x_{i,e} - x_{j,e}\right)}{\partial x_{i,e}} + \frac{\partial \frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2}}{\partial x_{i,e}} \left(x_{i,e} - x_{j,e}\right) \right) \\
&= \frac{-2}{c} \sum_{j \neq i} \left(\frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} \right. \\
&\quad \left. + \frac{\frac{\partial (\delta_{ij} - \|x_i - x_j\|_2)}{\partial x_{i,e}} \delta_{ij} \|x_i - x_j\|_2 - \left(\delta_{ij} - \|x_i - x_j\|_2\right) \frac{\partial \delta_{ij} \|x_i - x_j\|_2}{\partial x_{i,e}}}{\delta_{ij}^2 \|x_i - x_j\|_2^2} \left(x_{i,e} - x_{j,e}\right) \right) \\
&= \frac{-2}{c} \sum_{j \neq i} \left(\frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} \right. \\
&\quad \left. + \frac{\|x_i - x_j\|_2 \left(-\left(x_{i,e} - x_{j,e}\right) + \left(x_{i,e} - x_{j,e}\right)\right) - \delta_{ij} \left(x_{i,e} - x_{j,e}\right)}{\delta_{ij} \|x_i - x_j\|_2^3} \left(x_{i,e} - x_{j,e}\right) \right) \\
&= \frac{-2}{c} \sum_{j \neq i} \left(\frac{\delta_{ij} - \|x_i - x_j\|_2}{\delta_{ij} \|x_i - x_j\|_2} - \frac{\left(x_{i,e} - x_{j,e}\right)^2}{\|x_i - x_j\|_2^3} \right)
\end{aligned}
$$

The update is repeated iteratively, until convergence is reached.

1: Compute all pairwise distances δ_{ij} in the original D-dimensional data space.
2: Initialise the d-dimensional coordinates by performing MDS with the given distances.
3: Compute the update of Eq. (3.46) all individual entries of all points.
4: Update low dimensional coordinates.
5: Return to step 3 until the value of the stress function no longer decreases.

Algorithm 3.7: NLM Algorithm. Modified from [LV07].

The above-described process is the same for all methods in this class, the only difference is the kind of dissimilarity used in the high dimensional space and how the pairwise values are computed. A different dissimilarity was also used for each of the MDS approaches introduced earlier. The remainder of this section introduces a corresponding NLM method for each of these MDS methods.

3.5.2 Euclidean Nonlinear Mapping

The first introduced NLM method in this work is the original approach that laid the foundations for this class of DRMs.

3.5.2.1 Base Method

The original approach introduced by Sammon [Sam69] uses the respective $\|\cdot\|_2$ norm on the original data as the high dimensional dissimilarity, which can be easily calculated for all pairs:

$$\delta_{ij} = \|y_i - y_j\|_2$$

All further steps are done according to the general introduction given above. To distinguish between the different approaches, it is referred to as Euclidean Nonlinear Mapping (ENLM) in this work because it is preserving the Euclidean distances. Trivially, this method can skip the computation and be directly applied to the pairwise distances if the data is not given as high dimensional coordinates but these distances instead, similar to other DRMs introduced before. If the target dimension d is large enough, classic metric MDS can preserve all pairwise distances without any error. Since the low dimensional coordinates are initialised with this MDS result, the iteration in Eq. (3.46) of the ENLM method does not yield any update and thus ends after the first iteration. The method only differs, if the target dimension d is smaller than the rank $r < s$ of the input data set Y.

3.5.2.2 Extension for Comparative Analysis

Apart from the target dimension, the method has three known options to influence the outcome. The first is the initialisation of the low dimensional coordinates that is fixed to the corresponding MDS result in this work. Thus, NLM is used as a post processing step improving the MDS result rather than computing a completely different solution. Random initialisations entail the risk of yielding undesired result, mainly in the form of local minima of the stress function E_{NLM}, which have a much higher value than the global minimum. Similarly, the MDS initialisation could also result in a local minimum, but since the initial values are already optimal for E_{MDS}, see Eq. (3.39), the stress function E_{NLM} is expected to be much lower than for random values.
The second option to influence the outcome is the hyper parameter α: Using the Quasi-Newton method NLM determines in each iteration for every point the steepest descent of the stress function, i.e. the direction in which the point should be moved to best decrease the value of the target function. The magic factor here is the step length, which is limiting the update in the said direction. Larger values can speed up the convergence process while also increasing the risk of instabilities as was outlined in [LV07]. In the same book, a value of $\alpha \in [0.3, 0.4] \subset \mathbb{R}$ is recommended as best practice, but for an arbitrary initialisation of the data. Here, a value of $\alpha = 0.25$ was fixed to decrease the number of parameters to be chosen by the user. It is smaller than the literature recommendation to make the results more reliable and less prone to instabilities. The slower convergence rate is acceptable since the initial low dimensional coordinates already yield a small value for the stress function.
The third option is the stopping criterion for the iteration or how to determine,

whether convergence was reached. Two criteria were applied here. First, the change of the stress function from one iteration to the next was calculated and the iteration was stopped if the absolute value was below a threshold of 5×10^{-9}. Second, the number of iterations was counted, and the method aborted when a total count of 3 000 was exceeded, since in the investigations for this work required at most a few hundred iterations even for data sets exceeding a thousand samples.

In addition to the known options, some extensions were needed for the analysis of simulation results. The original method has no equivalent of importance factors, but they were obtained in an additional post processing step. After the iteration had converged, classic metric MDS was applied to the low dimensional coordinates. This way, the coordinates were aligned with the most important directions and the importance factors for these corrected directions were also calculated.
For the calculation of virtual simulations, the modified LLI method described in Section 3.4.3.2 was used, in which large weights for far apart neighbours are penalised. This is motivated by the fact, that NLM approaches are relatively more precise on short distances, so that only those should be used for interpolation. But since the ENLM method does not have a neighbourhood size k, the total number of samples was used for this method, which is closer to the PCA approach, which can yield the same results if the intrinsic dimension is overestimated. This is especially important since this is usually the case in the analysis of simulation results.

3.5.3 Graph-Based Nonlinear Mapping

The second method in the class of NLM approaches is an alternative to the one in the last subsection by utilising a different high dimensional dissimilarity.

3.5.3.1 Base Method

This NLM variant aims to preserve geodesic distances and was introduced in [Yan04] as Geodesic Nonlinear Mapping. It uses the same graph approximation of Isomap for the unknown true geodesic distances as described in Section 3.4.3.1. To differentiate from the following method in the next subsection, which also approximates the geodesic distances, it is referred to as Graph-Based Nonlinear Mapping (GNLM) in this work.
In contrast to ENLM, which yields the same result as its MDS counterpart PCA if the target dimension is large enough, GNLM usually does not match Isomap's result because of the spurious geodesic curvature: The zigzags in the paths can turn the preservation of the distances in the Euclidean low dimensional space into an overdetermined system [LV07], which means that the stress function is minimised in a least squares sense. Hence, the additional weights of E_{NLM} can heavily affect the outcome. This is very desirable because the more edges a path contains, the more artificial curvature can be added through detours and therefore these longer distances should

not be preserved as precise as the shorter ones. Albeit this mitigates the problem of spurious geodesic curvature, it does not resolve it completely.

3.5.3.2 Extension for Comparative Analysis

The GNLM approach inherits all options to affect the performance of the DRM from the Isomap and the ENLM methods because the graph distances were calculated in the same way and the steps of the NLM approaches are the same. To increase the comparability all options were treated in the same way as in these two methods, resulting in the usage of the modified k-rule, see Section 3.4.3.2, as well as the initialisation with the Isomap result, hyper parameter $\alpha = 0.25$ and the two criteria definition of convergence, see Section 3.5.2.2.

The method was extended for the Comparative Analysis in the same way as ENLM by calculating importance factors via an additionally application of classic metric MDS on the iteration result as described in Section 3.5.2.2. The calculation of virtual simulations was also done using the LLI method with additional distance constraint, see Section 3.4.3.2, but here only the kNN were used for interpolation with the same k that was used for the graph distances.

3.5.4 Parallel Transport Nonlinear Mapping

The third method in the NLM class is the last DRM introduced in this work.

3.5.4.1 Base Method

As is pointed out in [LV07], any dissimilarity can be used in NLM and its stress function of Eq. (3.45) for the high dimensional distances. The approach to use an approximation of the geodesic distances has shown good results [LV07] but was hindered by the quality of the approximation by graph distances. With the introduction of the parallel transport distances in [BYF+19] a better approximation is available, but so far it was not combined with the NLM approach. Thus, this combination is newly introduced in this work as Parallel Transport Nonlinear Mapping (PTNLM), closing the gap in the existing methods.
The combination of parallel transport distances as described in Section 3.4.4.1, and the weighted stress function of NLM in Section 3.5 harmonises well, since parallel transport distances lose accuracy for paths consisting of many edges, and these tend to be longer distances [BYF+19], whose weight is relatively small, see Eq. (3.45).
Furthermore, if PTU can preserve all pairwise distances without any errors, PTNLM will return the same result, as the iterations in Eq. (3.46) will not yield any update. This is especially interesting since PTU is linearly precise and it returns the exact solution on linear manifolds, if the tangent spaces are estimated correctly. Thus, PTNLM inherits this property of being linearly precise from the PTU approach under the same circumstances.

3.5.4.2 Extension for Comparative Analysis

Similarly, PTNLM inherits all options to influence the outcome from both PTU and ENLM. As for all methods before, the options are treated in the same manner to increase comparability, i.e. the neighbourhood building and tangent space estimation were done as described in Section 3.4.4.2, and the NLM configuration is the same as for ENLM, see Section 3.5.2.2.

The importance factors for the Comparative Analysis were once again deducted by performing the classic metric MDS on the iteration result. But in contrast to the other two NLM methods that used the LLI method, the creation of virtual simulations was done according to the LAI interpolation described in Section 3.3.3.2 and Section 3.4.4.2. Similar to PTU, PTNLM is defining the distances in terms of paths in local tangent spaces, only the preservation of these distances is weighted differently in the NLM variant. As the distances to the kNN is usually small compared to the other points and shorter distances are better preserved than longer ones, PTNLM also preserves local tangent spaces relatively well. This is why the creation of virtual simulations is done with the tangent space-based LAI.
With the last addition, all methods in the class of NLM approaches are ready for the Comparative Analysis of simulation results.

3.6 Recapitulation

The last sections introduced three classes of DRMs with multiple different methods in each class. In order to better understand the commonalities and differences, the main features are briefly summarised before continuing with further theory in the next chapter.
Approaches that belong to the LM category compute properties of local neighbourhoods aggregate a global alignment matrix and compute a low dimensional embedding by calculating the eigenvectors corresponding to the smallest eigenvalues of this alignment matrix. This matrix is usually relatively sparse, since only entries for nearest neighbours are non-zero. Thus, local properties are better preserved than global structures, since only those are considered.
MDS approaches determine pairwise distances for all points and compute a low dimensional embedding by calculating the eigenvectors corresponding to the largest eigenvalues of the squared and double centred dissimilarity matrix. This matrix is usually dense, since all pairwise distances are computed. The Euclidean distances in the resulting low dimensional embedding match the high dimensional dissimilarities in a least squares sense. This means that longer distances are relatively better preserved than shorter ones, unravelling the global structures while sometimes neglecting local properties.
NLM methods compute the embedding in an iterative process, where the initial low dimensional node positions are obtained as an MDS result and then improved by minimising a stress function with a Quasi-Newton optimization. Methods in this class

can be viewed as a compromise: They are similar to MDS methods in the sense that they try to preserve pairwise distances. But in contrast to the previous class, the distances are weighted according to their size, allowing to be relatively more precise on smaller distances while still capturing the global structures.
The multiple methods within each class differ in the property or distance they aim to preserve. An overview of these properties can be found in Tab. 3.3. As stated before, the PCA approach is equivalent to classic metric MDS and thus an MDS method. Furthermore, it is the only linear method covered in this work, all other approaches are nonlinear.

Name	Class	Preserved Property		Preserved Distance		
		Local Weights	Tangent Space	Euclidean	Graph	Parallel Transport
PCA	MDS			✓		
LLE	LM	✓				
LTSA	LM		✓			
MLLE	LM	✓				
Isomap	MDS				✓	
PTU	MDS		✓			✓
ENLM	NLM			✓		
GNLM	NLM				✓	
PTNLM	NLM		✓			✓

Table 3.3: Overview of the DRMs covered in this work, their classes and what they aim to preserve in the low dimensional embedding.

All the introduced nonlinear methods were modified or extended for the usage in the analysis of simulation results. For some, this means that only intrinsic parameters are calibrated, such as the neighbourhood rule for Isomap. But, for others, fundamental changes were made, such as the scaled aggregation of the alignment for LTSA or the adaptive tangent space dimension in PTU. Thus, the performance may differ from the base approaches in the respective literature.
To be used in a Comparative Analysis, the nonlinear methods needed the addition of importance factors and virtual simulations. For LM approaches, importance factors were calculated by the scaling method described in Section 3.3.2.2, which aims to make the embeddings locally distance preserving. For all other methods, these were obtained from an EVD decomposition of a squared and double centred distance matrix of the MDS step, either because it was an MDS method or because classic metric MDS was used on the low dimensional coordinates as a post processing step.
The virtual simulations were either computed by the LAI method of Section 3.3.3.2 for tangent-based DRMs, namely LTSA, PTU and PTNLM, or via the LLI approach for all other methods. Here the version with an additional distance penalty described in Section 3.4.3.2 was used for Isomap, ENLM and GNLM.

4 Difference Dimensionality Reduction

In the last section several DRMs and their base models have been introduced. The concept of importance factors to determine the number of effects and their function in the low dimensional representation of the data was explained. This section elaborates how the correlation of effects on different parts, post values or states can be investigated utilising these low dimensional coordinates and their importance factors. This process of Difference Dimensionality Reduction (DDR) is introduced for the linear case first because a difference operation is already existing in literature [TM10], [BBT13]. Afterwards, this linear difference operation is generalised in order to transfer the methodology to nonlinear methods. Two specific implementations of this general difference approach are derived, motivated by the DRMs introduced in the last section. Both methods can be improved by a normalisation enhancement which is described subsequently. This section is concluded with a brief summary of the properties of the methods and the DRMs to which they can be applied. This summary highlights the distinction between the new approaches and the original process.

4.1 Difference Principal Component Analysis

The process of Difference Principal Component Analysis (DPCA) was first published in 2010 by Clemens-August Thole, Igor Nikitin, Lialia Nikitina and Tanja Clees [TNNC10] and is a linear method to investigate the dependence between two data sets. The DPCA is implemented in the commercial software DIFFCRASH and is filed as a patent [TM10].

4.1.1 Basic Definitions

The DPCA's base idea is to determine the correlation of underlying effects of one data set with the other data set in a two-step approach.
The first data set $Y \in \mathbb{R}^{D\times s}$ of s samples with an initial large dimension of D is referred to as the source. Under the assumption of a linear generating function, it is in a first step decomposed into its underlying d effects by the truncated SVD in the linear PCA approach, see also Eq. (3.6):

$$Y \approx Y_d := U|_d \Sigma|_d (V|_d)^\mathsf{T} \quad \text{, with } d < s$$

Each row of the right matrix $V|_d^\mathsf{T} \in \mathbb{R}^{d\times s}$ is associated with an effect or mode, as described in Section 3.2. This means that an effect is defined by the samples in which it is manifesting through the weights of the linear combination given by the corresponding row.
In the second step, the relation to an additional target data set is investigated.

Definition 4.1 (Source and target data set)
Given a so-called source data set $Y \in \mathbb{R}^{D\times s}$, an additional high dimensional data set $\mathcal{Y} \in \mathbb{R}^{\mathcal{D}\times s}$

$$\mathcal{Y} =: (\mathcal{y}_1 \ \dots \ \mathcal{y}_s) \ , \ \textit{with } \mathcal{y}_i \in \mathbb{R}^{\mathcal{D}} \ \forall \ i \in \{1, \dots, s\}$$

with possibly different dimension $\mathcal{D} \neq D$ but identical number of samples s is called the target of the difference operation.

Specifically, the correlation to the first $1 \leq e \leq d$ effects of the source are determined by "subtracting" [TNNC10] them from the target. This is done by comparing the eigenvalues of the original Gram matrix $G_{\mathcal{Y}} \in \mathbb{R}^{s\times s}$ of the target data set with the positive eigenvalues of the τ-modified Gram matrix $G_{\mathcal{Y},V|_e,\tau}$.

Definition 4.2 (τ-modified Gram matrix)
Given a weight factor $\tau \in \mathbb{R}$, a diagonal matrix $\Sigma|_e$, a matrix $V|_d^{\mathsf{T}} \in \mathbb{R}^{d\times s}$ with orthonormal columns and a Gram matrix $G_{\mathcal{Y}} \in \mathbb{R}^{s\times s}$ with

$$G_{\mathcal{Y}} = \frac{1}{\mathcal{D}} \mathcal{Y}^{\mathsf{T}} \mathcal{Y}$$

the τ-modified Gram matrix $G_{\mathcal{Y},V|_e,\tau} \in \mathbb{R}^{s\times s}$ is defined as:

$$G_{\mathcal{Y},V|_e,\tau} \ := \ G_{\mathcal{Y}} - \tau V|_e \Sigma|_e V|_e^{\mathsf{T}} \ \overset{V \textit{ orth.}}{=} \ G_{\mathcal{Y}} - \tau \left(\sum_{i=1}^{e} \sigma_i v_i v_i^{\mathsf{T}} \right) \tag{4.1}$$

Here, $V|_e$ and $\Sigma|_e$ are both obtained from the truncated SVD, where the number of effects e can be chosen by the analyst, usually under consideration of the importance factors. A lower bound for the weight factor τ is later given in Eq. (4.5) of Section 4.1.3, though it is usually overestimated in practical applications.
The eigenvalues of these two matrices are used to quantify the correlation by computing and evaluating two difference measures.

Definition 4.3 (Difference measures)
Given two symmetric matrices $G_{\mathcal{Y}}, G_{\mathcal{Y},V|_e,\tau} \in \mathbb{R}^{s\times s}$ and their EVDs

$$G_{\mathcal{Y}} =: \mathcal{V} \ \mathrm{diag}\,(\mu_1, ..., \mu_s)\, \mathcal{V}^{\mathsf{T}}$$
$$G_{\mathcal{Y},V|_e,\tau} =: W \mathrm{diag}\,(\lambda_1, ..., \lambda_s)\, W^{\mathsf{T}}$$

with corresponding eigenvalues μ_i and λ_i in descending order, the functions

$$\delta_{\text{spec}}\left(G_{\mathcal{Y}}, G_{\mathcal{Y},V|_e,\tau}\right) := 1 - \frac{\sqrt{\max\{\lambda_1, 0\}}}{\sqrt{\mu_1}} \tag{4.2}$$

$$\delta_{\text{var}}\left(G_{\mathcal{Y}}, G_{\mathcal{Y},V|_e,\tau}\right) := 1 - \frac{\sum_{i=1}^{s} \max\{\lambda_i, 0\}}{\sum_{i=1}^{s} \mu_i} \tag{4.3}$$

are called difference measures.

The first difference measure δ_{spec} quantifies the relative reduction in terms of the spectral norm of the input data set $\|\mathcal{Y}\|_2 = \sqrt{\mu_1}$. This norm is an indicator for the maximum scatter of the data set because of its geometric interpretation as the maximum stretch factor: Any vector multiplied by this matrix can be elongated at most by this factor. Furthermore, the square root of the first eigenvalue is also the largest importance factor in the linear case.
While the first difference measure is already used in several publications, for example, [TNNC10] or [BST15], the second difference measure was newly introduced in this work. As described in Section 3.2.2, the eigenvalues contain the information about the variance of the data set and their sum is the total variance. The second difference measure δ_{var} thus quantifies the relative reduction in terms of total variance of $\mathcal{Y}$. Together, these two measures express which portions of the second data set are correlated to the e effects of the first data set.

The core assumption is that the data was generated with a linear function and the SVD is returning the correct underlying modes as well as number of effects d. In the case of a nonlinear generating function, the number of effects is usually overestimated, and the modes can represent a fraction or a mixture of underlying effects, which is demonstrated later in Section 5.1.3.

4.1.2 Application to Simulation Results

The DPCA was developed specifically for the analysis of simulation results. The source of the operation is always an extracted post value on a part at a number of states in the results. Here, a part is an arbitrary subset of nodes or elements as introduced in Section 3.2.3. It could, for example, be the displacements of all nodes in a group at one or more states. This group may consist of several PIDs or a user-defined set at crucial states. Alternatively, it could also be the internal energy of the elements in a vital area.
To increase the comparability among different parts and to decrease the sensitivity to the size of the part the Gram matrices are generally normalised through division by the square root of the number of values used in the underlying vector-vector multiplications. This number of values is usually the number of elements multiplied with the number of states, though more complex combinations are theoretically possible.
The target is also a post value on another part and possibly a different state of the simulation results. In the commercial version, the difference operation was limited to using the same post value on source and target, though this is purely for implementation reasons and no theoretical requirement. As findings from this work have been integrated into the product version, this limitation has been removed.
One important requirement for the DPCA to yield meaningful results is that the samples are extracted from the same set of simulations in identical order as the source: Since effects are defined as linear combinations of these samples, they can be evaluated on all parts, states, and post values. If the samples and their order is fixed

the correlation between those entities can be determined and "subtracted" by the difference operation.
The challenging part of the analysis is the selection of these different parts, post values, active states, and the correct number of effects for the target as well as for the source. The DPCA method was designed to aid experts like simulation engineers in their work, which is the reason, why the target part is usually easy to find. It is commonly one of the crucial parts for the simulation discipline of the user, which, for example, may be an area of a car's load bearing structures that shows variation in the displacements or the internal energy in a set of occupant safety simulations. When the target part and value are identified, the states for the analysis need to be chosen.
Although theoretically a set of multiple states is possible, the analysis is usually conducted for a single state. This is mostly due to the fact that the analyst wants to identify isolated effects. If multiple states are considered in the analysis, multiple effects can be merged and obfuscate the underlying structure. The importance factors can be consulted to determine the correct states.
In Fig. 4.1 the development of the first three linear importance factors over time is visualised. This development can be computed by performing the DR for each state and compute the importance factors for low dimensions or modes for every state.

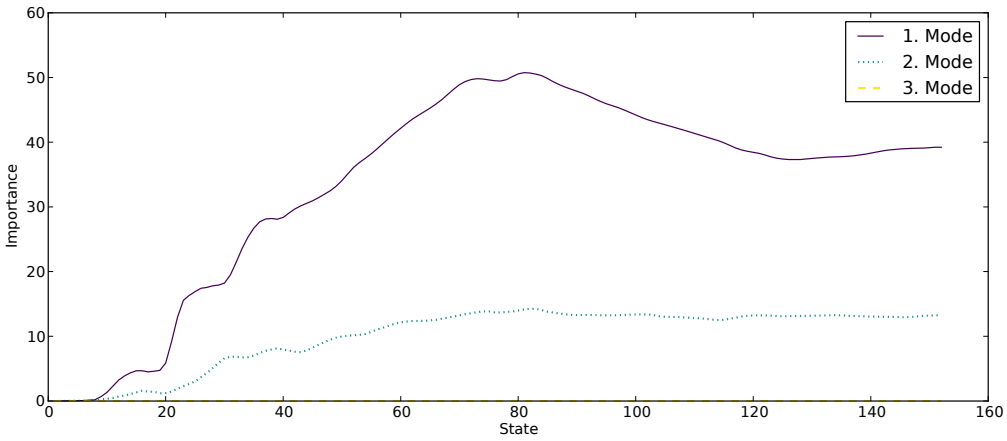


Figure 4.1: Example of the importance factors over time for the Silverado longitudinal rails example featured in Chapter 3. The curves represent the importance of the first, second and third coordinate direction in the different states. As this example only contains three simulation results, the third curve is constantly equal to zero.

Such development curves can be used to identify possible candidates for the analysis states. A change in the behaviour of the model can usually be found in local maxima or steep increases of these curves. When searching for a suitable state for the target of the analysis, the analyst should pick a state that is some time after the start of the variation, but still after it had time to manifest. Possible states for the first mode,

represented by the continuous line in Fig. 4.1, would be at around states 60 to 67 as they are at a point of the curve with large importance factors, but before the two peaks at states 70 to 85. Selecting states at local peaks is usually not recommended because the decreasing gradient indicates the starting of one or more stabilising effects.
Possible states for source parts of the analysis should trivially be picked before the target state. Beyond that, however, it is recommended to select a state shortly before a peak or further increase of the importance factor. Possible states for the second mode, represented by the dotted line in Fig. 4.1, would be around states 25 to 27 as they are at the beginning of the curve, but with a significant size at this time and before the local maximum at around state 30.
Once the source and target with corresponding states are identified, the actual difference operation itself is performed.

4.1.3 Connection to Orthogonal Projection

The difference operation in the DPCA approach is closely connected to the projection onto the orthogonal complement of a set of vectors.

Definition 4.4 (Orthogonal complement projection)
Given a matrix $V|_e \in \mathbb{R}^{s\times e}$*, the symmetric matrix* $P \in \mathbb{R}^{s\times s}$

$$P := I_s - V|_e V|_e^{\mathsf{T}}$$

is called the projection onto the orthogonal complement of $V|_e$*., with* $P^2 = P$ *and* $PV|_e = \mathbb{0}$*.*

Since this matrix P is a projection, it holds that its eigenvalues are always either 0 or 1, and since it is an orthogonal projection, it also holds that $\mathrm{im}(P) \perp \ker(P)$ [LM12]. Multiplying the Gram matrix $G_{\mathcal{Y}}$ from both sides with this projection P is the same as projecting the data set first and calculating the Gram matrix afterwards:

$$PG_{\mathcal{Y}}P = \frac{1}{\mathcal{D}}P^{\mathsf{T}}\mathcal{Y}^{\mathsf{T}}\mathcal{Y}P = \frac{1}{\mathcal{D}}\left(\mathcal{Y}P\right)^{\mathsf{T}}\left(\mathcal{Y}P\right) = G_{\mathcal{Y}P} \tag{4.4}$$

The relation of the Gram matrix $G_{\mathcal{Y}P}$ for the projected data set and the modified matrix $G_{\mathcal{Y},V|_e,\tau}$ of the DPCA is formulated by the following theorem.

Theorem 4.1
Under the assumption that τ *is chosen such that*

$$\tau \geq \frac{\|G_{\mathcal{Y}}\|_2}{\sigma_e} \in \mathbb{R} \tag{4.5}$$

holds, the following relation is existing between the projected Gram matrix of Eq. (4.4) and the result of the difference operation in Eq. (4.1):
Eigenvectors for positive eigenvalues of $G_{\mathcal{Y}P}$ *can only be eigenvectors with positive eigenvalues of* $G_{\mathcal{Y},V|_e,\tau}$ *and eigenvectors for zero eigenvalues of* $G_{\mathcal{Y}P}$ *can only be eigenvectors with non-positive eigenvalues of* $G_{\mathcal{Y},V|_e,\tau}$*.*

Proof. Let $w \in \mathbb{R}^s$ be an eigenvector of $G_{\mathcal{Y}P}$ such that $G_{\mathcal{Y}P}w = \eta w$ and $w^\mathsf{T}w = 1$. Since $G_{\mathcal{Y}P}$ is a Gram matrix, it is symmetric and positive semi-definite. Thus, its eigenvectors w form an orthonormal basis of $\mathbb{R}^s$. Furthermore with $1 \leq i \leq e$ it follows that

$$G_{\mathcal{Y}P}\ v_i = P^\mathsf{T}G_{\mathcal{Y}}P\,v_i = P^\mathsf{T}G_{\mathcal{Y}}\,\mathbb{0}_s = 0 \cdot v_i$$

and thus, that all $v_1, ..., v_e$ are eigenvectors of $G_{\mathcal{Y}P}$ with eigenvalues 0. Therefore, if $w \in \mathbb{R}^s$ is an eigenvector of $G_{\mathcal{Y}P}$, it holds that either $Pw = \mathbb{0}_s$ or $Pw = w$, because of $\mathrm{im}(P) \perp \ker(P)$.
Since the matrix is positive semi-definite, there are two possibilities for its eigenvalues η. In the first case of $\eta > 0 \in \mathbb{R}$, it holds that $Pw = w$ and $v_i^\mathsf{T}w = 0\ \forall 1 \leq i \leq e$ and thus the product of w with $G_{\mathcal{Y},V|_e,\tau}$ is

$$\begin{aligned} w^\mathsf{T}G_{\mathcal{Y},V|_e,\tau}w &= w^\mathsf{T}\left(G_{\mathcal{Y}} - \tau V|_e\Sigma|_eV|_e^\mathsf{T}\right)w \\ &= w^\mathsf{T}G_{\mathcal{Y}}w - \tau w^\mathsf{T}V|_e\Sigma|_eV|_e^\mathsf{T}w \\ &= w^\mathsf{T}P^\mathsf{T}G_{\mathcal{Y}}Pw - \tau\,\mathbb{0}_e^\mathsf{T}\Sigma|_e\mathbb{0}_e \\ &= \eta \end{aligned}$$

meaning that if w is an eigenvector of the τ-modified Gram matrix the associated eigenvalue would be η as well.
In the second case, where $\eta = 0 \in \mathbb{R}$ there are two subcases: In the first subcase were $Pw = \mathbb{0}_s$, it holds that $V|_eV|_e^\mathsf{T}w = w$ and that the scalar product can be estimated as:

$$\begin{aligned} w^\mathsf{T}G_{\mathcal{Y},V|_e,\tau}w &= w^\mathsf{T}\left(G_{\mathcal{Y}} - \tau V|_e\Sigma|_eV|_e^\mathsf{T}\right)w \\ &= w^\mathsf{T}G_{\mathcal{Y}}w - \tau\sum_{i=1}^{e}\sigma_i w^\mathsf{T}v_iv_i^\mathsf{T}w \\ &\leq \mu_1 - \tau\sigma_e\sum_{i=1}^{e}w^\mathsf{T}v_iv_i^\mathsf{T}w \\ &= \mu_1 - \tau\sigma_e \\ &\overset{4.5}{\leq} \mu_1 - \frac{\|G_{\mathcal{Y}}\|_2}{\sigma_e}\sigma_e \\ &= 0 \end{aligned}$$

For the second subcase where $Pw = w$ the following holds

$$\begin{aligned} w^\mathsf{T}G_{\mathcal{Y},V|_e,\tau}w &= w^\mathsf{T}\left(G_{\mathcal{Y}} - \tau V|_e\Sigma|_eV|_e^\mathsf{T}\right)w \\ &= w^\mathsf{T}G_{\mathcal{Y}}w - \tau w^\mathsf{T}V|_e\Sigma|_e\mathbb{0}_e \\ &= w^\mathsf{T}P^\mathsf{T}G_{\mathcal{Y}}Pw \\ &= 0 \end{aligned}$$

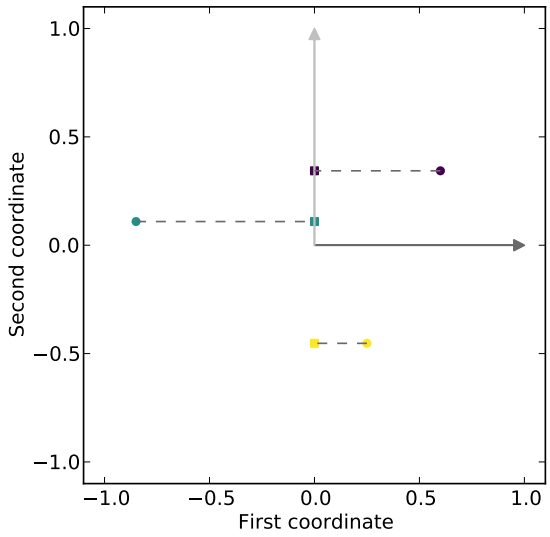
With the last case complete it is proven that eigenvectors for positive eigenvalues of $G_{\mathcal{Y}P}$ can only be eigenvectors with positive eigenvalues of $G_{\mathcal{Y},V|_e,\tau}$ and eigenvectors

for zero eigenvalues of $G_{\mathcal{Y}P}$ can only be eigenvectors with non-positive eigenvalues of $G_{\mathcal{Y},V|_e,\tau}$.

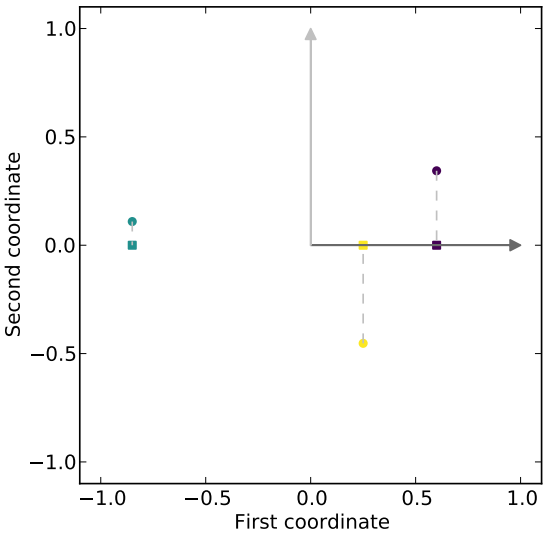
□

Because of this relation an EVD restricted to positive eigenvalues will yield the same result on both matrices $G_{\mathcal{Y},V|_e,\tau}$ and $G_{\mathcal{Y}P}$, if τ satisfies the lower bound given in Eq. (4.5). Since the difference measures δ_{spec} and δ_{var} are discarding eigenvalues $\lambda_i < 0$, see Eq. (4.2) and Eq. (4.3), they would be the same for the τ-modified Gram matrix as well as for the projected one.
This means that the DPCA is eliminating those parts of the second data set, which are correlated with vectors associated with the effects of the first data set. Hence only the uncorrelated portions of information remain in the target data set. The elimination is achieved by projection along the vectors onto the span of their orthogonal complement, but here all vectors are expressed as linear combinations of the second data set, see Fig. 4.2 for a visual example.
There are some benefits and drawbacks in performing the weight-based Gram matrix modification over the orthogonal projection. The first benefit of the modification is that only symmetric matrices are involved in each step. This allows the operation to be carried out very efficiently, both in terms of memory requirements and the number of instructions, as only vector-vector multiplications and no matrix-matrix multiplications are required. A second advantage is that for a fixed τ subtracting events with eigenvalues close to 0 does not yield a big impact. This prevents the user from subtracting unimportant effects that can represent noise in the data.
For the orthogonal projection, caution must be taken not to subtract too many effects, which can be emphasised with a simple example: Because the coefficient matrix V is orthogonal, subtracting all s effects would eliminate the complete data set in any case, whereas the τ modification only considers eigenvectors referring to non-zero eigenvalues. In practice for the analysis of node displacements, a default value of $\tau = 10\,000$ is usually a good choice and is used by many analysts as it is the default value used by DIFFCRASH software.
But this weight factor of τ can also be a major disadvantage: Since it depends on the fraction of two eigenvalues of different data sets, it can lead to instabilities. This could be the case, if one data set shows large deviations and another one only small differences. If the eigenvalues in the numerator and denominator are very different, τ needs to be chosen very large and the default value might be too small. However, if it is chosen too large, summing up all terms in Eq. (4.1) on a computer with limited precision will basically overwrite the content of $G_{\mathcal{Y}}$. This problem is not very susceptible to the sizes of the underlying parts, since the input data is divided by this number, see Section 3.2.3, but it can be very critical, especially if different post values are involved.
In this work, the orthogonal projection is used since the investigated sets rarely exceed or even reach thousands of simulations, which is nowadays not a problem in terms of matrix-to-matrix multiplications, so the first benefit is not necessary. More-

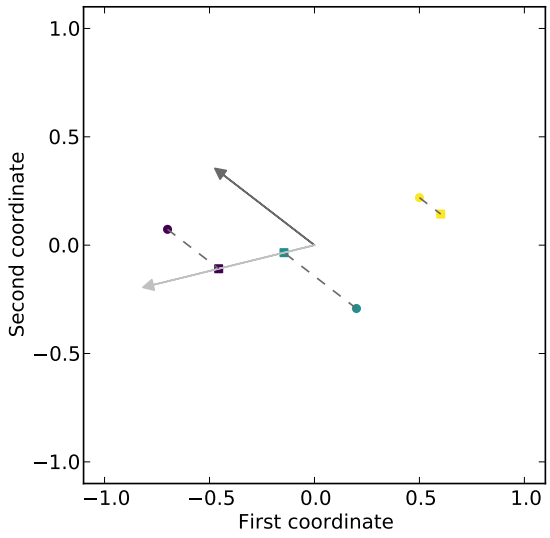
over, the correlation between different post values will be investigated, which was not done in earlier work and poses a challenge in the choice of the correct parameter τ. Additionally, the number of effects was carefully chosen to be small enough so that not noise effects are subtracted.
A graphical example of the orthogonal projection and how it is affecting the data is given in Fig. 4.2. The first row shows the effects on source data set X and the second row shows the impact onto a target data set $\mathcal{Y}$. As the modes are defined through linear combinations v_1, v_2 of the sample points, the source modes can also be expressed as a linear combination of the target data. The difference operation $\left(I_s - v_1 v_1^{\mathsf{T}}\right)$ is projecting the data points along the original modes onto the span of the complement vector, which may no longer be orthogonal on the target data set $\mathcal{Y}$.



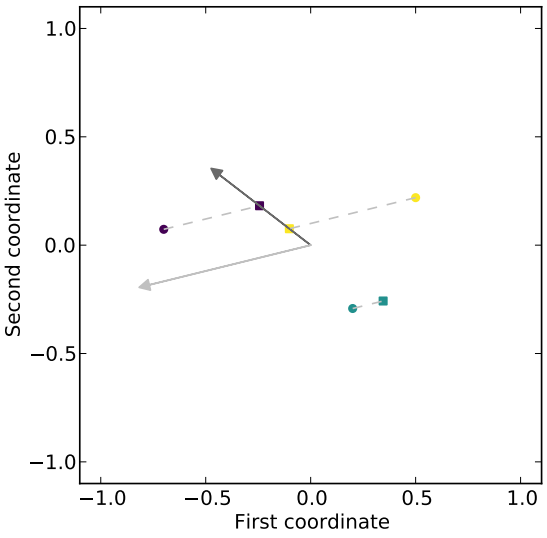
(a) First mode $X\left(I_s - v_1 v_1^{\mathsf{T}}\right)$ on source X



(b) Second mode $X\left(I_s - v_2 v_2^{\mathsf{T}}\right)$ on source X



(c) First mode $\mathcal{Y}\left(I_s - v_1 v_1^{\mathsf{T}}\right)$ on target $\mathcal{Y}$



(d) Second mode $\mathcal{Y}\left(I_s - v_2 v_2^{\mathsf{T}}\right)$ on target $\mathcal{Y}$

Figure 4.2: Visual example of the DPCA operation: The arrows represent the modes of the source but expressed as linear combinations of the sample points. The circles are the original data points, and the squares are the images of the projection onto the orthogonal complement of the respective mode, which is defining the difference operation.

4.2 Generalised Difference Dimensionality Reduction

As described in the last section, the aim of the DPCA method is to remove those portions of a data set that are correlated with e effects that are defined as a linear combination of samples. These effects were determined utilising the low dimensional embedding X obtained by a specific DRM. In this section, this concept is firstly generalised in order to secondly extend it to other DR approaches.
Given a source data set $Y \in \mathbb{R}^{D\times s}$ and its low dimensional embedding $X \in \mathbb{R}^{d\times s}$ such that:

$$\begin{aligned} Y &= F(X) \\ \text{with } X &= \begin{pmatrix} X|_e \\ X|_{d-e} \end{pmatrix} \end{aligned}$$

It is assumed without loss of generality that the first e effects are to be subtracted.

Definition 4.5 (Generalised Difference Dimensionality Reduction)
Given a low dimensional embedding $X \in \mathbb{R}^{d\times s}$, the Generalised Difference Dimensionality Reduction of a target data set $\mathcal{Y} \in \mathbb{R}^{\mathcal{D}\times s}$ is defined as the matrix multiplication

$$\Delta_{\mathrm{DDRM}}(\mathcal{Y}) := \mathcal{Y}(I_s - M)$$

with a suitable mode matrix $M \in \mathbb{R}^{s\times s}$.

Definition 4.6 (Ideal mode matrix)
An ideal mode matrix $M \in \mathbb{R}^{s\times s}$ for the data set $X \in \mathbb{R}^{d\times s}$ and the generating function $F : \mathbb{R}^{d\times s} \to \mathbb{R}^D \times s$ is a matrix with the following properties:

$$\begin{pmatrix} \mathbb{0}_e \\ X|_{d-e} \end{pmatrix} = X(I_s - M) \tag{4.6}$$

$$F(X(I_s - M)) = F(X)(I_s - M) \tag{4.7}$$

The property in Eq. (4.6) means that the low dimensional coordinates associated with the underlying e effects are eliminated from the source data set and thus the corresponding rows are equal to zero after the operation. The second property shown in Eq. (4.7) is demanding that the operation defined by right multiplication of $(I_s - M)$ should be invariant under the function $F(\cdot)$ for the given intrinsic coordinates $X \in \mathbb{R}^{d\times s}$. This way, the difference operation can be performed either in the low or in the high dimensional representation and the subtraction of effects can be evaluated at the complete samples, regardless of whether the underlying effects were derived from a subset or on the complete data. If such a matrix M is found, the difference operation can be evaluated on a new target data set $\mathcal{Y}$ by computing the difference measures δ_{spec} and δ_{var} as introduced in Eq. (4.2) and Eq. (4.3) for the EVDs of the original Gram matrix $G_{\mathcal{Y}}$ and the Gram matrix for the data set modified by the right-hand multiplication with the mode matrix $G_{\mathcal{Y}(I_s-M)}$.

In practise, the exact intrinsic coordinates X as well as the true generating function $F(\cdot)$ are usually unknown and only the approximations $\widetilde{X}$ and $\widetilde{F}(\cdot)$ computed by a DRM are available. Trivially both, the computed data representation and the function, are depending on the DRM used to obtain this embedding. Hence the resulting mode matrix M must be tailored to the utilised method and the assumptions on its generating function $\widetilde{F}(\cdot)$.
For the PCA or classic metric MDS approach, the underlying data model is linear, meaning that the function can be rewritten as a matrix multiplication from the left. The approximations for the function and the d-dimensional coordinates are obtained by a constrained SVD as described in Section 3.2

$$\begin{aligned} Y &= \widetilde{F}_{\mathrm{PCA}}\big(\widetilde{X}\big) \\ &:= U|_d \cdot \Sigma|_d V|_d^{\mathsf{T}} \end{aligned}$$

with $\widetilde{F}_{\mathrm{PCA}}(\cdot) = U|_d \cdot$ and $\widetilde{X} = \Sigma|_d V|_d^{\mathsf{T}}$. The linear model ensures that the any matrix multiplication from the right hand is in fact invariant under the function $\widetilde{F}_{\mathrm{PCA}}$ satisfying the second property by design for all data sets $X \in \mathbb{R}^{d\times s}$. One suitable difference matrix is $M_{\mathrm{DPCA}} = V|_e V|_e^{\mathsf{T}}$. The compliance with the first property can be easily shown by performing the multiplication:

$$\begin{aligned} X\left(I_s - M_{\mathrm{DPCA}}\right) &= X\left(I_s - V|_e V|_e^{\mathsf{T}}\right) \\ &= \Sigma|_d V|_d^{\mathsf{T}}\left(I_s - V|_e V|_e^{\mathsf{T}}\right) \\ &= \Sigma|_d \begin{pmatrix} V|_e^{\mathsf{T}} \\ V|_{d-e}^{\mathsf{T}} \end{pmatrix}\left(I_s - V|_e V|_e^{\mathsf{T}}\right) \\ &= \begin{pmatrix} \mathbb{0}_e \\ X|_{d-e} \end{pmatrix} \end{aligned}$$

The matrix $(I_s - M_{\mathrm{DPCA}})$ is the orthogonal projection introduced as P in Section 4.1.3.
However, in practical application to real data while using nonlinear DRMs, this approach contains several challenges. First, the existence of a matrix M, such that both properties in Eq. (4.6) and Eq. (4.7) hold, is not guaranteed: Since $M \in \mathbb{R}^{s\times s}$, the number of variables is s^2, while the total number of equations is $(d + D)\, s$. This could yield an overdetermined system of equations with a required invariance for an arbitrary, possibly nonlinear function $F(\cdot)$ and any data set X. Hence, the second property is relaxed to:

$$\widetilde{F}\big(\widetilde{X}\left(I_s - M\right)\big) \approx \widetilde{F}\big(\widetilde{X}\big)\left(I_s - M\right) \tag{4.8}$$

This means that the invariance needs to hold only for the given data set, the specific embedding $\widetilde{X}$ calculated by the DRM and only approximately instead of exactly.
After relaxing the property, the next problem arises that M is often not unique, if no further assumptions or constraints are active: Since the second set of equations given

in Eq. (4.7) was removed, only the first set of Eq. (4.8) remains. With $\widetilde{X} \in \mathbb{R}^{d \times s}$, the number of given equations is $d \cdot s$, with $d < s$, but the number of variables is s^2, because $M \in \mathbb{R}^{s \times s}$, yielding an underdetermined system of linear equations. While this ambiguity poses a problem from an algorithmic point of view, it is also a chance to determine M in such a way that the resulting operation has several beneficial properties. For example, choosing the matrix M_{DPCA} for the linear approach yields the orthogonal projection P as stated before. As explained earlier in Section 4.1.3, this operation has the beneficial property of being a projection, which means that $P^2 = P$ holds. Thus, applying the identical difference operation multiple times is the same as applying it once, which is helpful from an analysis point of view because it is less prone to usage errors, for example. Furthermore, the eigenvalues of a projection are always either 0 or 1 and since P is an orthogonal projection, it also holds that $\mathrm{im}(P) \perp \ker(P)$ [LM12]. With these two traits it can be shown that $\|G_{\mathcal{Y}(I_s - M)}\|_2 \leq \|G_{\mathcal{Y}}\|_2$ and $\|G_{\mathcal{Y}(I_s - M)}\|_F \leq \|G_{\mathcal{Y}}\|_F$ for any matrix $\mathcal{Y}$ [Gal13]. This imposes the beneficial property that subtracting effects from a data set cannot increase the variance in that data set and thus the difference measures δ_{spec} and δ_{var} are non-negative and smaller than one.
While the matrix M_{DPCA} is well defined for the linear approach and has the two beneficial properties mentioned before, computing the equivalent for the nonlinear methods is more challenging and does generally not yield these beneficial properties. A difference method utilising a matrix M that is complying with the properties described in Eq. (4.6) and Eq. (4.8) is called a Generalised Difference Dimensionality Reduction Method (DDRM). The goal of this section is to construct such matrices for the nonlinear approaches introduced earlier that have similar, though not as strong beneficial properties as the linear approaches.

The linear DPCA approach describes the mode that should be eliminated as a linear combination of samples, which specifies the complete axis to be subtracted, see Fig. 4.2.
For nonlinear methods, this cannot be achieved for the full axis since it is nonlinear and only a finite number of samples is given. The meaning of said axis can only be interpreted at the available sample points and may change across the low dimensional data space. For the nonlinear variants, the operation is broken down into how it is affecting the individual points and defined by these responses instead. These responses can be formulated as an update that leads from an origin to an image.

Definition 4.7 (Origin and image)
Given that a DRM computed the low dimensional point $\tilde{x}_i \in \mathbb{R}^d$, called origin, the subtraction of the first e effects in the low dimensional data space results in the image point:

$$\widehat{x}_i := \begin{pmatrix} \mathbb{0}_e \\ \widetilde{x}_{e+1,i} \\ \vdots \\ \widetilde{x}_{d,i} \end{pmatrix}$$

Definition 4.8 (Generalised mode matrix)
Given a low dimensional approximation $\widetilde{X} \in \mathbb{R}^{d\times s}$ and the approximation for the generating function $\widetilde{F} : \mathbb{R}^{d\times s} \to \mathbb{R}^{D\times s}$ computed by a DRM, a matrix $M_{\mathrm{DDRM}} \in \mathbb{R}^{s\times s}$ is called generalised mode matrix, if the following properties hold for the i-th column and all $i \in \{1, \ldots, s\}$:

$$\widehat{x}_i = \widetilde{X}\,(e_i - m_i)$$
$$\widetilde{F}\big(\widetilde{X}\,(e_i - m_i)\,\big) \approx \widetilde{F}\big(\widetilde{X}\big)\,(e_i - m_i)$$

To apply the generalised difference operation, a suitable generalised mode matrix M_{DDRM} must be determined. With the origin and image known in the low dimensional space, the construction of m_i solely depends on the assumptions on $\widetilde{F}$ and this is where the two approaches introduced in this thesis differ.

4.2.1 Difference Local Linear Interpolation

The first example of a Generalised Difference Dimensionality Reduction Method is motivated by the LLE approach in combination with the LLI method for the evaluation of low dimensional data points, see Section 3.3.2. Essentially, the basic idea of the newly developed Difference Local Linear Interpolation (DLLI) is to define the generalised mode matrix M_{DLLI} in terms of nearest neighbour weights derived from the low dimensional embedding.
If the DRM used to compute this embedding is LLE or MLLE, the low dimensional coordinates were determined such that the following holds:

$$\widetilde{F}\left(\sum_{j\in\mathcal{N}(y_i)} w_{ij}\widetilde{x}_j\right) \approx \widetilde{F}\,(\widetilde{x}_i) \;=\; y_i \;\approx \sum_{j\in\mathcal{N}(y_i)} w_{ij}y_j = \sum_{j\in\mathcal{N}(y_i)} w_{ij}\widetilde{F}\,(\widetilde{x}_j)$$

This means that the generating function $\widetilde{F}$ is approximately invariant for a certain set of local linear combinations for LLE or for multiple specific sets in the case of MLLE. The concept of DLLI is to use this invariance in the difference operation. In order to utilise this property, two core assumptions need to be made.
Firstly, it is assumed that the invariance is approximately valid for all linear combinations of nearest neighbours and not only for the fixed sets of weights.
Secondly, it is assumed, that the nearest neighbours can also be determined in the computed low dimensional embedding, meaning that $\mathcal{N}(y_i)$ can be replaced by $\mathcal{N}(\widetilde{x}_i)$.
It is important to note, that embeddings computed by the aforementioned DRMs are in general only locally linear, i.e. they aim to preserve a fixed set of local linear combinations in a least squares sense, which is not to be confused with piecewise linearity. Thus, any other linear combination will in general not be preserved exactly, and a mathematical proof for the two assumptions cannot be given. However, tests have shown that the relations hold approximately, if the Dimensionality Reduction succeeded in computing a viable embedding, which is demonstrated in the evaluation

in Section 5.1.3. The first assumption is also fundamental for the work in [FZGK14], where it was evaluated in combination with Isomap and showed promising results. Changing the determination of the nearest neighbours from high to low dimensional data space in combination with an interpolation was also investigated for several DRMs in [Hah16], where it was shown, that it is applicable in several cases. Furthermore, the evaluation in Section 5.1.2 has also shown that using the low dimensional nearest neighbours is a viable option. Thus, the two properties are assumed to hold in order to introduce the new difference operation.
For each given origin point $\widetilde{x}_i$ and its image point $\widehat{x}_i$ the corresponding entries of the column m_i of M_{DLLI} can be determined in two steps: First, the low dimensional neighbourhoods $\mathcal{N}(\widetilde{x}_i)$ and $\mathcal{N}(\widehat{x}_i)$ are determined. Then, the interpolation weights for the nearest neighbour reconstruction are calculated. In this calculation, the approach of the constraint LLI described in Section 3.4.3.2 was used, which penalises large weights for nearest neighbours, which are far away. Here, the optimisation problem Eq. (3.40) is solved for $\widetilde{x}_i$ and $\widehat{x}_i$ with their respective neighbours yielding the two sets of weights $v_i, w_i \in \mathbb{R}^s$, such that

$$\begin{aligned}
\widehat{x}_i &\approx \sum_{j=1}^{s} v_{ij}\widetilde{x}_j \quad , v_{ik} = 0 \;\; , \; \forall k \notin \mathcal{N}(\widehat{x}_i) \\
\widetilde{x}_i &\approx \sum_{j=1}^{s} w_{ij}\widetilde{x}_j \quad , w_{ik} = 0 \;\; , \; \forall k \notin \mathcal{N}(\widetilde{x}_i)
\end{aligned}$$

With these weight vectors the desired column m_i is given by the following relation:

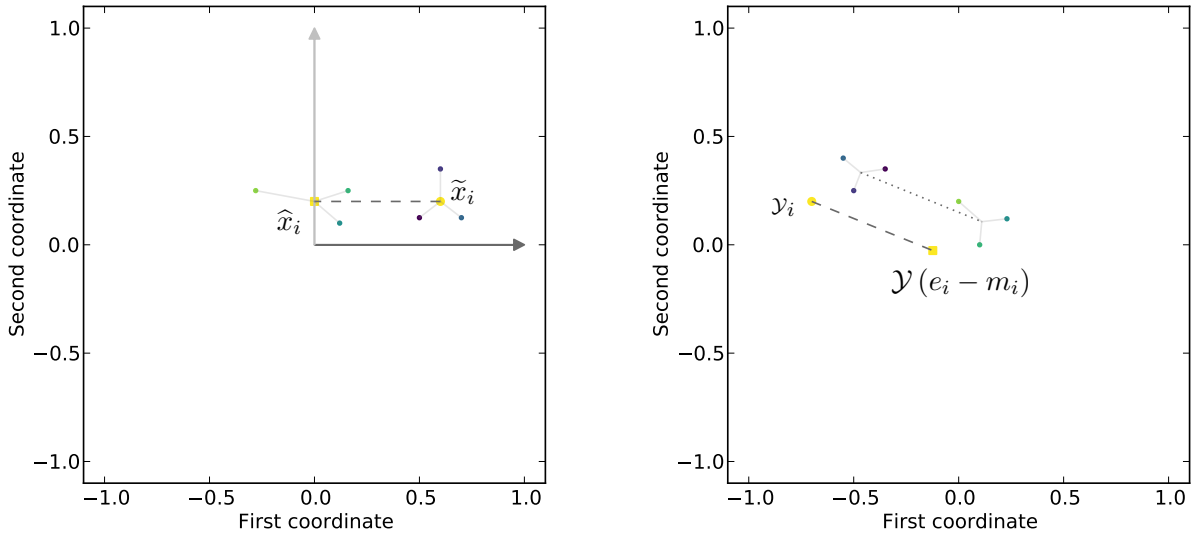
$$m_i := w_i - v_i \tag{4.9}$$

This relation approximately yields the elimination property of Eq. (4.6) for the corresponding column as is shown in the following. With $e_i \in \mathbb{R}^s$ being the i-th unit vector, it holds that:

$$\begin{aligned}
\widetilde{X}\,(e_i - m_i) &= \widetilde{x}_i - \widetilde{X} m_i \\
&= \widetilde{x}_i - \widetilde{X}\,(w_i - v_i) \\
&\approx \sum_{j=1}^{s} w_{ij}\widetilde{x}_j - \sum_{j=1}^{s} w_{ij}\widetilde{x}_j + \sum_{j=1}^{s} v_{ij}\widetilde{x}_j \\
&\approx \widehat{x}_i
\end{aligned}$$

This means that the difference operation can be viewed as an update vector that is added to the original node position. Starting and ending points of said update vector are defined by the linear combination of the nearest neighbours and can hence also be evaluated in the original high dimension Y as well as on a different target data set $\mathcal{Y}$.
A graphical example of this update is given in Fig. 4.3. In the left picture, the source of the operation is shown. Both the designated origin $\widetilde{x}_i$ marked by the circle and

the image $\widehat{x}_i$ of this difference operation marked by the square are expressed as a linear combination of their $k = 3$ nearest neighbours. In the right picture, the effect on the target data set $\mathcal{Y}$ is shown. For visualisation means $\mathcal{D} = 2$ is assumed. Here origin and image are reconstructed in this representation as the linear combination of the respective neighbours with the same weights as $\sum_{j=1}^{s} w_{ij} y_j$ and $\sum_{j=1}^{s} v_{ij} y_j$. The update is the dotted line and calculated as the difference between these two points. Finally, the square $\mathcal{Y}(e_i - m_i)$ is obtained by adding the update to the designated data point y_i in this data set.



(a) First mode $\widetilde{X}(I_s - M_{\mathrm{DLLI}})$ on source $\widetilde{X}$

(b) First mode $\mathcal{Y}(I_s - M_{\mathrm{DLLI}})$ on target $\mathcal{Y}$

Figure 4.3: Visual example of DLLI operation: The pictures show an excerpt of the low dimensional representation of the data sets with the two source modes highlighted as arrows. The circle is one designated point and the dots other original data points. The square is the image of the designated point, projected onto the orthogonal complement of the first mode. The dashed line is the update defined by this difference operation.

When applying this approach in combination with DRMs, several things need to be considered:
First, the core of this method is the preservation of local linear combinations. Hence, it should only be used with DRMs that reasonably comply with this property. While the LLE variants are constructed with this aim, the work in [FZGK14] has shown that this can also be true for methods that preserve pairwise Euclidean distances between nearest neighbours such as Isomap, ENLM and GNLM.
Second, the construction of the neighbourhoods should match the original rule used to compute the embedding. All methods introduced in Chapter 3 use the modified

k-rule, and hence all neighbourhoods for DLLI should be constructed by computing the k nearest neighbours using the identical value for k.
Finally, interpolation with the additional distance penalty was chosen with the aim to limit the result of the difference operation to the manifold. When removing variance from a data set, the result should be a subset of the original data. The distance penalty of Eq. (3.40) indirectly enforces a proximity to the nearest neighbours: Since large weights are penalised, extrapolations far away from the given sample points are also penalised.
With these additions, the DLLI is the first implementation of the generalised difference method that can be used in combination with nonlinear DRM approaches.

4.2.2 Difference Local Affine Interpolation

The second Generalised Difference Dimensionality Reduction Method is motivated by the LTSA approach and its creation of virtual simulations by interpolation in a local tangent space, which was introduced as LAI in Section 3.3.3.1. The core idea of this new method, called Difference Local Affine Interpolation (DLAI), is to define M_{DLAI} in terms of local affine subspace weights.
All generalised difference methods utilise the low dimensional coordinates to define the generalised mode matrix, which can be applied in the high dimensional representation as well. Embeddings that are computed by the LTSA approach aim to preserve local tangent spaces in a least squares sense. These tangent spaces are computed for each point y_i and its complete neighbourhood $\overline{\mathcal{N}}(y_i) = \mathcal{N}(y_i) \cup \{i\}$ as defined in Eq. (3.23) and Eq. (3.24). The matrix Y_i, containing the high dimensional coordinates for all points in this neighbourhood, is centralised and decomposed by a truncated SVD, where the right singular vectors W_i are defining the tangent space, see Section 3.3.3.1.

$$\left(Y_i - \overline{y}_i \mathbb{1}^{\mathsf{T}}_{|\overline{\mathcal{N}}(y_i)|}\right) \approx\colon U_i \Sigma_i W_i^{\mathsf{T}}$$

Each point y_i can be projected into its local tangent space and written as an affine linear combination of the sample points:

$$y_i^{\perp} := Y_i \omega_i$$

The weights $\omega_i \in \mathbb{R}^{|\overline{\mathcal{N}}(y_i)|}$ for this affine combination of the projection $y_i^{\perp}$ can be computed by subtracting the local mean value $\overline{y}_i$ and applying the pseudo inverse for the truncated SVD of the centralised local data matrix Y_i:

$$\omega_i := W_i \Sigma_i^{-1} U_i^{\mathsf{T}} \left(y_i - \overline{y}_i\right)$$

As the weight vectors ω_i are in the span of W_i and this matrix has been used to aggregate the global alignment matrix, the preservation of tangent spaces means that the relative position of the point given by these affine interpolation weights is the same in the high as well as in the low dimensional embedding. Hence, for the same

weights ω_i it holds that:

$$y_i^{\perp} = Y_i \omega_i$$
$$\widetilde{x}_i \approx \widetilde{X}_i \omega_i$$

Additionally, since only information in the tangent space is preserved and differences orthogonal to this space are discarded, it holds that:

$$Q_i \Sigma_i^{-1} U_i^{\mathsf{T}} \left(y_i - y_i^{\perp} \right) = \mathbb{0}_{|\overline{\mathcal{N}}(y_i)|}$$

Accordingly, the intrinsic coordinates for the projection are the same as for the original data point. Under the assumption that the point is close to its projection into its own tangent space, meaning that $y_i \approx y_i^{\perp}$, the following holds:

$$\widetilde{F}\left(\sum_{j \in \overline{\mathcal{N}}(y_i)} \omega_{ij} \widetilde{x}_j \right) \approx \widetilde{F}\left(\widetilde{x}_j\right) \;=\; y_j \;\approx\; y_j^{\perp} \;=\; \sum_{j \in \overline{\mathcal{N}}(y_i)} \omega_{ij} y_j = \sum_{j \in \overline{\mathcal{N}}(y_i)} \omega_{ij} \widetilde{F}\left(\widetilde{x}_j\right)$$

This means that in the case of an embedding computed by LTSA, the generating function $\widetilde{F}$ is approximately invariant for local affine interpolation weights of the sample points, if the point can be reasonably approximated by its own tangent space. Similar to the previous method, the DLAI aims to use this invariance in the difference operation and also needs two core assumptions.
First, it is assumed that the invariance is approximately valid for all points x in the affine subspace and not only for the given sample points $\widetilde{x}_i$.
Second, it is assumed that the nearest neighbours spanning the tangent space can also be determined in the computed low dimensional embedding, so that $\overline{\mathcal{N}}(y_i)$ can be replaced by $\overline{\mathcal{N}}(\widetilde{x}_i)$.
For each given origin point $\widetilde{x}_i$ and its image point $\widehat{x}_i$ the corresponding entries of the column m_i of M_{DLAI} can be determined in two steps: First, the complete low dimensional neighbourhoods are determined. While $\overline{\mathcal{N}}(\widetilde{x}_i)$ can be constructed without modifications, the complete neighbourhood for the image point $\widehat{x}_i$ cannot be determined in the same way. Since it is not an original sample point, the union $\mathcal{N}(y_i) \cup \{i\}$ is in general not meaningful as $\widehat{x}_i$ and $\widetilde{x}_i$ can be far apart. Instead, the $(k+1)$-nearest low dimensional neighbours are used to determine $\mathcal{N}(\widehat{x}_i)$ for the image point.
With the neighbourhoods constructed, the subspace weights for the local affine interpolation can be calculated. This calculation is done in the same manner as the LAI given in Section 3.3.3.2: The matrix containing the low dimensional coordinates of the nearest neighbours is centralised and then decomposed by an SVD, see Eq. (3.30). Then, the interpolation weights can be computed with the pseudo inverse of Eq. (3.31).
As explained in the last section, it is important to restrict the interpolations to the solution manifold by preventing extrapolations far away from the sample points. Here, these far extrapolations are prevented through a limitation of the subspace weights.

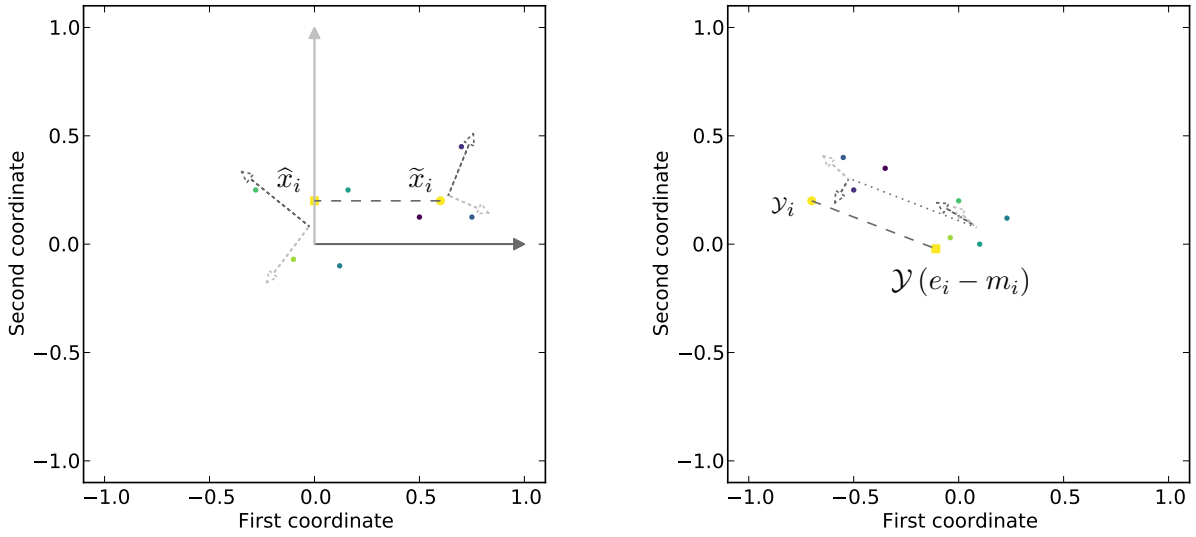
For the sample points, the interpolation weights ω_i in this specific subspace can be retrieved directly from the SVD. These sample weights are used to determine a bounding box in the subspace coefficients. The newly computed weights $\tilde{\omega}$ for point $x \in \{\widehat{x}_i, \widetilde{x}_i\}$ are then clamped to this bounding box, with component-wise min and max:

$$\begin{aligned} \omega_{\min} &:= \min_{j \in \mathcal{N}(x)} (\omega_j) \\ \omega_{\max} &:= \max_{j \in \mathcal{N}(x)} (\omega_j) \\ \tilde{\omega} &\leftarrow \max(\omega_{\min}, \min(\omega, \omega_{\max})) \end{aligned}$$

This process is performed for the origin and the image of each point and yields two sets of weights $v_i, w_i \in \mathbb{R}^s$, such that

$$\begin{aligned} \widehat{x}_i &\approx \sum_{j=1}^{s} v_{ij} \widetilde{x}_j \quad , v_{ik} = 0 \;\; , \; \forall k \notin \mathcal{N}(\widehat{x}_i) \\ \widetilde{x}_i &\approx \sum_{j=1}^{s} w_{ij} \widetilde{x}_j \quad , w_{ik} = 0 \;\; , \; \forall k \notin \overline{\mathcal{N}}(\widetilde{x}_i) \end{aligned}$$

These weights can be used to compute the corresponding i-th column m_i of M_{DLAI} with the same relation as for the DLLI approach, as described in Eq. (4.9). Multiplying a data set by $(I_s - M_{\mathrm{DLAI}})$ can also be viewed as an update vector added to the original node position. A visual explanation of this update is given in Fig. 4.4. In this figure, the left picture shows the effect of the difference operation on the source data set $\widetilde{X}$. The local tangent spaces for the origin $\widehat{x}_i$ and its three nearest neighbours and the one for the four nearest neighbours of the image $\widetilde{x}_i$ are computed. Both the origin and the image of the designated point are then defined in the basis of their respective local frames by computing the weights w_{ij} and v_{ij}. In the right picture the impact on the target data set $\mathcal{Y}$ is shown. The two previously computed tangent frames are also transferred to this data representation. The origin and the image are then reconstructed in the local coordinates of their respective tangent spaces by linear combination with the respective weights $\sum_{j=1}^{s} w_{ij} y_j$ and $\sum_{j=1}^{s} v_{ij} y_j$. The update is the dotted line and calculated as the difference between these two reconstructions. Finally, the square $\mathcal{Y}(e_i - m_i)$ is obtained by adding the update to the designated data point y_j in this data set.



(a) First mode $\widetilde{X}(I_s - M_{\mathrm{DLAI}})$ on source $\widetilde{X}$

(b) First mode $\mathcal{Y}(I_s - M_{\mathrm{DLAI}})$ on target $\mathcal{Y}$

Figure 4.4: Visual example of DLAI operation: An excerpt of the data sets is shown with the two source modes. The circle is one designated point and the dots other original data points. The square is the image of the designated point, projected onto the orthogonal complement of the first mode. The dotted arrows are the base vectors of the local tangent frames and the dashed line is the update defined by this operation.

Though the DLAI method is motivated by the tangent space preserving LTSA, it can also be meaningfully applied to embeddings computed by other DRMs as well. For example, the parallel transport variants PTU and PTNLM also approximately preserve tangent spaces, see Section 3.4.4.1. Moreover, the affine interpolation weights are a special case of nearest neighbour weights so the difference method can also be reasonably used in combination with local weight-based approaches similar to the DLLI.
This concludes the introduction of the two specific methods implementing the concept of a Generalised Difference Dimensionality Reduction.

4.2.3 Normalisation Enhancement

The two introduced difference methods can be combined with an additional enhancement that is motivated by the strong features of the original DPCA. These two features originate from the equivalence to an orthogonal projection and are the fact that multiple subtractions of the same effect yield the same result as a single subtraction, and the fact that applying the difference operation cannot increase the variance in the target data. To guarantee both of these properties, a function must be a

projection that is well defined by its kernel and its image, which must also be orthogonal to prohibit variance increase [Gal13]. But since the PCA is an optimal rank-d approximation, see Section 3.2, no better vectors can be found to describe the variance than those vectors provided by the PCA. Hence, the nonlinear methods cannot mirror these strong features and still improve on the linear approach. If a difference operation is to improve the explained amount of variance per mode, these properties need to be relaxed.

Even if the two properties cannot be fully guaranteed for nonlinear methods, the adding of variance should be avoided as best as possible. In order to do this, the norm of $I_s - M$ should ideally be limited. But since the effect on the given data set is prescribed, there are not many options. A direct manipulation of e.g. the eigenvalues is affecting the function too much.
What can be done, is a limitation of the left scalar products by trimming the norm of the columns of $c_i \in \mathbb{R}^s$ of $I_s - M$:

$$I_s - M =: (c_1 \dots c_s)$$

This can be achieved for i-th column by manipulating the construction of the matrix. For each point in the difference operation the origin and the image position are expressed as linear combinations of the given sample points:

$$\begin{pmatrix} \mathbb{0}_e \\ \widetilde{x}_i|_{d-e} \end{pmatrix} = \widehat{x}_i = \sum_j v_{ij}\widetilde{x}_j \qquad (4.10)$$
$$\widetilde{x}_i = \sum_j w_{ij}\widetilde{x}_j$$

Which means that according to Eq. (4.9) the i-th column can be rewritten as:

$$c_i = e_i - w_i + v_i$$

Furthermore, an additional trivial linear combination for the origin is available with $\widetilde{x}_i = 1 \cdot \widetilde{x}_i$. Thus, for all $\psi_i \in \mathbb{R}$ the combination

$$\widetilde{x}_i = \psi_i\widetilde{x}_i + (1 - \psi_i)\sum_j w_j\widetilde{x}_j \qquad (4.11)$$

is also a valid approximation. With the linear combinations of Eq. (4.10) and the trivial combination a new column $c_i(\psi_i) \in \mathbb{R}^s$ is introduced. The aim is to choose ψ_i such that the norm of the new column is less than one. Its entries are

$$\begin{aligned} c_{ii}(\psi_i) &:= 1 + v_{ii} - \psi_i - (1 - \psi_i)w_{ii} \\ c_{ij}(\psi_i) &:= v_{ij} - (1 - \psi_i)w_{ij} \quad , \ \forall j \neq i \end{aligned}$$

and for $\psi_i = 0$ it is equal to the original column with $c_i(0) = c_i$. This way the norm of the i-th column is a function in ψ_i, too

$$\begin{aligned}\|c_i(\psi_i)\|_2^2 &= c_{ii}(\psi_i)^2 + \sum_{j\neq i} c_{ij}(\psi_i)^2 \\ &= (1 + v_{ii} - \psi_i - (1-\psi_i)w_{ii})^2 + \sum_{j\neq i} (v_{ij} - (1-\psi_i)w_{ij})^2\end{aligned} \tag{4.12}$$

First the norm of the unmodified column with $\psi_i = 0$ is evaluated. If this value is larger than one, the following minimization problem is solved:

$$\min_{\psi_i \in \mathbb{R}} \ \|c_i(\psi_i)\|_2^2 - 1 \tag{4.13}$$

Theorem 4.2
With the following scalar values $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$ for shorter notation,

$$\begin{aligned}\alpha_i &:= \sum_j v_{ij} w_{ij} \\ \beta_i &:= 1 - 2w_{ii} + \sum_j w_{ij}^2 \\ \gamma_i &:= \left(1 + (v_{ii} - \alpha_i)\,\beta_i^{-1}\right)^2 - \left(1 + \left(2v_{ii} - 2\alpha_i - 1 + \sum_j v_{ij}^2\right)\beta_i^{-1}\right)\end{aligned}$$

the solution $\psi_i^* \in \mathbb{R}$ to Eq. (4.13) with $c_i(0) > 1$ and $\beta_i \neq 0$ is given by

$$\psi_i^* := 1 + (v_{ii} - \alpha_i)\,\beta_i^{-1} \pm \sqrt{\max(\gamma_i, 0)} \tag{4.14}$$

Proof. For the equality of $c_i(\psi_i)$ and 1 the quadratic form can be analytically solved:

$$\begin{aligned}& 1 \stackrel{!}{=} \|c_i(\psi_i)\|_2^2 \\ \Leftrightarrow\ & 1 = (1 + v_{ii} - \psi_i - (1-\psi_i)w_{ii})^2 + \sum_{j\neq i} (v_{ij} - (1-\psi_i)w_{ij})^2 \\ \Leftrightarrow\ & 1 = 2\left(v_{ii} - \psi_i - (1-\psi_i)w_{ii} - \psi_i v_{ii} + \psi_i(1-\psi_i)w_{ii} - (1-\psi_i)\sum_j v_{ij}w_{ij}\right) \\ & \quad + \sum_j v_{ij}^2 + (1-\psi_i)^2 \sum_j w_{ij}^2 + \psi_i^2 + 1\end{aligned}$$

By subtracting 1 and with $\iota_i := \sum_j v_{ij}^2$ and $\rho_i := \sum_j w_{ij}^2$ being the squared lengths of the weight vectors for image and origin, this expression can be simplified to:

$$\begin{aligned}\Leftrightarrow\ & 0 = 2\left(v_{ii} - \psi_i - (1-\psi_i)w_{ii} - \psi_i v_{ii} + \psi_i(1-\psi_i)w_{ii} - (1-\psi_i)\sum_j v_{ij}w_{ij}\right) \\ & \quad + \iota_i + (1 - 2\psi_i + \psi_i^2)\rho_i + \psi_i^2 \\ \Leftrightarrow\ & 0 = -2\psi_i\left(1 - 2w_{ii} + \rho_i + v_{ii} - \sum_j v_{ij}w_{ij}\right) \\ & \quad + (1 - 2w_{ii} + \rho_i)\,\psi_i^2 + \iota_i + \rho_i + 2v_{ii} - 2w_{ii} - 2\sum_j v_{ij}w_{ij}\end{aligned}$$

With the introduction of $\alpha_i := \sum_j v_{ij} w_{ij}$ and $\beta_i := 1 - 2w_{ii} + \rho_i$ the expression can be further altered, if the latter is non zero:

$$\begin{aligned} &\Leftrightarrow \quad 0 = -2\psi_i \left(\beta_i + v_{ii} - \alpha_i\right) + \beta_i \psi_i^2 + \iota_i + \beta_i - 1 + 2v_{ii} - 2\alpha_i \\ &\overset{\beta_i \neq 0}{\Leftrightarrow} \quad 0 = \psi_i^2 - 2\left(1 + \left(v_{ii} - \alpha_i\right)\beta_i^{-1}\right)\psi_i + \left(\iota_i + 2v_{ii} - 2\alpha_i - 1\right)\beta_i^{-1} + 1 \end{aligned} \tag{4.15}$$

Eq. (4.15) is a quadratic equation in monic form and its scaled discriminant

$$\gamma_i := \left(1 + \left(v_{ii} - \alpha_i\right)\beta_i^{-1}\right)^2 - \left(1 + \left(\iota_i + 2v_{ii} - 2\alpha_i - 1\right)\beta_i^{-1}\right) \tag{4.16}$$

can be negative in some cases. In order to solve the minimisation problem of Eq. (4.13), the closest possible real solution is chosen. Finally, the optimal ψ_i^* can be obtained as

$$\psi_i^* := 1 + \left(v_{ii} - \alpha_i\right)\beta_i^{-1} \pm \sqrt{\max(\gamma_i, 0)}$$

□

With this optimal solution, the column of M can be reconstructed with the modified weights for the origin, such that the column norm is limited.

1: **for all** columns c_i in $(I_s - M)$ **do**
2: **if** $\|c_i(0)\|_2^2 > 1$ **then**
3: **if** $\beta_i > 0$ **then**
4: Compute ψ_i^* according to Eq. (4.14).
5: Update the weights for the origin, see Eq. (4.11) and reconstruct the column.
6: **end if**
7: **end if**
8: **end for**

Algorithm 4.1: Normalisation Enhancement Algorithm for Generalised Difference Methods

While this updated formulation does not affect the difference operation on the source data set, it can have a big impact on the target data set, especially, if the approximation of the origin by the weights is not very precise.

4.3 Recapitulation

In the previous sections in this chapter, the base concept of the Generalised Difference Dimensionality Reduction was introduced. In this process, the correlation of a target data set with a fixed number of low dimensional modes of a source data set is investigated. This is achieved, by first computing a generalised mode matrix M_{DDRM}

for the source data set. Then the target data set is multiplied with this mode matrix and the difference measures δ_{spec} and δ_{var} are computed for the result. For this general concept of a difference operation, three specific implementations, with different constructions of the mode matrix have been explained in this section. One of these is linear and two are nonlinear methods.
The linear DPCA is based on the orthogonal projection onto the orthogonal complement of the modes, which are to be subtracted and can be described globally by a linear combination of samples. It is the only method featured in this work that already exists in literature. The equivalence to the orthogonal projection imbues some strong beneficial properties, but unfortunately, the linear model means that it is only meaningfully applicable to linear manifolds.
The newly introduced DLLI method is based on local neighbourhood weights. For each sample point, the difference operation is described exclusively by the nearest neighbours of the origin and the image of the modification vector. Thus, the method is only meaningfully applicable to nonlinear DRMs that preserve these linear combinations of nearest neighbours reasonably well. An additional regularisation in the computation of the weights is penalising large weights for relatively distant points, indirectly enforcing proximity to manifold at the risk, of probably having a less precise position of the points.
Finally, the new DLAI approach is based on sample positions in local tangent spaces. Again, the difference operation is described in terms of origin and image of the modification vector, but here the points are described by their position in the local tangent frames. These tangent frames are determined by local PCAs and the points are then projected into the spanned subspace of the first principal components. Large weights for base vectors of the subspace are allowed up to a bounding box, thus explicitly enforcing the proximity. The position in this tangent space is exact up to this bounding box, though the projection into this space may discard some information. Although the method was designed for DRMs that preserve these local tangent spaces sufficiently well, it can also be applied to methods, which preserve local weights, since the positions in the subspaces are also expressed in local neighbour weights.
Finally, both linear approaches were extended by an additional normalisation enhancement to mitigate the drawback of not being an orthogonal projection like the linear DPCA.
With the addition of nonlinear difference methods, the nonlinear DRMs can be utilised in the Extended Workflow.

5 Evaluation

The DRMs explained in Chapter 3 were altered from their base versions in literature and the nonlinear difference methods introduced last in Chapter 4 are completely new, therefore the performance of both must be carefully evaluated. Hence, the approaches in this chapter are thoroughly tested under certain aspects. Before their capabilities in the Extended Workflow on complex simulation data are evaluated, their performance is first tested individually on simpler artificial data sets. The performance on these simple data sets is also used to select methods for the actual analysis task. All methods introduced in Chapter 3 are evaluated on the simple artificial examples and depending on their performance, some are discarded for further investigations.

5.1 Performance on Artificial Data

Artificial data sets are created to contain certain properties and are widely used in literature to evaluate DRMs [ST02], [LV07], [ZQZ11], [BYF+19]. These data sets provide the means to test the two steps of the analysis separately, first the Dimensionality Reduction and then the difference operation. Furthermore, the performance of different methods can be compared and evaluated with regard to the ideal result that is known for these data sets. In this section, a selected set of artificial examples is tested to evaluate the performance of the new methods on manifolds with specific properties. All utilised artificial examples are introduced in this section, but are additionally summarised in Section B.1 of the appendix.

5.1.1 Creating Artificial Data Sets

In this thesis, artificial data sets are created by first sampling intrinsic coordinates in the unit hypercube $[0,1]^d \subset \mathbb{R}^d$ and then applying known generation functions $f : [0,1]^d \to \mathbb{R}^D$. This has two important effects:
First, all data sets of same sampling mode and same intrinsic dimension are intrinsically the same. This means that the original low dimensional coordinates for the sample points are identical and only the high dimensional representation is different. This property is crucial for testing the difference operations.
Second, this way of creating data sets increases the reproducibility of the tests performed: Because both sampling and generating function used to calculate the high dimensional representation can be explicitly stated, an interested reader could recreate the identical data sets.

Several different kinds of sampling a data space are known in literature and a good overview is given in [SS06]. Three of the methods introduced in [SS06] are used in this work: The so-called Deterministic, the "Quasi-Random" and the "Pseudo-Random"

sampling methods. Though all of these are in fact deterministic, they have different properties.
Data, referred to here as deterministic, is sampled by placing it in a regular pattern in order to provide equal coverage of the domain. In this work, the data is aligned in a grid that is equally spaced in all directions. This equal spacing guarantees a space-filling distribution of the samples. The regularity of this pattern has the benefit of providing easy means of following the local effects of the generating function, since the pattern can be identified also in the high dimension. But it can also pose a challenge, especially if one dimensional direction is eliminated, several points can collapse onto each other.
Pseudo-Random data is usually generated using a random number generator with the aim of getting a uniform distribution $\mathcal{U}(0,1)$. It has the benefit of not generating regular patterns and also no repetitions. Here it was generated with the build in "rand"-function of the C-Standard-lib version "GNU libc 2.17" and a seed of 1, see [Fre20]. A drawback of this random number approach is that data can have dense clusters on the one hand as well as holes in the domain on the other hand.
Quasi-Random sampled data can be viewed as a compromise between the two previous methods. In this approach, the data is generated in an incremental way, adaptively refining the sampling of the domain, without repetition of values or obvious patterns. One possibility to create such samples are Latin hypercube samplings that provide uniform coverage and avoid correlation patterns [HTP06]. Since an even coverage can be achieved by the deterministic sampling and correlation patterns are avoided by the Pseudo-Random sampling, a different approach was chosen as a compromise in this work. Here, the e-th coordinate direction x_{ie} is sampled using a Halton-Sequence [SS06] for the e-th prime number. This sequence can be computed utilising the expansion of a number $n \in \mathbb{N}$ in base $b \in \mathbb{N}$, with the coefficients $c_k \in \mathbb{N}$ and the radical-inverse function for this base $\nu_b : \mathbb{N} \to [0,1) \subset \mathbb{R}$:

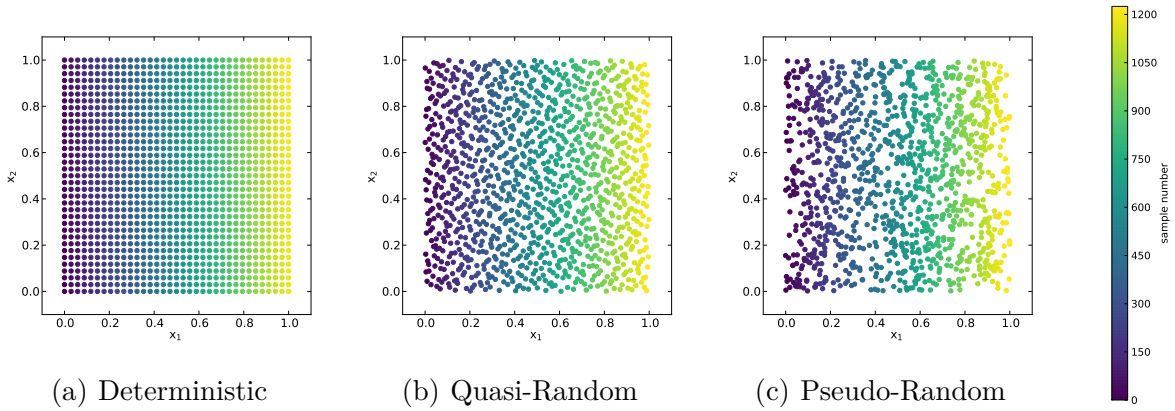
$$n =: \sum_{0}^{j} c_k b^k$$

$$\nu_b(i) := \sum_{k=0}^{j} c_k b^{-k-1}$$

For a selected dimension d and a set of prime numbers $\{p_1, ...p_d\}$ the i-th value of the corresponding Halton-Sequence can be computed as:

$$x_i := (\nu_{p_1}(i), \ldots, \nu_{p_d}(i))^{\mathsf{T}} \tag{5.1}$$

All data sets investigated in this thesis have an intrinsic dimension of $d \leq 6$ and the Quasi-Random sampled data was generated using the Halton-Sequence for prime numbers $\{2, 3, 5, 7, 11, 13\}$.
Examples for the different sampling methods are visualised in Fig. 5.1. For better perception, all low dimensional data points are sorted lexicographically and coloured according to their number.



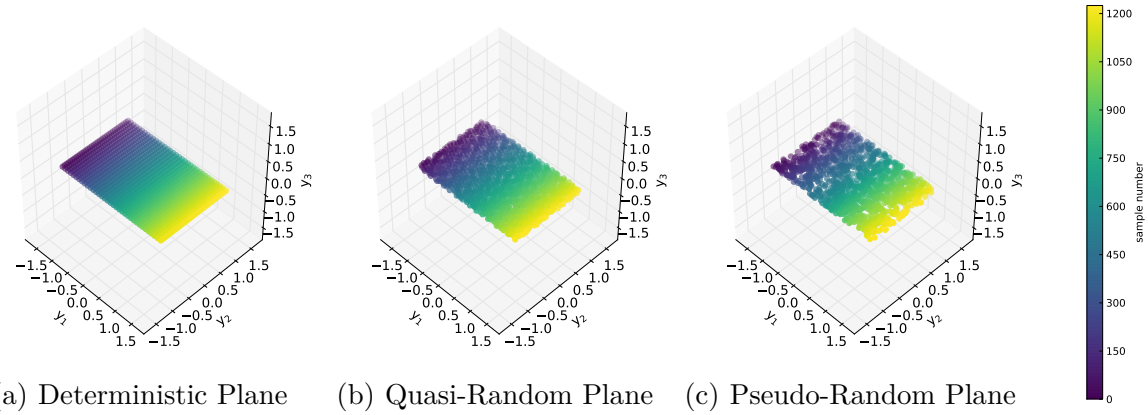
(a) Deterministic (b) Quasi-Random (c) Pseudo-Random

Figure 5.1: Example of sampling low dimensional data. A total of 1 225 two dimensional points were sampled with the three given sampling methods. The colour reflects the sample number, starting dark at one and and getting lighter as the number increases.

Two possible examples of artificial data sets with intrinsic dimension $d = 2$ are the Plane and the S-Shape, which are used in the further evaluation sections.
The Plane example is an ideal two dimensional linear manifold. Given the low dimensional coordinates $x_1, ..., x_s \in [0, 1]^2$ the high dimensional representation of the data set can be created for any desired large dimension $D \geq 2$ using the generating function $f_{\text{plane}} : [0, 1]^2 \to \mathbb{R}^D$:

$$f_{\text{plane}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} 3x_1 - 1.5 \\ 2x_2 - 1 \\ \mathbb{0}_{D-2} \end{pmatrix} \tag{5.2}$$

The results for the three different sampling methods are visualised in Fig. 5.2. They show the coloured rectangle in which the centre of the mass is approximately in the origin of the coordinate system, depending on the distribution of the sample points. One edge of the Plane, which is associated with the first high as well as low dimensional coordinate, is longer than the other, which is associated with the second coordinates.



(a) Deterministic Plane (b) Quasi-Random Plane (c) Pseudo-Random Plane

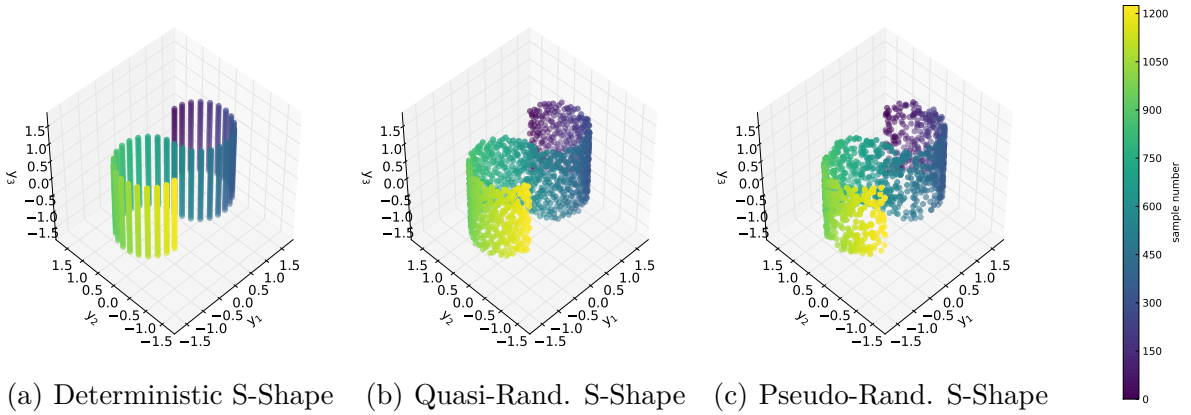
Figure 5.2: Linear example of artificial data. The 1 225 low dimensional data points of Fig. 5.1 are projected to the high dimensional space using the Plane function of Eq. (5.2). Displayed are the first three high dimensional coordinates. Equivalent to Fig. 5.1, the colour corresponds to the sample number.

The second intrinsically two dimensional manifold example is the S-Shape data set. This nonlinear manifold can be created from given low dimensional sample points $x_1, ..., x_s \in [0,1]^2$ with the generating function $f_{\text{sshape}} : [0,1]^2 \to \mathbb{R}^D$ and $\phi_{\text{sshape}} : [0,1] \to \mathbb{R}$ for any $D \geq 3$.

$$\phi_{\text{sshape}}(x_1) := \begin{cases} \frac{3}{4}(\cos(3\pi x_1) - 1) \ , & x_1 > \frac{1}{2} \\ -\frac{3}{4}(\cos(3\pi x_1) - 1) \ , & x_1 \leq \frac{1}{2} \end{cases}$$

$$f_{\text{sshape}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \phi_{\text{sshape}}(x_1) \\ \frac{3}{4}\sin(3\pi x_1) \\ 2x_2 - 1 \\ \mathbb{0}_{D-3} \end{pmatrix} \tag{5.3}$$

The first two high dimensional coordinates y_1 and y_2 depend on the first intrinsic coordinate x_1 only and describe the position of the sample point on the arc of a curve shaped like the letter “S”. The third high dimensional coordinate y_3 is associated with the remaining low dimensional coordinate x_2 and describes the position in the extrusion direction of the curve, as can be seen in Fig. 5.3.



(a) Deterministic S-Shape (b) Quasi-Rand. S-Shape (c) Pseudo-Rand. S-Shape

Figure 5.3: Nonlinear example of artificial data. The 1 225 low dimensional data points of Fig. 5.1 are projected to the high dimensional space using the Plane function of Eq. (5.3). Plotted are the first three high dimensional coordinates. Equivalent to Fig. 5.1, the colour corresponds to the sample number.

Since the data points were first identically sampled in the low dimension and only thereafter projected into the larger space, this means that the data in Fig. 5.2.a is intrinsically the same as Fig. 5.3.a, although the high dimensional representation is different. The same holds for Fig. 5.2.b and Fig. 5.3.b, which are intrinsically identical, as well as for Fig. 5.2.c and Fig. 5.3.c. This is crucial for the evaluation of the difference methods in Section 5.1.3. But before the second step of the difference operation is investigated, the first step of Dimensionality Reduction is evaluated.

5.1.2 Assessing Embedding Quality

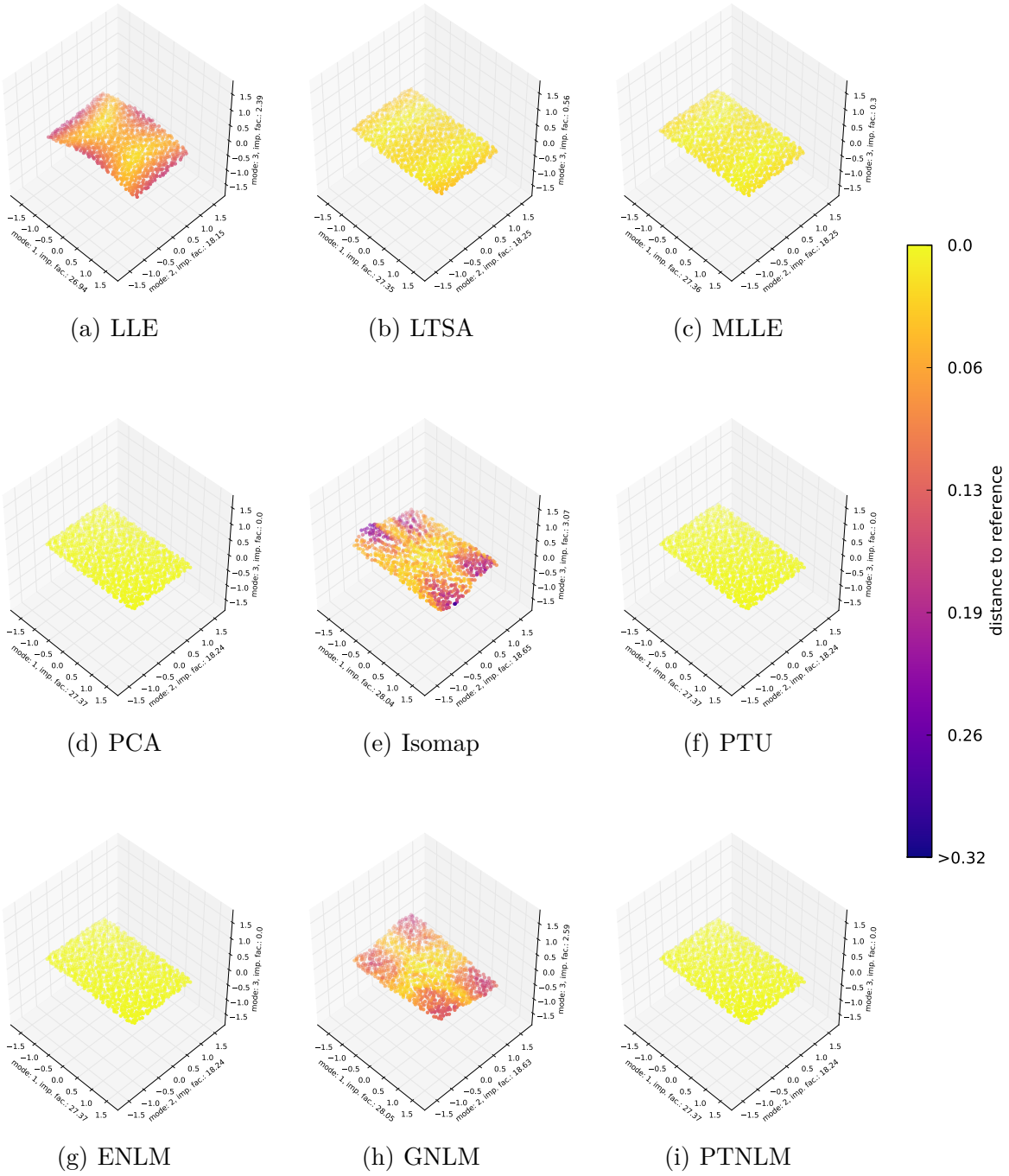
In order to evaluate the Dimensionality Reduction step, it is important to assess the quality of the obtained low dimensional embedding. As the second step is defined on the basis of the calculated coordinates, it is important to yield good embeddings, though different measures could be consulted to evaluate whether an embedding is good or bad. Two measures are employed in the following.
For artificial data sets, the desired outcome is usually known, thus the DRM results can be compared with the target, both in terms of importance factors and coordinates. In this section, two data sets are utilised to investigate the performance of the different DRMs: a linear and a nonlinear data set.
The linear example is the Plane introduced in the last subsection, see Fig. 5.2. In literature, nonlinear methods are seldom evaluated on linear manifolds, since the PCA already yields the rank-d optimal embedding, see Section 3.2, there is no need to apply more complex methods. Because PCA or classic metric MDS is optimal for linear manifolds, its result is the desired outcome of the DR. As the desired outcome is known, the distance to this reference can be computed for the results of the nonlinear methods, in order to evaluate their performance. Since the nature of the manifold is

unknown in practical applications, the nonlinear methods should ideally also perform reasonably well on linear manifolds.
The Plane data set is intrinsically two dimensional, but here it is embedded using the linear function in Eq. (5.2) into a higher dimensional space with $D = 5$. The newly computed coordinates y_1 have a direct linear dependency on the intrinsic coordinates x_1, as well as y_2 and x_2 respectively. After calculating these high dimensional representations, the different DRMs are applied. During this application, the methods were instructed to compute data sets of low dimension $d = 4$. This target dimension was intentionally overestimated compared to the original intrinsic dimension of two. The first assessment is, whether the number of significant importance factors matches the two of the linear approach and if the sizes are also similar. All methods used a neighbourhood size of $k = 10$ which is in literature often used for intrinsically two dimensional data sets, e.g. [ZW07], [BYF+19].
The resulting importance factors are displayed in Tab. 5.1, where an entry of "-" indicates that the DRM determined fewer dimensions. For the LMs the newly developed distance scaling approach, as described in Section 3.3.2.2, was used. For the MDS and NLM methods, the eigenvalues of the double centred dissimilarity matrix were used, as described in Section 3.4.3.2 and Section 3.5 respectively. Most of the DRMs match the desired result of the linear PCA very well. Only the LLE, Isomap and GNLM approaches have a noticeable but small third importance factor.

	PCA	LLE	LTSA	MLLE	Isomap	PTU	ENLM	GNLM	PTNLM
Imp. factor 1	27.371	26.944	27.351	27.363	28.045	27.371	27.371	28.050	27.371
Imp. factor 2	18.249	18.156	18.258	18.252	18.652	18.249	18.249	18.630	18.249
Imp. factor 3	-	2.391	0.569	0.301	3.070	-	-	2.590	-
Imp. factor 4	-	0.001	0.001	0.002	-	-	-	-	-

Table 5.1: Importance factors for the Plane data set computed by the different DRM approaches with $k = 10$.

For a fixed method, the importance factors for the different dimensions are of different magnitude. This means that the low dimensional coordinates are well defined up to their sign. Hence, these coordinates can be compared to the desired result, after the orientation has been validated by a scalar product and corrected if necessary. The result of this comparison is displayed in Fig. 5.4. Since the PCA is the reference, the error is zero for all points and because the target dimension is large enough to preserve all pairwise Euclidean distances, ENLM yields the same result, see Section 3.5.2. The manifold is well sampled, and the neighbourhood size is sufficient to correctly estimate the tangent spaces, hence the PTU and PTNLM are linearly precise, see Section 3.4.4.1, and also yield the correct result. The graph-based approaches yield the worst results, while the LMs are mostly better, but not as precise as the parallel transport variants. Overall, all methods provide usable low dimensional embeddings.



(a) LLE (b) LTSA (c) MLLE

(d) PCA (e) Isomap (f) PTU

(g) ENLM (h) GNLM (i) PTNLM

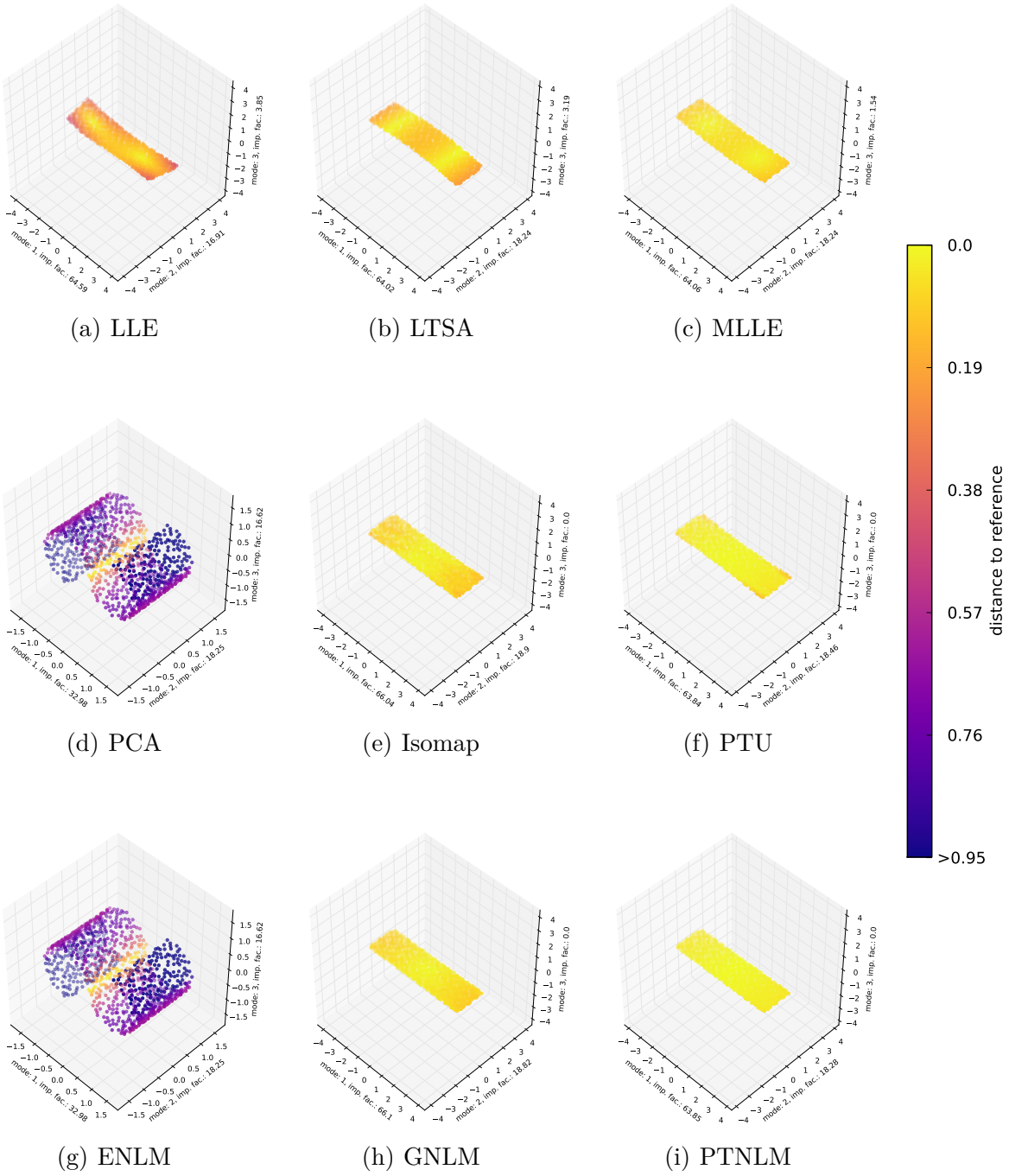
Figure 5.4: DRM results for the Plane data set. The 1 000 data points were sampled with the Quasi-Random method and then projected using the formula of Eq. (5.2). The first row shows the result of the LM-approaches, the second those of the MDS-methods, and the last row those of the NLM-variants.

The second example evaluated in this subsection is the S-Shape data set introduced in Eq. (5.3). This data set is again intrinsically two dimensional and embedded into a larger space with $D = 5$. Now, three original dimensions are non-zero with a linear dependency between x_2 and y_3, corresponding to the extrusion direction, and a nonlinear relation between x_1 and y_1, y_2, which is running along the curve of the S-shape. This nonlinear relation can be described as three semicircles of radius $\frac{3}{4}$, since it is the shape of the curve. Hence, the ideal result would be a centred rectangle with one side having length 1 and one side having a length of $\frac{9}{4}\pi$. The intrinsic coordinates of the data set can be scaled and centred to match this plane. Afterwards the importance factors of a PCA of this plane can be compared to the ones of computed by the DRMs. Since the data is intrinsically two dimensional as in the previous example, the neighbourhood size was also chosen as $k = 10$ for all approaches and the results are show in Tab. 5.2.

	Ideal	PCA	LLE	LTSA	MLLE	Isomap	PTU	ENLM	GNLM	PTNLM
Imp. factor 1	64.942	32.987	64.60	64.020	64.063	66.045	63.847	32.987	66.100	63.850
Imp. factor 2	18.249	18.251	16.91	18.248	18.241	18.906	18.460	18.251	18.820	18.280
Imp. factor 3	-	16.628	3.857	3.194	1.542	-	-	16.628	-	-
Imp. factor 4	-	-	2.764	0.001	0.787	-	-	-	-	-

Table 5.2: Importance factors for the S-Shape data set computed by the different DRM approaches with $k = 10$.

This time the PCA deviates strongly from the ideal result, it is even the worst result together with ENLM which results in the same embedding, since the target dimension is large enough, the method is performing like a linear method.
All nonlinear approaches perform better, while the graph-based methods Isomap and GNLM overestimate the importance factors and the tangent-based methods LTSA, PTU and PTNLM slightly underestimate them. The LMs perform slightly worse than the nonlinear approaches of the other classes, as they have significant third and even fourth dimensions, but they are still much better than the one computed by PCA.
As for the linear example, the importance factors are of different magnitude, so the low dimensional coordinates should be well defined up to their sign. Likewise, the low dimensional coordinates can be aligned if needed and then easily compared. This comparison is shown in Fig. 5.5.
The shortfall of the PCA and ENLM methods to obtain a two dimensional embedding is clearly visible from the dark colour. Here the ”S” is just rotated, but not unrolled. For the nonlinear methods, LLE is the worst, but still significantly better than the linear variants. The LMs seem to perform worse than the pairwise distance-based classes MDS and NLM. Finally, the best results are obtained with the parallel transport variants, with the PTNLM result being close to optimal.



(a) LLE (b) LTSA (c) MLLE

(d) PCA (e) Isomap (f) PTU

(g) ENLM (h) GNLM (i) PTNLM

Figure 5.5: DRM results for the S-Shape data set. The 1 000 data points were sampled with the Quasi-Random method and then projected using the formula of Eq. (5.3). The first row shows the result of the LM-approaches, the second those of the MDS-methods, and the last row those of the NLM-variants.

This method of evaluating embeddings by computing sample distances can be employed when the ideal result is known. For practical applications, where the desired outcome is generally unknown, different criteria must be applied, to assess the embedding quality. Several criteria are published in literature, a good overview is given in [LV10]. Most of these criteria focus on different aspects of the embedding, e.g., whether an embedding preserves neighbourhood ranks or not [LV09]. Ideally, the utilised criterion should match the purpose or aim for which the DR is employed.
In this work, DR is used to understand underlying effects and to measure correlations with the difference operation. The core idea for the visualisation of the underlying effects as well as for the nonlinear difference measures is the approximation of high dimensional data points as an image of the generating function $\widetilde{F}(\tilde{x})$ for possibly new low dimensional coordinates $\tilde{x}$. Thus, the embeddings should be assessed with regard to their capabilities to obtain such a high dimensional data point for low dimensional coordinates. A new criterion is introduced, to measure these capabilities.
Given an input data set Y and low dimensional coordinates $\widetilde{X}$ computed by a DRM, a reduction score is computed based on the mean reconstruction residual for all sample points y_i and $\tilde{x}_i$, respectively. For the linear PCA, the reconstruction $\widetilde{F}(\widetilde{x_i})$ can be computed as described in Section 3.2. For the nonlinear methods, the LLI and LAI approaches are used, as described in Section 3.3.2.2 and Section 3.3.3.2, depending on the preserving properties of the method. Both of these interpolation approaches use the neighbourhood $\mathcal{N}(\tilde{x}_i)$, which may be different from $\mathcal{N}(y_i)$ and, most importantly in this work, it holds that $i \notin \mathcal{N}(\tilde{x}_i)$. Hence, the reconstruction error is in general not zero for the sample points. The mean of the reconstruction errors is computed and afterwards divided by the mean of the pairwise Euclidean distances in the high dimensional space. This yields a new score ξ_{DRM}, which is independent of the unit scale in which the data is given:

$$\xi_{\mathrm{DRM}}\left(Y, \widetilde{X}\right) := (s-1)\frac{\sum_{i=1}^{s}\|y_i - \widetilde{F}(\tilde{x}_i)\|_2}{\sum_{i=1}^{s}\sum_{j=1}^{i}\|y_i - y_j\|_2} \tag{5.4}$$

The smaller this score is, the better are the generating properties of the DRM for the given samples. While this new score cannot evaluate whether the dimensionality has been captured correctly, it provides an aid for the analyst to decide whether the embedding should be used for further analysis steps. And since it depends only on the given sample points and the computed approximations, it can also be consulted for data sets where the intrinsic structure is unknown. The scores for the two manifolds and the different DRMs are listed in Tab. 5.3.

DRM / Data	PCA	LLE	LTSA	MLLE	Isomap	PTU	ENLM	GNLM	PTNLM
Plane	3.73e-14	1.19e-4	3.15e-6	7.37e-5	3.33e-3	2.05e-7	5.34e-7	2.18e-3	2.05e-7
S-Shape	4.25e-14	5.75e-3	6.27e-3	5.35e-3	8.66e-3	3.36e-3	1.29e-6	3.76e-3	2.42e-3

Table 5.3: Reduction scores for the different DRMs and the two artificial examples, computed with $k = 10$.

For both data set, all methods yield good results. On the linear Plane data, LLE, Isomap and GNLM perform slightly worse than the other methods. The nonlinear S-Shape data set shows a deficit of this score: Though the PCA and the ENLM failed to determine the two dimensional intrinsic structure, their generating properties are very good as they utilise a three dimensional embedding. Despite of this deficit, this score can still be helpful to decide when not to continue with an analysis using a low dimensional embedding or comparing the different embedding of the same dimension with each other.

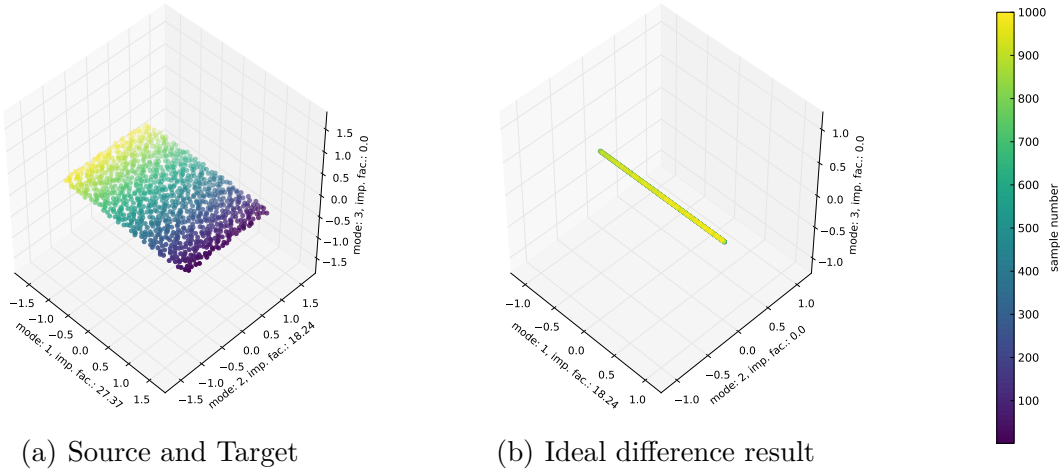
While the optimal method for DR for the first step of the analysis of simulation results is yet not known, the evaluation in this section has shown two initial insights: First, MLLE did outperform LLE on all investigated examples. Since both methods have the same goal of preserving local neighbourhood weights, this work focusses on the MLLE.
Second, if the target low dimension d is large enough for the pairwise Euclidean distances to be preserved without loss, PCA and ENLM provide the same result. As the target dimension is usually overestimated in the analysis of simulation results, this will often be the case. For this reason, this thesis focusses on the PCA.

5.1.3 Evaluating the Results of Difference Operations

The second step of the Extended Workflow is the difference operation, which investigates the correlation between a source and a target data set. Two nonlinear difference operations have been introduced, the DLLI in Section 4.2.1 and the DLAI in Section 4.2.2. Even though the DLAI is designed for tangent space-based methods and is expected to perform better on these approaches, both operations can theoretically be combined with all nonlinear DRMs in the first step. Thus, all combinations of difference operations and reduction methods are computed. For each of the combinations, a modification matrix $M \in \mathbb{R}^{s \times s}$ is obtained and applied to the new data set $\mathcal{Y} \in \mathbb{R}^{\mathcal{D} \times s}$. The resulting $\mathcal{Y}(I_s - M)$ is then evaluated.
Similar to the results obtained from the different DRMs in the last section, the results of the two difference operations can also be judged in two aspects: For the artificial data sets, the ideal results are known, so the low dimensional new coordinates resulting from the difference operation can be compared to these ideal results.
In practical applications, where these ideal results are unknown, the difference measures δ_{spec} and δ_{var} as introduced in Eq. (4.2) and Eq. (4.3) can be computed in order to get an easily understandable quantitative comparison.
In the following three examples with different dependencies between source and target data set are evaluated: a perfectly linear dependency, a nonlinear dependency, and a random, nearly uncorrelated relation.

5.1.3.1 Linear Example

The first correlation to be investigated is the perfectly linear example. Source Y and target $\mathcal{Y}$ for the difference operations in this example are the same data set, namely the Plane data set of Eq. (5.2). The 1 000 samples were generated with the Quasi-Random sampling introduced in Section 5.1.1. Since the data set is a linear manifold, PCA can perfectly capture the underlying structure, and since target and source are identical, the dependency is also linear and the DPCA yields the ideal result.
In this test, the first effect is subtracted. Since there is a clear one-to-one relation, subtracting the first mode of the source data set should eliminate the first mode of the target data set, setting the first coordinate of all points to zero. The second coordinate should remain unchanged, i.e., the second mode becomes the first mode with the same importance factor. Both the given data set and the ideal result as obtained by the DPCA are visualised in Fig. 5.6. For illustration purposes, the PCA is then applied to the result to compute the remaining importance factors as well.

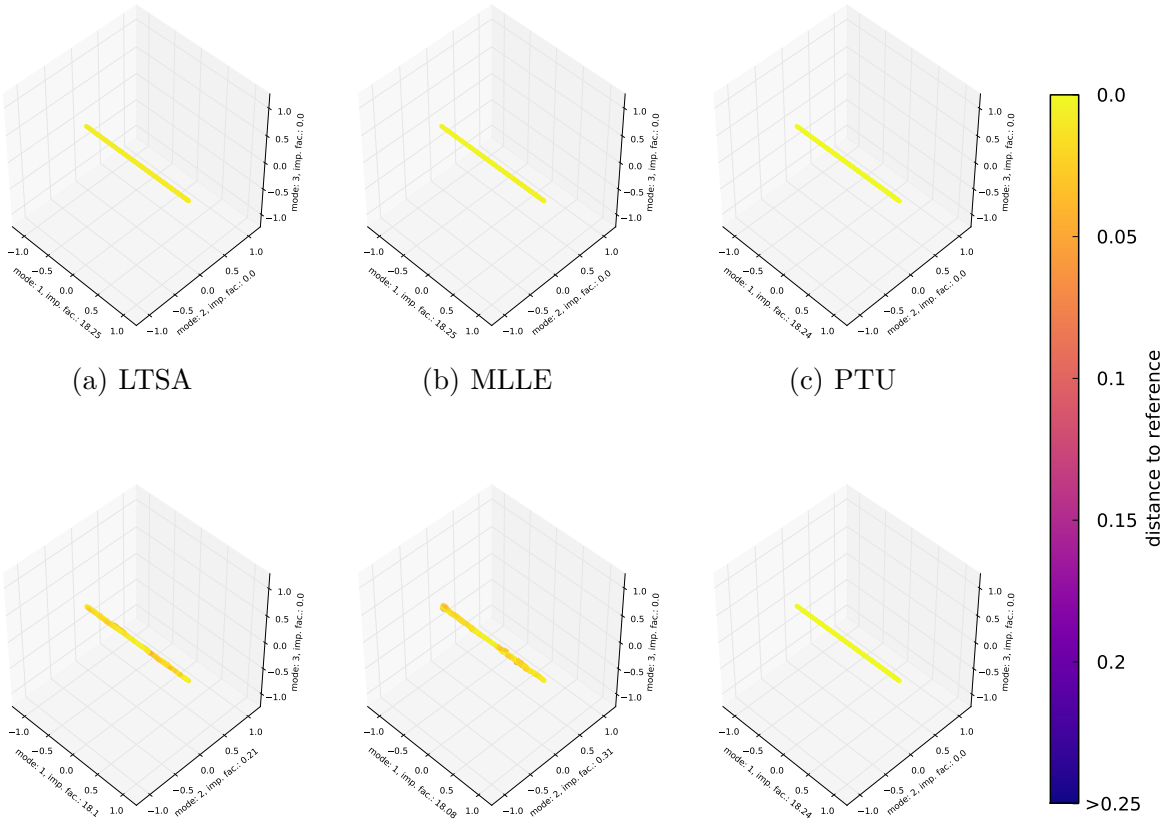


(a) Source and Target (b) Ideal difference result

Figure 5.6: Difference example Plane. The first picture shows the PCA result for the Plane data, that is source and target for the difference operation. The second picture shows the desired outcome where the data is projected onto the orthogonal complement of the first principal component.

The first nonlinear difference approach is the DLLI. First all nonlinear DRMs were applied to the input data set with $k = 10$, yielding the results displayed in the last section. Then, the first mode was subtracted using the respective M_{DLLI} calculated for the same neighbourhood size $k = 10$ as introduced in Section 4.2.1 with the additional normalisation enhancement of Section 4.2.3. The resulting low dimensional coordinates, displayed in Fig. 5.7, were then oriented with PCA, to obtain a standardised representation and comparable importance factors. In this case, the linear approach is always used to get an isolated nonlinear effect from the difference opera-

tion only and no side effects from an additional nonlinear DR.
While the overall results are good, the graph-based approaches Isomap and GNLM have local errors as a consequence of the small local distortions that were already visible in Fig. 5.4. The parallel transport variants yield results that are visually indistinguishable from the perfect result.



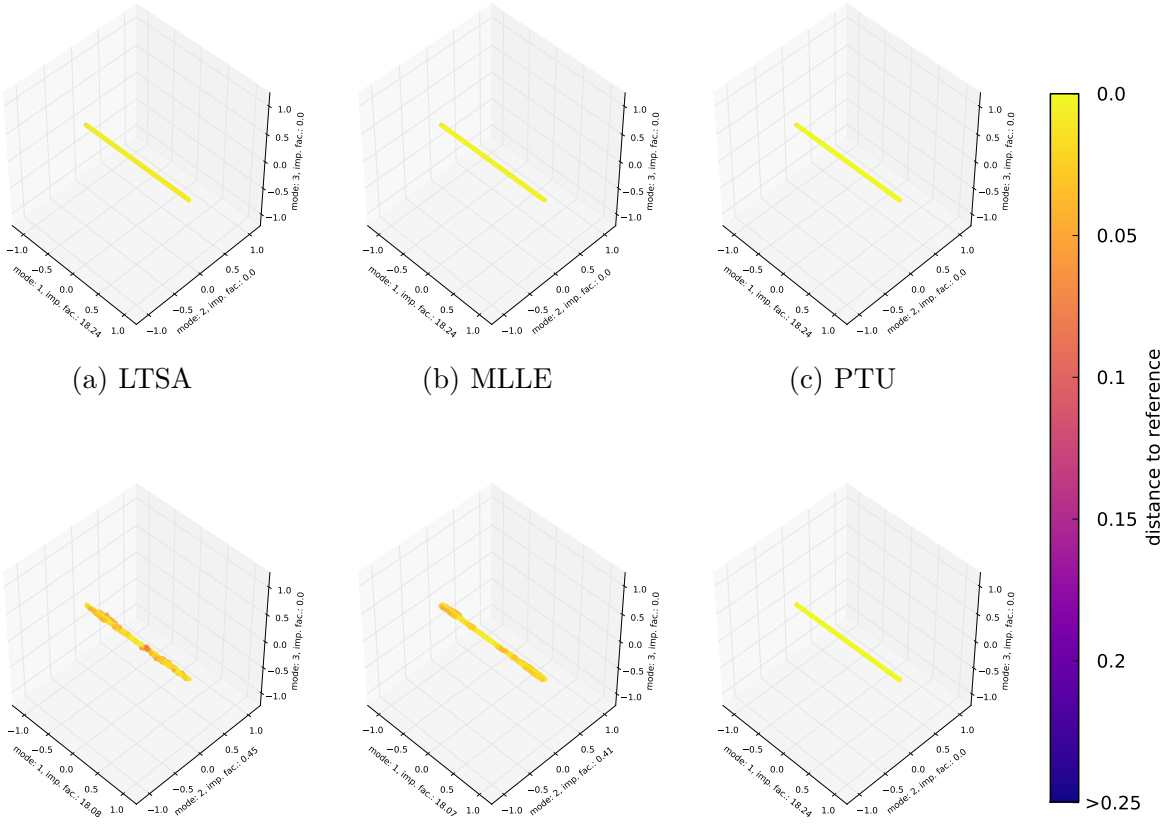
(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.7: DLLI results for Plane data set. The colour shows the difference to the DPCA result of Fig. 5.6.b, which is the reference for the desired outcome.

The second difference method is the DLAI approach. It was tested in the same way as the last method: The first mode of all DRMs for the data set was determined and then M_{DLAI} with $k = 10$ and normalisation enhancement was calculated, as explained in Section 4.2.2 and Section 4.2.3 respectively. As can be seen from the visualisation in Fig. 5.8, the results are in general comparable to the ones obtained from the last difference method. Surprisingly, the results for the MLLE approach are marginally better than the ones of the LTSA method, even though the DLAI was specifically designed for tangent-based methods such as LTSA, validating the evaluation of all possible combinations of DRMs and difference methods. The results for the graph-based DRMs appear to be slightly worse than the ones obtained in

combination with DLLI, but a detailed inspection has shown that these are also due to the local distortions mentioned before.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.8: DLAI results for Plane data set. The colour is again the difference to the DPCA result of Fig. 5.6.b.

Though small errors were visible for some DRMs, both difference methods performed well on the linear example. This was to be expected, as the manifold is very well sampled, and all DR approaches were able to sufficiently capture the underlying structure. Nonetheless, since the actual nature of the solution manifold is unknown in practical applications, it is important for the nonlinear methods to also handle linear dependencies satisfactory.

5.1.3.2 Nonlinear Example

The second example contains a nonlinear dependency between two different high dimensional data sets that are intrinsically the same. First, the 1 000 sample points are Quasi-Randomly sampled in $\mathbb{R}^2$ and then projected into the high dimensional space with two different generating functions.
The source data set is the S-Shape as introduced in Eq. (5.3) and its first mode should

be subtracted from the target data set. For the target data set, the Heated Swissroll manifold is used. It is a variant of the very popular Swissroll example, which is an Archimedean spiral extruded into one direction [TDSL00]. The "Heated" version is additionally curved in its extrusion direction [LV07].
Several variants of the Swissroll example exist in literature, e.g. in [ZW07], [ZQZ11] and [BYF+19], and the various publications often differ in two parameters: the extrusion length and the offset to the centre of the spiral. For the investigation in this section, a constant offset of $c = 0.05$ was used. The Heated Swissroll with extrusion length of one and offset $c \in \mathbb{R}$ is defined as $f_{\text{hroll},c} : [0,1]^2 \to \mathbb{R}^D$, with $\phi_{\text{hroll}} : [0,1] \to \mathbb{R}$:

$$\begin{aligned} \phi_{\text{hroll}}(x_2) &:= \frac{3}{2}(x_2 - \frac{1}{2}) \\ f_{\text{hroll},c}(x) &:= \begin{pmatrix} 2(1+\phi_{\text{hroll}}(x_2)^2)\sqrt{x_1+c}\ \cos(4\pi\sqrt{x_1+c}) \\ 2(1+\phi_{\text{hroll}}(x_2)^2)\sqrt{x_1+c}\ \sin(4\pi\sqrt{x_1+c}) \\ 2x_2 - 1 \\ \mathbb{0}_{D-3} \end{pmatrix} \end{aligned} \tag{5.5}$$

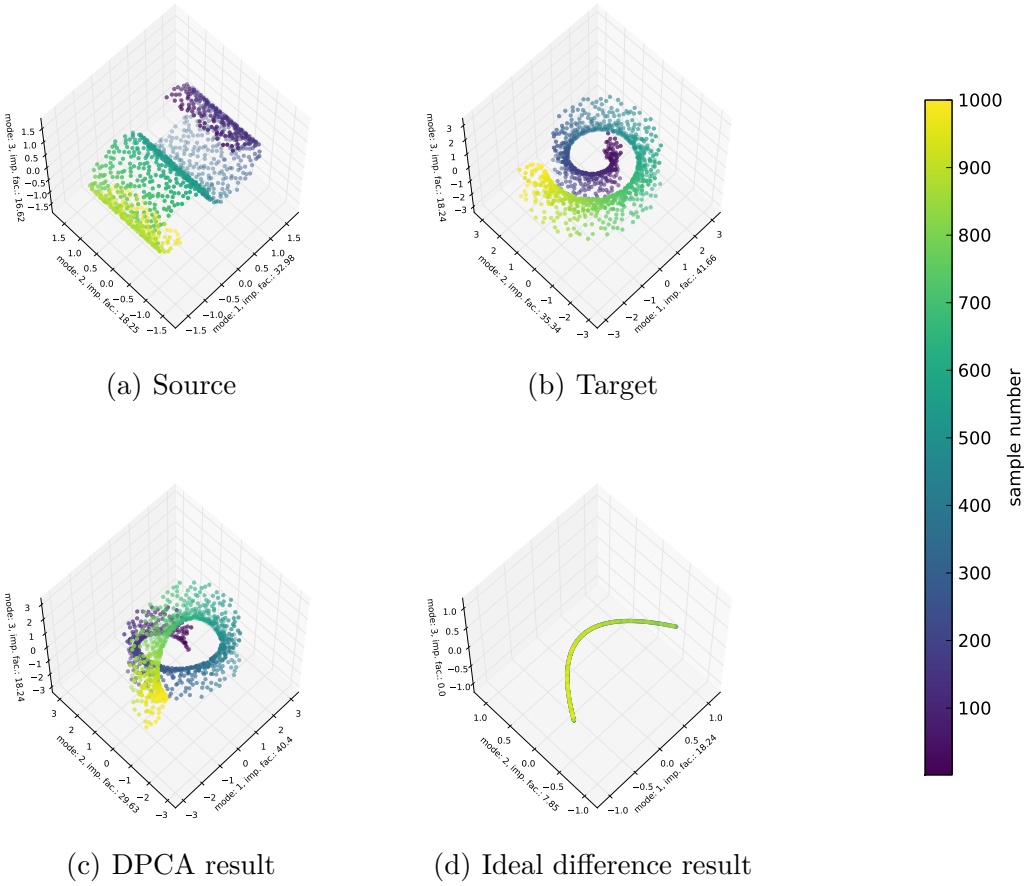
Before computing the difference operations by subtracting the first mode, the desired outcome can be calculated by performing a modified sampling and then applying the generating function of the target: The data is first "medianified" in the low dimensional data space. This can be viewed as the opposite of centralising, where the mean is subtracted, here everything is set to the mean value:

$$\check{X} := \frac{1}{s} X \mathbb{1}_s \mathbb{1}_s{}^\mathsf{T}$$

Then, the ideal outcome of the difference operation, i.e. removing the variance in the first e modes, can be computed by:

$$f_{\text{hroll},c}\Big(\text{diag}\big(\underbrace{0,...,0}_{e},\underbrace{1,...,1}_{d-e}\big)X + \text{diag}\big(\underbrace{1,...,1}_{e},\underbrace{0,...,0}_{d-e}\big)\check{X}\Big)$$

Since the first $e = 1$ most important effect should be removed, this means that the variation along the spiral should be removed, while variation along the extrusion curve should be preserved. The input data sets and the ideal outcome are visualised in Fig. 5.9. Furthermore, the result of the DPCA approach is displayed, showing that the linear difference operation completely fails to detect the dependency between the data sets, resulting in a self-intrusion of the spiral, far of the ideal curve.



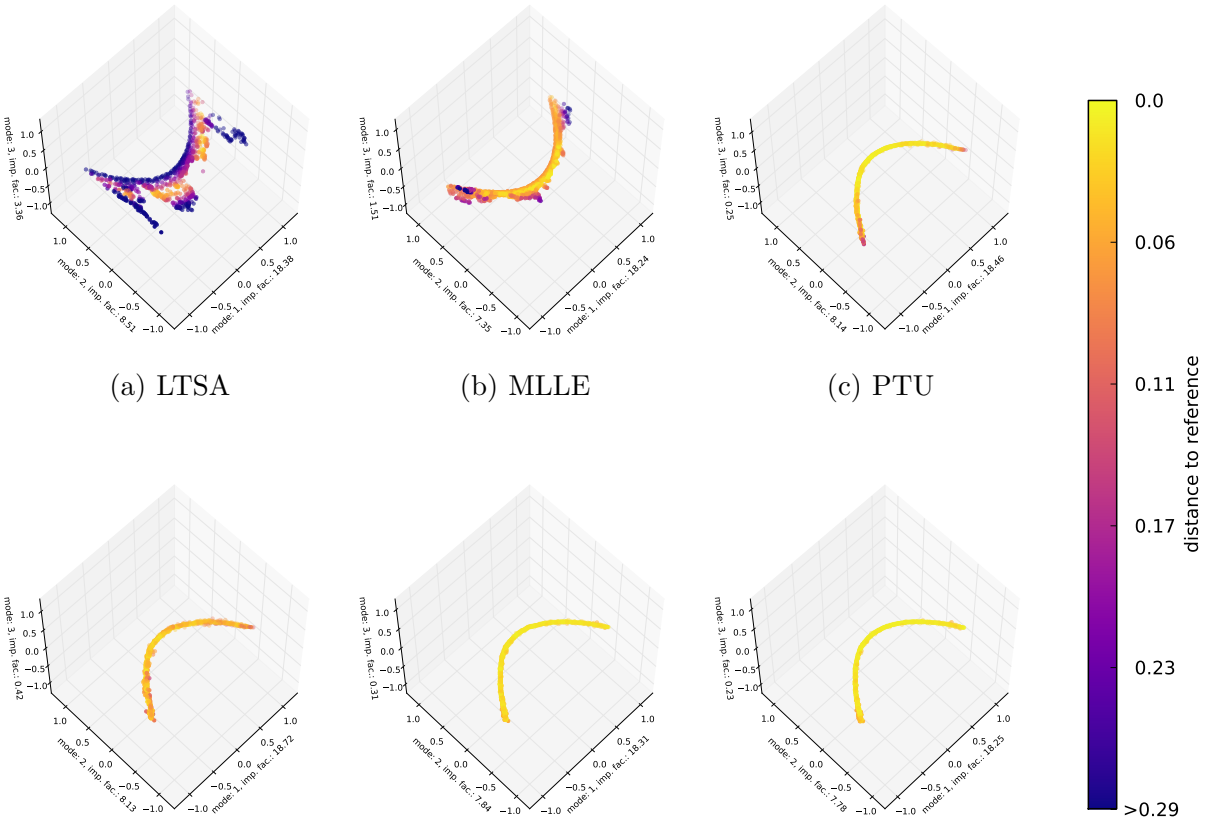
(a) Source (b) Target

(c) DPCA result (d) Ideal difference result

Figure 5.9: Difference example for S-Shape (a) and Heated Swissroll (b). The first picture shows the source and the second the target for the difference operation. The third picture (c) shows the DPCA result, while (d) finally shows the ideal outcome of a difference operation that was calculated by medianifying the intrinsic coordinates prior to projection into the high dimensional space. The points are coloured to their sample number, showing which points are intrinsically identic.

The disastrous result of the linear method underlines the need for nonlinear methods. The first nonlinear difference methods is the DLLI approach. Similar to the last example, all nonlinear DRMs are initially applied to the first data set with $k = 10$ resulting in the low dimensional embeddings visualised in Fig. 5.5. Then the corresponding modifications $\mathcal{Y}(I_s - M_{\text{DLLI}})$ were computed and afterwards oriented with a final application of PCA to increase comparability. These oriented results are visualised in Fig. 5.10.
The LMs show large errors for multiple sample points with LTSA being the worst of all methods. MDS approaches yield better results and NLM methods generate curves comparable to the ideal result. For all classes, the errors tend to increase towards the ends of the curve, which can be explained by the fact, that the curvature increases

along the spiral of the Heated Swissroll. This means that the variation is stronger towards the ends while being relatively smaller towards the centre of the curve.

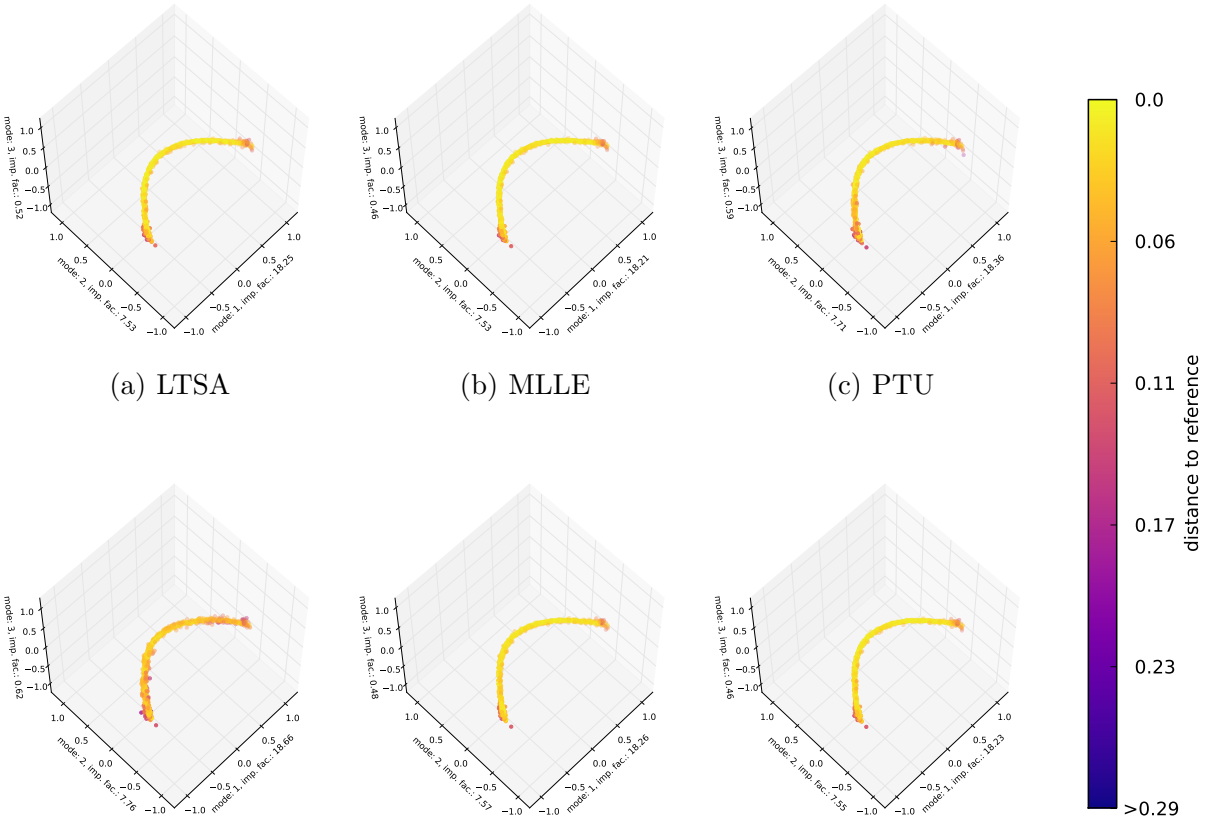


(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.10: DLLI results for the S-Shape and Heated Swissroll data sets. The colour shows the difference to the ideal result shown in Fig. 5.9.c

Similarly, the DLAI was applied in combination with the different low dimensional embeddings and the same target data set. The results, which were subsequently oriented by a PCA application, are displayed in Fig. 5.11. Here all methods provide comparable and very good results, with the MDS variants being slightly worse than the others. Again, an error concentration towards the ends of the low resulting curve can be observed, caused by the aforementioned increasing curvature along the Swissroll's spiral.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.11: DLAI results for the S-Shape and Heated Swissroll data sets. Colour is again the difference to the ideal result shown in Fig. 5.9.c

After the very simple linear example in Section 5.1.3.1, the newly developed difference methods also performed well on the more challenging nonlinear example. Here, the DLLI provided results of very different quality, while the DLAI obtained similar and good results for all investigated nonlinear DRMs.

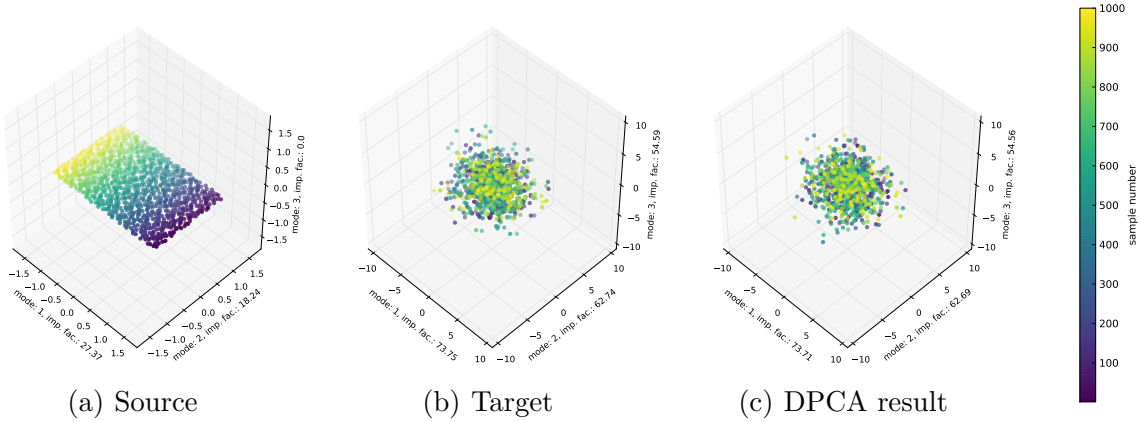
5.1.3.3 Random Example

The last example is the most challenging application, where the difference methods are applied to two data sets of 1 000 samples that are intrinsically as different as possible. Here the source data set is again the Quasi-Randomly sampled two dimensional plane data set of Eq. (5.2). The target is a set of five dimensional Orientable Noise data. Orientable Noise is a new data set where all coordinates are generated with a normal distribution and expectation of zero $\mathcal{N}(0, \sigma^2)$, but with an increasing standard deviation σ for each dimension. This way, the orientation of the PCA result for this data set is unique up to the sign of the axis. The generating function can be stated as $f_{\text{onoise}} : [0,1]^0 \to \mathbb{R}^D$, where $[0,1]^0$ indicates, that it is random and not depending

on any input value other than the number of samples to be generated.

$$f_{\text{onoise}}(x) := \begin{pmatrix} \mathcal{N}(0,1) \\ \vdots \\ \mathcal{N}(0,i) \\ \vdots \\ \mathcal{N}(0,D) \end{pmatrix} \tag{5.6}$$

The data set in this investigation was generated with $D = 5$ and there is no significant correlation with the expected two dimensional plane data. Thus, the importance factors should not be noticeably affected when subtracting the first mode of the source from the target. However, local coordinate changes may occur, and since the data is randomly generated, some minor amount of correlation cannot be prevented. Since the Orientable Noise can be interpreted as a linear five dimensional data set with random sampling and the identity as the generating function, the DPCA is assumed to provide the correct result, since two close to uncorrelated linear manifolds are involved. A visualisation of the data sets as well as the DPCA result is given in Fig. 5.12.



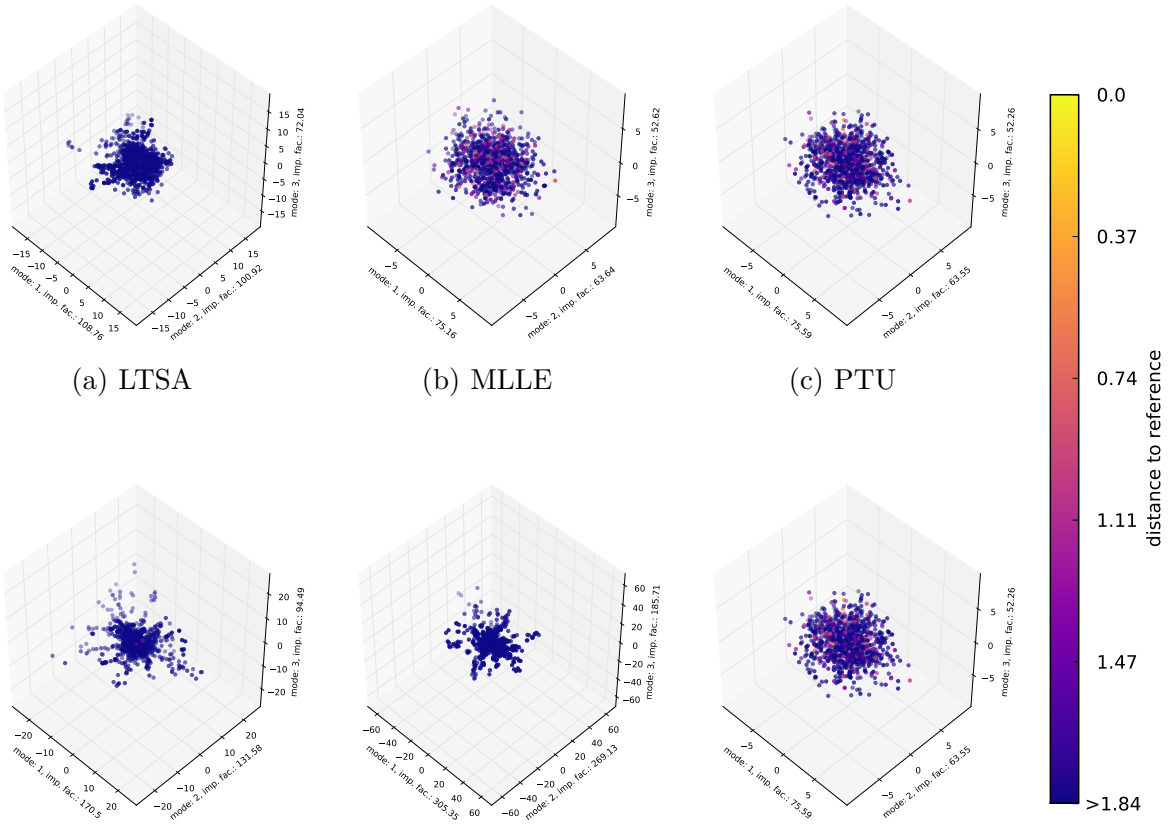
(a) Source (b) Target (c) DPCA result

Figure 5.12: Difference example for Plane and Orientable Noise. The first picture shows the source and the second the target for the difference operation. The third picture shows the outcome of the liner DPCA operation, which is assumed to be the correct result.

The investigation in Section 5.1.2 has shown that the nonlinear DRMs can capture the intrinsic structure of the Plane with a neighbourhood size of $k = 10$ very well. When subtracting the first mode of the Plane from the new randomly generated target with the DLLI and the same number of neighbours $k = 10$, the results in Fig. 5.13 are obtained.

Though the source data set is the same as in the linear dependency example of Section 5.1.3.1, the results for this example are much worse. In combination with LTSA, Isomap and GNLM, the DLLI significantly increases the variance in the target data

set, multiplying the largest importance factor by up to a factor of four, e.g., for GNLM, where the importance factor should actually have stayed unchanged. It is noteworthy that the graph-based results have star-shaped pattern where some groups of points are pushed outside from the centre of the manifold. This is most likely caused by extrapolations with large weights for nearest neighbours in the plane. The parallel transport variants perform better, and MLLE is the best performing DRM in combination with DLLI, while still increasing the first two importance factors.



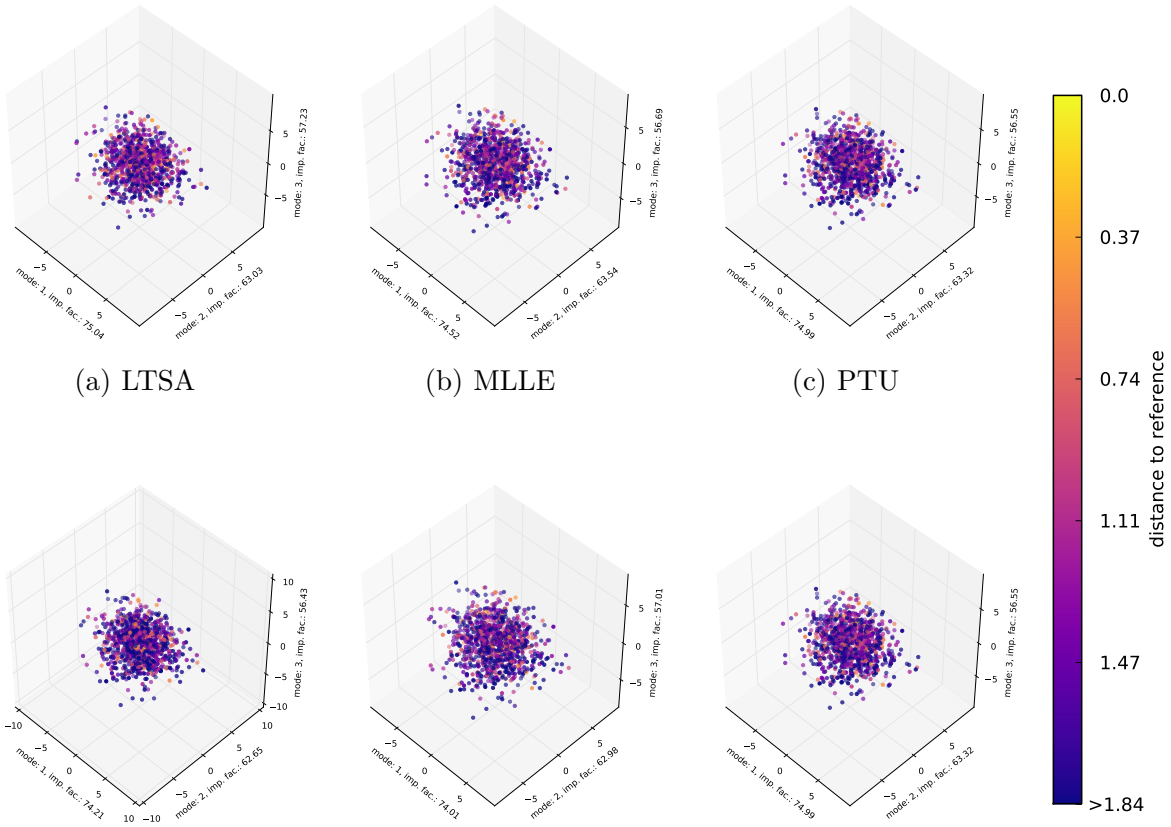
(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.13: DLLI results for the Plane and Orientable Noise data sets. The colour shows the difference to the DPCA result of Fig. 5.1.3.3.c, which is the desired outcome as the reference. The reference importance factors are $73.71, 62.69$ and 54.56.

Next, the DLAI approach in combination with the normalisation enhancement is evaluated on this example with challenging random data. All parameters were chosen as in the previous examples to guarantee good comparability. The results for the different DRMs and a neighbourhood size of $k = 10$ is displayed in Fig. 5.14. In contrast to the last nonlinear method, this nonlinear approach is yielding comparable results in combination with all different DR approaches. Visually, LTSA is yielding the best result, while GNLM has the least increase in importance factors. Though the

importance factors still marginally increase for all DRMs, the overall performance is much better and closer to the linear result. A slight increase of importance factors in the case of uncorrelated data set might not be avoidable as the generalised difference operations are no orthogonal projections, see Section 4.2.3.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.14: DLAI results for the Plane and Orientable Noise data sets. The colour is again the difference to Fig. 5.1.3.3.c. The importance factors of the reference are 73.71, 62.69 and 54.56.

The investigations in this section have led to further insights: In the best case of a clean linear relation between target and source, both nonlinear difference operations yield similar results close to the ideal result. Furthermore, the nonlinear methods are capable of handling nonlinear correlations. In general, the results of the DLLI method are varying stronger for the different DRMs than those of the DLAI approach. Most importantly, the DLLI strongly increases the importance factors for random or unstructured data sets.

Based on these insights, only the DLAI method will be investigated for further examples, as it showed superior capabilities. The fact that the best DRM was always different for each example underlines the necessity to apply different approaches and not to rely on a single method.

5.1.4 Methodology Impact

The different DRM approaches vary in the methodology of the applied reduction, i.e. how they aim to capture the underlying structure of the data. This methodology determines which properties are preserved and affects the outcome of the reduction process. As this outcome is used to define the difference operation, this second step is also influenced by the underlying model of the reduction method.

In [BYF+19] an example of a nonlinear Petals data set is used to highlight the different properties of the reduction methods. A similar data set can be created with the generating function $f_{\text{petals}} : [0,1]^2 \to \mathbb{R}^D$, with $\phi_{\text{petal},j} : [0,1]^2 \to \mathbb{R}^2, j \in \{1,2,3,4\}$:

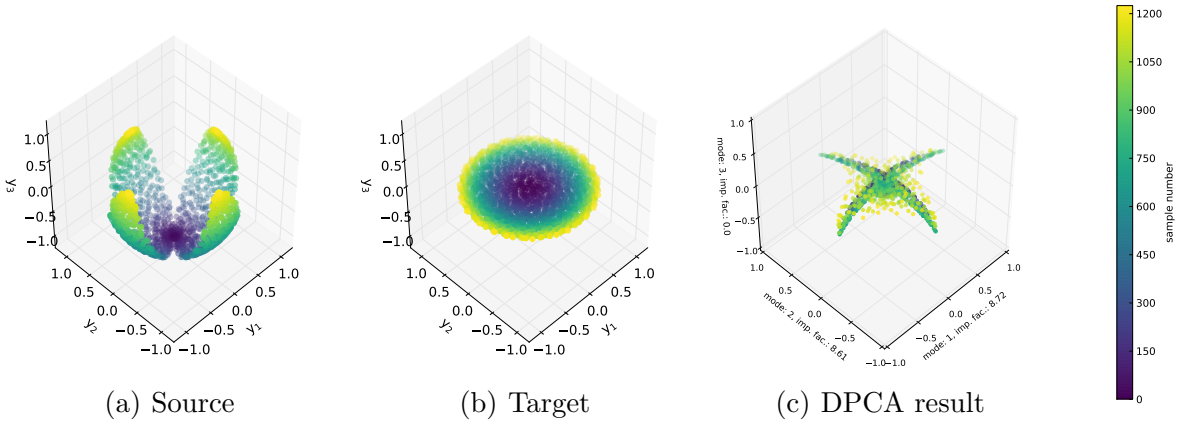
$$
\begin{aligned}
\phi_{\text{petal},1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2-\frac{1}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2-\frac{1}{2}\right)\right) \end{pmatrix} \\
\phi_{\text{petal},2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} -\sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2-\frac{3}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2-\frac{3}{2}\right)\right) \end{pmatrix} \\
\phi_{\text{petal},3}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2-\frac{5}{2}\right)\right) \\ -\sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2-\frac{5}{2}\right)\right) \end{pmatrix} \\
\phi_{\text{petal},4}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2-\frac{7}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2-\frac{7}{2}\right)\right) \end{pmatrix} \\
f_{\text{petals}}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \begin{cases} \phi_{\text{petal},1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, 0 \le x_2 < \frac{1}{4} \\ \phi_{\text{petal},2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{1}{4} \le x_2 < \frac{1}{2} \\ \phi_{\text{petal},3}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{1}{2} \le x_2 < \frac{3}{4} \\ \phi_{\text{petal},4}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{3}{4} \le x_2 \le 1 \end{cases} \\ -\cos\left(\frac{2}{3}\pi x_1\right) \\ \mathbb{0}_{D-3} \end{pmatrix}
\end{aligned}
\tag{5.7}
$$

The resulting data set for 1 225 Quasi-Randomly sampled points is displayed in Fig. 5.15.a. A suitable target data set, to show the effect on the following difference operation, is the Disk data set created with the generating function $f_{\text{disk}} : [0,1]^2 \to \mathbb{R}^D$:

$$
f_{\text{disk}}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \sqrt{x_1}\cos\left(2\pi x_2\right) \\ \sqrt{x_1}\sin\left(2\pi x_2\right) \\ \mathbb{0}_{D-2} \end{pmatrix} \tag{5.8}
$$

Both of these data sets, the Petals as well as the Disk, are intrinsically two dimensional. This means that subtracting the first $e = 2$ modes of the Petal data set from

the Disk data set displayed in Fig. 5.15.b should ideally eliminate all variance and collapse all samples onto one point, if both were generated from the same intrinsic coordinates. The DPCA result in Fig. 5.15.c shows that the linear approach fails to obtain this result.

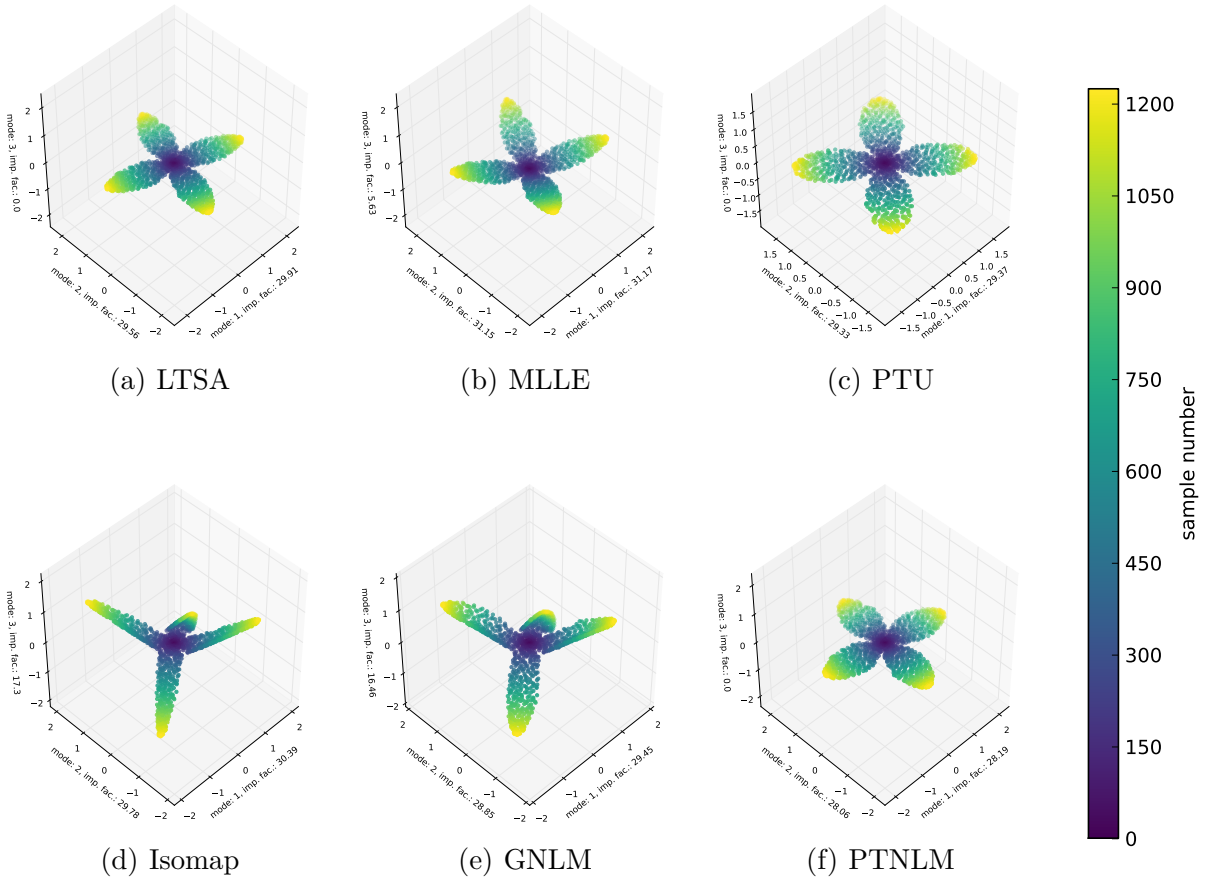


(a) Source (b) Target (c) DPCA result

Figure 5.15: Difference example for the Petals and the Disk data sets. The first picture shows the source and the second the target for the difference operation. The third picture shows the outcome of the liner DPCA operation for the first $e = 2$ modes.

The Petals data set is challenging for DR because of three properties. First, the high dimensional values f_{petals} depend nonlinearly on both intrinsic coordinates x_1 and x_2. These types of manifolds are defined as "non-developable" in [LV07] and can be challenging for some DRM approaches.
Second, the manifold is containing gaps between one petal and its neighbours. These gaps are problematic for graph-based approaches such as Isomap, because paths in the manifold are elongated by the detours around these gaps.
Lastly, the manifold contains bottlenecks in the centre and at the tips of the Petals, with a high density of sample points. This high density may pose a challenge for local methods such as LLE, because the local properties like interpolation weights can fail to capture the global structure, if all nearest neighbours are concentrated in a small area.
When applying the different DRMs, the resulting embedding is also different depending on the underlying methodology. The linear PCA fails to capture the intrinsic two dimensional structure of the data and just reorients the data set. The results for the nonlinear DRMs are displayed in Fig. 5.16.
Methods based on parallel transport distances, such as PTU and PTNLM, perform best, as they result in two dimensional embeddings with undistorted individual segments. These two approaches yield very similar embeddings up to a global rotation, which can occur when all dimensions are equally important.
The LM approaches LTSA and MLLE unroll the Petals data, but also truncate the individual segments, yielding shorter and slimmer leaves. Additionally, the MLLE

method also provides a significant but undesired third dimension.
An even bigger third dimension is present for the graph-based Isomap and GNLM approaches, which perform worst of all nonlinear methods. The detours in the manifold paths induced by the gaps between the individual segments cause an undesired distortion of the embedding that wrongly orientates and thins the segments.



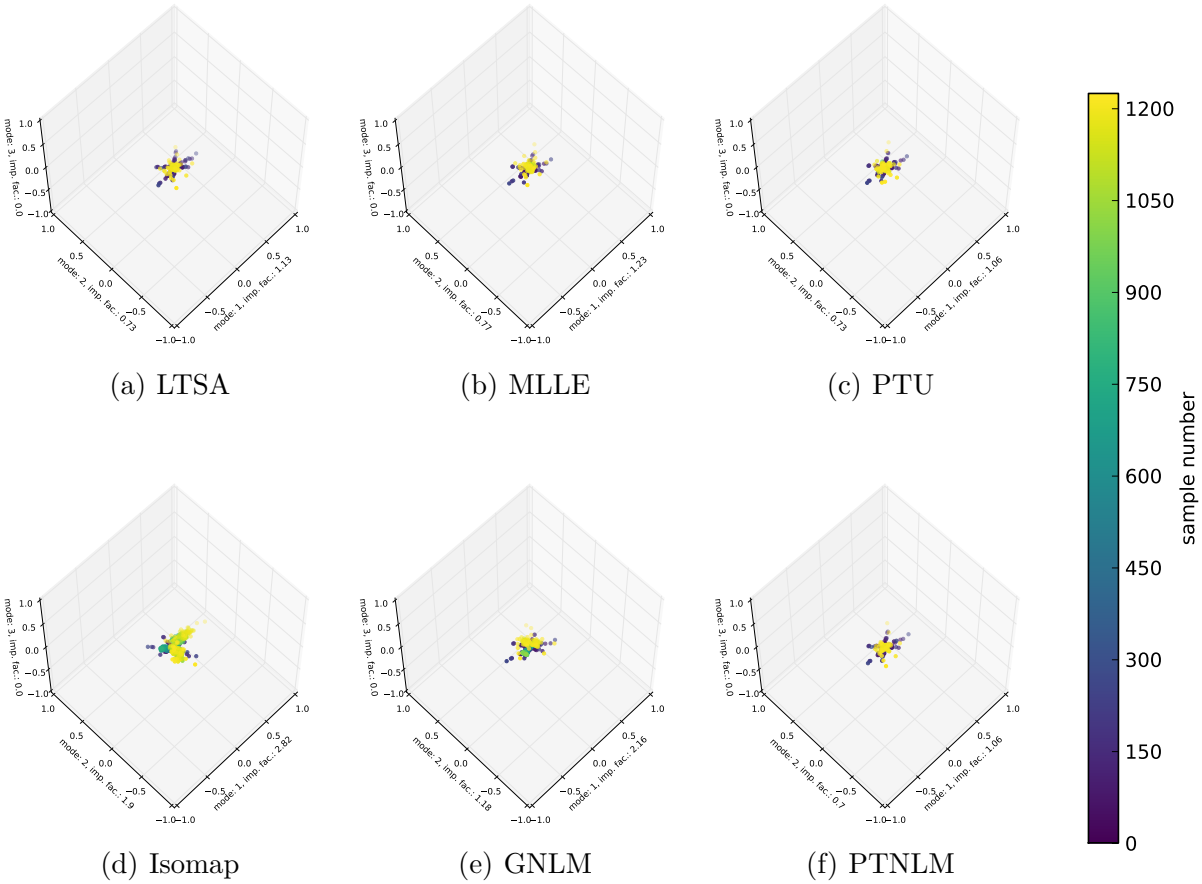
(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.16: DR results for the Petals data set for the different nonlinear approaches. The points are coloured according to their sample number.

The difference in the computed low dimensional embeddings directly affects the outcome of any following difference operation. For the nonlinear DRMs in combination with the DLAI approach as the difference operation, the results are shown in Fig. 5.17 and the corresponding difference measures are listed in Tab. 5.4.
Ideally, all points would be superposed in one single location and the δ-measures should be close to 100%. The better a DRM captures the underlying structure of the Petals data set, the more variance is removed from the Disk data set, i.e. the smaller the remaining importance factors are.
Although all nonlinear approaches yield better results than the linear DPCA, the graph-based Isomap and GNLM approaches yield worse results than the other meth-

ods. The LMs perform almost as good as the parallel transport variants with MLLE being slightly worse than LTSA. This matches the findings of the reduction step, where the methods had a similar order.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.17: DLAI results for the Petals and Disk data sets for the different nonlinear approaches.

Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
δ_{spec}	50.1%	93.5%	92.9%	83.8%	93.9%	87.6%	93.9%
δ_{var}	75.4%	99.7%	99.7%	98.1%	99.7%	99.0%	99.7%

Table 5.4: Difference result of the different approaches for the Petals as the source and the Disk as the target.

5.1.5 Additional Complexity

The investigations in the last section have shown that the capabilities of the DRM to capture the intrinsic structure of the source data set is essential and has a crucial influence on the performance of the following difference operation. Before approaching real simulation data, the reduction step should be evaluated on more challenging artificial examples with additional complexity. Amongst others, two properties can

be observed frequently in real application data that can affect the performance of the DRMs: The first is the existence of noise in the samples and the other is that the assumptions made in Section 3.1.4 are not entirely met. Both cases are briefly evaluated in this section.
It is important to note that both topics are only briefly investigated in a heuristic study and an in-depth evaluation would exceed the limited scope of this thesis. Hence, the cases are addressed to the extent that is relevant for the data used in this work.

5.1.5.1 Noisy Data Sets

Data sets in practical applications often contain some degree of noise. The reasons for the presence of noise in simulation data results are diverse, but, amongst others, include physical and numerical uncertainties [Mar99] as well as simulation process-related reasons [TM03].
Since the amount of noise is a priori unknown, a data analysis method should be able to produce reliable results up to a point where the noise is in the same magnitude as the information incorporated in the data. Hence, this section investigates the behaviour of the DRMs under noise.
For this investigation, the linear Plane data set of Eq. (5.2) was chosen as this is the only data set where the linear PCA can be reasonably included in the comparison to put the results of the nonlinear methods into a relation. Again, 1 000 sample points $x_i \in \mathbb{R}^2$ were sampled with the Quasi-Random approach of Section 5.1.1 and projected into the high dimensional data space using $y_i = f_{\text{plane}}(x_i) \in \mathbb{R}^D$ with $D = 5$. Furthermore, white Gaussian noise [LV07] with a varying standard deviation was added to the projected data. The standard deviation was chosen relative to the maximum value of the high dimensional coordinates. For a given noise level $\nu \in \mathbb{R}$, the data was generated by:

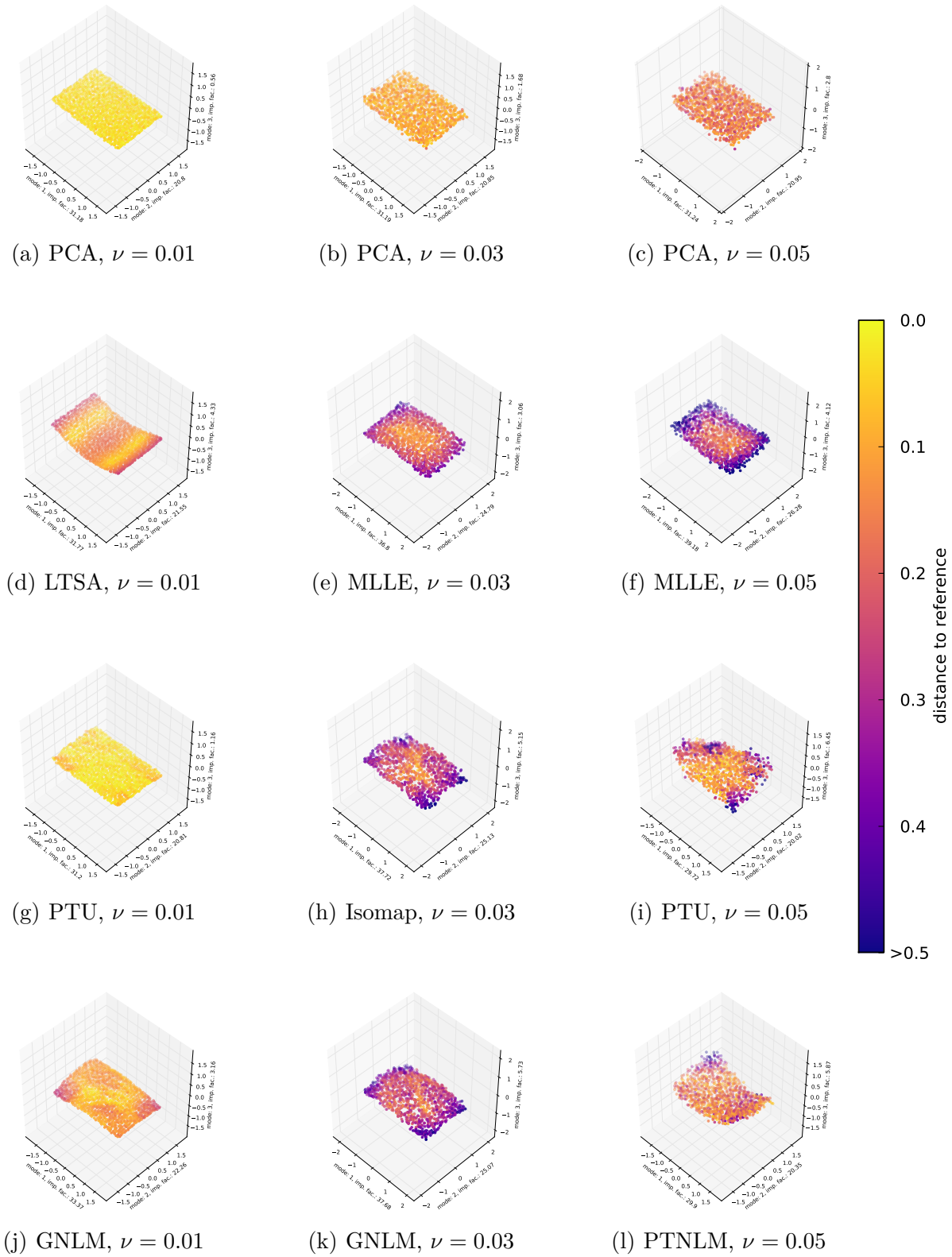
$$\begin{aligned} x_{\max} &:= \max_{ij} |x_{ij}| \\ \widetilde{y}_{ij} &:= \mathcal{N}(0, \nu x_{\max}) + y_{ij} \end{aligned}$$

It is important to note that while only the first two coordinates y_{i1} and y_{i2} contain actual information, all five coordinates y_{ij} per sample are affected by the noise.
Since the nonlinear methods performed reasonably well with a neighbourhood size of $k = 10$ for the unperturbed Plane in Section 5.1.2, the same number of neighbours was also used for the noisy data sets. The noise level ν was varied three times and the DRMs applied to the data set each time, with a target dimension of $d = D = 5$. To compare the different performances, the reduction scores for ξ_{DRM} of Eq. (5.4) were then computed for each method and each noise level. The resulting scores are listed in Tab. 5.5.

Noise \ DRM	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$\nu =$ 1e-2	2.37e-15	1.89e-2	1.89e-2	2.35e-2	2.28e-2	2.10e-2	2.26e-2
$\nu =$ 3e-2	1.05e-15	5.71e-2	5.65e-2	7.32e-2	6.66e-2	5.73e-2	5.24e-2
$\nu =$ 5e-2	6.52e-16	9.53e-2	8.64e-2	1.13e-1	1.18e-1	6.74e-2	8.83e-2

Table 5.5: DRM scores for the different approaches and the Plane data set with increasing white Gaussian noise. A neighbourhood size of $k = 10$ was used for all nonlinear DRMs and noise levels ν.

For the nonlinear DRMs, the growth of the score values matches the increase in the noise level very well. It is important to note, that the noise levels ν are multiplied with the maximum value, which for this data set is 1.497. The ξ_{DRM} scores are relative with regard to the average distance. That average distance is 1.318 and smaller than the maximum value.
Since the target dimension $d = 5$ is large enough to embed the data without any loss, the linear PCA only re-orients the data, yielding close to perfect scores for all noise levels. The nonlinear methods, on the other hand, perform an actual DR, i.e. the computed low dimension is smaller than five, resulting in errors depending on the information removed from the data set.
At the lowest noise level, the LM class with LTSA and MLLE perform best on this example. On the other hand, the NLM approaches yield the best scores at the highest noise level. The nonlinear MDS variants are resulting in the worst scores for all noise levels.
For the most part, these findings also match the visual representation in Fig. 5.18. Here, each column shows the results for one noise level. In each row the DRM with the best performance in the respective class is displayed, with the linear PCA in the first row, LMs in the second, nonlinear MDS approaches in the third and NLM methods in the last row.
The reference for the embeddings is the Plane without any noise and the PCA result shows the evenly distributed noise over the complete manifold.
On the contrary, the nonlinear methods provide results where the error is concentrating in certain areas. In general, LMs provide rather smooth manifolds as all local properties are preserved in a global least squares solution. The MDS methods can provide results where single points are poking out of the manifold, if the local noise was large in the input. As the NLM methods can be seen as a compromise between the two other classes, the results are also smoother than the MDS variants, but not as smooth as those of the LMs.
Overall, the visual impression matches the results of the ξ_{DRM} reduction scores, where methods with a low score also yield small errors in the embedding. The only exception is the PTU result for the lowest noise level, which produced a very good distance in the embedding but relatively bad reduction score. This may be due to the fact that the reduction score is calculated with regards to the noisy Plane data and the colouring of the embedding with respect to the noiseless data set.



(a) PCA, $\nu = 0.01$ (b) PCA, $\nu = 0.03$ (c) PCA, $\nu = 0.05$

(d) LTSA, $\nu = 0.01$ (e) MLLE, $\nu = 0.03$ (f) MLLE, $\nu = 0.05$

(g) PTU, $\nu = 0.01$ (h) Isomap, $\nu = 0.03$ (i) PTU, $\nu = 0.05$

(j) GNLM, $\nu = 0.01$ (k) GNLM, $\nu = 0.03$ (l) PTNLM, $\nu = 0.05$

Figure 5.18: DRM results for the Plane with increasing noise levels. The colour indicates the distance to the noiseless data set.

In conclusion, the linear approach seems to handle noisy data sets better than the nonlinear methods, at least for linear manifolds. The nonlinear DRMs can handle the small noise levels investigated in this section reasonably well. These small levels are motivated by the degree of noise in the application data used in this work, see Section 5.2.2 and Section 5.2.3. A study of stronger noise levels beyond the scale typically found in practical applications is left for further research.

5.1.5.2 Relaxing the Assumptions

In Section 3.1.4 several assumptions were made for the given data set. Some of these are easily satisfied in practical applications. For example, if the data is constant, i.e. all samples are the same, there is no need to analyse it, and if the data has no expectancy of zero, it can be centralised before applying the analysis methods. But other assumptions are not as easily satisfiable.
One more complex assumption is that the data lies on a single connected manifold of a fixed dimension. This assumption cannot easily be satisfied, if the data violates it and it is thus relaxed in this section to investigate the implications. Similar to the investigation in the last section, the topic can only be visited heuristically in the scope of this thesis. Hence, the violation of the assumption is only evaluated to a scale, which is relevant for the example data featured in this work. Data sets can violate the assumption in two different ways: connectivity and constant dimension.

First, the manifold can be disconnected, meaning that it consists of more than one connected component. While the recommendation is to investigate each component separately, the analyst may want to conduct a global analysis, resulting in an artificial connection of the multiple components. Such a case can be demonstrated with an artificial data set example consisting of two orthogonal planes, which are placed with a gap in between, as displayed in Fig. 5.19.

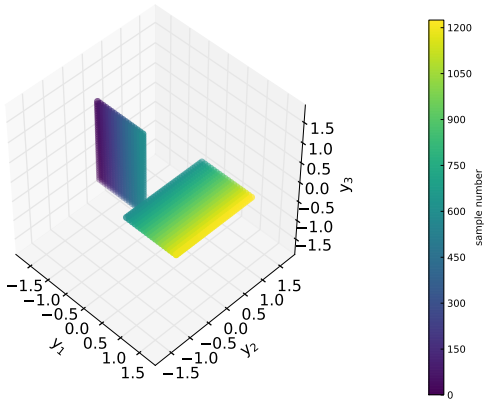


Figure 5.19: Visualisation of the example data set of the two orthogonal planes. The points are coloured according to their sample number.

This intrinsically two dimensional data set can be created with the following generating function $f_{\text{twoplanes}} : [0,1]^2 \to \mathbb{R}^D$, with $\phi_{\text{left}}, \phi_{\text{right}} : [0,1]^2 \to \mathbb{R}^3$:

$$\begin{aligned}
\phi_{\text{left}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} 3x_1 - 1.75 \\ 0 \\ 2x_2 - 1 \end{pmatrix} \\
\phi_{\text{right}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} 3x_1 - 1.25 \\ 2x_2 - 1 \\ 0 \end{pmatrix} \\
f_{\text{twoplanes}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \begin{cases} \phi_{\text{right}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 \geq \frac{1}{2} \\ \phi_{\text{left}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 < \frac{1}{2} \end{cases} \\ \mathbb{0}_{D-3} \end{pmatrix}
\end{aligned} \tag{5.9}$$

To evaluate the performance of the DRMs, an example data set with 1 225 sample points was generated with the deterministic grid-base sampling method introduced in Section 5.1.1 and then projected using the formula of Eq. (5.9). The results of the nonlinear DRMs for this data set are shown in Fig. 5.20.

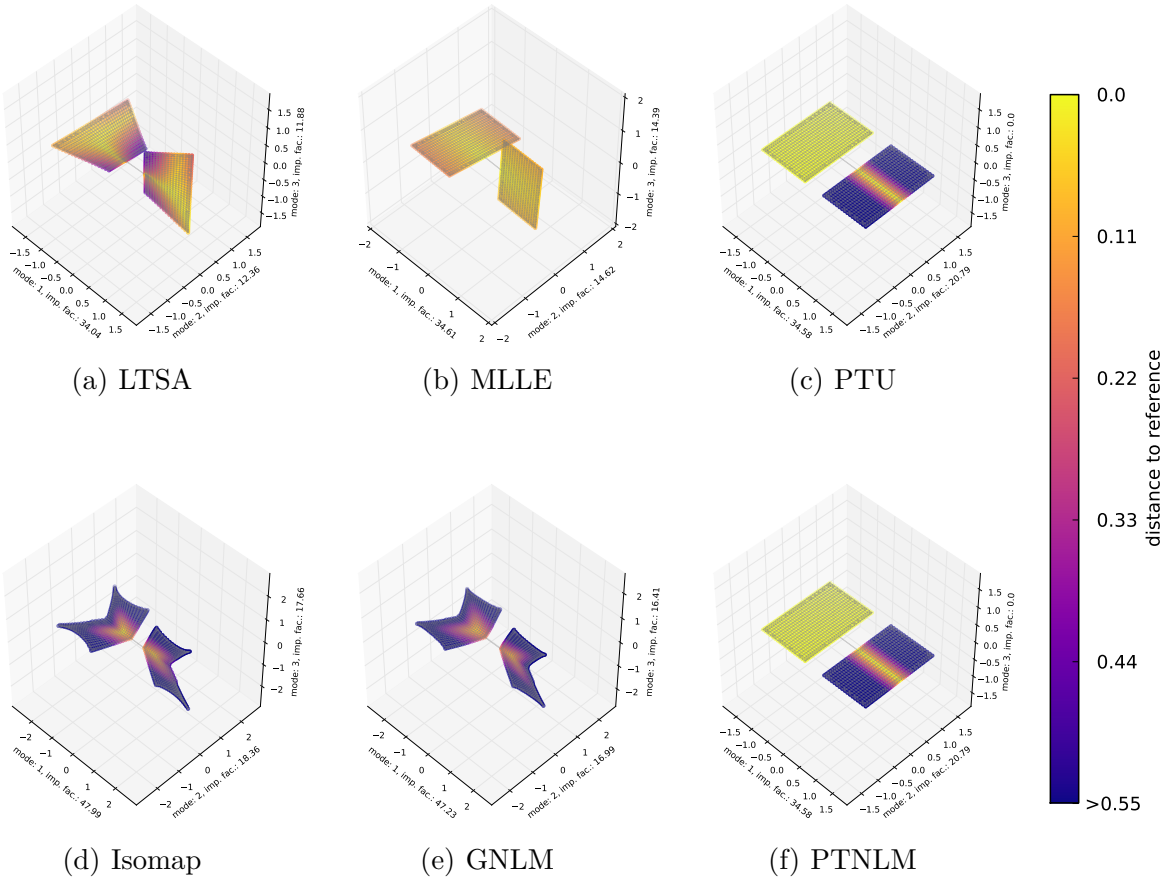
The challenge of this data set is that the two separate, disconnected manifolds must remain undistorted and must be correctly orientated to each other. Since the data set is linear, the linear PCA yields the correct result, which is identical to the input data set. All nonlinear approaches are affected by the neighbourhood graph constructed with an intentionally small neighbourhood size of $k = 4$, resulting in an enforced connection of the manifolds by a single edge.

The approaches of the LM class cope with the fact that two neighbourhoods contain one disproportionately large distance between the nearest neighbours, which locally distorts the manifolds. Apart from these local distortions, the results are acceptable.

In the results of the graph distance-based approaches Isomap and GNLM, global deformations can be observed, which result from the detours induced by the gaps in the neighbourhood graph. These detours will always align the two planes orthogonal to each other, even if they would be oriented different.

On the contrary, the parallel transport variants always align the two planes in the same orientation, regardless how they were originally placed. Fortunately, the correction of the geodesic paths prevents any distortion of the individual planes.

When interpreting results with more than one connected component, it is important to consider that the orientation of patches to each other depends more on the used nonlinear DRM approach than on the actual position to each other. This behaviour can be corrected by increasing the number of neighbours so that there are enough connections to determine the orientation of the components relative to each other.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

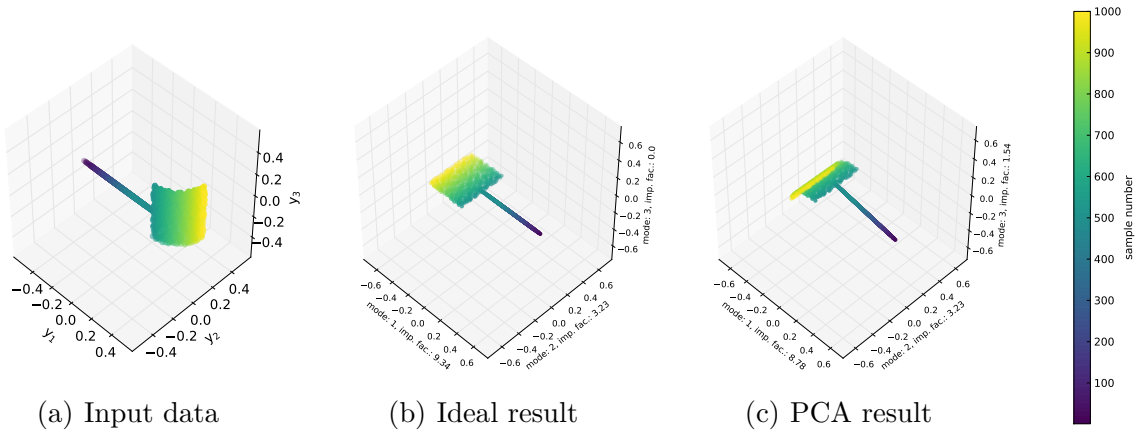
Figure 5.20: DRM results for the data set of Fig. 5.19 with two orthogonal planes. The neighbourhood graph is visualised as grey lines and the points are coloured according to the distance to the PCA result, which is the ideal outcome.

The second way in which the assumption that the data is lying on a single connected manifold of a fixed dimension can be violated is the case of mixed dimension data sets. In practical applications, the dimension can change throughout the data set, with one portion being of a different intrinsic dimension than other areas. Ideally, the analysis would be applied to each portion separately, but dividing a data set into neat clusters is not always be possible. To investigate the behaviour in this case, a new data set of the so-called Shovel example is introduced. This data set is consisting of a blade, which is a nonlinear two dimensional manifold, that is connected to a handle represented by a one dimensional linear manifold. The generating function can be

stated as $f_{\text{shovel}} : [0,1]^2 \to \mathbb{R}^D$, with $\phi_{\text{blade}}, \phi_{\text{handle}} : [0,1]^2 \to \mathbb{R}^3$:

$$\begin{aligned}
\phi_{\text{blade}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} -\frac{1}{5} - \frac{1}{4}\cos(\pi x_1) \\ \frac{1}{4} - \frac{1}{4}\sin(\pi x_1)) \\ \frac{1}{2}x_2 - \frac{1}{4} \end{pmatrix} \\
\phi_{\text{handle}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} x_1 + \frac{1}{4}x_2 - \frac{1}{2} \\ 0 \\ 0 \end{pmatrix} \\
f_{\text{shovel}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &:= \begin{pmatrix} \begin{cases} \phi_{\text{blade}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 \geq \frac{1}{2} \\ \phi_{\text{handle}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 < \frac{1}{2} \end{cases} \\ \mathbb{0}_{D-3} \end{pmatrix}
\end{aligned} \tag{5.10}$$

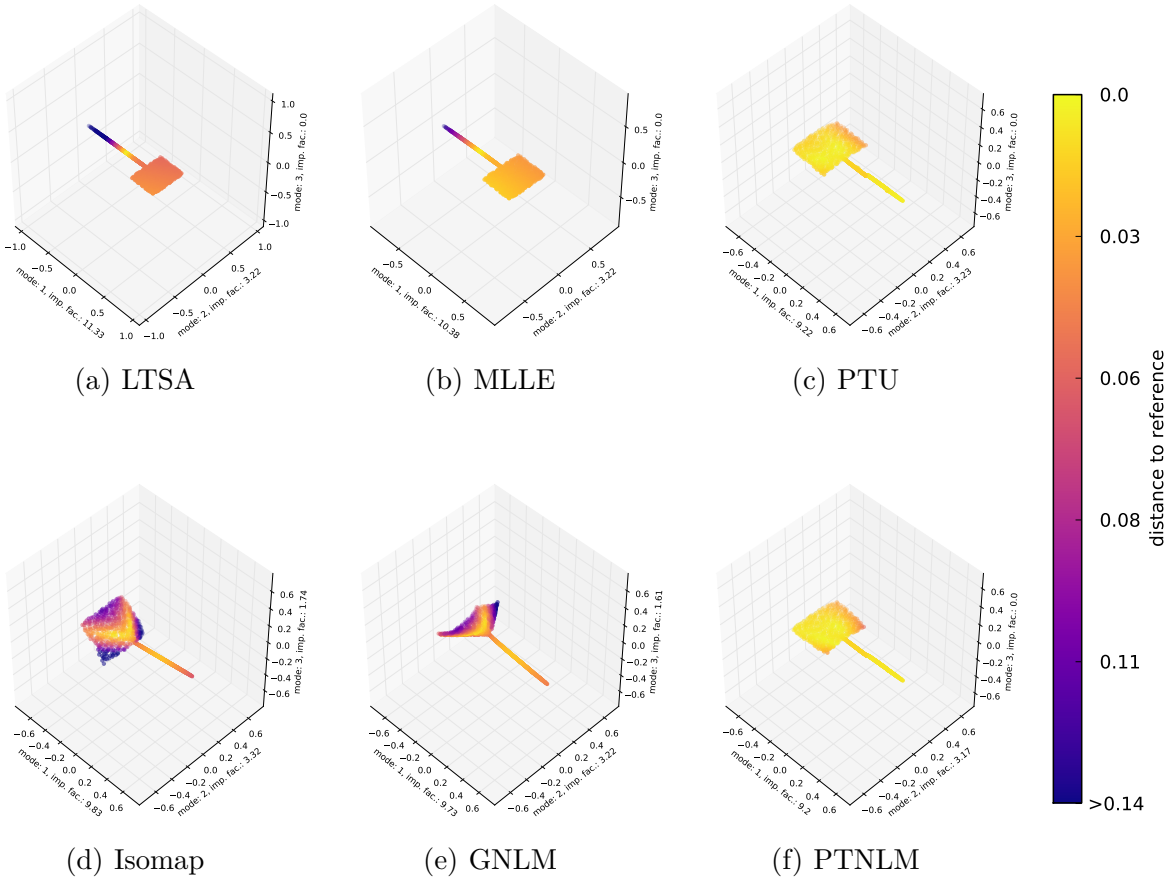
To evaluate the performance of the different DRMs on this example, a data set with 1 000 points was sampled with the Quasi-Random method introduced in Section 5.1.1 and then projected using the formula of Eq. (5.10). This input data set is visualised in Fig. 5.21.a and the ideal DR result in Fig. 5.21.b. Ideally, the Shovel is unrolled into the low dimensional plane with the handle staying straight and a rectangular connection with the blade.



(a) Input data (b) Ideal result (c) PCA result

Figure 5.21: Visualisation of the Shovel data set. The points are coloured according to their sample number.

When applying the linear PCA, the result is just a rotation of the input data set, capturing the two dimensional intrinsic structure. All nonlinear approaches were applied with a neighbourhood size of $k = 10$, as this has proven to be a good choice for intrinsically two dimensional manifolds such as the blade of the Shovel. The resulting embeddings for the different nonlinear DRM approaches are closer to the desired outcome than the PCA result and displayed in Fig. 5.22.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.22: DRM results for the Shovel data set of Fig. 5.21.a. The colour shows the distance to the ideal result shown in Fig. 5.21.b.

The LMs LTSA and MLLE provide usable low dimensional embeddings for this example. Only the handle of the Shovel is slightly curved, because of the sample density, which prevents the local neighbourhoods from reflecting the global orientation.
Similar to the disconnected example, the graph distance-based methods Isomap and GNLM provide the worst results with the largest errors, as they yield three dimensional instead of two dimensional objects, with local distortions at the corners of the Shovel blade.
Encouragingly, the parallel transport variants provide close to ideal results, with only small errors at the blade corners near the handle. This is due to a graph connection between the nodes in the blade near the handle with the nodes in the handle, which results in a slight rotation of the local tangent frame.

Both ways to violate the assumption have been heuristically investigated, and for small gaps in the manifold as well as a small difference in the intrinsic dimensions, the modified nonlinear DRMs can still be applied in this work. These small violations can frequently be observed in application data, but an in-depth investigation of stronger violations beyond these practical requirements is left for future research.

5.2 Performance on Crash Simulation Data

With the basic functionality of the new methods demonstrated on artificial data sets in the last sections, this section features the application of the Extended Workflow on simulation result data. In contrast to the artificial manifolds, where the ideal outcome is known and samples can be distributed as desired, these properties are in general not given for the analysis of real simulation examples. The three examples in this section have an increasing complexity, introducing more difficulty with each new data set. All data sets shown here were simulated using LS-DYNA explicit [LC20a] version "smp d R8.0.0" in revision 95309, though any simulation code could be used.

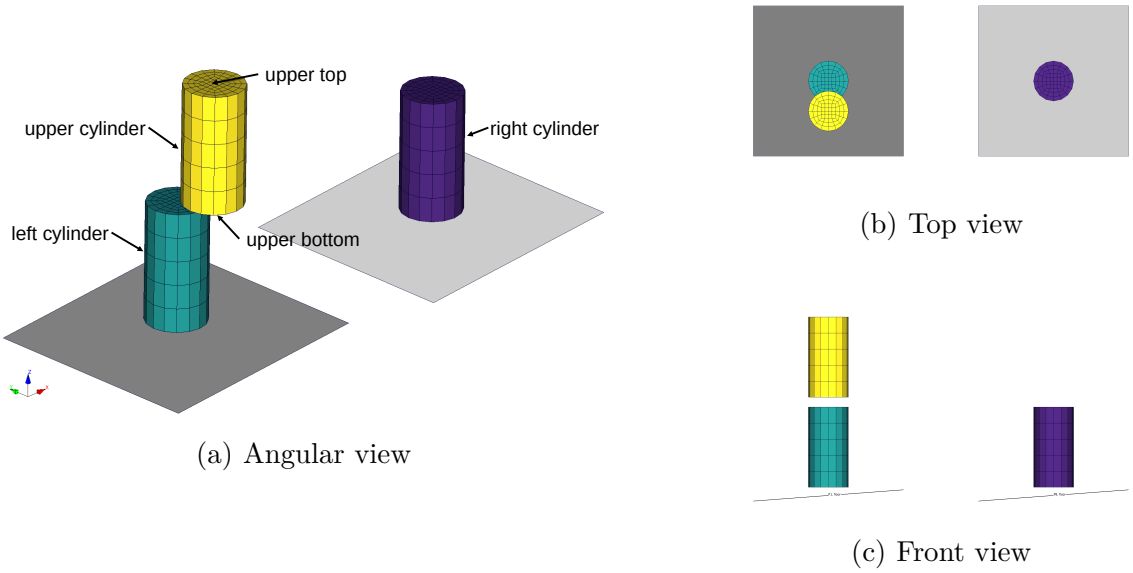
5.2.1 Cylinders Example

The first data set consisting of simulation results is the Cylinders example, which was specifically constructed for this work. This simple example is used to explain the different steps of the analysis and to compare the results for different analysis targets. Since this data set was specifically constructed for this thesis, it offers a unique opportunity as the dependencies between the parts involved are simple and known before applying the analysis. The known intrinsic structure of this data is used to highlight the limitations of the linear approach and the capabilities of the nonlinear methods. Each step of the analysis is investigated very detailed for this first example and shorter for the following data sets, since the steps are similar and further explanations would be redundant. All simulations involved were created using a small set of scripts, which can be found in Section B.2 of the appendix and can thus be entirely recreated by the interested reader.

5.2.1.1 Structure of the Data Set

The Cylinders data set consists of 91 variants of a simulation that involves three identical cylinders and two identical floor segments. Each cylinder has a height of 80 mm and a radius of 20 mm. It is modelled by solid elements, while the top and the bottom faces have an additional layer of shell elements that share the nodes with the solid elements. The material is assumed to be an idealised aluminium modelled by a MAT_24 card and assigned to all elements. The exact material specification can be found in Section B.2. Each floor is modelled by a single shell element which spans 150 mm in x and y direction and is inclined by 7.5% in z direction with increasing values of x. The two floors are placed 75 mm apart from each other and the first and the second cylinders are centred above each floor with a distance of 7.5 mm in z-direction, measured in the centre of the cylinder. A third cylinder is placed above the left cylinder with an offset of 10 mm in z-direction and -30 mm in y-direction. This initial geometry is displayed in Fig. 5.23. The starting conditions for the simulation are that the floors are fixed, and the cylinders are falling with a constant speed. Here, all floor nodes are fully constrained so that no degree of freedom is left, and the nodes in the cylinders are unconstrained with an initial velocity of -10 ms in z-direction.

A result state is written to the output file every 1 ms in order to get a fine time resolution of the process. This basic set-up with two interacting cylinders on the left and one independent cylinder on the right is the same amongst all simulations.



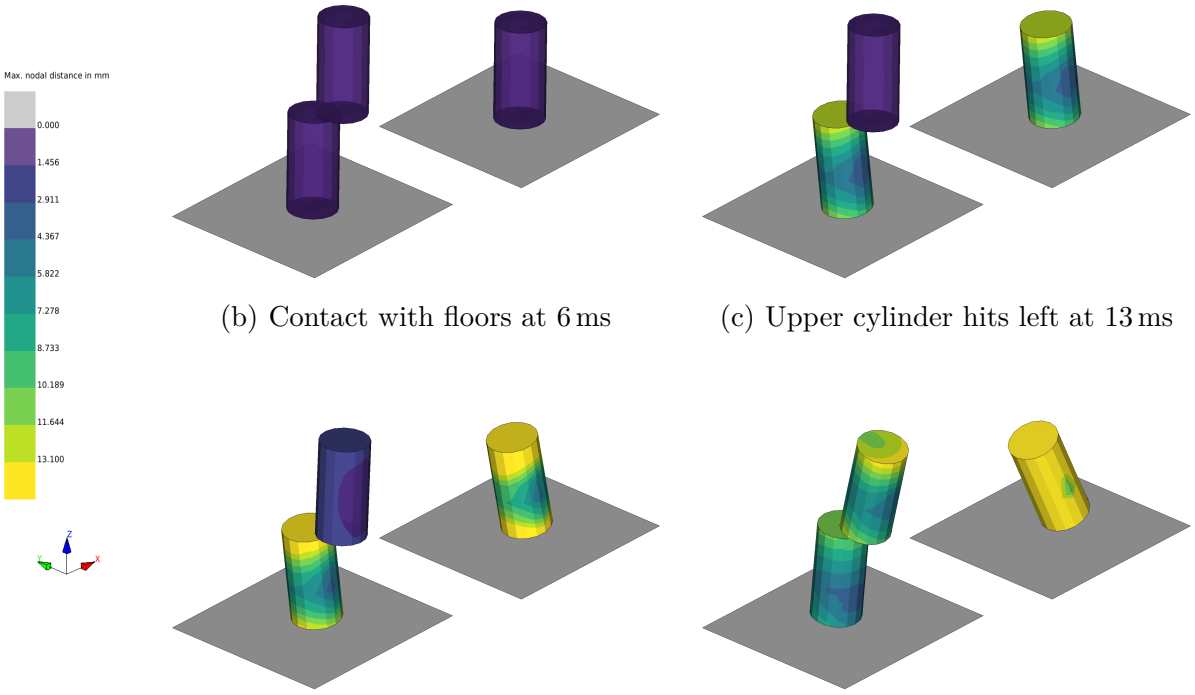
(a) Angular view

(b) Top view

(c) Front view

Figure 5.23: Initial geometry of the Cylinders example.

The first simulation is this unmodified set-up, and the other 90 variations were calculated as follows: The left floor was rotated in the xy-plane around its centre by two degrees more in each variant until it was rotated by a full 180° in the last simulation. The right floor was also rotated in the xy-plane around its centre, but by a random angle between 0° and 180° in each simulation run. These floors act as triggers to induce a controlled variation in the behaviour of the cylinders. The resulting behaviour is comparable in all variants, i.e. first all cylinders are in a free fall until the lower cylinders hit the floors. Depending on the rotation of the floors, the two lower cylinders start to tilt in different directions in each variant. A few milliseconds later, the upper cylinder hits the lower left cylinder and stabilises it after its tilting was allowed to unfold. But, since it was tilted in a different direction for each variant, the upper cylinder also starts to tilt in a different way in each simulation, as their faces are hitting at a different point and angle.
The behaviour of the right cylinder is completely independent of the other cylinders since there is no contact and the angles are created randomly. This cylinder hits the floor and starts to tilt as described earlier, but since there is no further impact, it does never stabilise. Snapshots of the described behaviour of the cylinders are displayed in Fig. 5.24.



(b) Contact with floors at 6 ms

(c) Upper cylinder hits left at 13 ms

(d) Tilting of upper cylinder at 18 ms

(e) Last evaluated state at 30 ms

Figure 5.24: Different states of the cylinders in the first simulation result. The colour represents the maximum nodal distance of the same node at this state in all different simulation results. This measurement was not done for the floors as they were rotated and are expected to be different in the variants.

For a showcase analysis, the correlation of the variance in the movement of the upper cylinder with the left and right lower cylinders is analysed. In the upper cylinder, the nodal displacements are chosen as the target of the analysis. Multiple quantities of the lower cylinders are investigated as possible sources for the variance of this target. Specifically, the correlation to three candidates is investigated: These are firstly the nodal displacements of the lower left cylinder, secondly the nodal displacements of the lower right cylinder and finally the internal energy of the bottom face of the lower left cylinder.

5.2.1.2 Selecting the Analysis States

Selecting the right states for target and source is an important part of the analysis. As explained in Section 4.1.2, the possible choices for these states can be determined using the importance factors calculated by a DRM. These importance factors can be calculated for the desired entity on the parts of interest for all states and then plotted as a curve to inspect the development over time. Fig. 5.25 visualises the development of the first linear importance factor over time for the nodal displacements of different

parts in the example. Though the linear method can sometimes fail to determine the correct dimension of the scatter, it can still be used to determine the state, in which the variance starts to occur and how it develops. There are three curves present for each of the cylinders, since the cylinder itself is modelled by solid elements and the top and bottom face, modelled by shells, are separate PIDs, which are used as parts in this analysis.
The target part in this example was fixed as the solid elements of the upper cylinder. For the analysis, the target state is chosen as 18 since the scatter had time to develop there after it first occurred at around state 15. But this state is before the second increase at around state 21.

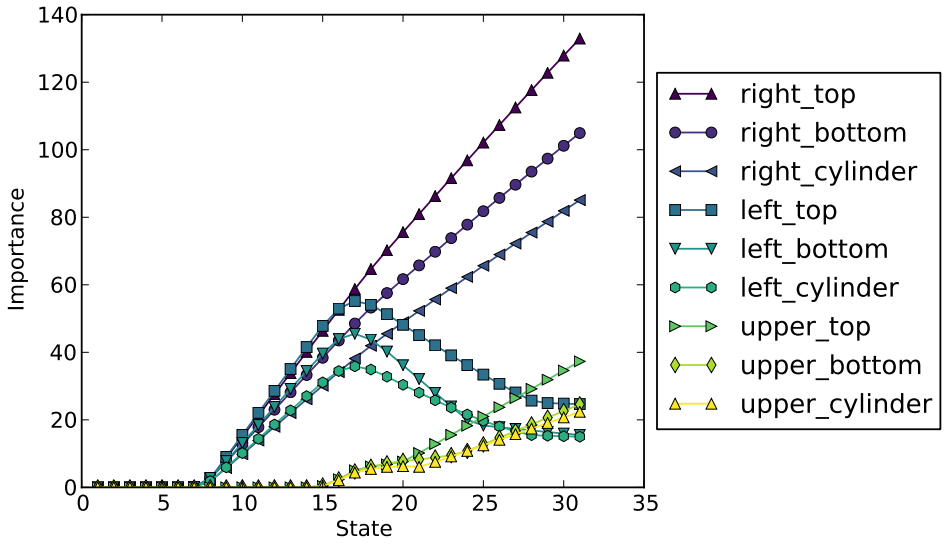


Figure 5.25: Importance of the first PCA mode over time for the displacements of different parts in the Cylinders example.

Two possible candidates of nodal displacement as sources of this scatter are investigated. First, the left cylinder, as it interacts with the upper cylinder and - in this simple set-up - is the known sole reason for the difference on the target. The second investigated source is the right cylinder's displacement as false positive test. Because it was randomly varied and there is no interaction with the upper cylinder, there must be no significant correlation between this second source and the target. The scatter for each of these PIDs starts at about state 8 and is developing afterwards. The left starts to stabilise at around state 16, so the state of the source should be set before the stabilising, therefore, number 14 was chosen for both lower cylinders.
As the variance in the displacement of upper cylinder can also be correlated with other post values, e.g. the internal energy, this correlation should also be investigated. The internal energy can be evaluated in this example in the top and bottom faces of the cylinders, modelled by shell elements, and the development of their linear importance factors is plotted in Fig. 5.26. The impacts of the cylinders are clearly visible in this plot, as the variance for the energy of the bottom faces increases drastically, when

the left and right cylinders hit the ground at around state 7. The impact between the upper cylinders bottom face and the left cylinders top face is emphasised by the increase in the importance factors at around state 15. As the bottom face of the left cylinder is chosen as the third possible source, the corresponding contact state 7 is chosen as the analysis state.

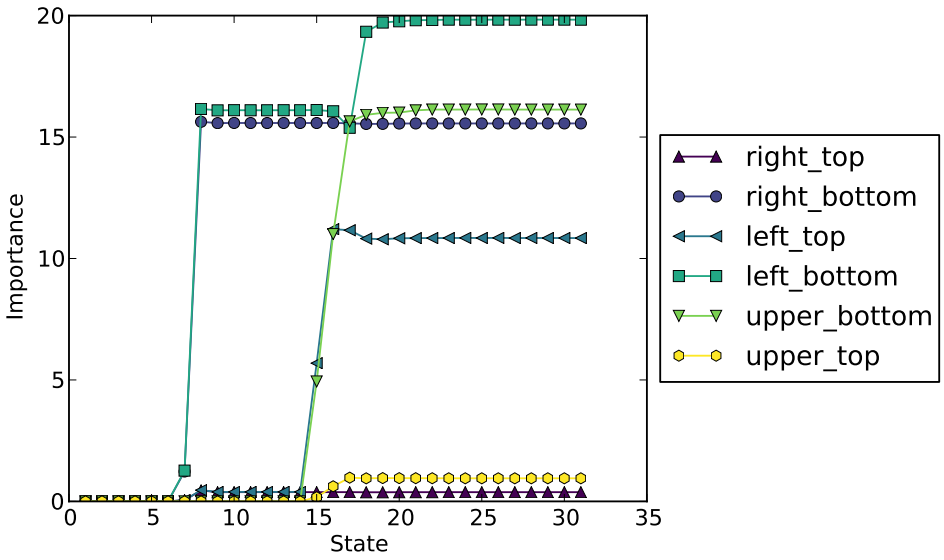


Figure 5.26: Importance of the first PCA mode over time for the internal energy of the cylinder faces.

With the analysis states determined, the CA can be performed for the following parts and quantities listed in Tab. 5.6.

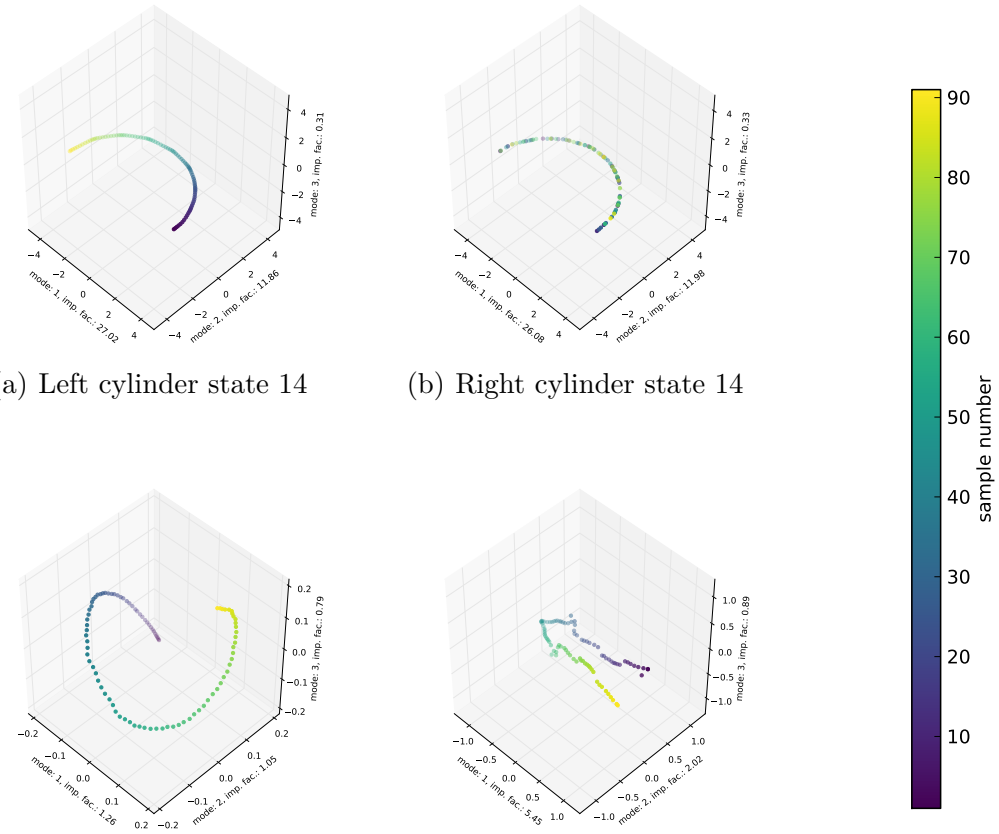
Type	Part	Quantity	State	Time
Target	Upper cylinder	Nodal displacements	18	17 ms
Source 1	Lower left cylinder	Nodal displacements	14	13 ms
Source 2	Lower right cylinder	Nodal displacements	14	13 ms
Source 3	Bottom left face	Shell internal energy	14	13 ms

Table 5.6: List of target and sources for the Cylinders example in the Extended Workflow.

5.2.1.3 Application of the Extended Workflow

The Extended Workflow comprises two steps, the Dimensionality Reduction and the difference operation, which are investigated separately in this section. First, the performance of the DR step is investigated. The DR results of the PCA approach for the nodal displacements of the target and the two sources at the specified states are shown in Fig. 5.27. Starting with the sources, the linear PCA approach estimates two significant modes for both lower cylinders and fails to capture the one dimensional intrinsic structure. The two linear modes describe, how much a cylinder tilts in the x or y direction, although the behaviour could just as well be parametrised

one dimensionally by the angle in which the cylinder is tilts. The resulting virtual simulations for the first two linear modes showing the different tilting behaviour are visualised in Fig. 5.28.



(a) Left cylinder state 14

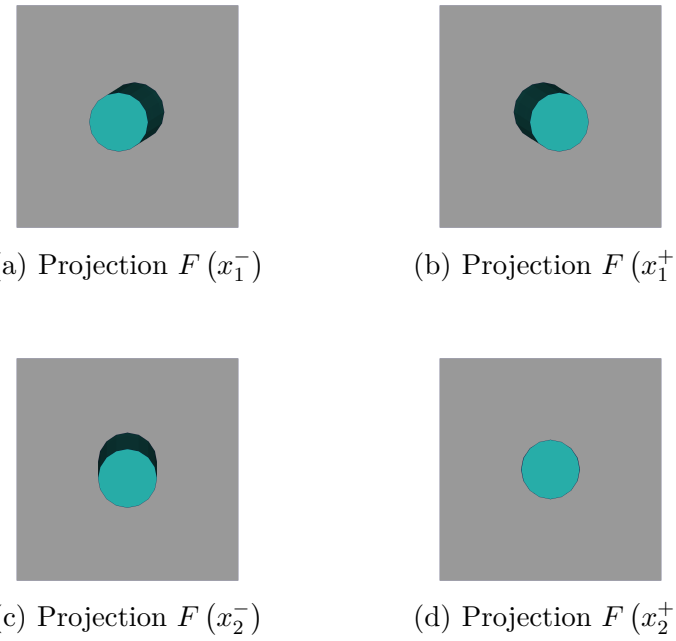
(b) Right cylinder state 14

(c) Left bottom state 14

(d) Upper cylinder state 18

Figure 5.27: Low dimensional embeddings of the three investigated sources in figures a, b, c and the target in figure d. The points are coloured according to their sample number and show the structured rotation in the left cylinders as well as the random rotation in the right cylinder.

The intrinsic one dimensional structure is obvious from the arc-shaped manifolds shown in Fig. 5.27. While the shape of the manifolds looks similar for the two lower cylinders, the different colour distribution shows the incremental and the random rotation of the cylinders.



(a) Projection $F\left(x_1^-\right)$ (b) Projection $F\left(x_1^+\right)$

(c) Projection $F\left(x_2^-\right)$ (d) Projection $F\left(x_2^+\right)$

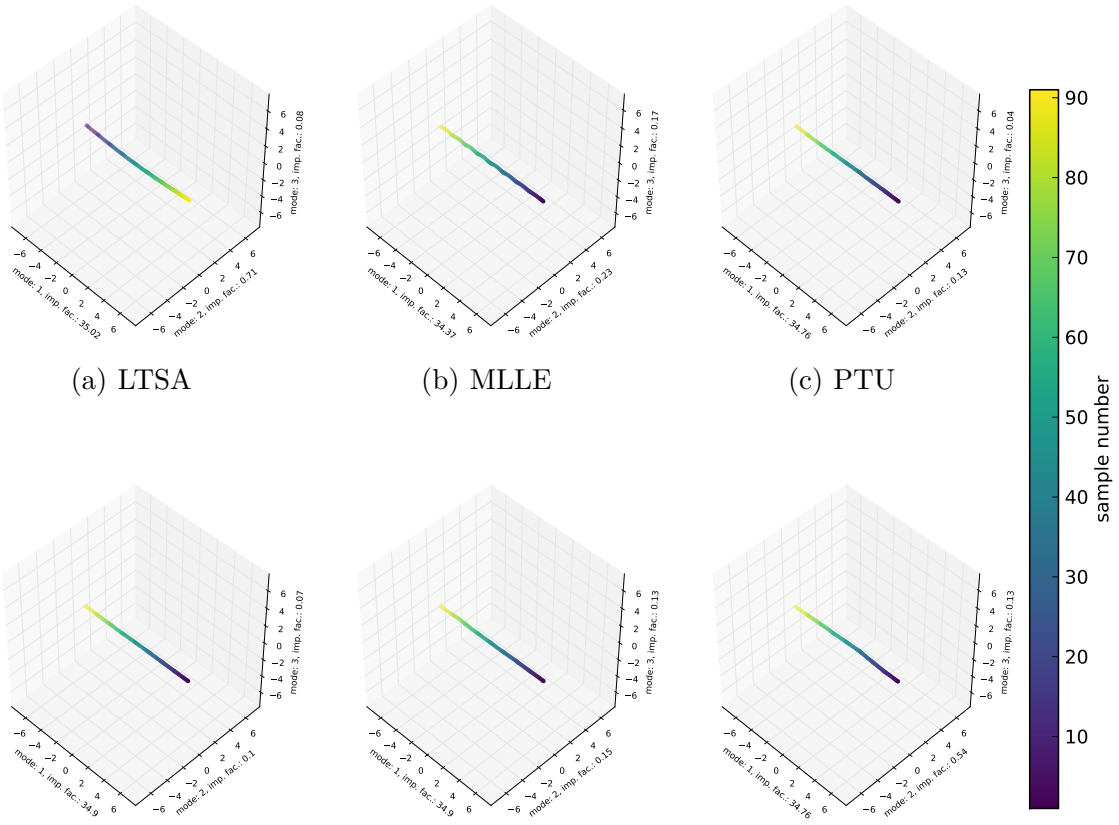
Figure 5.28: Top view of the lower left cylinder in the virtual simulation results. The first mode in the upper row shows the tilting in negative or positive x direction with a constant tilting in y direction. The second mode in the bottom row is purely associated with tilting in negative the y direction.

With the goal to better capture the underlying properties, the nonlinear DRMs were applied to the source candidates. The neighbourhood size chosen was $k = 6$, as this proved to be a good choice for intrinsically one dimensional data sets. Smaller values sometimes yield disconnected graphs, and larger values include already distant points in local neighbour hoods. This size was chosen identically for all methods to increase the comparability of the results. The results for the displacements of the left cylinder are exemplarily visualised in Fig. 5.29.
All nonlinear approaches unravel the one dimensional intrinsic structure with minor differences in the actual coordinates. The intrinsic dimension is also reflected in the computed importance factors and their relative difference in Tab. 5.7: Though more than one and up to 90 importance factors were calculated, only one is of significant size for the nonlinear methods, while there are two noticeably large factors for the linear PCA. As these factors are sorted in descending order, only the first five are listed in the table to emphasise that only one or two are important and all further modes are negligible.

	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
Imp. factor 1	27.023	35.027	34.374	34.902	34.77	34.902	34.760
Imp. factor 2	11.865	0.717	0.001	0.103	0.132	0.151	0.544
Imp. factor 3	0.319	0.001	0.001	0.075	0.042	0.132	0.130
Imp. factor 4	0.286	0.001	0.001	0.060	0.019	0.107	0.093
Imp. factor 5	0.105	0.001	0.001	0.043	0.018	0.079	0.051

Table 5.7: Importance factors for the first five modes of the lower left cylinders displacements at state 14 computed by different DRMs.

The results for the other two sources are similar and are hence not displayed, as only the order of the samples is different for the displacement of the right cylinder and the magnitude of the importance factors for the energy of the bottom face is smaller.



(a) LTSA (b) MLLE (c) PTU

(d) Isomap (e) GNLM (f) PTNLM

Figure 5.29: Low dimensional embeddings of the lower left cylinder at state 14 computed by different nonlinear DRMs.

This concludes the highlighting of the differences in the first step of the workflow.

The varied capturing of the intrinsic dimension in the low dimensional embedding of the first step has a strong impact on the second step of difference operation, since the operation is defined by these low dimensional coordinates and their importance

factors. In the following, the three different sources and their effect on the target are discussed individually. Two tests are performed for each of the sources: Initially, the single first mode and then the first two modes are subtracted from the target. Since the intrinsic dimension is one, subtracting the first, most important mode should already reveal the correlation between source and target, but since the PCA has two modes with substantial importance factors, both are subtracted and a comparison is made. The subtraction is performed with the orthogonal projection DPCA approach for the linear PCA method and with the new DLAI approach for the nonlinear DRMs. After applying the difference operation, the difference measures δ_{spec} and δ_{var} are computed as introduced in Section 4.1.
The first investigated source is the displacement of the lower left cylinder, which is interacting with the target upper cylinder, and its different behaviour is the sole reason for scatter of the target. The results of the difference operation for this source are displayed in Tab. 5.8. Subtracting only the first linear mode computed by the PCA method using the DPCA approach does not yield a significant reduction in terms of δ_{spec} or δ_{var}. Using the first two linear modes in the difference operation yields large values for both deltas, but not as large as expected: Since the PCA determined only two significant importance factors and this source is the only reason for the targets scatter, subtracting both underlying modes should eliminate all the variance in the target, ideally yielding a δ_{var} of close to 100%.
The nonlinear DRMs perform much better in this example: Subtracting only the single first mode already yields much higher deltas, which are closer to the ideal 100% and better than subtracting two modes with the linear method. Furthermore, increasing to two modes does not change the outcome, since they only determined one important mode. Five of the six methods yield identical delta results up to the first decimal, except for LTSA, which is marginally worse.

Modes	Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$e = 1$	δ_{spec}	0.0004%	94.1%	94.4%	94.4%	94.4%	94.4%	94.4%
$e = 2$	δ_{spec}	83.4%	94.1%	94.4%	94.4%	94.4%	94.4%	94.4%
$e = 1$	δ_{var}	11.5%	99.5%	99.6%	99.6%	99.6%	99.6%	99.6%
$e = 2$	δ_{var}	95.3%	99.5%	99.6%	99.6%	99.6%	99.6%	99.6%

Table 5.8: Difference result with the left cylinder as the source and the upper cylinder as the target part.

The second investigated source is the displacement of the right cylinder, which was randomly rotated and has no interaction with the target cylinder. This source should not yield significant correlation and hence only small values for the delta measures. The resulting difference measures are listed in Tab. 5.9. Here the linear DPCA performs very well, even when both significant modes of the source are subtracted, although the delta values increase marginally, but not to any noticeable size.
The nonlinear DRMs also yield small, although slightly higher delta values. But in contrast to the linear DRM, the values do not increase when two instead of one mode are subtracted. Similar to the Orientable Noise data set in Section 5.1.3.3, the

rotation of the cylinders was determined randomly. Though no high correlation with the deterministic data set is expected, small local correlations can occur, which is emphasised by the fact that the linear approach has small but non-zero delta values that increase with rising number of modes. Again, all nonlinear methods perform very similarly, with the exception of the LTSA approach, which is slightly worse.

Modes	Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$e = 1$	δ_{spec}	0.06%	2.12%	0.29%	0.27%	0.27%	0.25%	0.27%
$e = 2$	δ_{spec}	0.17%	2.12%	0.29%	0.27%	0.27%	0.25%	0.27%
$e = 1$	δ_{var}	0.17%	5.34%	2.55%	2.51%	2.51%	2.48%	2.51%
$e = 2$	δ_{var}	0.76%	5.33%	2.54%	2.51%	2.51%	2.48%	2.51%

Table 5.9: Difference result right to upper cylinder.

The last investigated source is the internal energy of the bottom face in the left cylinder. As this face is the one in contact with the floor, which is the original trigger of the variance in the left cylinders, this part should yield a significant correlation with the target part. The results are displayed in Tab. 5.10. The performance of the linear DRM is comparable to the PCA results obtained when subtracting the displacements of the lower left cylinder, but slightly worse as the delta values are a little bit smaller. This result is to be expected, as it can be seen from Fig. 5.27.c that there are not only two, but three substantial modes computed from the energies: The low dimensional embedding is not a flat semicircle, but a spiral protruding from the two dimensional plane. Hence, only up to two modes are not enough to describe the full variance in the source.
When utilising nonlinear DRMs, the results are even closer to the performance on the first source. Here, the intrinsic one dimensional structure was captured in the same way, yielding almost identical results to the first test. The only exception to this similarity is the MLLE method, which performs significantly worse in this case, especially when only a single mode is subtracted, but not as bad as the linear PCA.

Modes	Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$e = 1$	δ_{spec}	0.000299%	94.0%	1.13%	94.2%	94.4%	94.3%	94.4%
$e = 2$	δ_{spec}	67.3%	94.0%	92.0%	94.2%	94.4%	94.3%	94.4%
$e = 1$	δ_{var}	10.5%	99.5%	11.8%	99.6%	99.6%	99.6%	99.6%
$e = 2$	δ_{var}	86.4%	99.5%	98.7%	99.6%	99.6%	99.6%	99.6%

Table 5.10: Difference result bottom face to upper cylinder.

The above stated findings can be summarised as follows: In this simple simulation example, the causality was clear and thus, the desired results for the target could be formulated. For the three candidates, two correlated sources and one uncorrelated source were investigated.
The correlated sources have shown the problems of the linear DPCA approach, as the underlying one dimensional dependency between the lower left cylinder and the upper cylinder could not be confirmed, as the resulting values for δ_{spec} and δ_{var} were

insignificant. On the contrary, the nonlinear DR approaches in combination with DLAI correctly determined this one dimensional dependency. This dependency was confirmed in two different quantities: The nodal displacements and the shell internal energy estimated an almost identical correlation, as they produced very similar values for the deltas. In this case, the different approaches yielded approximately the same result, although single methods sometimes deviated from the others.
For the uncorrelated source in the form of the nodal displacement of the right cylinder, the results of the linear method were as expected as it produced very small values for δ_{spec} and δ_{var}. The nonlinear methods resulted in slightly higher delta values, which was to some extent expected as they contain more information in fewer modes. The absolute sizes of the difference measures were still reasonably small for a purely random dependency.
Overall, the performance on this first simulation data example is satisfactory, as the nonlinear approaches exceeded the linear method while successfully passing all performed tests.

5.2.2 Rocker Example

The second simulation data example in this work is the Rocker example, which is a variant of the original setup initially introduced by Christopher Ortmann in [OS13]. The analysis of this example is closer to the actual application, as the exact dependencies are not known before-hand, but unlike, e.g., a full car application, the interactions can be derived from the straightforward set-up.

5.2.2.1 Structure of the Data Set

The example in this section is a simplified set-up, derived from a lateral pole impact of a vehicle as defined by the European New Car Assessment Programme (NCAP) [Eur19]. It consists of a rocker section that is connected to a short segment of the seat cross member. A rigid wall with a mass of 85 kg is attached to the far end of the seat member and the displacements of the nodes at this end are constrained in all directions except for a translation in y direction. The nodes at the corners of the rocker section are fixed in z direction, but all other movements are free. Both the rocker and the seat member have an initial velocity of $-29\,\text{km/h}$ and move in y direction towards a rigid pole. A more detailed explanation can be found in [Ort15] and [OS14]. In these publications, the automated insertion of walls into the inner part of the rocker is investigated in order to optimise its performance in certain load cases. The objective for the optimisation is to minimise the rigid wall forces, and the optimisation method is the so-called Graph and Heuristic Based Topology Optimization (GHT) [OS13]. In this method, the walls are incrementally inserted into the topology according to heuristics derived from expert knowledge. The resulting geometry is visualised in Fig. 5.30.

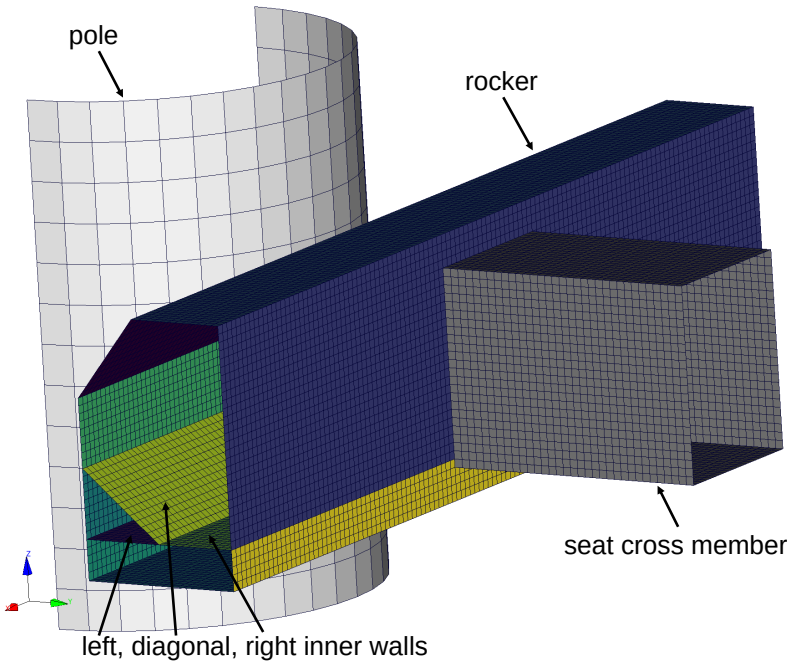
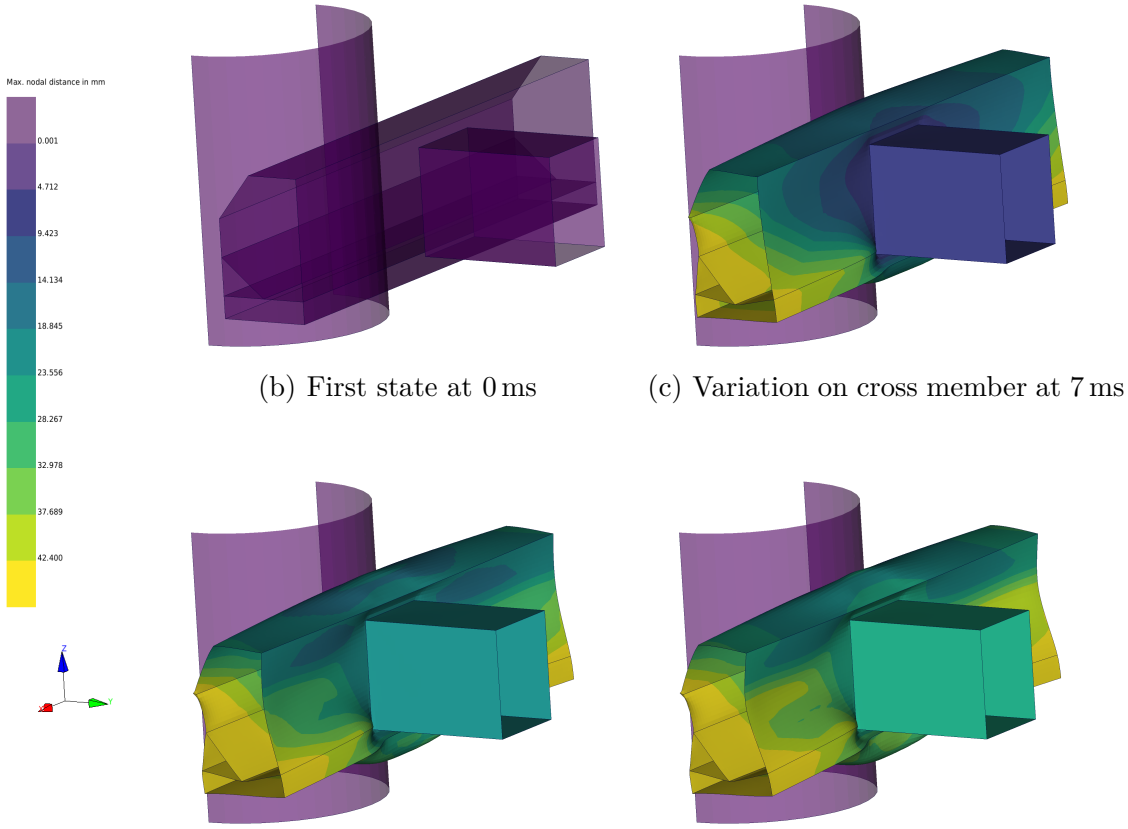


Figure 5.30: Initial geometry of the rocker example. The thickness of the grey PIDs remains fixed, while the coloured inner and outer wall segments of the rocker are varied in the optimisation process.

In each iteration, multiple insertions generated by different heuristics are evaluated and the three best results advance to the next iteration, where the heuristics are applied again, and further insertions are made until the design does not improve any more. After each insertion and before the next wall is inserted, a Shape and Sizing Optimization (SSO) is performed, adjusting the parameters of the design. The 360 simulations analysed in this thesis were generated by Dominik Schneider from the University of Wuppertal and are the runs computed in such a final SSO at the end of an optimisation. After two GHT iterations, three walls are inserted into the inner part of the rocker, as shown in Fig. 5.30. The profile of the rocker at this stage consists of 11 walls with individual thicknesses. The aim of the final SSO is to find the optimal distribution of these 11 wall thickness values so that the rigid wall forces are minimal for a given constant mass. This thickness optimisation was conducted using LS-OPT with a domain reduction strategy [LC20b], meaning that the design space was adaptively sampled, yielding a concentration of data points towards the computed optimum. This differs significantly from the other examples featured so far, where the sample points were evenly distributed in the solution space.
For all 360 evaluation points of the optimisation a simulation was run and a state written to the result file every 1 ms. These simulations were terminated, once the seat cross member came to a complete stop, and as this occurred at different times for the different designs, the number of states available in the results is not identical for the samples. Thus, the analysis is only conducted up to the last commonly available state.

The optimisation varies the wall thicknesses to modify the performance of the rockers and to find the best in these different behaviours. These differences in behaviour are visualised in Fig. 5.31. After the initial contact between the rocker and the pole, the different wall thicknesses affect the deformation of the individual walls, resulting in variation on the rocker that is smaller in the inner section and increases towards the corners. The different overall deformation of the rocker affects the displacement of the seat cross member, which is closer to the pole in some simulations and further away in others for each fixed state. This varying behaviour of the seat cross member is relevant to the user from an application point of view: The purpose of the NCAP pole impact test is to validate the capability of a car to protect the occupants [Eur19], who are positioned in the seats mounted on this cross member. Hence, its behaviour and scatter are vital for the test rating.



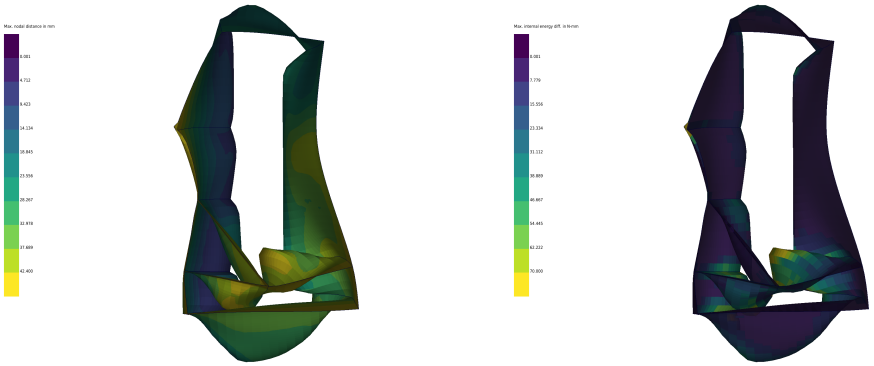
(b) First state at 0 ms (c) Variation on cross member at 7 ms

(d) Increasing differences at 12 ms (e) Last common state at 14 ms

Figure 5.31: Different states of the first simulation result of the Rocker example. The colour represents the maximum nodal distance of the same node at this state amongst all different simulation results.

Crucial to the overall behaviour of the complete model are the three connected walls inside the rocker. These inner walls were inserted at this position into the original hollow design by the GHT in order to drastically modify the performance of the

design in the given optimisation problem. These topology modifications had a better impact on the outcome than other evaluated changes. Otherwise, this would not have been the optimal design. These crucial parts also show a large variance in the nodal displacements as well as the shell internal energy, see Fig. 5.32.



(a) Displacement variation (b) Energy variation

Figure 5.32: Side view of the Rocker example with highlighted difference. The colour represents the maximum nodal distance in Figure (a) and the maximum shell internal energy difference in Figure (b), for the same entity at the same state in different simulation results.

Because of its significance for the application, the variance in the nodal displacements of the seat cross member is considered as the target for the analysis in this example. The dependency of this variance to the inner walls of the rocker is investigated by calculating the correlation to two possible sources: The difference in the nodal displacements and shell internal energies of all three inner walls, where the three PIDs are considered as one part for the analysis.

5.2.2.2 Selecting the Analysis States

With the target and possible sources fixed, appropriate states must be selected for the analysis. For this purpose, the importance factors and their development over time can be utilised, similar to the other example shown before. The development of the first linear importance factor over time for the variation in the nodal displacements of the different PIDs is visualised in Fig. 5.33.
For the target part of the seat cross member, the associated curve is monotonically increasing, with an inclining slope starting shortly after the impact of the rocker and the pole. State 13 at 12 ms is selected as the target state for the analysis, because the scatter could unfold but it is not the maximal value for this curve.
Though the three connected inner walls are considered as one part in the analysis, three individual curves are plotted that show the very similar behaviour in their displacements: Shortly after the impact of the rocker and the pole in the first state,

the difference starts to increase until state 8. As described in Section 4.1.2, the source state should be chosen before the target state and shortly before maximum of the curves so that the scatter was able to unfold. Hence, state 7 is used as the source for the nodal displacements of the inner walls.

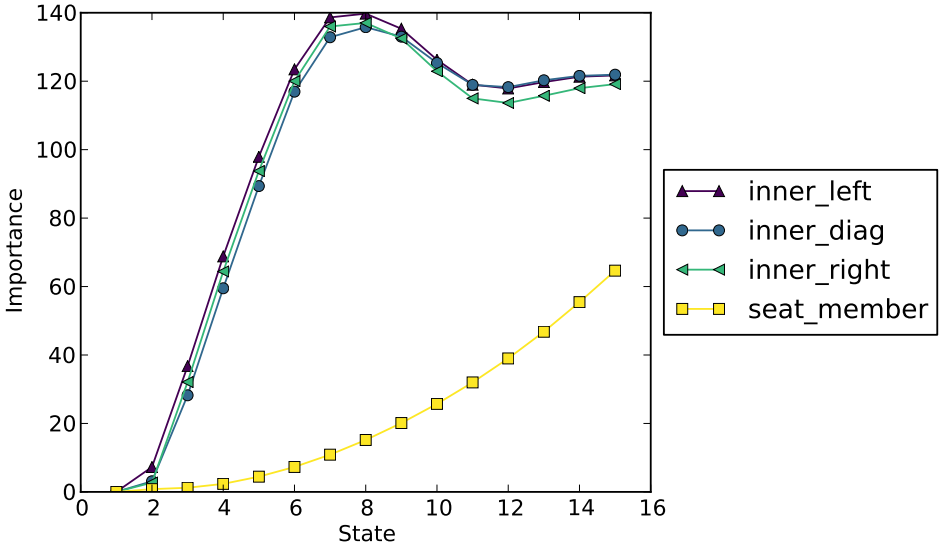


Figure 5.33: Importance of the first mode over time for the nodal displacements of the left, right and diagonal inner walls of the rocker and the seat cross member segment.

Fig. 5.34 visualises the development of the first linear importance factor for the difference in the shell internal energy of the PIDs. These curves are overall comparable to the curves for the nodal displacements, there are only minor differences.
The curve for the target importance factor still increases monotonically, but the steepest ascent is directly after the impact rather than towards the end of the simulation. However, the importance factors for the three inner walls show different behaviour. The curve associated with the diagonal wall also increases monotonically, while the curve corresponding to the right wall has a distinct peak, similar to the curves for the nodal displacements. Finally, the curve for the left wall can be seen as a mixture of the other two, as it has a peak, but less prominent than the right wall and shows an increase over time, similar to the diagonal wall. With the knowledge about the behaviour of the nodal displacements in mind, the analysis state for the internal energy source was also chosen as 7, due to the overall development of the importance factors of the walls.

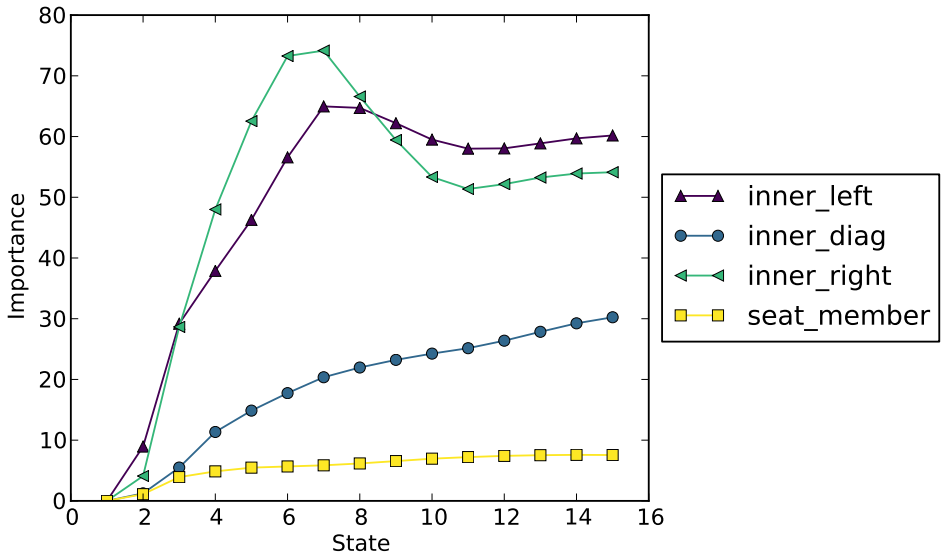


Figure 5.34: Importance of the first mode over time for the shell element internal energy of the four parts in the Rocker example.

With the source and target parts as well as the relevant states for the analysis as listed in Tab. 5.11, the preconditions for the analysis are complete.

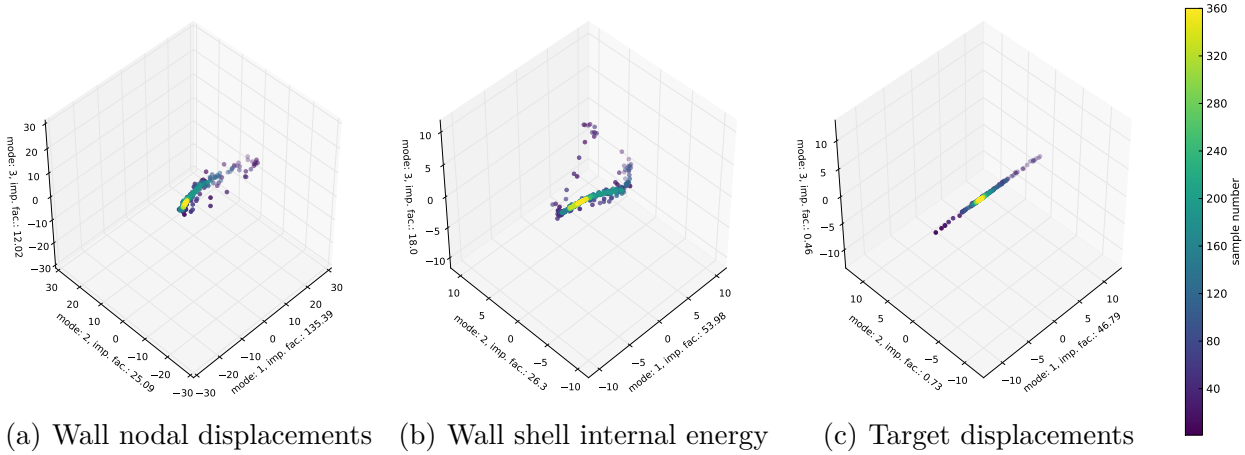
Type	Part	Quantity	State	Time
Target	Seat cross member	Nodal displacements	13	12 ms
Source 1	Inner walls	Nodal displacements	7	6 ms
Source 2	Inner walls	Shell internal energy	7	6 ms

Table 5.11: List of target and sources for the Rocker example in the Extended Workflow.

5.2.2.3 Application of the Extended Workflow

In this example, the Extended Workflow is applied from the target point of view. While investigating the Cylinders example in Section 5.2.1, the source was known and the analysis was conducted with this knowledge from the source point of view. In practical applications, however, the target point of view is more common as the source is usually to be determined.
The DR of the nodal displacements of the seat cross member at state 13 yields an almost perfect line, which means that the data is very close to a one dimensional linear manifold as can be seen in Fig. 5.35.c. As the manifold is basically linear, the PCA can capture the structure sufficiently well.
The virtual simulations for this target are visualised in Fig. 5.36 and show the difference associated with this mode: The rocker moves in a straight line and is very close to the pole for simulations on the one end of the mode spectrum and further away on the other end. As the pole is rigid and the seat member does not deform noticeably, the inner walls have to give way to this movement of the cross member.

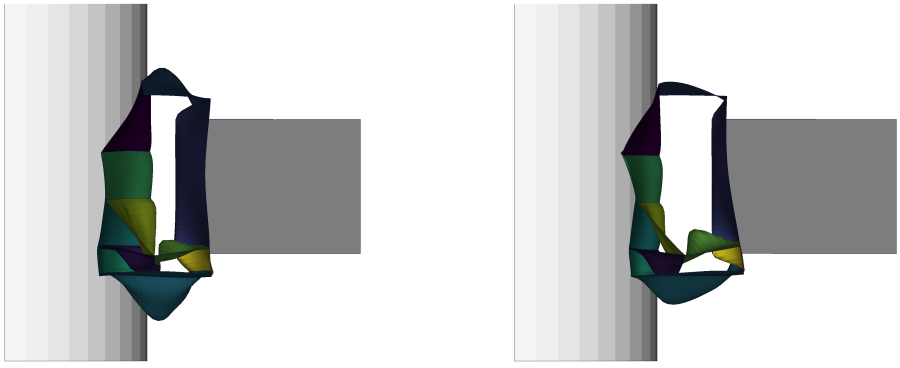
Thus, a significant correlation between the inner walls and the variance of the seat cross member displacement is to be expected, though the actual percentage for the difference measures cannot be stated beforehand.



(a) Wall nodal displacements (b) Wall shell internal energy (c) Target displacements

Figure 5.35: Low dimensional embeddings of the two investigated sources in Figures (a) and (b) and the target in Figure (c). The points are coloured according to their sample number and show the concentration of the points as they were sampled adaptively during the converging optimisation process.

To investigate this correlation, the Dimensionality Reduction is first applied to the two possible sources. The low dimensional embeddings computed by the PCA for the nodal displacement as well as the shell element internal energy of all three inner walls considered as one part are visualised in Fig. 5.35.a and Fig. 5.35.b, respectively.



(a) Projection $F\left(x_1^-\right)$ (b) Projection $F\left(x_1^+\right)$

Figure 5.36: Side view of the rocker in the virtual simulation results. The first mode of the seat cross member shows the segments different displacement in the y direction. The seat member moves in a straight line towards the rigid pole while the inner walls deform and give way.

The Dimensionality Reduction was performed with the nonlinear DRMs as well as for the linear method and the resulting importance factors for the nodal displacements of the inner walls computed by the different methods are listed in Tab. 5.12. All nonlinear methods used a neighbourhood size of $k = 13$, because tests and the linear DR results showed that the intrinsic dimension is larger than two, where a size of $k = 10$ would be used. The exact intrinsic dimension for this data set is not known as it is difficult to distinguish between noise and small but valid variance.
All approaches find one very important mode, though the number of less important but still significant dimensions varies among the methods. While the linear PCA has a relatively smooth descent of importance factors after the first factor, the nonlinear methods have different progressions, with the results of MLLE, Isomap and PTU being closest to the linear factors. Strongly deviating are the LTSA approach, which has four almost equally important lesser modes with a rather sudden drop afterwards, and the GNLM and PTNLM methods, where a second very important mode is closer to the first.
Because of this discrepancy, the difference operations were conducted with two different scenarios: First, only the single most important mode $e = 1$ of the sources was subtracted from the target part, as most DRMs determined one outstanding importance factor. Secondly, the first $e = 5$ modes were subtracted, as this is the maximum number of modes, which was still significant for all methods: The importance factor of the sixth mode of the LTSA approach is too small to be chosen as a valid, non-noise dimension.

	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
Imp. factor 1	135.399	116.574	116.581	152.227	133.347	146.868	181.315
Imp. factor 2	25.098	18.125	26.577	24.433	20.270	69.588	124.783
Imp. factor 3	12.024	14.754	16.631	16.065	11.316	29.245	29.678
Imp. factor 4	8.644	14.177	11.680	13.552	6.238	23.376	20.885
Imp. factor 5	6.410	13.008	10.338	8.014	4.271	17.165	10.464
Imp. factor 6	4.554	0.007	8.936	5.957	3.926	15.161	9.828
Imp. factor 7	4.008	0.007	0.001	4.660	3.612	12.412	9.057

Table 5.12: Importance factors for the first seven modes of the inner rocker walls displacements at state 7, computed by different DRMs.

The difference operation for the two number of modes was performed using the DPCA method for the linear PCA and the DLAI approach with the aforementioned neighbourhood size of $k = 13$ for the nonlinear DRMs.
When subtracting the first linear mode of the nodal displacement, the impact is very minimal as δ_{spec} and δ_{var} are very small, as can be seen in Tab. 5.13. Increasing the number of modes does increase the amount of correlated variance, but not to a significant extent. This result is not consistent with the expectation that the walls should have a significant correlation with the movement of the target. Contrary, the nonlinear methods all obtained higher values for both difference measures for $e = 1$ as well as for $e = 5$ modes. It is noteworthy that although the Isomap approach yields smaller delta values for $e = 1$ modes, it yields comparably results to the other

methods for $e = 5$. Apart from this deviation, the results for the nonlinear DRMs are of the same magnitude and conclude a relevant correlation between the nodal displacements of the walls and the seat member.

Modes	Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$e = 1$	δ_{spec}	0.90%	33.0%	32.9%	7.21%	31.7%	26.3%	26.0%
$e = 5$	δ_{spec}	6.56%	41.2%	32.5%	43.1%	45.9%	32.6%	41.8%
$e = 1$	δ_{var}	1.79%	55.2%	54.9%	13.9%	53.4%	45.7%	45.2%
$e = 5$	δ_{var}	12.7%	65.4%	54.4%	67.7%	70.7%	54.6%	66.1%

Table 5.13: Difference result for the nodal displacements of the inner rocker walls to the displacements of the seat cross member.

A similar investigation was performed, where the target was still the displacement of the seat member, but the source was chosen as the shell internal energies of the inner walls. Again, for better comparability, the single $e = 1$ and the $e = 5$ largest modes were subtracted with the same neighbourhood size of $k = 13$. The corresponding results are listed in Tab. 5.14. Again, the linear method yields rather small values for the difference deltas, though larger than those for the nodal displacements, especially for $e = 5$. Analogous to the displacements, all nonlinear methods yield higher results for the difference measures and interestingly, the Isomap approach is again the only deviating method. But the absolute difference between $e = 1$ and $e = 5$ modes is higher for the internal energies, suggesting an intrinsic dimension that is considerably greater than one.

Modes	Measure	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
$e = 1$	δ_{spec}	1.5%	16.3%	18.5%	9.08%	16.1%	19.8%	16.2%
$e = 5$	δ_{spec}	16.9%	32.8%	40.0%	62.5%	32.2%	47.9%	32.4%
$e = 1$	δ_{var}	2.97%	29.9%	33.5%	17.4%	29.5%	35.8%	29.7%
$e = 5$	δ_{var}	31.0%	54.9%	64%	85.9%	54.1%	72.8%	54.3%

Table 5.14: Difference result for the shell internal energy of the inner rocker walls to the displacements of the seat cross member.

The Rocker example has shown the capabilities of the new methods for an example from a real application that was not specifically constructed for this thesis. It highlighted the short comings of the linear approach to find the expected correlation between the inner walls and the seat cross member. The nonlinear methods yielded higher correlation values and comparable results for the nodal displacements as well as for the shell internal energies. This supports the assumption of a dependency between the involved parts. The special sampling structure of the simulation results, which originates from an incremental optimisation process, showed that the new methods can also handle data sets with varying sample density. This is important as it poses a challenge from a DR point of view and can always be encountered in real applications.

5.2.3 Silverado Example

The last simulation data example featured in this work is the Silverado example, based on the model already introduced in Section 3. While the previous examples consisted of few parts, this example is a full car application. The new methods are here tested in a scenario close that is very close to the application in a practical use case.
In this complex scenario, dependencies found between different parts of the simulation results are reasoned for, even though other influences can never be completely excluded. The model itself, but also this specific data set already been used in several publications [BJST16], [MSJ20] and the findings from previous work [BST15] are utilised in this thesis. Some of the already known relations between source and target parts are reviewed using the developed nonlinear methods. Newly discovered insights that go beyond the published knowledge are then validated with a plausibility check.

5.2.3.1 Structure of the Data Set

The Silverado data set used in this work is based on the open model published by the National Crash Analysis Center of the George Washington University [PRMB09], which is still available for public download from [Nat19]. It models the full-frontal impact of a 2007 Chevrolet Silverado crashing into a rigid wall barrier at 35 mph, which is motivated by a US-NCAP test [PRMB09]. The finite element model has been validated and compared to actual crash test results and has shown to reproduce the physical behaviour reasonably well [MSCK12]. It is composed of 679 PIDs of various element types.
The 77 simulations in the example data set used in this work were generated by a parameter variation, where the element thicknesses of 13 selected PIDs were changed. The varied parts are highlighted in Fig. 5.37 and a complete list of all changed PIDs and original element thicknesses can be found in Section B.3 of the appendix. Each of the original element thicknesses was multiplied with a different random variable $v \sim \mathcal{U}(0.8, 1.2)$ to induce some variation into the initial design of model and to feign manufacturing tolerances.
This variation of the elements thicknesses also results in a different crash behaviour in the simulations, with varying intensity over the crash duration and in different areas of the car. To capture these variances in detail, a result state was written to the result file every 1 ms. While these variances are to be expected, the overall performance should ideally not be affected by parameter changes within the manufacturing tolerance magnitude. From an application point of view, occupant protection and thus all parts in the immediate vicinity of the passengers are of special interest. One of the known crucial parts in this field is the firewall of the car [Nat12]. Among the 77 simulations, this part shows large variation in the displacements of the nodes on the driver's side, as can be seen from Fig. 5.38 for an advanced state.

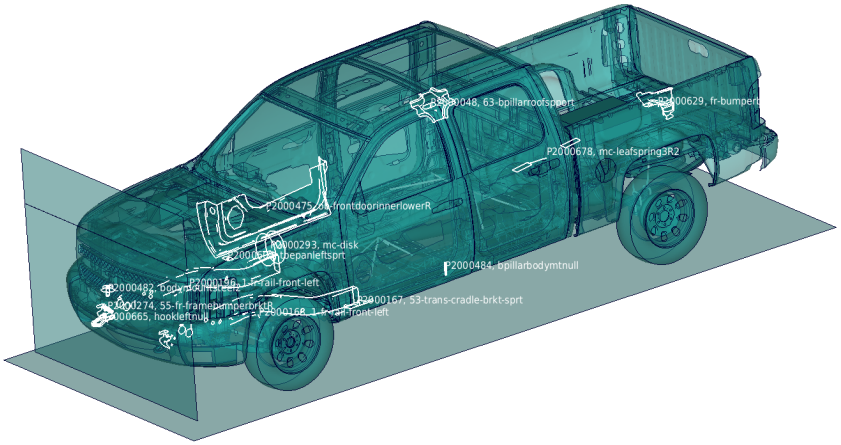


Figure 5.37: Varied PIDs in the Silverado example: The thicknesses of the highlighted parts were changed randomly within ±20% of their original value. The parts are on purpose distributed over the whole car to yield some thicknesses that have a huge impact on the behaviour, but also some with minor influence.

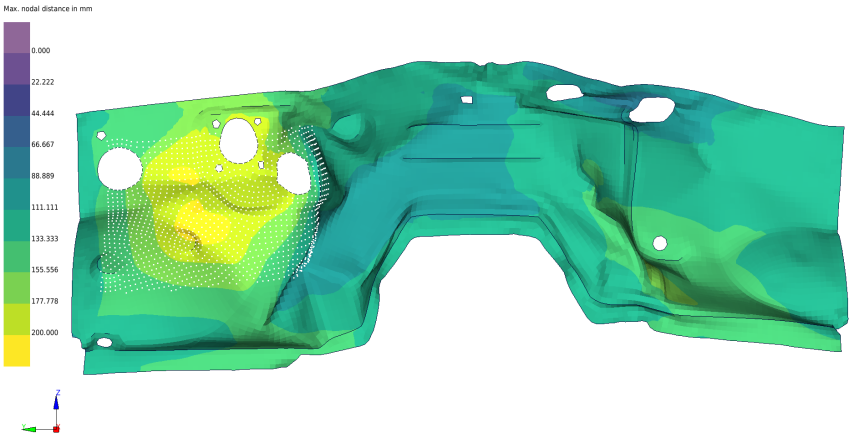
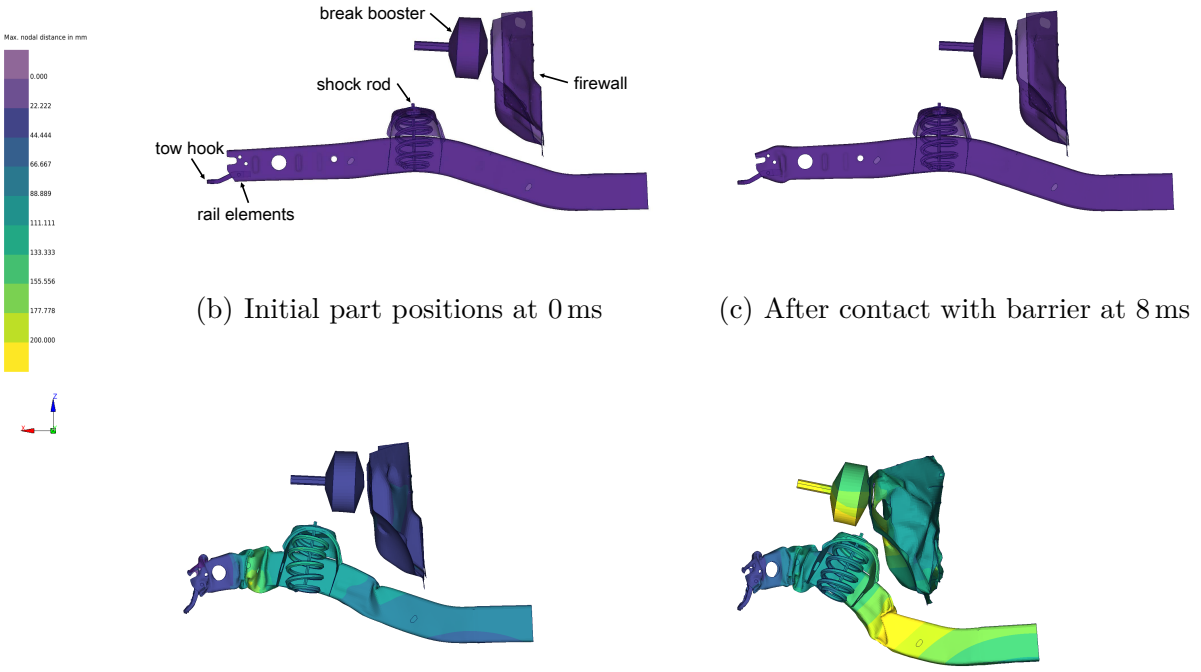


Figure 5.38: Target for the Silverado example: The nodal displacements for the firewall show significant variation, highlighted by the colour, which indicates the maximum distance for a node in one simulation to the same node in another simulation at the same state. The distance exceeds 200 mm for some nodes on the driver's side. The displacements of the selected node group are taken as the target for the analysis.

This type of challenge is usually the case in practical applications: A crucial part is showing variation in a significant magnitude and is thus taken as the target for the analysis, while the origin for this scatter is to be found. While highlighting the existing variances, together with the engineer's expertise, can help to identify possible sources for this scatter, it is important to also check possible correlations before constructional changes are made.
It is to be expected that the load bearing structures, for example the longitudinal rail, on the left side of the car have some kind of influence on the behaviour of the firewall on the driver's side. In this particular case, the shock rod and the break booster have also played an important role in previous works [BST15]. For some selected parts on the driver's side, the behaviour over time is displayed in Fig. 5.39.
The tow hook on the far left is the first part of the car to make contact with the rigid barrier. Depending on the individual part thicknesses, the behaviour over time is different for all the simulations, as the thickness of the displayed longitudinal rail is amongst the varied ones and as the total mass of the car and its distribution is affected by the overall combination of different thicknesses. This different behaviour in the results is noticeable on the longitudinal rail, but very prominent on the other parts: In some simulations an interlocking of the shock rod and the break booster can be observed, while they pass each other in different simulations. Whether they make contact or not determines, whether the break booster is pushed into the firewall or not, which causes some of the visible variation on the target part. This causality was already found in [BST15] and should also be reproducible with the nonlinear methods. Thus, the displacement of this part is taken as a first possible source for the investigation.
In previous works only the correlation between the same post value, e.g. nodal displacement, on different parts has been investigated. While the variation in the nodal displacements is very prominent, the difference in the shell element plastic strain also offers interesting insights: The elements in the longitudinal rail show significant differences over the duration of the crash and these are visible at a very early stage when the displacements are still very similar.
A very local effect can be spotted here at the connection between the tow hook and the rail, where a single triangular element shows very different plastic strain values at the same state in different simulations. The colour highlighting in Fig. 5.40 helps to recognise this triangular and its neighbouring mesh elements. The curious position of a single triangular directly adjacent to a hole in the rail and at the connection to the hook, combined with these huge differences, raises the question of how big the impact of this area is. Thus, the plastic strain of the longitudinal rails shell element in the area of this connection is considered as a second possible source in the analysis.



(b) Initial part positions at 0 ms

(c) After contact with barrier at 8 ms

(d) Rail and rod differences at 37 ms

(e) Variation on booster and wall at 67 ms

Figure 5.39: Different states of selected parts in the Silverado in the first simulation result. The colour represents the maximum nodal distance of the same node at this state amongst all different simulation results.

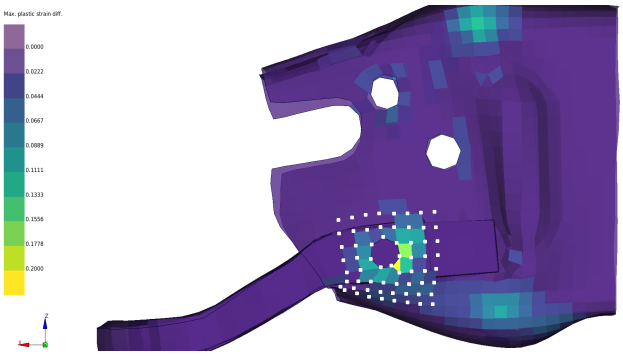


Figure 5.40: Detailed view of the connection between the tow hook and the longitudinal rail in the Silverado example. The colour indicates the maximum difference in the plastic strain for the element at the same state in different simulations. The nodes of the elements in the area of the connection between the two PIDs, which are considered as the second source for the analysis are highlighted.

5.2.3.2 Selecting the Analysis States

Once the target and two possible sources for the analysis are identified, the corresponding states for the analysis must be selected, similar to the two simulation examples mentioned before.
Again, the development of the linear importance factors is utilised to determine the relevant states for the analysis. Although the linear DRM may not capture the exact intrinsic dimension of the problem, it is still useful to determine at what point variance starts to occur in the data. Fig. 5.41 visualises the development of the first linear importance factor over time for the selected parts. In addition to the target nodes of the firewall and the two source parts, which are the break booster and the elements of the longitudinal rail near the connection to the tow hook, the importance factors for the tow hook itself and the shock rod are also plotted. This is done to emphasise the similarity between the curve for the firewall nodes and the curve for the break booster by demonstrating that development of the importance factors is not so similar for all selected parts.
The maximum importance factor for the target part of the firewall nodes is reached at approximately state 80, thus the analysis state is chosen slightly earlier at state 78, as explained in Section 4.1.2.
State 68 is chosen for the source part of the break boosters, since the importance factors for the nodal displacements of this part show a very similar behaviour as for the target part, but the source state must be chosen before the target state. Furthermore, the curve for the break booster shows a gradient change until state 62, which is why the source state should be selected on the rather straight segment of the curve afterwards.
The development of the first linear importance factors for the plastic strain of the same parts is plotted in Fig. 5.42. It is obvious from the plots that the difference in the plastic strain is most prominent in the elements of the longitudinal rail, which was identified as the second source. These differences occur shortly after the impact at state 7 and show a steep increase at around state 18. Because of the change in the ascent, the source state for the plastic strain of the elements in the longitudinal rail is selected as 9 in order to get the earliest possible and isolated effect.
With the target and possible source parts, with the relevant quantities as well as corresponding states identified and listed in Tab. 5.15, the Extended Workflow can be applied to the Silverado example as well.

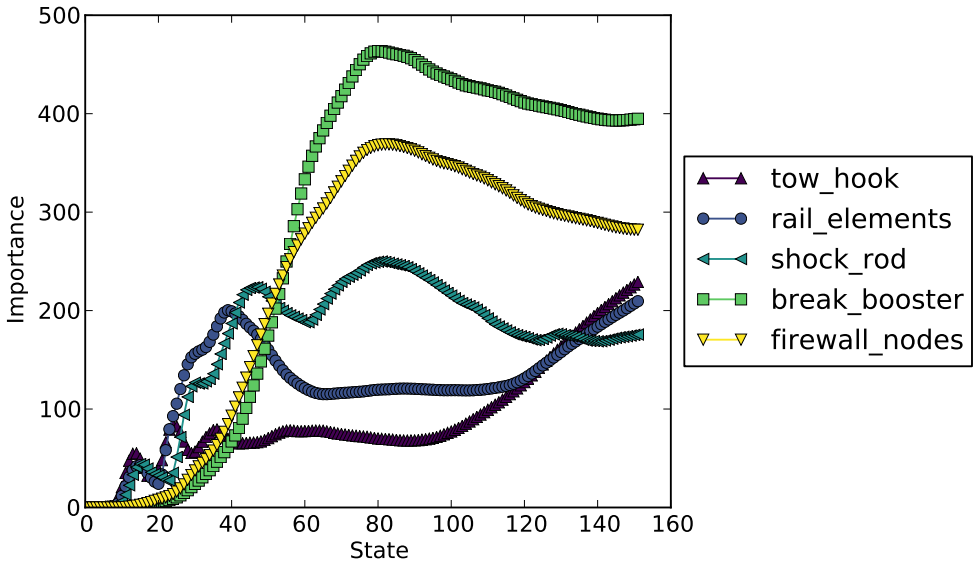


Figure 5.41: Importance of the first linear mode over time for the nodal displacement of selected parts in the Silverado example.

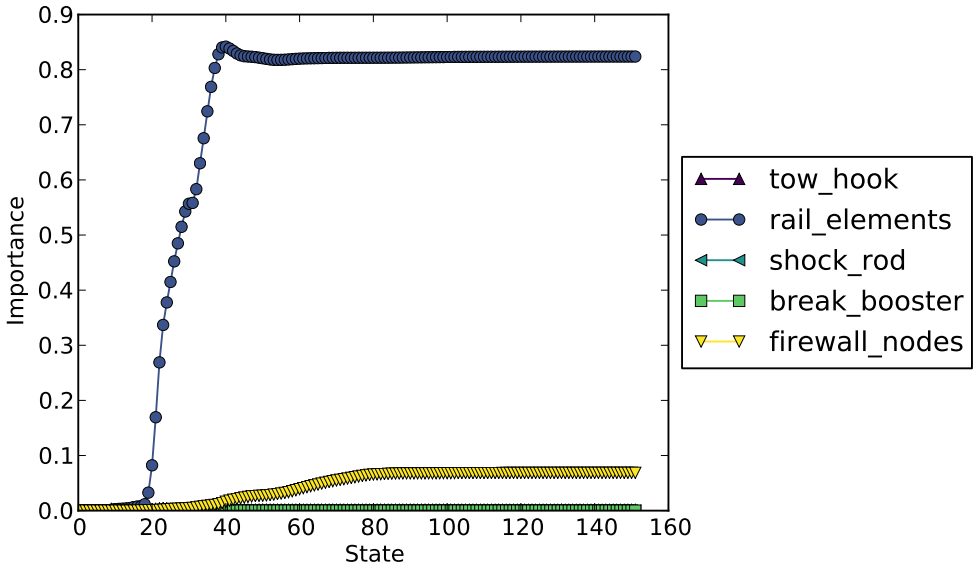


Figure 5.42: Development of the first importance factor for the plastic strain of selected parts in the Silverado example as computed with the linear PCA approach. The very stiff parts such as the tow hook, the shock rod and the break booster show hardly any plastic strain and thus hardly any difference, which is why the importance factors for all three are close to zero.

Type	Part	Quantity	State	Time
Target	Fire wall node group	Nodal displacements	78	77 ms
Source 1	Break booster	Nodal displacements	68	67 ms
Source 2	Longitudinal rail node group	Shell plastic strain	9	8 ms

Table 5.15: List of target and sources for the Silverado example in the Extended Workflow.

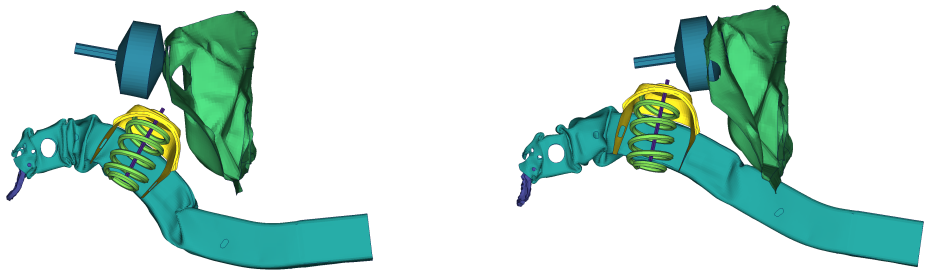
5.2.3.3 Application of the Extended Workflow

The Extended Workflow is applied for this example starting from the target point of view. First, the Dimensionality Reduction Methods are used to determine the intrinsic structure of the variance on the displacements of the firewall node group. Tab. 5.16 lists the first nine importance factors for the different DRMs. The neighbourhood size was set to $k = 10$ for all approaches, as this value yielded the best results. All approaches determined one very important dimension and a varying number of minor modes, with no more than eight dimensions worth mentioning for the nonlinear methods. The additional minor modes show that the movement of the wall is more complex than in the other examples investigated before, but the single most important direction underlines that the behaviour is mostly one dimensional.

DRM	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
Imp. factor 1	365.229	286.150	296.539	471.007	365.671	468.825	365.099
Imp. factor 2	82.945	83.658	87.752	96.758	81.820	90.924	80.065
Imp. factor 3	67.435	76.512	54.224	64.099	67.176	57.328	65.671
Imp. factor 4	38.522	53.580	53.423	54.926	39.886	47.221	35.155
Imp. factor 5	26.914	34.786	41.582	45.313	2.693	38.448	21.689
Imp. factor 6	20.254	28.682	33.086	19.358	0.141	29.202	13.248
Imp. factor 7	18.131	15.535	25.611	12.883	-	25.820	-
Imp. factor 8	9.404	15.224	12.392	7.608	-	15.147	-
Imp. factor 9	8.370	0.003	0.002	-	-	-	-

Table 5.16: Importance factors for the nodal displacements of the target firewall segment at state 78 computed by the different DRMs. An entry of "-" indicates that the method stopped with fewer dimensions.

Since all DRMs determined that the underlying manifold is essentially one dimensional, the first mode is crucial for the behaviour of the firewall. Fig. 5.43 displays the selected parts of the virtual simulation results for the first linear mode. The displayed state shows the interlocking of the shock rod or its housing with the break booster. Depending on this interlocking, the break booster is pushed into the firewall or not: In some simulations the former can be observed, in others the latter. Thus, a significant correlation between the variance of the two parts is to be expected.



(a) Projection $F\left(x_1^-\right)$ (b) Projection $F\left(x_1^+\right)$

Figure 5.43: State 68 of the virtual simulation results for the first linear mode of the displacements of the firewall node group. The difference in the firewall is mainly a translation in x-direction, which can be seen from the extent of the wall in this side view.

As stated before, this interaction was already found with the linear approach in [BST15] and is the reason why the break booster's displacement was chosen as a first source. In addition to the linear PCA, the nonlinear DRMs were also applied to the displacement of the break booster at state 68 and the results are listed in Tab. 5.17. Similar as for the target, the number of neighbours was chosen as $k = 10$ for all nonlinear approaches as it provided the best results. The calculated importance factors for the break booster show a large first mode, but also further significant modes. The actual number of relevant modes varies among the different methods, but the work in [BST15] showed that subtracting only the first mode already yields a very strong correlation and hence no further modes need to be subtracted.

DRM	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
Imp. factor 1	404.785	310.774	336.740	524.286	386.571	521.115	385.624
Imp. factor 2	176.022	184.399	193.035	196.661	181.436	187.504	177.467
Imp. factor 3	74.197	106.557	69.106	88.160	88.143	79.207	86.099
Imp. factor 4	37.988	54.695	49.002	77.980	58.295	62.827	55.606
Imp. factor 5	25.618	44.767	47.843	64.045	1.688	51.463	23.484
Imp. factor 6	11.515	41.755	33.278	5.577	0.001	42.753	15.19
Imp. factor 7	6.475	8.391	25.389	4.197	-	24.139	-
Imp. factor 8	3.364	0.001	0.001	-	-	-	-

Table 5.17: Importance factors for the nodal displacements of the break booster at state 68 in the Silverado example.

In contrast to the two simulation data examples before, the investigation of the correlation between the parts in this example was performed slightly different:
First, only the $e = 1$ single largest mode was subtracted and the resulting difference measures δ_{spec} and δ_{var} are listed in Tab. 5.18. Choosing a larger second number of modes as for the other two data sets is difficult since MLLE, for example, has a substantial seventh mode, while PTU can subtract at most five modes. Thus, only the single largest mode among several DRM approaches can be reasonably compared.

Secondly, for the linear method, both DPCA variants were evaluated. The two variants are the orthogonal projection-based variant and the τ-modified Gram matrix variant with the industry standard $\tau = 10\,000$, see Section 4.1.3. For the other data sets, the results were very similar and τ could have been chosen sufficiently large so that including the modification variant would not have yielded further insights compared to the orthogonal projection introduced in this work. But, since the investigation in [BST15] was performed using the modification variant with $\tau = 10\,000$, it is important to include it in this evaluation to reproduce the published results.
The results for the difference deltas confirm the previous findings and what was visually prominent in the virtual simulations: The correlation between the difference in the displacements of the break booster at an earlier state and the movement of the firewall nodes at a slightly later state is very high. When subtracting the first mode, the linear methods already obtain a reduction of the target variance of $\delta_{\text{var}} \approx 80\%$. The nonlinear methods all yield even higher values, thus confirming a strong correlation between the nodal displacements and concluding the investigation of the first source.

Measure	PCA_τ	$\text{PCA}_\perp$	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
δ_{spec}	60.8%	61.2%	70.1%	76.8%	86%	77.2%	84.2%	81.8%
δ_{var}	79.8%	80.1%	84.6%	89.9%	95.4%	91%	95.3%	92.9%

Table 5.18: Difference result break booster to wall nodes for the different DRMs. The nonlinear variants are utilising the DLAI difference method. The $\text{PCA}_\perp$ uses the orthogonal projection based DPCA and PCA_τ the modification variant with $\tau = 10\,000$.

The second possible source is the plastic strain of the elements on the front part of the longitudinal rail at the early state 9 and was newly investigated in this work. For this post value, the number of neighbours was chosen as $k = 6$ as the linear importance factors suggest a one dimensional intrinsic structure. Tab. 5.19 lists the resulting importance factors. The smaller range of values for the plastic strain and the earlier state yield much smaller overall importance factors compared to the nodal displacements: Since the range of values is smaller, the differences are also smaller, and since the initial geometry of the simulations is the same, the differences start to unfold over the duration of the crash, so the early state also dampens the differences. The relative size of the importance factors for the plastic strain compared to each other essentially indicates an intrinsic dimension of one.

DRM	PCA	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
Imp. factor 1	0.368	0.354	0.373	0.432	0.387	0.431	0.387
Imp. factor 2	0.081	0.071	0.025	0.051	0.053	0.045	0.052
Imp. factor 3	0.030	0.042	0.019	0.037	0.022	0.034	0.020
Imp. factor 4	0.015	0.024	0.012	0.024	-	0.022	-
Imp. factor 5	0.010	0.0	0.010	0.014	-	0.015	-
Imp. factor 6	0.009	0.0	0.005	0.013	-	0.011	-
Imp. factor 7	0.005	0.0	0.001	0.009	-	0.009	-
Imp. factor 8	0.004	0.0	0.001	0.001	-	0.006	-

Table 5.19: Importance factors for the second source in the Silverado example. The values are computed using the shell element plastic strain of at the connection between the longitudinal rail and the tow hook at state 9.

Because of this intrinsic dimension of one and to increase the comparability to the previous source part, again only the $e = 1$ single largest mode is subtracted. The results of this operation are displayed in Tab. 5.20 and show significant disparities between the different approaches:
The linear PCA_τ method using the τ-based DPCA difference operation with the industry standard of $\tau = 10\,000$ does not yield any significant correlation between the plastic strain of the source elements and the targets nodal displacements. This can partially be explained, by the different magnitude of the eigenvalues involved and the resulting requirements on τ. As the formula derived in Section 4.1.3 of this thesis yields a required τ of

$$\tau \overset{!}{\geq} \frac{\|G_\mathcal{Y}\|_2}{\sigma_e} = \frac{365.229^2}{0.368} \approx 362\,478.865$$

which means, that the default value of τ is too small for this application. This emphasises that caution is advised when using the modification variant of DPCA in connection with quantities of different magnitude.
The linear $\text{PCA}_\perp$ approach utilising the orthogonal projection-based DPCA yields substantially higher difference measures, reducing the target's variance by almost half with a $\delta_{\text{var}} = 49.9\%$ and suggesting a strong correlation. From an application point of view, if half of the variance is correlated with the investigated source, the analyst might expect at least a second source of scatter.
The nonlinear approaches, on the contrary, all yield δ_{var} reductions of the target's variance by more than two thirds and up to 83.2%, which would suffice to qualify as the only source in practical applications.

Measure	PCA_τ	$\text{PCA}_\perp$	LTSA	MLLE	Isomap	PTU	GNLM	PTNLM
δ_{spec}	0.75%	32.0%	60.8%	60.8%	57.3%	46.2%	62.2%	52.2%
δ_{var}	1.73%	49.9%	81.8%	81.7%	79.5%	69.6%	83.2%	75.3%

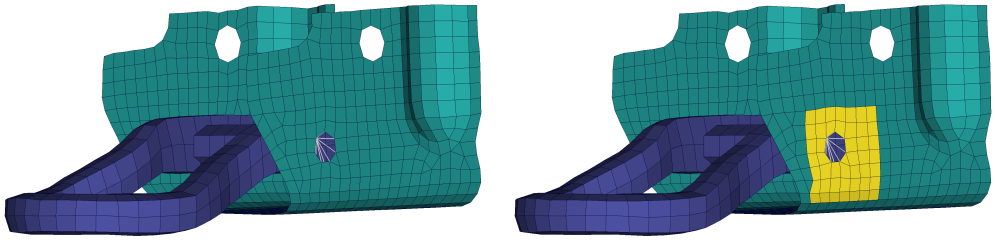
Table 5.20: Difference result of the different approaches for the element groups plastic strain as a source and the firewall nodes displacement as a target.

The Silverado example has shown that the newly developed methods can also handle more complex data sets in the scale of full car applications. It has highlighted the importance of choosing the correct τ when working with the DPCA approach and has shown the possibilities of the orthogonal projection-based alternative. With regards to the results from earlier publications, the new methods were able to confirm the existing findings of a linear dependency. Furthermore, including a different post value, e.g. plastic strain, can provide additional insights, especially in combination with the nonlinear methods. As already explained for the Cylinder example in Section 5.2.1.3, different post values can help to detect an effect at earlier states. In this example, a nodal variance at state 78 was correlated with a plastic strain difference at an early state of 9.

5.2.3.4 Plausibility Check

The newly found correlation of the target scatter on the important area of the firewall with a very local effect at a very early state raises the question of whether these findings are valid or not.
While the other findings of the Cylinders example of Section 5.2.1 were justified constructively and the results for the Rocker of Section 5.2.2 could be reasoned for with the help of the simple structure and the generation by the GHT optimisation, the results for this complex example can only be validated by a plausibility check.

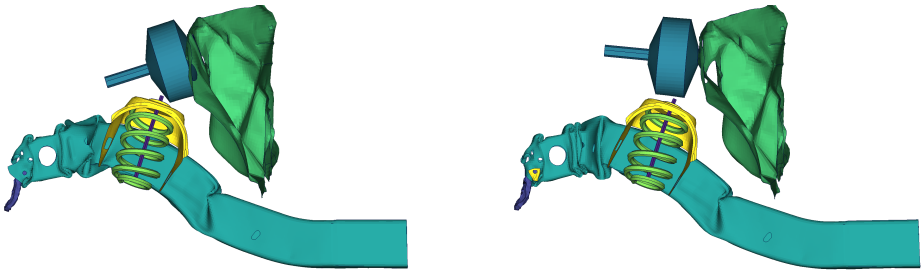
The findings of the new source revolve around a single triangular element and its neighbours in a peculiar position near a hole, which has an unusually high post value variance compared to the other elements in the rail. For practical applications, it is crucial to find such local areas that have a large impact on the performance of the simulation.
The mesh structure of the model is used to perform a minimally invasive plausibility check: The 77 simulation results of the example were generated by varying the element thicknesses of a baseline design by $\pm 20\%$ within their original value. This baseline design, as available for download in [Nat19], was simulated once without any modifications to obtain a reference run. Afterwards, a new model was generated by taking the same baseline design and introducing a new part, consisting only of the elements near the connection to the hook, see Fig. 5.44. Without changing any mesh connectivity or nodal positions, the thickness of this newly introduced part was reduced to 80% of the initial rail thickness, matching the minimum of all given samples. The contacts to all parts were handled in the same manner as for the original rail. Although further modifications could be done, only this minimal change was made to have an as local as possible impact on the simulation. The modified design was then simulated on the identical machine with the same configuration as the reference run.



(a) Original model (b) Modified model

Figure 5.44: Modification in the element mesh of the Silverado example. The node positions and the connectivity of the elements are untouched, only the thickness of the highlighted elements near the tow hook was reduced.

As explained in the last section, the behaviour of the break booster correlates strongly with the variance on the firewall, since in some simulations the booster interlocks with the shock rod and is thus pushed into the firewall. When inspecting the reference run for this behaviour, it can be seen, that the two parts do indeed interlock, see Fig. 5.45.a. The simulation with identical connectivity and initial node positions but only locally minimal changed thicknesses shows no such interlocking, see Fig. 5.45.b.



(a) Original simulation (b) Modified simulation

Figure 5.45: Selected parts at state 68 in the two simulation results computed for the plausibility check. The modified simulation has a local change in the initial thicknesses.

This means that the behaviour of the target part in the simulation was qualitatively changed by modifying the thickness of a small group of elements. These elements were identified by the new approaches developed in this thesis as having a strong correlation to the target part, underlining their capabilities to find these unexpected dependencies. Once identified these dependencies can be utilised to improve an existing model. In this specific example, the complete connection between the longitudinal rail and the tow hook, as well as all local geometry, is modelled differently in the newer 2017 variant of the Silverado.

The plausibility check of the last simulation example concludes the evaluation of the newly developed methods.

5.3 Recapitulation

In the last sections, the capabilities of the new nonlinear approaches have been demonstrated and compared to the linear method.

The controlled environment of artificial data sets enabled an individual evaluation of the two steps of the Extended Workflow and a comparison to the ideal result.
In the first reduction step, all DRMs could handle linear manifolds very well, but only the nonlinear approaches were able to capture the intrinsic structure of the nonlinear data sets. The number of possible DRMs for this step was subsequently reduced since PCA and ENLM yielded redundant results and LLE and MLLE aimed to preserve similar properties.
In the second step, all these possible DRMs were combined with the difference operation methods. Here, the linear PCA was always used in combination with the linear DPCA approach, and all nonlinear DRMs were combined with both the DLLI and the DLAI method. While all difference operation methods were able to detect linear dependencies very well, only the nonlinear methods were able to handle nonlinear dependencies. The investigation of the random dependency showed that the DLLI method failed to obtain meaningful results for this almost uncorrelated relation, while the DLAI method performed much better on this example. It is noteworthy that this chaotic relation is one of the most challenging tests for these methods. Because of this difference in the performance, the focus of the further evaluations was placed on the DLAI for nonlinear methods.
Then, the impact of the different methodologies of the DRMs approaches on the DLAI operation was demonstrated with two nonlinear non-developable manifolds.
Before evaluating the method's performance on simulation data, it was demonstrated that additional complexities such as noise or minor violations of the underlying assumptions can be handled by the methods. This heuristic investigation was conducted for noise levels and differences in the intrinsic dimension, which are usually found in practical applications.

The complete workflow was then evaluated on three simulation data examples. These three investigated examples include data from different applications and have an increasing size as well as complexity.
The constructed Cylinders example underlined the limitations of the linear approach and the benefit of the nonlinear methods. All nonlinear DRMs in combination with the DLAI obtained almost perfect results.
For the Rocker example, which was simulated during an optimisation process, the linear DPCA did not find any significant correlation between the source parts and the target part. This is unexpected because the source parts were specifically inserted by the optimisation to influence the behaviour of said target part. The nonlinear

methods found a significant correlation, and while most of the methods obtained comparable results, the Isomap approach deviated from this majority outcome.
Finally, in the Silverado example, a known linear dependency was confirmed by the nonlinear methods and a new nonlinear dependency was discovered. This new dependency was additionally confirmed by a plausibility check. Again, most of the DRMs obtained similar results, but with a slightly wider range than in previous examples.

The deviation of single methods from the majority outcome underlines the need to use several different approaches in an analysis. If the number of DRMs is to be reduced further, the above stated findings recommend to use at least one method from each class of DRM, because the different classes have different strengths and weaknesses. Because of the overall performance, a minimal set of DRMs could be MLLE for the class of LMs, PTU for the MDS class and anyone of GNLM or PTNLM for the NLM methods. These nonlinear DRMs in combination with the DLAI difference operation have shown to reliably identify linear and nonlinear correlations between the behaviour of different parts in a given set of simulation results.

6 Conclusion

Finally, the investigations and findings in this thesis are summarised and the central research question is revisited. Conclusively, the limitations of the presented study are highlighted and opportunities for future work are addressed.

6.1 Summary

Prior work on the Comparative Analysis of simulation results has shown the capabilities of Dimensionality Reduction Methods in these engineering analyses. For example, in the context of car crash simulation results, where, amongst others, the publications [BBT13] and [Oka15] have highlighted their benefit in practical applications.
However, the Difference Principal Component Analysis [TNNC10], which is used by several car manufacturers worldwide to investigate correlations between different parts of a simulation, is based on a linear Dimensionality Reduction concept, while the underlying data contains many nonlinearities. Other publications such as [BGIT+13] and [GIT15] have shown the advantages of nonlinear methods, often concentrating on one or two approaches and never in combination with the DPCA workflow. The central research question of this thesis was how the basic idea of this workflow can be extended to certain nonlinear methods and what functional differences these new approaches yield.
To answer this question, this thesis combined the two steps Dimensionality Reduction and Difference Dimensionality Reduction of the DPCA's Extended Workflow introduced in Section 2.2 with nonlinear DRMs. For the first step of the Dimensionality Reduction in Chapter 3, several generative nonlinear DRMs approaches were visited and successfully modified to be used in the CA. These approaches can be structured in three classes: In the first class of the so-called Local Method approaches the Locally Linear Embedding, Local Tangent Space Alignment and Modified Locally Linear Embedding methods were explained in Section 3.3. For the second class of the Multidimensional Scaling methods the Isomap and Parallel Transport Unfolding were covered in Section 3.4. The last class comprises the Nonlinear Mapping approaches of Section 3.5, where the Euclidean Nonlinear Mapping and Graph-Based Nonlinear Mapping methods were explained, and the Parallel Transport Nonlinear Mapping was newly introduced. All methods of this first step were extended by nonlinear importance factors and virtual simulation generation to be usable in the analysis and enhance the results.
For the second step of the difference operation in Chapter 4, the DPCA idea was first successfully abstracted to the Generalised Difference Dimensionality Reduction in Section 4.2 to formulate the concept in such a way that it can be transferred to nonlinear methods. Two specific new implementations of this abstract concept were

introduced, namely the Difference Local Linear Interpolation and the Difference Local Affine Interpolation.
With these modifications, the DPCA concept was extended to generative nonlinear reduction methods for the first time. The functional properties of these nonlinear methods were thoroughly tested in Chapter 5, first on artificial examples and then on simulation result data. In the process, both the DRMs and the difference methods were evaluated under certain aspects and the results of the different methods were compared with each other and with the ideal outcome, if known.
The evaluation of the methods has shown that the nonlinear approaches can find correlations between parts that were undiscovered by the linear state-of-the-art-method in both the constructed examples and the data collected from other applications. While the results of the different nonlinear methods were often in agreement, individual approaches sometimes deviated from the majority outcome, which underlines the importance of utilising several methods rather than focussing on a single approach.
The findings in this thesis extend those of [BST15] and confirm that DRMs can be utilised to identify dependencies between different parts of a simulation, as the results for the linear dependency were reproduced. Furthermore, the capabilities of nonlinear methods in the analysis of simulation results, which were already shown for some of the approaches in [BGG16], [IT16] and [MSJ20], can now be utilised in this specific application due to the extensions made in this work.
With these nonlinear methods, correlations between parts were found that would have gone unnoticed by the linear approach, as the application on all simulation data examples in Section 5.2 showed. Especially the importance of a small area with a few elements was correctly identified, as the plausibility check showed that it had a strong impact on the overall performance of the model, though it was easy to overlook.
This thesis therefore shows that the newly developed methods help to understand the variation and dependencies in a given set of simulation results and can hence be utilised in a wide field of applications, e.g. in understanding the solution space of an optimisation problem as in the Rocker example or in a manufacturing tolerance motivated parameter variation as in the Silverado example.

6.2 Outlook

Certain aspects or limitations of the presented work motivate future research. First of all, a finite number of representatives of three classes of DRMs in combination with two difference methods was investigated in this thesis. Future research could focus on additional approaches for the Dimensionality Reduction step as well as for the Difference Dimensionality Reduction operation. With the large number of different approaches and underlying concepts already available in the literature, e.g. Local Orthogonality Preservation [LLW$^+$16] or Latent Variable Models [Sau20], many interesting combinations are possible.
A second aspect of the presented work are the assumptions specified in Section 3.1.4. The brief investigation in Section 5.1.5 covered this topic up to a level that is rel-

evant for the given examples. Research beyond this level could be an interesting subject for future work. While the topic of noise in application data is often covered to some degree in the introductory papers of the different DRMs, most DRMs share the assumption that the data lies on a single connected manifold of fixed dimension. Though the recommendation is to treat each component separately, the question on how to handle mixed dimension data sets in an integrated analysis was raised several times, e.g. with the "barbell" in [SR03]. The question is still open for many approaches and could motivate the development of additional methods.
A further aspect is the composition of target and source for the analysis. In this work, both source and target have always been a single post value at single state. Theoretically, more complex combinations are possible, but an in-depth investigation of these combinations is needed before utilising them in an analysis. While the extension to multiple states is relatively simple, the combination of different post values, e.g. nodal displacements and element strains, poses the challenge of equilibrating different metric units. Furthermore, the appropriate instances need to be chosen, which leads to the last aspect.
Many steps of the presented workflow contain one or several manual components. On a small scale, this involves, for example, the selection of the number of neighbours to construct a neighbourhood or the respective state for the analysis. On a larger scale, the selection of post value and part for source as well as target is also left to the analyst. Most of these manual interactions could theoretically be automated. During the research for this work, a simple approach to getting an estimate for the number of neighbours was utilised, but the presented number was always subsequently chosen manually. Often the final number of neighbours was identical to the simple estimate of Section 3.3.2.2, showing that an automation of this step is possible, though further research on this topic is needed.
The selection of the appropriate post value or part for source and target of the analysis could also be automated. If the target of the analysis is already known, as in the case of the Silverado's firewall or the seat cross member of the Rocker, all possible sources for this target could be investigated, e.g. by a brute force or heuristic approach. In the case where the target is also unknown, perhaps the best possible source for as many targets as possible could be searched, potentially revealing critical components of a model.
For both approaches, it is crucial to identify dependencies between sources and targets, which is the purpose of the new methods developed in this work. With further automation, these new methods and their improved capability to identify correlations between different parts could provide interesting and encouraging new insights.

A Appendix

A.1 Projected Eigenvalue Decomposition

The LMs of Section 3.3 discard the eigenvector associated with the smallest eigenvalue, because it is assumed that this vector is $\mathbb{1}_s$ and the only eigenvector with eigenvalue 0. In theory, if the local properties do not yield an overdetermined system, multiple eigenvectors to the value 0 can exist. In this case, the matrix can be projected onto the orthogonal complement of the constant $\mathbb{1}_s$ vector prior to solving the eigenvalue system. For this, determine any orthogonal matrix $B \in \mathbb{R}^{s \times s}$ with:

$$B = (b_1, \ldots, b_{s-1}, \mathbb{1}_s)$$

By multiplying the original alignment matrix Φ_{LM} with the rectangular matrix $B|_{s-1}$ from both sides, the corresponding rows and columns are removed. Then the eigenvalue problem can be solved for this reduced matrix and the eigenvectors can be projected back into the original domain

$$\begin{aligned} & w \text{ EV of } B|_{s-1}^{\mathsf{T}} \Phi_{\text{LM}} B|_{s-1} \\ \Rightarrow & B|_{s-1}\ w \end{aligned}$$

to get eigenvectors of Φ_{LM} which are orthogonal to $\mathbb{1}_s$.

B Data Sets

B.1 Summary of Artificial Data Sets

Definition B.1
Plane. $f_{plane} : [0,1]^2 \to \mathbb{R}^D$

$$f_{plane}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} 3x_1 - 1.5 \\ 2x_2 - 1 \\ \mathbb{0}_{D-2} \end{pmatrix}$$

Definition B.2
S-shape. $f_{sshape} : [0,1]^2 \to \mathbb{R}^D$, *with* $\phi_{sshape} : [0,1] \to \mathbb{R}$

$$\phi_{sshape}(x_1) := \begin{cases} \frac{3}{4}(\cos(3\pi x_1) - 1) \ , & x_1 > \frac{1}{2} \\ -\frac{3}{4}(\cos(3\pi x_1) - 1) \ , & x_1 \leq \frac{1}{2} \end{cases}$$

$$f_{sshape}(x) := \begin{pmatrix} \phi_{sshape}(x_1) \\ \frac{3}{4}\sin(3\pi x_1) \\ 2x_2 - 1 \\ \mathbb{0}_{D-3} \end{pmatrix}$$

Definition B.3
Heated Swissroll with constant offset $c \in \mathbb{R}$. $f_{hroll,c} : [0,1]^2 \to \mathbb{R}^D$, *with* $\phi_{hroll} : [0,1] \to \mathbb{R}$*:*

$$\phi_{hroll}(x_2) := \frac{3}{2}(x_2 - \frac{1}{2})$$

$$f_{hroll,c}(x) := \begin{pmatrix} 2(1 + \phi_{hroll}(x_2)^2)\sqrt{x_1 + c} & \cos(4\pi\sqrt{x_1 + c}) \\ 2(1 + \phi_{hroll}(x_2)^2)\sqrt{x_1 + c} & \sin(4\pi\sqrt{x_1 + c}) \\ 2x_2 - 1 & \\ \mathbb{0}_{D-3} & \end{pmatrix}$$

Definition B.4
Orientable Noise $f_{onoise} : [0,1]^0 \to \mathbb{R}^D$

$$f_{onoise}(x) := \begin{pmatrix} \mathcal{N}(0,1) \\ \vdots \\ \mathcal{N}(0,i) \\ \vdots \\ \mathcal{N}(0,D) \end{pmatrix}$$

Definition B.5
Petals $f_{petals} : [0,1]^2 \to \mathbb{R}^D$, *with* $\phi_{petal,j} : [0,1]^2 \to \mathbb{R}^2, j \in \{1,2,3,4\}$:

$$\phi_{petal,1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2 - \frac{1}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2 - \frac{1}{2}\right)\right) \end{pmatrix}$$

$$\phi_{petal,2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} -\sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2 - \frac{3}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2 - \frac{3}{2}\right)\right) \end{pmatrix}$$

$$\phi_{petal,3}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2 - \frac{5}{2}\right)\right) \\ -\sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2 - \frac{5}{2}\right)\right) \end{pmatrix}$$

$$\phi_{petal,4}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \sin\left(\frac{2}{3}\pi x_1\right)\cos\left(\arccos\left(x_1\right)\left(4x_2 - \frac{7}{2}\right)\right) \\ \sin\left(\frac{2}{3}\pi x_1\right)\sin\left(\arccos\left(x_1\right)\left(4x_2 - \frac{7}{2}\right)\right) \end{pmatrix}$$

$$f_{petals}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \begin{cases} \phi_{petal,1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, 0 \le x_2 < \frac{1}{4} \\ \phi_{petal,2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{1}{4} \le x_2 < \frac{1}{2} \\ \phi_{petal,3}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{1}{2} \le x_2 < \frac{3}{4} \\ \phi_{petal,4}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \frac{3}{4} \le x_2 \le 1 \end{cases} \\ -\cos\left(\frac{2}{3}\pi x_1\right) \\ \mathbb{0}_{D-3} \end{pmatrix}$$

Definition B.6
Disk $f_{disk} : [0,1]^2 \to \mathbb{R}^D$:

$$f_{disk}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \sqrt{x_1}\cos\left(2\pi x_2\right) \\ \sqrt{x_1}\sin\left(2\pi x_2\right) \\ \mathbb{0}_{D-2} \end{pmatrix}$$

Definition B.7
Two planes $f_{twoplanes} : [0,1]^2 \to \mathbb{R}^D$, *with* $\phi_{left}, \phi_{right} : [0,1]^2 \to \mathbb{R}^3$:

$$\phi_{left}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} 3x_1 - 1.75 \\ 0 \\ 2x_2 - 1 \end{pmatrix}$$

$$\phi_{right}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} 3x_1 - 1.25 \\ 2x_2 - 1 \\ 0 \end{pmatrix}$$

$$f_{twoplanes}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \begin{cases} \phi_{right}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 \ge \frac{1}{2} \\ \phi_{left}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 < \frac{1}{2} \end{cases} \\ \mathbb{0}_{D-3} \end{pmatrix}$$

Definition B.8
Shovel $f_{shovel} : [0,1]^2 \to \mathbb{R}^D$, *with* $\phi_{blade}, \phi_{handle} : [0,1]^2 \to \mathbb{R}^3$*:*

$$\phi_{blade}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} -\frac{1}{5} - \frac{1}{4}\cos(\pi x_1) \\ \frac{1}{4} - \frac{1}{4}\sin(\pi x_1)) \\ \frac{1}{2}x_2 - \frac{1}{4} \end{pmatrix}$$

$$\phi_{handle}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} x_1 + \frac{1}{4}x_2 - \frac{1}{2} \\ 0 \\ 0 \end{pmatrix}$$

$$f_{shovel}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} \begin{cases} \phi_{blade}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 \geq \frac{1}{2} \\ \phi_{handle}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, & x_1 < \frac{1}{2} \end{cases} \\ \mathbb{0}_{D-3} \end{pmatrix}$$

B.2 Creating the Cylinders Example

The Cylinders example introduced in Section 5.2.1 was created using the following scripts. Since formatting is important for all of these files, white spaces are highlighted with the special symbol "␣" and must be replaced when storing these files. First, two LS-PREPOST [LC20a] command files must be run to create the LS-DYNA input files `floor.k` and `cylinder.k` for the respective components.

B.2.1 Needed Files

The contents of the command files are:

floor.cfile

```
$#␣LS-PrePost␣command␣file
cemptymodel
meshing␣4pshell␣create␣1␣1␣-75␣-75␣-3.75␣75␣-75␣7.5␣75␣75␣7.5␣-75␣75␣-3.75
ac
meshing␣4pshell␣accept␣1␣1␣1␣floor
ac
genselect␣target␣node
genselect␣clear
genselect␣node␣add␣node␣1/0
genselect␣node␣add␣node␣2/0
genselect␣node␣add␣node␣3/0
genselect␣node␣add␣node␣4/0
spcdata␣createnodeset␣1␣1␣NODESET(SPC)␣1
$genselect␣clear
spcdata␣create␣set␣1␣0␣1␣1␣1␣1␣1␣1
KEYWORD␣INPUT␣1
*MAT_PIECEWISE_LINEAR_PLASTICITY_TITLE
floor_mat
$#␣␣␣␣␣mid␣␣␣␣␣␣␣␣ro␣␣␣␣␣␣␣␣␣e␣␣␣␣␣␣␣␣pr␣␣␣␣␣␣␣sigy␣␣␣␣␣␣etan␣␣␣␣␣␣fail␣␣␣␣␣␣tdel
␣␣␣␣␣␣␣␣␣1␣7.8500E-9␣2.0000E+5␣␣0.300000␣215.00000␣␣␣␣␣0.0001.0000E+21␣␣␣␣␣0.000
$#␣␣␣␣␣␣␣c␣␣␣␣␣␣␣␣␣p␣␣␣␣␣␣␣lcss␣␣␣␣␣␣␣lcsr␣␣␣␣␣␣␣␣␣vp
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣0.000
$#␣␣␣␣eps1␣␣␣␣␣␣␣eps2␣␣␣␣␣␣␣eps3␣␣␣␣␣␣␣eps4␣␣␣␣␣␣␣eps5␣␣␣␣␣␣␣eps6␣␣␣␣␣␣␣eps7␣␣␣␣␣␣␣eps8
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000
$#␣␣␣␣␣es1␣␣␣␣␣␣␣␣es2␣␣␣␣␣␣␣␣es3␣␣␣␣␣␣␣␣es4␣␣␣␣␣␣␣␣es5␣␣␣␣␣␣␣␣es6␣␣␣␣␣␣␣␣es7␣␣␣␣␣␣␣␣es8
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000
*END
keyword␣updatekind
MAT_PIECEWISE_LINEAR_PLASTICITY
KEYWORD␣INPUT␣1
*SECTION_SHELL
```

```
$#␣␣␣secid␣␣␣␣elform␣␣␣␣␣␣␣shrf␣␣␣␣␣␣␣␣nip␣␣␣␣␣␣propt␣␣␣␣qr/irid␣␣␣␣␣␣icomp␣␣␣␣␣␣setyp
␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣2␣␣␣1.000000␣␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣1
$#␣␣␣␣␣␣␣t1␣␣␣␣␣␣␣␣␣t2␣␣␣␣␣␣␣␣␣t3␣␣␣␣␣␣␣␣␣t4␣␣␣␣␣␣␣nloc␣␣␣␣␣␣marea␣␣␣␣␣␣␣idof␣␣␣␣␣edgset
␣␣␣1.000000␣␣␣1.000000␣␣␣1.000000␣␣␣1.000000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣␣␣0
*END
keyword␣␣updatekind
SECTION_SHELL
KEYWORD␣␣INPUT␣␣1
*PART
$#␣␣title
floor
$#␣␣␣␣␣␣␣pid␣␣␣␣␣␣secid␣␣␣␣␣␣␣␣␣mid␣␣␣␣␣␣␣eosid␣␣␣␣␣␣␣hgid␣␣␣␣␣␣␣grav␣␣␣␣␣adpopt␣␣␣␣␣␣␣tmid
␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0
*END
keyword␣␣updatekind
PART_PART
save␣␣outversion␣␣2
save␣␣keyword␣␣"floor.k"
quit
```

cylinder.cfile

```
$#␣␣LS-PrePost␣␣command␣␣file
cemptymodel
left
meshing␣␣cylindersolid␣␣create␣␣20.0␣␣80.0␣␣16␣␣5␣␣0.0␣␣0.0,0.0,0.0,0.0,1.0
ac
meshing␣␣cylindersolid␣␣accept␣␣2␣␣10␣␣10␣␣cylinder
ac
elgenerate␣␣shelltype␣␣2
genselect␣␣target␣␣segment
loadstate␣␣1;
loadquat␣␣0.707107␣␣0.0␣␣0.0␣␣0.707107;
loadgrot␣␣1.0␣␣0.0␣␣0.0␣␣0.0␣␣0.0␣␣1.0␣␣0.0␣␣0.0␣␣0.0␣␣0.0␣␣1.0␣␣0.0␣␣0.0␣␣0.0␣␣0.0␣␣1.0␣␣;
loadzoom␣␣0.90;
loadeyepos␣␣0.0␣␣0.0␣␣0.0;
loadupvect␣␣0.0␣␣1.0␣␣0.0;
genselect␣␣element␣␣add␣␣region␣␣in␣␣0.349554␣␣0.952775␣␣0.764801␣␣1.011599
elgenerate␣␣shell␣␣solidface␣␣3␣␣330
genselect␣␣clear
elgenerate␣␣accept
genselect␣␣element␣␣add␣␣region␣␣in␣␣0.318329␣␣0.047225␣␣0.716951␣␣-0.101906
elgenerate␣␣shell␣␣solidface␣␣4␣␣394
genselect␣␣clear
elgenerate␣␣accept
iniveldata␣␣operationtype␣␣2
genselect␣␣target␣␣node
genselect␣␣clear
genselect␣␣node␣␣add␣␣part␣␣2/0
iniveldata␣␣create␣␣0␣␣0␣␣-10000␣␣0␣␣0␣␣0␣␣0␣␣1
genselect␣␣clear
KEYWORD␣␣INPUT␣␣1
*MAT_PIECEWISE_LINEAR_PLASTICITY_TITLE
cylinder_mat
$#␣␣␣␣␣␣mid␣␣␣␣␣␣␣␣␣ro␣␣␣␣␣␣␣␣␣␣e␣␣␣␣␣␣␣␣␣pr␣␣␣␣␣␣␣sigy␣␣␣␣␣␣␣etan␣␣␣␣␣␣␣fail␣␣␣␣␣␣␣tdel
␣␣␣␣␣␣␣␣␣2␣␣7.8500E-9␣␣2.0000E+5␣␣0.300000␣␣215.00000␣␣␣␣␣␣0.0001.0000E+21␣␣␣␣␣␣0.000
$#␣␣␣␣␣␣␣␣c␣␣␣␣␣␣␣␣␣␣p␣␣␣␣␣␣␣␣lcss␣␣␣␣␣␣␣lcsr␣␣␣␣␣␣␣␣␣vp
␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣0.000
$#␣␣␣␣␣eps1␣␣␣␣␣␣␣␣eps2␣␣␣␣␣␣␣␣eps3␣␣␣␣␣␣␣␣eps4␣␣␣␣␣␣␣␣eps5␣␣␣␣␣␣␣␣eps6␣␣␣␣␣␣␣␣eps7␣␣␣␣␣␣␣␣eps8
␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000
$#␣␣␣␣␣␣es1␣␣␣␣␣␣␣␣es2␣␣␣␣␣␣␣␣es3␣␣␣␣␣␣␣␣es4␣␣␣␣␣␣␣␣es5␣␣␣␣␣␣␣␣es6␣␣␣␣␣␣␣␣es7␣␣␣␣␣␣␣␣es8
␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000
*END
keyword␣␣updatekind
MAT_PIECEWISE_LINEAR_PLASTICITY
KEYWORD␣␣INPUT␣␣1
*SECTION_SOLID_TITLE
cylinder
$#␣␣␣secid␣␣␣␣elform␣␣␣␣␣␣␣␣aet
␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣0
*END
keyword␣␣updatekind
SECTION_SOLID
KEYWORD␣␣INPUT␣␣1
*SECTION_SHELL_TITLE
cylinder_faces
$#␣␣␣secid␣␣␣␣elform␣␣␣␣␣␣␣shrf␣␣␣␣␣␣␣␣nip␣␣␣␣␣␣propt␣␣␣␣qr/irid␣␣␣␣␣␣icomp␣␣␣␣␣␣setyp
␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣␣2␣␣␣1.000000␣␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣1
$#␣␣␣␣␣␣␣t1␣␣␣␣␣␣␣␣␣t2␣␣␣␣␣␣␣␣␣t3␣␣␣␣␣␣␣␣␣t4␣␣␣␣␣␣␣nloc␣␣␣␣␣␣marea␣␣␣␣␣␣␣idof␣␣␣␣␣edgset
␣␣␣1.000000␣␣␣1.000000␣␣␣1.000000␣␣␣1.000000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣␣␣0
```

```
*END
keyword␣updatekind
SECTION_SHELL
KEYWORD␣INPUT␣1
*PART
$#␣title
cylinder
$#␣␣␣␣␣pid␣␣␣␣␣secid␣␣␣␣␣␣␣mid␣␣␣␣␣eosid␣␣␣␣␣␣hgid␣␣␣␣␣␣grav␣␣␣␣adpopt␣␣␣␣␣␣tmid
␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
*END
keyword␣updatekind
PART_PART
KEYWORD␣INPUT␣1
*PART
$#␣title
top
$#␣␣␣␣␣pid␣␣␣␣␣secid␣␣␣␣␣␣␣mid␣␣␣␣␣eosid␣␣␣␣␣␣hgid␣␣␣␣␣␣grav␣␣␣␣adpopt␣␣␣␣␣␣tmid
␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
*END
keyword␣updatekind
PART_PART
KEYWORD␣INPUT␣1
*PART
$#␣title
bottom
$#␣␣␣␣␣pid␣␣␣␣␣secid␣␣␣␣␣␣␣mid␣␣␣␣␣eosid␣␣␣␣␣␣hgid␣␣␣␣␣␣grav␣␣␣␣adpopt␣␣␣␣␣␣tmid
␣␣␣␣␣␣␣␣␣4␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
*END
keyword␣updatekind
PART_PART
KEYWORD␣INPUT␣1
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
$#␣␣␣␣␣cid␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣title
␣␣␣␣␣␣␣␣␣1cylinder_to_top
$#␣␣␣␣ssid␣␣␣␣␣␣msid␣␣␣␣␣sstyp␣␣␣␣␣mstyp␣␣␣␣sboxid␣␣␣␣mboxid␣␣␣␣␣␣␣spr␣␣␣␣␣␣␣mpr
␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
$#␣␣␣␣␣␣fs␣␣␣␣␣␣␣␣fd␣␣␣␣␣␣␣␣dc␣␣␣␣␣␣␣␣vc␣␣␣␣␣␣␣vdc␣␣␣␣penchk␣␣␣␣␣␣␣␣bt␣␣␣␣␣␣␣␣dt
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣0␣␣␣␣␣0.0001.0000E+20
$#␣␣␣␣␣sfs␣␣␣␣␣␣␣sfm␣␣␣␣␣␣␣sst␣␣␣␣␣␣␣mst␣␣␣␣␣␣sfst␣␣␣␣␣␣sfmt␣␣␣␣␣␣␣fsf␣␣␣␣␣␣␣vsf
␣␣1.000000␣␣1.000000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣1.000000␣␣1.000000␣␣1.000000␣␣1.000000
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
$#␣␣␣␣␣cid␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣title
␣␣␣␣␣␣␣␣␣2cylinder_to_bottom
$#␣␣␣␣ssid␣␣␣␣␣␣msid␣␣␣␣␣sstyp␣␣␣␣␣mstyp␣␣␣␣sboxid␣␣␣␣mboxid␣␣␣␣␣␣␣spr␣␣␣␣␣␣␣mpr
␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣4␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
$#␣␣␣␣␣␣fs␣␣␣␣␣␣␣␣fd␣␣␣␣␣␣␣␣dc␣␣␣␣␣␣␣␣vc␣␣␣␣␣␣␣vdc␣␣␣␣penchk␣␣␣␣␣␣␣␣bt␣␣␣␣␣␣␣␣dt
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣0␣␣␣␣␣0.0001.0000E+20
$#␣␣␣␣␣sfs␣␣␣␣␣␣␣sfm␣␣␣␣␣␣␣sst␣␣␣␣␣␣␣mst␣␣␣␣␣␣sfst␣␣␣␣␣␣sfmt␣␣␣␣␣␣␣fsf␣␣␣␣␣␣␣vsf
␣␣1.000000␣␣1.000000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣1.000000␣␣1.000000␣␣1.000000␣␣1.000000
*END
keyword␣updatekind
CONTACT_AUTOMATIC_SURFACE_TO_SURFACE

save␣outversion␣2
save␣keyword␣"cylinder.k"
quit
```

The two created key files are combined in one additional input file to get the basic solvable input deck.

combine.k

```
$-------------------------------------------------------------------------------
*KEYWORD
*TITLE
Cylinders␣Example
$-------------------------------------------------------------------------------
$␣Patch␣Contacts
$-------------------------------------------------------------------------------
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
$#␣␣␣␣␣cid␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣title
␣␣␣␣␣␣␣␣␣1floor_to_bottom
$#␣␣␣␣ssid␣␣␣␣␣␣msid␣␣␣␣␣sstyp␣␣␣␣␣mstyp␣␣␣␣sboxid␣␣␣␣mboxid␣␣␣␣␣␣␣spr␣␣␣␣␣␣␣mpr
␣␣␣␣␣␣␣␣␣4␣␣␣␣␣␣␣␣␣1␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣3␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣0
$#␣␣␣␣␣␣fs␣␣␣␣␣␣␣␣fd␣␣␣␣␣␣␣␣dc␣␣␣␣␣␣␣␣vc␣␣␣␣␣␣␣vdc␣␣␣␣penchk␣␣␣␣␣␣␣␣bt␣␣␣␣␣␣␣␣dt
␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣0␣␣␣␣␣0.0001.0000E+20
$#␣␣␣␣␣sfs␣␣␣␣␣␣␣sfm␣␣␣␣␣␣␣sst␣␣␣␣␣␣␣mst␣␣␣␣␣␣sfst␣␣␣␣␣␣sfmt␣␣␣␣␣␣␣fsf␣␣␣␣␣␣␣vsf
␣␣1.000000␣␣1.000000␣␣␣␣␣0.000␣␣␣␣␣0.000␣␣1.000000␣␣1.000000␣␣1.000000␣␣1.000000
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
```

```
$#     cid                                                                 title
          2top_to_bottom
$#   ssid       msid       sstyp      mstyp      sboxid     mboxid     spr        mpr
         8          3          3          3          0          0          0          0
$#     fs        fd        dc        vc        vdc     penchk        bt        dt
    0.000     0.000     0.000     0.000     0.000          0     0.000 1.0000E+20
$#    sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf
  1.000000  1.000000  0.000000  0.000000  1.000000  1.000000  1.000000  1.000000
*CONTACT_AUTOMATIC_SURFACE_TO_SURFACE_ID
$#     cid                                                                 title
          3bottom_to_floor
$#   ssid       msid       sstyp      mstyp      sboxid     mboxid     spr        mpr
        13          9          3          3          0          0          0          0
$#     fs        fd        dc        vc        vdc     penchk        bt        dt
    0.000     0.000     0.000     0.000     0.000          0     0.000 1.0000E+20
$#    sfs       sfm       sst       mst      sfst      sfmt       fsf       vsf
  1.000000  1.000000  0.000000  0.000000  1.000000  1.000000  1.000000  1.000000
$---------------------------------------------------------------------------
$ Read Models
$---------------------------------------------------------------------------
*PARAMETER
R an_left       0.0
R an_right      0.0
R dx_right      225.0
R dz_lower      10.0
R dz_upper      100.0
R dy_upper      -30.0
$ Floor Left
*DEFINE_TRANSFORMATION
       1000
$ option &       dx&       dy&       dz&       px&       py&       pz&     angle&
    ROTATE     0.000     0.000     1.000     0.000     0.000     0.000  &an_left
*INCLUDE_TRANSFORM
floor.k
$#  idnoff    ideoff    idpoff    idmoff    idsoff    idfoff    iddoff
          0         0         0         0         0         0         0
$#  idroff
          0
$#  fctmas    fcttim    fctlen    fcttem  incout1
    0.000     0.000     0.000     0.000         0
$#  tranid
       1000
$ Cylinder Left Lower
*DEFINE_TRANSFORMATION
       1001
$ option &       dx&       dy&       dz&
    TRANSL       0.0       0.0 &dz_lower
*INCLUDE_TRANSFORM
cylinder.k
$#  idnoff    ideoff    idpoff    idmoff    idsoff    idfoff    iddoff
          0         0         0         0         0         0         0
$#  idroff
          0
$#  fctmas    fcttim    fctlen    fcttem  incout1
    0.000     0.000     0.000     0.000         0
$#  tranid
       1001
$ Cylinder Left Upper
*DEFINE_TRANSFORMATION
       1002
$ option &       dx&       dy&       dz&
    TRANSL       0.0 &dy_upper &dz_upper
*INCLUDE_TRANSFORM
cylinder.k
$#  idnoff    ideoff    idpoff    idmoff    idsoff    idfoff    iddoff
       1000      1000         4         4         4         0         0
$#  idroff
          0
$#  fctmas    fcttim    fctlen    fcttem  incout1
    0.000     0.000     0.000     0.000         0
$#  tranid
       1002
$ Floor Right
*DEFINE_TRANSFORMATION
       2000
$ option &       dx&       dy&       dz&       px&       py&       pz&     angle&
    ROTATE     0.000     0.000     1.000     0.000     0.000     0.000 &an_right
$ option &       dx&       dy&       dz&
    TRANSL &dx_right       0.0       0.0
```

```
*INCLUDE_TRANSFORM
floor.k
$#␣␣idnoff␣␣␣␣ideoff␣␣␣␣idpoff␣␣␣␣idmoff␣␣␣␣idsoff␣␣␣␣idfoff␣␣␣␣iddoff
␣␣␣␣␣␣␣2000␣␣␣␣␣␣␣2000␣␣␣␣␣␣␣␣␣␣8␣␣␣␣␣␣␣␣␣␣8␣␣␣␣␣␣␣␣␣␣8␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0
$#␣␣idroff
␣␣␣␣␣␣␣␣␣␣0
$#␣␣fctmas␣␣␣␣fcttim␣␣␣␣fctlen␣␣␣␣fcttem␣␣␣incout1
␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣␣0
$#␣␣tranid
␣␣␣␣␣␣␣2000
$␣Cylinder␣Right␣Lower
*DEFINE_TRANSFORMATION
␣␣␣␣␣␣␣2001
$␣option␣&␣␣␣␣␣␣␣dx&␣␣␣␣␣␣␣␣dy&␣␣␣␣␣␣␣␣dz&
␣␣␣␣TRANSL␣&dx_right␣␣␣␣␣␣␣␣0.0␣&dz_lower
*INCLUDE_TRANSFORM
cylinder.k
$#␣␣idnoff␣␣␣␣ideoff␣␣␣␣idpoff␣␣␣␣idmoff␣␣␣␣idsoff␣␣␣␣idfoff␣␣␣␣iddoff
␣␣␣␣␣␣␣2000␣␣␣␣␣␣␣2000␣␣␣␣␣␣␣␣␣␣9␣␣␣␣␣␣␣␣␣␣9␣␣␣␣␣␣␣␣␣␣9␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0
$#␣␣idroff
␣␣␣␣␣␣␣␣␣␣0
$#␣␣fctmas␣␣␣␣fcttim␣␣␣␣fctlen␣␣␣␣fcttem␣␣␣incout1
␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣␣␣␣␣0
$#␣␣tranid
␣␣␣␣␣␣␣2001
$---------------------------------------------------------------------------------
$␣Control␣Cards
$---------------------------------------------------------------------------------
*CONTROL_ENERGY
$#␣␣␣␣hgen␣␣␣␣␣␣␣rwen␣␣␣␣␣slnten␣␣␣␣␣␣rylen
␣␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣2␣␣␣␣␣␣␣␣␣␣1
*CONTROL_MPP_IO_NODUMP
*CONTROL_MPP_IO_NOFULL
*CONTROL_TERMINATION
$#␣␣endtim␣␣␣␣endcyc␣␣␣␣␣␣dtmin␣␣␣␣endeng␣␣␣␣endmas
␣␣0.003000␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000␣␣␣␣␣␣0.000
$-------------------------------------------------------------------------------------
$␣Output
$---------------------------------------------------------------------------------
*DATABASE_BINARY_D3PLOT
$#␣␣␣␣␣␣␣dt␣␣␣␣␣␣␣lcdt␣␣␣␣␣␣␣beam␣␣␣␣␣␣npltc␣␣␣␣␣psetid
␣1.0000E-4␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣0
$#␣␣␣␣ioopt
␣␣␣␣␣␣␣␣␣␣0
*DATABASE_FORMAT
$#␣␣␣␣iform␣␣␣␣ibinary
␣␣␣␣␣␣␣␣␣␣0␣␣␣␣␣␣␣␣␣␣1
*END
```

The variation and simulation of this basic set-up was done with the following Python script, which can be run by Python2 as well as by Python3.

varyAndSimulate.py

```
import␣os
from␣random␣import␣random,␣seed

lsdynaExec␣=␣""␣#␣"/home/user/Software/ls-dyna/R8.0.0/ls-dyna_smp_d_r800_x64_redhat59_ifort131"
femzipExec␣=␣""␣#␣"/home/user/Software/femzip/femzip_dyna"
inputFile␣=␣"combine.k"
numberOfIntervals␣=␣90
maxAngle␣=␣180
threads=2

inputFolder␣=␣os.path.dirname(inputFile)
if␣inputFolder␣==␣"":
␣␣␣␣inputFolder␣=␣".."
print("Input:␣"+inputFile)
seed(10)
for␣i␣in␣range(0,numberOfIntervals+1):
␣␣␣␣angle␣=␣i␣*␣maxAngle␣/␣numberOfIntervals
␣␣␣␣angle2␣=␣random()␣*␣maxAngle
␣␣␣␣if␣i␣==␣0:
␣␣␣␣␣␣␣␣angle2␣=␣angle
␣␣␣␣print("Processing␣angle:␣"+str(angle))
␣␣␣␣f␣=␣open(inputFile,␣"r")
␣␣␣␣directoryName␣=␣"run_"+str("%02d"␣%␣(i))
␣␣␣␣if␣not␣os.path.exists(directoryName):
␣␣␣␣␣␣␣␣os.mkdir(directoryName)
```

```
    os.chdir(directoryName)
    n = open("input.k", "w")
    for line in f:
        if line[:9] == "R an_left":
            line = "R an_left      {:5.1f}\n".format(angle)
        elif line[:10] == "R an_right":
            line = "R an_right     {:5.1f}\n".format(angle2)
        n.write(line)
    f.close()
    n.close()
    os.system("cp "+str(inputFolder)+"/floor.k .")
    os.system("cp "+str(inputFolder)+"/cylinder.k .")
    if lsdynaExec != "":
        os.system(lsdynaExec + " i=input.k NCPU= "+str(threads))
    if femzipExec != "":
        os.system(femzipExec + " -I d3plot -O d3plot.fz -C ../precisions.par -X")
    os.chdir("..")
print("Finished")
```

B.2.2 Order of Commands

With the above given files stored to a folder, the needed commands are:

```
lsprepost floor.cfile
lsprepost cylinders.cfile
python varyAndSimulate.py
```

B.3 Varying the Silverado Example

The thicknesses of the following 13 parts were randomly varied within ±20% of their original value in the Silverado example:

PID	Thickness in mm
2000048	1.051
2000167	3.175
2000168	3.000
2000196	3.000
2000274	2.980
2000293	6.000
2000475	1.800
2000482	2.000
2000484	2.000
2000605	1.529
2000629	3.980
2000665	0.500
2000678	15.000

Table B.1: Silverado PIDs and original thicknesses.

C Registers

List of Figures

List of Tables

List of Algorithms

List of References

[AGHH08] Ackermann, Sascha ; Gaul, Lothar ; Hanss, Michael ; Hambrecht, Thomas: Principal component analysis for detection of globally important input parameters in nonlinear finite element analysis. In: *Optimisation and Stochastic Days* 5 (2008)

[BBT13] Brown, Richard ; Borsotto, Dominik ; Thole, Clemens-August: Relating scatter in occupant injury time histories to instability in airbag behaviour. In: *Procs. 9th European LS-DYNA Users' Conference, Manchester, UK*, 2013

[Bel03] Bellman, R.E.: *Dynamic Programming*. Dover Publications, 2003 (Dover Books on Computer Science Series). – ISBN 9780486428093

[BFG05] Böttcher, Curd-Sigmund ; Frik, Steffen ; Gosolits, Bernd: *20 years of crash simulation at Opel-experiences for future challenges*. 2005

[BGG16] Bohn, Bastian ; Garcke, Jochen ; Griebel, Michael: A sparse grid based method for generative dimensionality reduction of high-dimensional data. In: *Journal of Computational Physics* 309 (2016), S. 1–17

[BGIT+13] Bohn, Bastian ; Garcke, Jochen ; Iza-Teran, Rodrigo ; Paprotny, Alexander ; Peherstorfer, Benjamin ; Schepsmeier, Ulf ; Thole, Clemens-August: Analysis of car crash simulation data with nonlinear machine learning methods. In: *Procedia Computer Science* 18 (2013), S. 621–630

[BJST16] Borsotto, Dominik ; Jansen, Lennart ; Strickstrock, Robin ; Thole, Clemens-August: Use of Data Reduction Methods for Robust Optimization. In: *Procs. 14th International LS-DYNA Users' Conference, Dearborn, USA*, 2016, S. 22–7–22–16

[BLTD17] Budninskiy, Max ; Liu, Beibei ; Tong, Yiying ; Desbrun, Mathieu: Spectral Affine-Kernel Embeddings. In: *Computer Graphics Forum* Bd. 36 Wiley Online Library, 2017, S. 117–129

[BM+76] Bondy, John A. ; Murty, Uppaluri Siva R. u. a.: *Graph theory with applications*. Bd. 290. Macmillan London, 1976

[BST+02] Balasubramanian, Mukund ; Schwartz, Eric L. ; Tenenbaum, Joshua B. ; Silva, Vin de ; Langford, John C.: The isomap algorithm and topological stability. In: *Science* 295 (2002), Nr. 5552, S. 7–7

[BST15] Borsotto, Dominik ; Strickstrock, Robin ; Thole, Clemens-August: Improving robustness of Chevrolet Silverado with exemplary design adaptations based on identified scatter sources. In: *10th European LS-DYNA conference*, 2015, S. C–1–4–22–C–1–4–27

[BYF+19] Budninskiy, Max ; Yin, Gloria ; Feng, Leman ; Tong, Yiying ; Desbrun, Mathieu: Parallel Transport Unfolding: A Connection-Based Manifold Learning Approach. In: *SIAM Journal on Applied Algebra and Geometry* 3 (2019), Nr. 2, S. 266–291

[CA02] Cichocki, Andrzej ; Amari, Shun-ichi: *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002

[Die19] Diez, Constantin: *Process for Extraction of Knowledge from Crash Simulations by means of Dimensionality Reduction and Rule Mining*, Bergische Universität Wuppertal, Dissertation, 2019. `http://d-nb.info/1182555063/34`

[DSAC10] Daza-Santacoloma, Genaro ; Acosta, Carlos D. ; Castellanos, Germán: Regularization parameter choice in locally linear embedding. In: *Neurocomputing* 73 (2010), Nr. 10-12, S. 1595–1605

[DWHS16] Diez, Constantin ; Wieser, Christian ; Harzheim, Lothar ; Schumacher, Axel: Automatic Generation of Robustness Knowledge for Selected Crash Structures. In: *Procs. 14th German LS-DYNA Forum, Bamberg, Germany*, 2016

[EKM+13] Eck, Christiana ; Kovalenko, Yevgeniya ; Mangold, Oliver ; Prohl, Raphael ; Tkachuk, Anton ; Trickov, Vladimir: Reduction of numerical sensitivities in crash simulations on HPC-computers (HPC-10). In: *High Performance Computing in Science and Engineering '13*. Springer, 2013, S. 679–697

[Eur19] European New Car Assessment Programme: *Oblique Pole Side Impact Testing Protocol For 2020 Implementation*. `http://www.euroncap.com/for-engineers/protocols/`. Version: 2019. – last checked 07.02.2021

[FHT15] Fahrmeir, Ludwig ; Hamerle, Alfred ; Tutz, Gerhard: *Multivariate statistische Verfahren*. 2. Walter de Gruyter GmbH & Co KG, 2015

[Fra16] Franz, Thomas: *Reduced-order modeling for steady transonic flows via manifold learning*, DLR, Deutsches Zentrum für Luft-und Raumfahrt, Dissertation, 2016

[Fre20] Free Software Foundation: *The GNU C Library Reference Manual*. `https://www.gnu.org/software/libc/manual/html_mono/libc.html`. Version: 2020. – Free Software Foundation, Inc, last checked 20.02.2021

[FZGK14] Franz, Thomas ; Zimmermann, Ralf ; Görtz, Stefan ; Karcher, Niklas: Interpolation-based reduced-order modelling for steady transonic flows via manifold learning. In: *International Journal of Computational Fluid Dynamics* 28 (2014), Nr. 3-4, S. 106–121

[Gal13] Galántai, Aurél: *Projectors and projection methods*. Bd. 6. Springer Science & Business Media, 2013

[GIT14] Garcke, Jochen ; Iza-Teran, Rodrigo: Maschinelle Lernverfahren zur effizienten und interaktiven Auswertung großer Mengen von CAE-Modellvarianten. In: *17. Kongress Simulation und Erprobung in der Fahrzeugentwicklung 2014 : Berechnung, Prüfstands- und Straßenversuch, Baden-Baden, 18. und 19. November 2014 / VDI Fahrzeug- und Verkehrstechnik*. Düsseldorf : VDI-Verl, 2014. – ISBN 978–3–18–092224–9, S. 395–406

[GIT15] Garcke, Jochen ; Iza-Teran, Rodrigo: Machine Learning Approaches for Repositories of Numerical Simulation Results. In: *Procs. 10th European LS-DYNA Users' Conference, Würzburg, Germany*, 2015

[GIT16] Garcke, Jochen ; Iza-Teran, Rodrigo: Datenanalysemethoden zur Auswertung von Simulationsergebnissen im Crash und deren Abgleich mit dem Experiment. In: *18. Kongress SIMVEC - Simulation und Erprobung in der Fahrzeugentwicklung 2016 : Berechnung, Prüfstands- und Straßenversuch : Baden-Baden, 22. und 23. November 2016 / VDI Fahrzeug- und Verkehrstechnik*. Düsseldorf : VDI-Verl, 2016. – ISBN 978–3–18–092279–9

[Hah16] Hahner, Sara: *Untersuchung einer inversen Abbildung zu nichtlinearen Dimensionsreduktionen mit Anwendung auf Simulationsdaten*, Institut für Numerische Simulation, Universität Bonn, Bachelorthesis, 2016

[Hen98] Henn, Hans-Wolfgang: Crash tests and the head injury criterion. In: *Teaching mathematics and its applications* 17 (1998), Nr. 4, S. 162–170

[Hot33] Hotelling, Harold: Analysis of a complex of statistical variables into principal components. In: *Journal of educational psychology* 24 (1933), Nr. 6, S. 417

[HTP06] Hess, Stephane ; Train, Kenneth E. ; Polak, John W.: On the use of a Modified Latin Hypercube Sampling (MLHS) method in the estimation of a Mixed Logit model for vehicle choice. In: *Transportation Research Part B: Methodological* 40 (2006), Nr. 2, S. 147–163

[IT16] Iza-Teran, Rodrigo: *Geometrical Methods for the Analysis of Simulation Bundles*, Institut für Numerische Simulation, Universität Bonn, Dissertation, 2016. `http://hss.ulb.uni-bonn.de/2017/4648/4648.htm`

[ITMHG20] Iza-Teran, Rodrigo ; Morand, L ; Helm, D ; Garcke, Jochen: Learning Product Properties with Small DataSets in Forming Simulations. In: *International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes (NUMISHEET 2020)*, 2020

[JBT17] Jansen, Lennart ; Borsotto, Dominik ; Thole, Clemens-August: Combined Analysis of LS-DYNA Crash-Simulations and Crash-Test Scans. In: *11th European LS-DYNA conference*, 2017

[KDM10] Karbauskaitė, Rasa ; Dzemyda, Gintautas ; Marcinkevičius, Virginijus: Dependence of locally linear embedding on the regularization parameter. In: *Top* 18 (2010), Nr. 2, S. 354–376

[KGE08] Kübler, Lars ; Gargallo, Saimon ; Elsässer, Konrad: Characterization and evaluation of frontal crash pulses with respect to occupant safety. In: *9th International Symposium and Exhibition on Sophisticated Car Occupant Safety Systems*, 2008

[KGS20] Kracker, David ; Garcke, Jochen ; Schumacher, Axel: Automatic analysis of crash simulations with dimensionality reduction algorithms such as PCA and t-SNE. In: *16th International LS-DYNA Users Conference*, 2020

[KW78] Kruskal, Joseph B. ; Wish, Myron: *Multidimensional Scaling*. Bd. 11. Sage Publications, 1978

[LC20a] LST-Corp.: *LS-DYNA*. `https://www.lstc.com/products/ls-dyna`. Version: 2011-2020. – Livermore Software Technology, an Ansys company, last checked 04.01.2021

[LC20b] LST-Corp.: *LS-OPT*. `https://www.lstc.com/products/ls-opt`. Version: 2011-2020. – Livermore Software Technology, an Ansys company, last checked 05.02.2021

[LLW+16] Lin, Tong ; Liu, Yao ; Wang, Bo ; Wang, Li-Wei ; Zha, Hong-Bin: Nonlinear Dimensionality Reduction by Local Orthogonality Preserving Alignment. In: *Journal of Computer Science and Technology* 31 (2016), Nr. 3, S. 512–524

[LM12] Liesen, Jörg ; Mehrmann, Volker: *Lineare Algebra*. Vieweg + Teubner Verlag, Wiesbaden, 2012. – ISBN 978–3–8348–8290–5

[LV07] Lee, John A. ; Verleysen, Michel: *Nonlinear Dimensionality Reduction*. Springer Science & Business Media, 2007. – ISBN 978–0–387–39350–6

[LV09] Lee, John A. ; Verleysen, Michel: Quality assessment of dimensionality reduction: Rank-based criteria. In: *Neurocomputing* 72 (2009), Nr. 7-9, S. 1431–1443

[LV10] Lee, John A. ; Verleysen, Michel: Scale-independent quality criteria for dimensionality reduction. In: *Pattern Recognition Letters* 31 (2010), Nr. 14, S. 2248–2257

[Mar99] Marczyk, Jacek: *Principles of simulation-based computer-aided engineering*. FIM Publications, 1999

[MCR+18] Macri, Domenico ; Claesson, Emil ; Rydberg, Simon ; Aspenberg, David ; Borsotto, Dominik: Automotive Crash Simulation Robustness using Principal Component Analysis (PCA) Approach. In: *NAFEMS Nordic 2018, Göteborg, Sweden, 24-25 April, 2018*, 2018

[MSCK12] Marzougui, Dhafer ; Samaha, Randa R. ; Cui, Chongzhen ; Kan, CD: Extended validation of the finite element model for the 2007 Chevrolet Silverado pick-up truck. In: *Ashburn: US National Crash Analysis Center* (2012)

[MSJ20] Mertler, Stefan ; Schumacher, Axel ; Jansen, Lennart: Reduced Order Modeling for Correlation Analysis of Crash Structures. In: *Automotive CAE Grand Challenge*, 2020

[MT05] Meil, Liquan ; Thole, Clemens-August: Clustering algorithms for parallel car-crash simulation analysis. In: *Modeling, simulation and optimization of complex processes*. Springer, 2005, S. 331–340

[MT08] Mei, Liquan ; Thole, Clemens-August: Data analysis for parallel car-crash simulation results and model optimization. In: *Simulation modelling practice and theory* 16 (2008), Nr. 3, S. 329–337

[Nat12] National Highway Traffic Safety Administration: Laboratory Test Procedure for New Car Assessment Program Frontal Impact Testing. In: *US Department of Transportation* (2012)

[Nat19] National Highway Traffic Safety Administration: *LS-DYNA FE Crash Simulation Vehicle Models*. `https://www.nhtsa.gov/crash-simulation-vehicle-models`. Version: 2019. – last checked 12.03.2021

[Oka15] Okamura, Masahiro: Robustness Analysis of a Vehicle Front Structure Using Statistical Approach. In: *10th European LS-DYNA conference*, 2015

[Oka17] Okamura, Masahiro: Improvement of Response Surface Quality for Full Car Frontal Crash Simulations by Suppressing Bifurcation using Statistical Approach. In: *11th European LS-DYNA conference*, 2017

[OOB19] Okamura, Masahiro ; Oda, Hodaka ; Borsotto, Dominik: Comparison of Laser-Scanned Test Results and Stochastic Simulation Results in Scatter Mode Space. In: *12th European LS-DYNA conference*, 2019

[Ort15] Ortmann, Christopher: *Entwicklung eines graphen- und heuristikbasierten Verfahrens zur Topologieoptimierung von Profilquerschnitten für Crashlastfälle*. Düren, Fachbereich D – Architektur, Bauingenieurwesen, Maschinenbau, Sicherheitstechnik, Bergische Universität Wuppertal, Dissertation, 2015

[OS13] Ortmann, Christopher ; Schumacher, Axel: Graph and heuristic based topology optimization of crash loaded structures. In: *Structural and Multidisciplinary Optimization* 47 (2013), Nr. 6, S. 839–854

[OS14] Ortmann, Christopher ; Schumacher, Axel: Branching strategies for the application of heuristics to the topology optimization of crash loaded structures. In: *Proceedings of the 11th World Congress on Computational Mechanics (WCCM XI), Barcelona*, 2014, S. 20–25

[Pea01] Pearson, Karl: LIII. On lines and planes of closest fit to systems of points in space. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901), Nr. 11, S. 559–572

[PHHV08] Porte, J De l. ; Herbst, BM ; Hereman, W ; VanDerWalt, SJ: An introduction to diffusion maps. In: *Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008), Cape Town, South Africa*, 2008, S. 15–25

[PRMB09] Pradeep, M ; Ritter, M ; Marzougui, D ; Brown, D: Modeling, testing, and validation of the 2007 Chevy Silverado finite element model. In: *Transportation Research Board 89th Annual Meeting, CDROM, TRB, Washington, DC, USA*, 2009, S. 1–18

[RBG+16] Richter, Justus ; Büchse, Matthias ; Graf, Wolfgang ; Thiele, Marko ; Löbner, Clemens ; Liebscher, Martin: Compression Methods for Simulation Models in SDM Systems. In: *Procs. 14th German LS-DYNA Forum, Bamberg, Germany*, 2016, S. 198–215

[RS00] Roweis, Sam T. ; Saul, Lawrence K.: Nonlinear dimensionality reduction by locally linear embedding. In: *Science* 290 (2000), Nr. 5500, S. 2323–2326

[Sam69] Sammon, John W.: A nonlinear mapping for data structure analysis. In: *IEEE Transactions on computers* 100 (1969), Nr. 5, S. 401–409

[Sau20] Saul, Lawrence K.: A tractable latent variable model for nonlinear dimensionality reduction. In: *Proceedings of the National Academy of Sciences* 117 (2020), Nr. 27, S. 15403–15408

[Sch20] Schumacher, Axel: Exemplarische Anwendungen und weitergehende Forschungen. In: *Optimierung mechanischer Strukturen*. Springer, 2020, S. 281–318

[SR00] Saul, Lawrence K. ; Roweis, Sam T.: *An Introduction to Locally Linear Embedding*. `https://cs.nyu.edu/~roweis/lle/papers/lleintro.pdf`. Version: 2000. – Department of Computer Science at Columbia University, last checked 11.05.2020

[SR03] Saul, Lawrence K. ; Roweis, Sam T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. In: *Journal of machine learning research* 4 (2003), Nr. Jun, S. 119–155

[SS06] Seydel, Rüdiger ; Seydel, Rudiger: *Tools for computational finance*. Bd. 3. Springer, 2006

[ST02] Silva, Vin ; Tenenbaum, Joshua: Global versus local methods in nonlinear dimensionality reduction. In: *Advances in neural information processing systems* 15 (2002), S. 721–728

[TDSL00] Tenenbaum, Joshua B. ; De Silva, Vin ; Langford, John C.: A global geometric framework for nonlinear dimensionality reduction. In: *science* 290 (2000), Nr. 5500, S. 2319–2323

[Ten98] Tenenbaum, Joshua B.: Mapping a manifold of perceptual observations. In: *Advances in neural information processing systems*, 1998, S. 682–688

[TM03] Thole, Clemens-August ; Mei, Liquan: Reasons for scatter in crash simulation results. In: *Procs. 4th European LS-DYNA Users' Conference, Ulm, Germany*, 2003, S. B–III–11–B–III–20

[TM10] Thole, Clemens-August ; Mierendorff, Hermann: *Method for determining joint causes of scatter of simulation results and/or measurement results*. 2010. – Patent: EP2510478A2

[TNNC10] Thole, Clemens-August ; Nikitin, Igor ; Nikitina, Lialia ; Clees, Tanja: Advanced mode analysis for crash simulation results. In: *Procs. 9th International LS-DYNA Users' Conference, Bamberg, Germany*, 2010, S. II11–II20

[Wal98] Walk, Kerry: *Writing Resources: Strategies for Essay Writing - How to Write a Comparative Analysis*. `https://writingcenter.fas.harvard.edu/pages/how-write-comparative-analysis`. Version: 1998. – Writing Center at Harvard University, last checked 09.05.2018

[WB12] Wolff, Sebastian ; Bucher, Christian: Recent Developments for Random Fields and Statistics on Structures. In: *Weimar Optimization and Stochastic Days* 9 (2012)

[WRR03] Wall, Michael E. ; Rechtsteiner, Andreas ; Rocha, Luis M.: Singular value decomposition and principal component analysis. In: *A practical approach to microarray data analysis*. Springer, 2003, S. 91–109

[Yan04] Yang, Li: Sammon's nonlinear mapping using geodesic distances. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Bd. 2 IEEE, 2004, S. 303–306

[ZL14] Zhang, Teng ; Lerman, Gilad: A novel m-estimator for robust pca. In: *The Journal of Machine Learning Research* 15 (2014), Nr. 1, S. 749–808

[ZQZ11] Zhang, Peng ; Qiao, Hong ; Zhang, Bo: An improved local tangent space alignment method for manifold learning. In: *Pattern Recognition Letters* 32 (2011), Nr. 2, S. 181–189

[ZW07] Zhang, Zhenyue ; Wang, Jing: MLLE: Modified locally linear embedding using multiple weights. In: *Advances in neural information processing systems*, 2007, S. 1593–1600

[ZZ04] Zhang, Zhen-yue ; Zha, Hong-yuan: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. In: *Journal of Shanghai University (English Edition)* 8 (2004), Nr. 4, S. 406–424

Curriculum Vitae

Personal Details

Stefan Matthias Mertler

Birthplace	Radevormwald

Education

04/2011 - 02/2014	Master of Sciences (M. Sc.) in Economathematics at the *University of Cologne*
10/2007 - 04/2011	Bachelor of Sciences (B. Sc.) in Economathematics at the *University of Cologne*

Work Experience

since 04/2013	Software developer at *Simulation Data Analysis and Compression Technologies (SIDACT) GmbH* in Bonn
02/2012 - 03/2013	Student assistant at *Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)* in Sankt Augustin