High-Order Retractions for Reduced-Order Modeling and Uncertainty Quantification

by

Aaron Charous

Sc.B., Brown University (2019)

Submitted to the Center for Computational Science & Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computational Science & Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2021

Author .....
Center for Computational Science & Engineering
January 15, 2021

Certified by .....
Pierre F.J. Lermusiaux
Professor, Department of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Youssef M. Marzouk
Co-Director, Center for Computational Science & Engineering

High-Order Retractions for Reduced-Order Modeling and Uncertainty Quantification

by

Aaron Charous

Submitted to the Center for Computational Science & Engineering
on January 15, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computational Science & Engineering

Abstract

Though computing power continues to grow quickly, our appetite to solve larger and larger problems grows just as fast. As a consequence, reduced-order modeling has become an essential technique in the computational scientist's toolbox. By reducing the dimensionality of a system, we are able to obtain approximate solutions to otherwise intractable problems. And because the methodology we develop is sufficiently general, we may agnostically apply it to a plethora of problems, whether the high dimensionality arises due to the sheer size of the computational domain, the fine resolution we require, or stochasticity of the dynamics. In this thesis, we develop time integration schemes, called retractions, to efficiently evolve the dynamics of a system's low-rank approximation. Through the study of differential geometry, we are able to analyze the error incurred at each time step. A novel, explicit, computationally inexpensive set of algorithms, which we call perturbative retractions, are proposed that converge to an ideal retraction that projects exactly to the manifold of fixed-rank matrices. Furthermore, each perturbative retraction itself exhibits high-order convergence to the best low-rank approximation of the full-rank solution. We show that these high-order retractions significantly reduce the numerical error incurred over time when compared to a naive Euler forward retraction. Through test cases, we demonstrate their efficacy in the cases of matrix addition, real-time data compression, and deterministic and stochastic differential equations.

Thesis Supervisor: Pierre F.J. Lermusiaux
Title: Professor, Department of Mechanical Engineering

Acknowledgments

Research is a disorderly process: there are so many stumbling blocks to overcome, an unfortunate number of dead-ends in which to fruitlessly invest time, and an overwhelming number of ideas to explore. Without guidance, one can easily get lost in the labyrinth that is the pursuit of new knowledge. So I will forever be grateful for those who supported me while completing my SM at MIT. Either through inadvertent yet fortuitous research discussions or by direct advice, I've been given invaluable direction that has steered me to this point.

To Professor Pierre Lermusiaux, I am most thankful that you allowed me to pursue my interests even when at first it was unclear where that would take me. There is an inherent risk when graduate students are given freedom to explore what they please. Perhaps they will endlessly wander between different fields, never committing to a specific area of study. But it is a priceless gift to study what intrigues you, what you think is important, and what you find meaningful. I appreciate your limitless enthusiasm and generosity with your time. And finally, I am grateful for the abstract ideas you patiently explained to me over and over again. You mentioned using information from the geometry of the low-rank manifold to increase the order of accuracy of numerical schemes many times before it clicked in my mind, and I could not have written this thesis without our many discussions.

To the MSEAS students, Abhinav, Akis, Aman, Chinmay, Clara, Corbin, Manan, Manmeet, Mike, Jacob, Jing, Tony, and Wael, you are the reason MIT is what it is to me. Everyone was exceedingly warm and welcoming upon my arrival, and instantaneously becoming one amongst a loyal group of friends is of immeasurable value. I cannot overstate how much I appreciate all of your kindness. I also thank you for both the illuminating research discussions and the inconsequential conversations about nothing. I look forward to many more to come.

I'd also like to thank Dr. Pat Haley, Dr. Chris Mirabito, Kate Nelson, and Lisa Mayer. In addition, I'm grateful for the friends I've made in my CSE cohort. You have all helped me navigate through these three semesters in so many different capacities.

I am grateful to the Office of Naval Research (ONR) for research support under grants N00014-19-1-2693 (IN-BDA) and N00014-19-1-2664 (Task Force Ocean, DEEP-AI) and to the MIT Lincoln Laboratory for research support under the Wide Area Ocean Floor Mapping project, each to the Massachusetts Institute of Technology.

Finally, I want to thank my family. You have grounded me and given me a sense

of belonging. Mom, Dad, Jason, and Brian, thank you for supporting my decision to pursue my passions in academia, for taking an interest in my work and life via countless hours of texting and calling, and for your unconditional, unrelenting love and support.

Contents

List of Figures

List of Tables

Introduction

The complexity of the problems we choose to tackle continuously grows as time passes. We researchers like to complain that all the "easy" problems have been solved a few hundred years ago by the likes of Joseph Fourier, Pierre-Simon Laplace, and Jean le Rond d'Alembert. Among their other invaluable contributions, they solved the constant-coefficient differential equations on simple domains we study as undergraduates. But life is not as simple as the nice closed-form expressions they derived [1]. Domains are rough with intricate geometry. Coefficients are spatially-dependent and random.

Biological phenomena, quantum mechanics, fluid flow, electromagnetic and acoustic wave propagation, climate modeling, and ocean dynamics are among the countless areas in which computational models are developed to predict future outcomes and understand underlying truths about the world. The advent of the computer allowed for a leap forward in terms of the problems we could solve. But the algorithms which we use to approximate solutions limit the scope of the problems which we can study. There is typically a trade-off between solution accuracy and computational cost/runtime. Of course, if we wait long enough – either for a computer to chug along or, on a longer time scale, for computer engineers to build us faster machines – we may be able to get an answer to a large problem. But humans are impatient, and for some cases, the time for a solution could be on the order of years (though it's easy to think of a problem that will take arbitrarily long to solve).

The next leap forward is then to come from better computational algorithms and modeling techniques. One approach is to use reduced-order models, such as the polynomial chaos expansion [2, 3, 4, 5, 6], where a very high-dimensional model is projected onto a low-dimensional space. Another typical approach is to use the Karhunen-Loève expansion or proper orthogonal decomposition [7, 8, 9, 10, 11], which will be our starting point. We start with a square-integrable stochastic process $\Phi(x;\omega)$ defined over a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ and with $x \in \mathcal{D}$ denoting our spatial variable. Ω is the sample space, the set of all possible outcomes, and ω represents

a simple event. $\mathcal{F}$ is the σ-algebra, a collection of subsets of Ω satisfying certain regularity conditions. $\mathbf{P}$ is the probability measure, a function on the σ-algebra that maps to the interval from zero to one. Then, Φ may be represented as follows.

$$\Phi(x;\omega) = \mathbb{E}[\Phi(x;\omega)] + \sum_{i=1}^{\infty} \varphi_i(x)\zeta_i(\omega)$$

Above, $\mathbb{E}$ denotes the expectation operator. This decomposition is the stochastic analogue to separation of variables technique to solve partial differential equations [12] and for function approximation [13]. That is, instead of expressing a deterministic function as $f(x_1, x_2) = \sum_{i=1}^{\infty} g_i(x_1)h_i(x_2)$, we allow x_2 to denote a simple event in the event space (denoted $\omega \in \Omega$). The modes $\phi_i(x)$ are deterministic functions given by the solution to the following eigenvalue problem.

$$\int_{\mathcal{D}} \mathbb{E}[\Phi(s;\omega)\Phi(x;\omega)]\varphi_i(s)ds = \lambda_i\varphi_i(x)$$

Because the kernel above is a Mercer kernel, λ_i will be non-negative, and φ_i will be orthonormal. The stochastic coefficients $\zeta_i(\omega)$ are defined as the projection of Φ on the modes φ_i.

$$\zeta_i(\omega) = \int_{\mathcal{D}} \Phi(x;\omega)\varphi_i(x)dx$$

One can show that ζ_i are zero-mean and mutually uncorrelated. That is, $\mathbb{E}[\zeta_i\zeta_j] = \delta_{ij}\lambda_j$, where δ_{ij} denotes the Kronecker delta function. The truncated Karhunen-Loève expansion yields the best approximation to Y in that it minimizes the total mean square error. As such, by truncating the number of modes and stochastic coefficients in the expansion, we can obtain a very accurate reduced-order model of the stochastic process. Note that this is the continuous analogue of the singular value decomposition, which will be discussed later in this thesis.

Sapsis and Lermusiaux extended this framework to the spatially-varying time-dependent case via the dynamically orthogonal equations [14]. Essentially, the mean, modes, and the stochastic coefficients in the Karhunen-Loève expansion are allowed to vary in time.

$$\Phi(x,t;\omega) = \mathbb{E}[\Phi(x,t;\omega)] + \sum_{i=1}^{\infty} \varphi_i(x,t)\zeta_i(t;\omega)$$

When given a stochastic partial differential equation of the form $\frac{\partial \Phi}{\partial t} = \mathscr{L}(\Phi, x, t; \omega)$, we wish to write out differential equations to describe how the modes and coefficients evolve in time so that a reduced-order model may be evolved without explicitly reconstructing Y. Since the coefficients and modes are both allowed to evolve in time, there are ambiguous degrees of freedom in how the mode and coefficient differential equations may be written. To get rid of the ambiguity, a gauge condition known as the dynamically orthogonal condition insists that $\langle y_i, \frac{\partial \varphi_j}{\partial t} \rangle = 0 \ \forall i, j$, where $\langle \cdot, \cdot \rangle$ denotes the inner product in space. With this, the following dynamically orthogonal equations may be written.

$$\frac{\partial \mathbb{E}[\Phi(x,t;\omega)]}{\partial t} = \mathbb{E}[\mathscr{L}(\Phi; x, t; \omega)]$$

$$\frac{d\zeta_i}{dt} = \langle \mathscr{L}(\Phi, \bullet, t; \omega) - \mathbb{E}[\mathscr{L}(\Phi, \bullet, t; \omega)], \varphi_i(\bullet, t) \rangle$$

$$\frac{\partial}{\partial t} \begin{pmatrix} \varphi_1(x,t) \\ \varphi_2(x,t) \\ \vdots \end{pmatrix}^T = \mathscr{P}_{\varphi}^{\perp} \left[\mathbb{E} \left[\mathscr{L}(\Phi, x, t; \omega) \begin{pmatrix} \zeta_1(t;\omega) \\ \zeta_2(t;\omega) \\ \vdots \end{pmatrix}^T \right] \right] \begin{pmatrix} \mathbb{E}[\zeta_1\zeta_1] & \mathbb{E}[\zeta_1\zeta_2] & \cdots \\ \mathbb{E}[\zeta_2\zeta_1] & \mathbb{E}[\zeta_2\zeta_2] & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}^{-1}$$

Above, $\mathscr{P}_{\varphi}^{\perp}[A]$ denotes the orthogonal projection of A onto the modes φ_i, which is formally defined below.

$$\mathscr{P}_{\varphi}^{\perp}[A] = A - \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \end{pmatrix}^T \begin{pmatrix} \langle A, \varphi_1 \rangle \\ \langle A, \varphi_2 \rangle \\ \vdots \end{pmatrix}$$

These equations produce a coupled system of differential equations which may be solved numerically. We note that the terms still include Φ; the Dirac-Frankel variational principle [15, 16] replaces Φ with its low-rank approximation in calculating the mode and coefficient time derivatives and will be discussed further in this thesis.

Previously, these equations have been discretized directly (see, e.g., [17, 18]); however, analogous equations may be derived in the finite-dimensional case where $\mathcal{D}$ and Ω are discrete. In contrast to the spatially continuous dynamically orthogonal equation, the spatially discrete dynamically orthogonal equations allow for a non-intrusive implementation which flattens the learning curve for reduced-order modeling and eliminates the mental burden of deriving new reduced-order equations for each test case. This was first analyzed in [19], where the *dynamical low-rank approximation* was proposed to solve time-dependent matrix initial value problems. The connec-

tion between the dynamical low-rank approximation and the dynamically orthogonal equations was made in [20]; the dynamically orthogonal equations can be thought of as instantaneously projecting the full-rank dynamics onto a low-rank manifold. The following matrix differential equations were proposed.

$$\dot{U} = \mathscr{P}_U^{\perp} \mathscr{L} Z (Z^T Z)^{-1}$$
$$\dot{Z} = \mathscr{L}^T U$$

Here, $\mathscr{P}_U^{\perp} = I - UU^T$, and the columns of U, u_i, correspond to φ_i. Similarly, the columns of Z, z_i, correspond to realizations of ζ_i. Then, UZ^T is our low-rank approximation to the discrete analogue of Φ. One small difference in these equations is we have not explicitly included the mean of Φ; this may be included as an additional column in U and a column of ones in Z.

It is here we note that though stochastic differential equations have been our motivation, our low-rank approximation applies equally as well to a deterministic case in, for instance, matrix differential equations or two-dimensional time-dependent partial differential equations. Instead of thinking of Ω as a stochastic event space, we may replace it with another physical space $\tilde{\mathcal{D}}$. The expectation operator can be thought of as an inner product weighted by a probability measure; so in the deterministic case, the expectation operator just becomes an inner product over $\tilde{\mathcal{D}}$. We've already mentioned that the proper orthogonal decomposition is the stochastic analogue of separation of variables, so this is not a large intellectual leap. As we will see in the thesis, the interpretation of $\mathcal{D}$ and Ω are unimportant to our analysis and may be abstracted away. Now, we have a system of nonlinear matrix differential equations, and a key question that remains is how to integrate the equations. Should we hold the right-hand sides constant and integrate $\dot{U}$ and $\dot{Z}$ simultaneously? Should we fix Z and integrate $\dot{U}$ followed by fixing U and integrating $\dot{Z}$ in an alternating pattern? For the most efficient and accurate schemes, perhaps we should do something different altogether.

In this thesis, we discuss time-integration schemes for the discrete dynamically orthogonal equations. Restricting the rank of the solution introduces a new type of numerical error when integrating, referred to as *retraction error*, and concepts from differential geometry are especially helpful in our analysis. We will assume no prior knowledge of differential geometry, so chapter one will introduce key concepts and definitions as they relate to the dynamically orthogonal equations. Chapter two focuses on retractions, the essential elements in time-integration schemes. Several

novel algorithms are introduced to efficiently project onto the low-rank manifold. Furthermore, a new set of retractions, which we refer to as *perturbative retractions*, is derived that converges to the projection operator, and each perturbative retraction exhibits high-order convergence to the best low-rank approximation of the full-rank solution. Chapter three goes over several applications of these retractions including matrix addition, real-time data compression, and several differential equations. A comparative analysis of the retractions is given in each case. Lastly, the strengths and weaknesses of the retractions are given in the conclusion, and future research directions are discussed.

Chapter 1

The low-rank manifold and its tangent space

1.1 Introductory definitions from differential geometry

We'll start with some basic definitions and then introduce more terminology as needed. First, we loosely define what differential geometry is. Many are familiar with Euclidean geometry, which can be thought of the study of the structure of "flat," i.e. non-curved, space. It's what is taught in middle school and high school. Riemannian geometry generalizes this study to curved spaces so long as two features hold, which will be more formally defined below. Riemannian geometry is one branch of the much broader discipline of differential geometry, which covers the use of calculus and linear algebra on geometric structures.

Now, the central topological structure of interest is the manifold. It can be thought of as the generalization of curves and surfaces and must locally resemble a Euclidean, or "flat," space [21]. For our purposes, this is not at all restrictive since by the Nash embedding theorem, every Riemannian manifold can be isometrically embedded into a Euclidiean space, where isometric means that the distances of all paths in the manifold are preserved given some metric [22]. In one dimension, examples of manifolds are curves such as lines and circles. In two dimensions, examples include surfaces such as planes, ellipsoids, spheres, etc. We will be particularly interested in *embedded manifolds*, or *submanifolds*, which are manifolds that exist in some higher-dimensional space. Henceforth, in this thesis when we say "manifold," we are referring to a an embedded manifold unless otherwise specified. To define a *smooth manifold*, we first

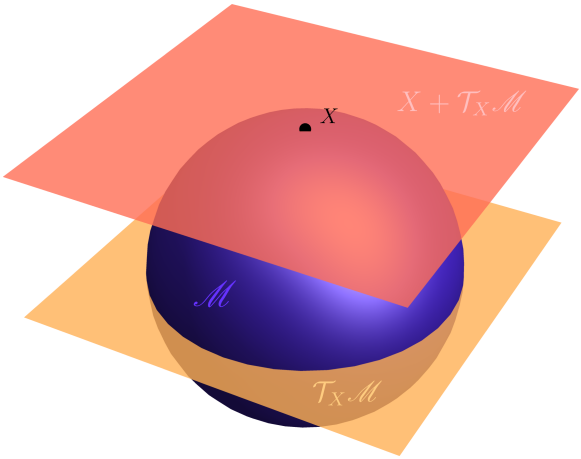


Figure 1-1: Tangent & affine tangent spaces of a sphere in $\mathbb{R}^3$

need to define a *graph*, and for both definitions we refer to [21].

Definition 1.1.1. A graph of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ takes a set of pairs $(x, y) \in (\mathbb{R}^n \times \mathbb{R}^m)$ such that $f(x) = y$.

Though seemingly abstract, we are quite familiar with graphs in two dimensions, where we plot univariate functions. For example, the function $f(x) = x^3$ has the graph with the set of pairs (x, x^3), and may be plotted with the first coordinate, x, on the x-axis and the second coordinate, x^3, on the y-axis.

Definition 1.1.2. $\mathscr{M} \subset \mathbb{R}^n$ is a smooth, k-dimensional manifold if it is locally the graph of a C^1 mapping expressing $n-k$ variables as functions of the other k variables.

Above, C^1 refers to the class of functions that are at least once continuously differentiable. This implies that the graph of a k-dimensional smooth manifold is a set of pairs $(x, y) \in (\mathbb{R}^{n-k} \times \mathbb{R}^k)$.

From [23], we adapt the following definition of a tangent space.

Definition 1.1.3. The tangent space $\mathcal{T}_X\mathscr{M}$ at point $X \in \mathscr{M}$ is the set of all vectors tangent to $\mathscr{M}$ at X.

Here, the notion of a vector may be generalized, and later in this thesis we'll consider a matrix to be a generalized vector. The tangent space is a vector space with the same dimension as the manifold; hence, the origin is always included. We often want to define the affine tangent space $X + \mathcal{T}_X\mathscr{M}_r$ at X, which, instead of including the origin, includes the point X.

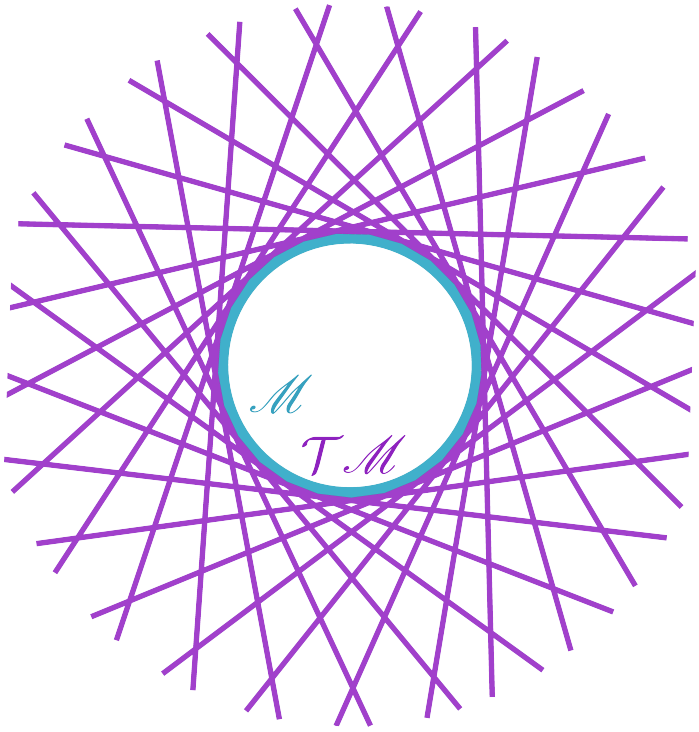


Figure 1-2: Tangent bundle of a circle in $\mathbb{R}^2$ (adapted from [24])

Definition 1.1.4. The tangent bundle $\mathcal{T}\mathscr{M}$ is the disjoint union of all tangent spaces of $\mathscr{M}$.

$$\mathcal{T}\mathscr{M} = \bigsqcup_{X \in \mathscr{M}} \mathcal{T}_X \mathscr{M} \tag{1.1}$$

Next, adapted from [25], we have the formal definition of a Riemannian metric.

Definition 1.1.5. Given a smooth manifold $\mathscr{M}$, a Riemannian metric g on $\mathscr{M}$ is a mapping that associates with each $X \in \mathscr{M}$ an inner product $g_X : \mathcal{T}_X \mathscr{M}_r \times \mathcal{T}_X \mathscr{M}_r \to \mathbb{R}$ which is differentiable on an open set of $\mathscr{M}$.

Finally, we can define a Riemannian manifold [25].

Definition 1.1.6. A Riemannian manifold is a real, smooth manifold with a given Riemannian metric.

We note here that a *Hermitian manifold* is the complex analogue to a Riemannian manifold, which, instead of assigning a differentiable, real, positive-definite inner product to the tangent space of each point on the manifold, assigns a differentiable, Hermitian inner product.

1.2 Parameterizing the low-rank manifold

Though numerical schemes may seem distant from the field of differential geometry, there has been a surge of interest in connecting the two in the recent past. The optimization community has used Riemannian geometry to develop new methods applied to manifolds of matrices. This allows for a more efficient search in a constrained setting where we only consider a particular class of matrices [26, 27, 28, 29]. More directly applicable to stochastic PDEs is the use of Riemannian geometry for initial value problems. In particular, we are interested in methods that restrict the solution to a *low-rank manifold*.

To be precise, we first state the definition of the rank of a matrix $X \in \mathbb{R}^{m\times n}$, and we'll follow [30] for this definition and the upcoming discussion of the singular value decomposition.

Definition 1.2.1. The rank r of matrix X, denoted $\text{rank}(X)$, is the smallest number such that there exists a decomposition

$$X = UZ^T, \qquad U \in \mathbb{R}^{m\times r}, Z \in \mathbb{R}^{n\times r}.$$

Alternatively, the rank of matrix X is the maximal number of linearly independent columns of X or the dimension of the column space of X. Note that the decomposition above is not unique: one only needs to consider an orthogonal transformation P applied as follows.

$$\begin{aligned} \tilde{U} = UP, \quad \tilde{Z} = ZP \\ \Rightarrow X = \tilde{U}\tilde{Z}^T \end{aligned}$$

These transformations give equivalent results [31], and the decomposition always exists, yielding the following lemma [32, 20]. Henceforth, let $\mathbb{R}^{n\times r}_*$ denote $\{A \in \mathbb{R}^{n\times r} : \text{rank}(A) = r\}$. Furthermore, let $\mathcal{V}_{m,r}$ denote the Stiefel manifold, defined below.

$$\mathcal{V}_{m,r} = \left\{Y \in \mathbb{R}^{m\times r} : Y^TY = I\right\}.$$

Lemma 1.2.1. Any matrix $X \in \mathscr{M}_r$ can be decomposed as $X = UZ^T$, where $U \in \mathcal{V}_{m,r}$ and $Z \in \mathbb{R}^{n\times r}_*$. This decomposition is unique up to an orthonormal rotation matrix Q, implying that $Q^TQ = QQ^T = I$. Moreover, if $U_1, U_2 \in \mathcal{V}_{m,r}$ and $Z_1, Z_2 \in$

$\mathbb{R}_*^{n\times r}$ (implying $\text{rank}(Z_1) = \text{rank}(Z_2) = r$), then

$$U_1 Z_1^T = U_2 Z_2^T \quad \Longleftrightarrow \quad \exists Q : U_1 = U_2 Q, \quad Z_1 = Z_2 Q. \tag{1.2}$$

One way of obtaining such a low-rank approximation to a matrix X is via the singular value decomposition (SVD). Suppose $\text{rank}(X) \leq k$ for some $k \in \mathbb{R}$. Then, X may be decomposed as follows.

$$X = U\Sigma V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Above, u_i are orthonormal vectors that form the columns of $U \in \mathbb{R}^{m\times k}$, v_i are orthonormal vectors that form the columns of $V \in \mathbb{R}^{n\times k}$, and $\Sigma \in \mathbb{R}^{k\times k}$ is a diagonal matrix with entries $\sigma_i \geq 0$. The number of nonzero singular values corresponds to the rank of matrix X. If we order the singular values, σ_i, from greatest to least, we can write an approximation X_r to X as follows for $r < k$.

Theorem 1.2.1. Let $\sigma_1 \geq ... \geq \sigma_k \geq 0$ be the singular values of a matrix X. For any $r < k$, the truncated SVD

$$X_r = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

provides the best approximation to X in the Frobenius norm. Furthermore,

$$||X - X_r||_F^2 = \sum_{i=r+1}^{k} \sigma_i^2,$$

and if $\sigma_r > \sigma_{r+1}$, then X_r is the unique best approximation of rank at most r.

This is known as the Eckart-Young-Mirsky theorem [33, 34, 35]. In short, it states that the best approximation with a lower rank to a matrix X is given by truncating the singular value decomposition of X. This is used in the statistics community for dimensionality reduction in principal component analysis. Taking $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_r \end{bmatrix}$ and $Z = \begin{bmatrix} \sigma_1 v_1 & \sigma_2 v_2 & \cdots & \sigma_r v_r \end{bmatrix}$ will then yield a low-rank approximation of the form $X = UZ^T$, and $U^T U = I$.

It is clear that for $r \ll m$ or $r \ll n$, it is beneficial to store X as the pair (U, Z) with the bilinear map $(U, Z) \to UZ^T$ since the pair requires the storage of $(m + n)r$ numbers rather than mn numbers. It is this fact that motivates the use of low-

rank approximations; when m and/or n are extremely large, the best feasible way to store X is through some low-rank decomposition. And if we can only store low-rank approximations to X, then we would like to restrict solutions of our problem to the low-rank manifold. In addition, by restricting solutions to the low-rank manifold, each computational step or basic linear algebra operation becomes much cheaper to compute.

Definition 1.2.2. The manifold of $m \times n$ real matrices of rank r is denoted as follows.

$$\mathscr{M}_r = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = r\}$$

From [30], we have that $\dim(\mathscr{M}_r) = (m + n - r)r$, and that $\mathscr{M}_r$ is a C^∞ smooth embedded submanifold of $\mathbb{R}^{m \times n}$.

1.3 Parameterizing the tangent space

In this section, we show how to paramterize any matrix of the tangent space of some point $X \in \mathscr{M}_r$. Koch and Lubich showed how to do so for decompositions of the form USV^T in [19], and Feppon and Lermusiaux derived similar equations for decompositions of the form UZ^T [20]. Here, we will follow the works above to obtain equations for the tangent space of a low-rank manifold, and in the next section we will define how to project onto the tangent space as well as how to define the space normal to the tangent space. In the process of doing so, we will introduce generalizations of the derivative, which will also be useful in the sections that follow.

To build some intuition, we'll first consider some toy problems. Take a curve S_1 defined by $f_1 : \mathbb{R} \to \mathbb{R}$. Say we are given a point x_1, and we would like to find the set of vectors that are tangent to S_1 at $(x_1, f_1(x_1))$. That is, we seek the affine tangent space of S_1 at x_1. From calculus, we know that the vector must have slope $\frac{df_1(x_1)}{dx}$. Hence, we can write the tangent space as follows.

$$\mathcal{T}(x_1, f_1(x_1)) = \left\{ \begin{bmatrix} \delta x & \frac{df_1(x_1)}{dx}\delta x \end{bmatrix}^T : \delta x \in \mathbb{R} \right\}$$

Hence, $\frac{df(x_1)}{dx}$ defines the tangent space. Generalizing this idea, imagine a surface S_2 defined by $f_2 : \mathbb{R}^2 \to \mathbb{R}$ embedded in $\mathbb{R}^3$. Now suppose we are given a point (x_2, y_2), and we would like to find the expression for the tangent plane $\mathcal{T}(x_2, y_2, f_2(x_2, y_2))$. In

a similar fashion, we can write the following.

$$\begin{aligned}\mathcal{T}(x_2, y_2, f_2(x_2, y_2)) &= \left\{ \begin{bmatrix} \delta x & \delta y & \frac{\partial f_2(x_2,y_2)}{\partial x}\delta x + \frac{\partial f_2(x_2,y_2)}{\partial y}\delta y \end{bmatrix}^T : \delta x, \delta y \in \mathbb{R} \right\} \\ &= \left\{ \begin{bmatrix} \delta_1 & \delta_2 & \delta^T \nabla f_2(x_2, y_2) \end{bmatrix}^T : \delta \in \mathbb{R}^2 \right\}\end{aligned}$$

Above, δ_1 and δ_2 refer to the first and second elements of δ, respectively. Moreover, δ defines the variation in the independent variables, and $\delta^T \nabla f$ defines the variation in the dependent variables. Hence, we see that the gradient of f completely defines the tangent plane; often times the variation in the independent variables is omitted, and the tangent space is written simply as follows.

$$\mathcal{T} f_2(x_2, y_2) = \left\{ \delta^T \nabla f_2(x_2, y_2) : \delta \in \mathbb{R}^2 \right\}$$

To generalize this further, we need to introduce generalizations to the derivative. We already know that the gradient generalizes the univariate derivative to multivariable functions. One step further is to generalize the gradient to vector-valued functions. For a function $F : \mathbb{R}^n \to \mathbb{R}^m$ with $F = \begin{bmatrix} f_1 & f_2 & \cdots & f_m \end{bmatrix}^T$, the Jacobian J is defined as

$$J = \begin{bmatrix} \frac{\partial F}{\partial x_1} & \frac{\partial F}{\partial x_2} & \cdots & \frac{\partial F}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

As expected, the Jacobian will define a tangent space to vector-valued multivariable functions. We need to go one step further, though, since we want to deal with matrix functions. The generalization to the Jacobian is known as the Fréchet derivative. From [36], we can define the Fréchet derivative of f, denoted Df, as follows.

$$f(x + k) = f(x) + (Df(x))k + o\left(||k||\right)$$

Above, $(Df(x))k$ is a generalized matrix-vector product of $Df(x)$ and k, and "little-o" notation describes the asymptotic behavior of a function, defined as follows.

$$f(x) = o(g(x)) \text{ as } x \to a \quad \iff \quad \lim_{x \to a} \frac{f(x)}{g(x)} = 0$$

That is, the growth in f is bounded by the growth of g. In essentially all of our

analysis, we implicitly let a above be equal to zero as we are interested in the limit when the step size of the numerical scheme goes to zero. Alternatively, the Fréchet derivative may be defined if there exists a bounded linear operator D of f at x such that

$$\lim_{||k||\to 0} \frac{||f(x+k) - f(x) - Df(x)||}{||k||} = 0.$$

Again, in the finite-dimensional case with vectors, the Fréchet derivative is simply the Jacobian. Without proof, we'll state that if the Fréchet derivative exists for a function f at a point x, then it is unique [36]. Hence, the Fréchet derivative of a function gives the tangent space of very general functions.

We are also interested in the generalization to the directional derivative. In vector spaces, the directional derivative gives how a field will change in a particular direction. The generalization is known as the Gateaux derivative, or Gateaux differential, which we'll denote as D_k for direction k.

$$D_k f(x) = \lim_{h\to 0} \frac{f(x+hk) - f(x)}{h} = \frac{d}{dh} f(x+hk)\bigg|_{h=0} \tag{1.3}$$

Again without proof, we'll note that the Fréchet derivative exists at $x = a$ if and only if all Gateaux derivatives are continuous functions of x at $x = a$ [36]. This means that there exist Gateaux differentiable functions that are not Fréchet differentiable as Fréchet differentiability requires Gateaux differentiability in all possible directions. If, however, both exist, then we have the following.

$$D_k f = (Df)k$$

Now we have the basic tools to understand the tangent space. We'll loosely follow the work of [19] and [20] directly below. Consider the low-rank decomposition of $X \in \mathbb{R}^{m\times n}$.

$$X = UZ^T$$

For X of rank r, we have that $U \in \mathbb{R}^{m\times r}$ and $Z \in \mathbb{R}^{n\times r}$. From definitions 1.2.1 and 1.2.2, we can write any $X \in \mathscr{M}_r$ as $X = UZ^T$, which is defined by the bilinear map $(U, Z) \to UZ^T$. To obtain the tangent space of the manifold, we would like the Fréchet derivative of the bilinear map. But, it's easier to compute the Gateaux derivative of the bilinear map; if we can define the Gateaux derivative in all directions,

this gives us equivalent information to the Fréchet derivative. We use the following well-known lemma.

Lemma 1.3.1. Consider an arbitrary linear map $g : V \times W \to Y$. The Gateaux derivative of g in directions k_1, k_2 is as follows.

$$D_{k_1,k_2} g(v, w) = g(v, k_2) + g(k_1, w)$$

Proof.

$$D_{k_1,k_2} g(v, w) = \frac{d}{dh} g(v + hk_1, w + hk_2)\bigg|_{h=0}$$

From bilinearity, we have

$$g(v + hk_1, w + hk_2) = g(v, w) + hg(k_1, w) + hg(v, k_2) + h^2 g(k_1, k_2). \tag{1.4}$$

Hence,

$$\begin{aligned} D_{k_1,k_2} g(v, w) &= \frac{d}{dh} \left[g(v, w) + hg(k_1, w) + hg(v, k_2) + h^2 g(k_1, k_2) \right] \Big|_{h=0} \\ &= g(k_1, w) + g(v, k_2). \end{aligned}$$

□

Now, for this particular bilinear map, $g(U, Z) = UZ^T$, and we have that all Gateaux derivatives $\delta_X \in \mathcal{T}_X \mathscr{M}_r$ take the following form

$$\delta_X = U\delta_Z^T + \delta_U Z^T$$

for any $\delta_U \in \mathbb{R}^{m \times r}$, $\delta_Z \in \mathbb{R}^{n \times r}$. Hence, we can define the tangent spaces as follows.

$$\mathcal{T}_X \mathscr{M}_r = \left\{ U\delta_Z^T + \delta_U Z^T : \delta_U \in \mathbb{R}^{m \times r}, \delta_Z \in \mathbb{R}^{n \times r} \right\}$$

The form above, however, is redundant. We could choose $(\delta'_U, \delta'_Z) \neq (\delta_U, \delta_Z)$ but still have $U\delta_Z'^T + \delta'_U Z^T = U\delta_Z^T + \delta_U Z^T$. Furthermore, we would like to have U semi-orthonormal ($U^T U = I$) and have tangent matrices that preserve this relationship. Semi-orthonormality of U is equivalent to insisting that $U \in \mathcal{V}_{m,r}$, recalling that $\mathcal{V}_{m,r}$ denotes the Stiefel manifold of real $m \times r$ matrices.

Insisting U stays in $\mathcal{V}_{m,r}$ is equivalent to enforcing that the Gateaux derivative of U^TU in the direction of δ_U is equal to zero; that way, U^TU stays constant and equal to I. From lemma 1.3.1, we need

$$\delta_U^T U + U^T \delta_U = 0.$$

Enforcing this condition is not restrictive enough to remove all of the redundant degrees of freedom. Instead, we can parameterize the tangent space by restricting the tangent vectors so that $U^T\delta_U = 0$. Clearly, this would imply that U remains in $\mathcal{V}_{m,r}$. This is known as the dynamically orthogonal condition, and [20] shows that this is a unique parameterization.

We can define the tangent space of $\mathcal{V}_{m,r}$ at U.

$$\mathcal{T}_U\mathcal{V}_{m,r} = \left\{\delta_U \in \mathbb{R}^{m\times r} : \delta_U^T U + U^T\delta_U = 0\right\} = \left\{\delta_U \in \mathbb{R}^{m\times r} : U^T\delta_U \in \mathrm{so}(r)\right\}$$

Above, $\mathrm{so}(r)$ denotes the set of skew-symmetric, real $r \times r$ matrices. Similarly, we can define the *DO space*, $\mathcal{U}_{m,r}$, as follows.

$$\mathcal{U}_{m,r} = \left\{\delta_U \in \mathbb{R}^{m\times r} : U^T\delta_U = 0\right\} \subset \mathcal{T}_U\mathcal{V}_{m,r} \tag{1.5}$$

With this, we can rewrite the tangent space to the manifold as follows.

$$\mathcal{T}_X\mathscr{M}_r = \left\{U\delta_Z^T + \delta_U Z^T : \delta_U \in \mathcal{U}_{m,r}, \delta_Z \in \mathbb{R}^{n\times r}\right\} \tag{1.6}$$

Note that $\mathcal{T}_X\mathscr{M}_r$ is fully parameterized by (δ_U, δ_Z). Furthermore, the linear map

$$\begin{aligned}\mathcal{U}_{m,r} \times \mathbb{R}^{n\times r} &\to \mathcal{T}_X\mathscr{M}_r \\ (\delta_U, \delta_Z) &\to U\delta_Z^T + \delta_U Z^T \equiv \delta_X\end{aligned}$$

defines an isomorphism (a map between structures that may be inverted). One direction of this map is already clear; below, we will derive the inverse map, going from $\delta_X \to (\delta_U, \delta_Z)$.

Theorem 1.3.1. Given $X = UZ^T \in \mathscr{M}_r$ and $\delta_X \in \mathcal{T}_X\mathscr{M}_r$, $(\delta_U, \delta_Z) \in \mathcal{U}_{m,r} \times \mathbb{R}^{n\times r}$ are given as

$$\delta_U = (I - UU^T)\delta_X Z(Z^TZ)^{-1}, \quad \delta_Z = \delta_X^T U.$$

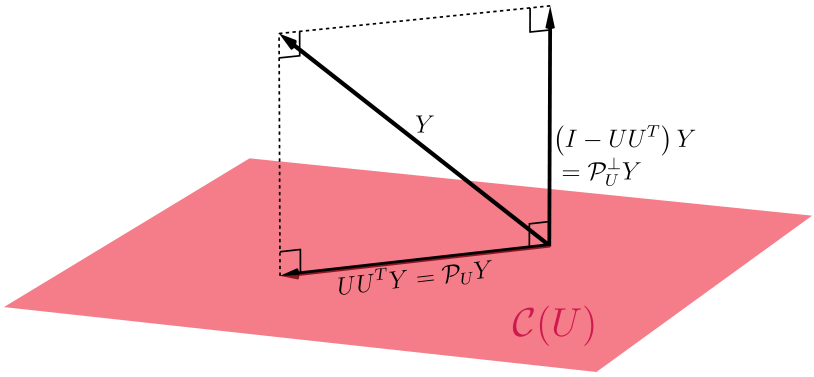


Figure 1-3: Graphical intuition for projection

Proof. From the definition of δ_X, we can easily find an expression for δ_Z.

$$\begin{aligned} \delta_X &= \delta_U Z^T + U\delta_Z^T \\ \Rightarrow U^T\delta_X &= \cancelto{0}{U^T\delta_U} Z^T + \cancelto{I}{U^T U}\delta_Z^T \\ \Rightarrow \delta_X^T U &= \delta_Z \end{aligned}$$

Now, we seek δ_U; we first take the orthogonal projection of δ_X on U, given by the operator $(I - UU^T)$.

$$\begin{aligned} (I - UU^T)\delta_X &= U\delta_Z^T + \delta_U Z^T - UU^T(U\delta_Z^T + \delta_U Z^T) \\ &= U\delta_Z^T + \delta_U Z^T - U\delta_Z^T \\ &= \delta_U Z^T \end{aligned}$$

Now, to get rid of the Z^T on the RHS, we multiply by Z and then the resulting matrix inverse. Note that this is just the Moore–Penrose right inverse.

$$\delta_U = (I - UU^T)\delta_X Z(Z^T Z)^{-1}$$

□

With this, we have a bijective map between (U, Z) and and element in the tangent space.

1.4 Projection onto the tangent space

Now, we may derive the differential equations for a dynamical system. Suppose we have the following initial value problem.

$$\frac{\partial A(t)}{\partial t} \equiv \dot{A}(t) = \mathscr{L}(t, A(t)) \tag{1.7}$$

$$A(0) = A_0 \tag{1.8}$$

We seek

$$\dot{X}(t) \in \mathcal{T}_{X(t)}\mathscr{M}_r \tag{1.9}$$

$$X(0) = X_0 \in \mathscr{M}_r \tag{1.10}$$

such that $||\dot{X}(t) - \mathscr{L}(t)||$ and $||A_0 - X_0||$ are minimized in some norm (which for us will be the Frobenius norm). We seek $\dot{X} \in \mathcal{T}_{X(t)}\mathscr{M}_r$ so that $X(t)$ remains on the low-rank manifold. Note that minimizing $||\dot{X}(t) - \mathscr{L}(t)||$ such that (1.9) holds is equivalent to the condition below [30].

$$\langle \dot{X} - \mathscr{L}, \delta_X \rangle = 0 \quad \forall \delta_X \in \mathcal{T}_X\mathscr{M}_r \tag{1.11}$$

Above, $\langle \cdot, \cdot \rangle$ denotes the inner product. This is a typical Galerkin condition insisting that the residual $\dot{X} - \mathscr{L}$ is orthogonal to the tangent space (or every element δ_X in the tangent space). This forces, instantaneously, the best approximation to $\mathscr{L}$ possible in the tangent space.

Many papers start from (1.9) without further justification. However, for pedagogical purposes, we'll provide further motivation as to why we seek a low-rank solution $X(t)$ whose derivative in the tangent space attempts to match $\mathscr{L}$. The best approximation to the solution $A(t)$ to the initial value problem (1.7, 1.8) would be the projection of $A(t)$ onto the low-rank manifold at all times t. This notion may actually be used as the definition of the projection operator, $\mathscr{P}_{\mathscr{M}_r}$. More precisely, for some arbitrary matrix $W \in \mathbb{R}^{n \times m}$,

$$\mathscr{P}_{\mathscr{M}_r} W = \underset{\tilde{W} \in \mathscr{M}_r}{\arg\min} ||W - \tilde{W}||. \tag{1.12}$$

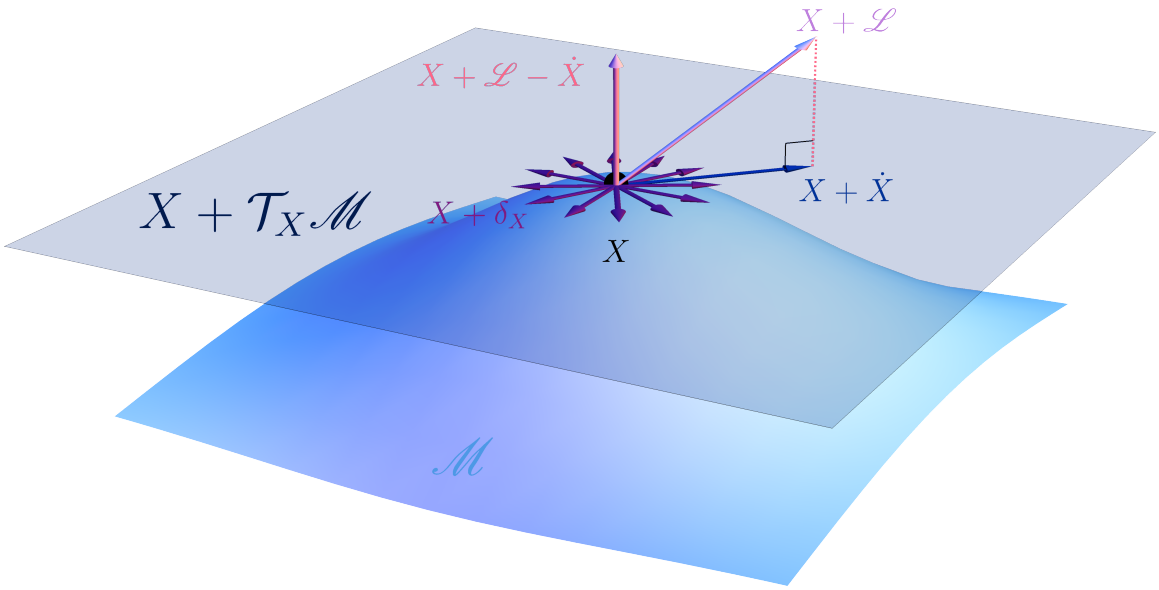


Figure 1-4: Orthogonal projection of $X + \mathscr{L}$ onto affine tangent space of manifold

Back to our problem, we seek X such that

$$X(t) = \mathscr{P}_{\mathscr{M}_r} A(t). \tag{1.13}$$

From here, we can deduce the initial condition for (1.10): $X_0 = \mathscr{P}_{\mathscr{M}_r} A_0$. But, that's as far as we can get so easily since we don't a priori know the solution $A(t)$ for $t > 0$. However, we do know how $A(t)$ changes with time, so we differentiate (1.13) with respect to time.

$$\frac{\partial X}{\partial t} = \frac{\partial}{\partial t} \mathscr{P}_{\mathscr{M}_r} A(t)$$

To proceed, we'll follow [20, p. 526]. Consider the definition of the derivative.

$$\frac{\partial}{\partial t} \mathscr{P}_{\mathscr{M}_r} A(t) = \lim_{h \to 0} \frac{\mathscr{P}_{\mathscr{M}_r} A(t+h) - \mathscr{P}_{\mathscr{M}_r} A(t)}{h}$$

Now, we can rewrite $A(t+h)$ using our knowledge of its derivative.

$$\frac{\partial}{\partial t} \mathscr{P}_{\mathscr{M}_r} A(t) = \lim_{h \to 0} \frac{\mathscr{P}_{\mathscr{M}_r} \left[A(t) + \int_t^{t+h} \frac{\partial A(s)}{\partial s} ds \right] - \mathscr{P}_{\mathscr{M}_r} A(t)}{h} \tag{1.14}$$

$$= \lim_{h \to 0} \frac{\mathscr{P}_{\mathscr{M}_r} \left[A(t) + h\mathscr{L}(t, A(t)) \right] - \mathscr{P}_{\mathscr{M}_r} A(t)}{h} \tag{1.15}$$

Above, we have used the fact that in the limit, the integral can be expressed as

$h\mathscr{L}(t, A(t))$ – think of this as approximating the integral in a Riemannian sense, using one box to approximate the area under the curve. Now, it's clear that this is just the Gateaux derivative of the projection operator in the direction of $\mathscr{L}(t, A(t))$ (see the definition of the Gateaux derivative, (1.3)). In other words, our expression is equivalent to how the projection of A changes in the direction of $\mathscr{L}$.

$$\frac{\partial}{\partial t}\mathscr{P}_{\mathscr{M}_r}A(t) = D_{\mathscr{L}(t,A(t))}\mathscr{P}_{\mathscr{M}_r}A(t)$$

It is known that at a point $X \in \mathscr{M}_r$, $D\mathscr{P}_{\mathscr{M}_r} = \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}$ [37, Lemma 4]. Intuitively, this is the case since we consider only an infinitesimal step in the $\mathscr{L}$ direction; then, since the manifold is smooth, it may be approximated (arbitrarily well as the step gets smaller and smaller) as a tangent space. How does the projection of $A(t)$ change in the direction of $\mathscr{L}$ on a tangent space? Well, since projecting onto tangent spaces is a linear process, the change in the projection onto the manifold is equal to the projection of $\mathscr{L}$ onto the tangent space. It may, in fact, be clearer if we go back to (1.15).

$$\lim_{h\to 0}\frac{\mathscr{P}_{\mathscr{M}_r}\left[A(t) + h\mathscr{L}(t, A(t))\right] - \mathscr{P}_{\mathscr{M}_r}A(t)}{h}$$
$$= \lim_{h\to 0}\frac{\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\left[A(t) + h\mathscr{L}(t, A(t))ds\right] - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}A(t)}{h}$$

Here, we've just used the fact that the manifold may be approximated as a tangent space in the limit. Next, we'll use the linearity of projection onto linear spaces.

$$\begin{aligned}
&= \lim_{h\to 0}\frac{\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}A(t) + \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}h\mathscr{L}(t, A(t)) - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}A(t)}{h}\\
&= \lim_{h\to 0}\frac{h\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}(t, A(t))}{h}\\
&= \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}(t, A(t))
\end{aligned}$$

Hence, we have shown that we seek to solve $\dot{X} = \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}(t, A(t))$. Depending on the problem, we may not know $A(t)$, so if we use X to approximate A, we have $\dot{X} = \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\mathscr{L}(t, X(t))$. This is known as the Dirac-Frenkel time-dependent variational principle [15, 16]. In either case, this is equivalent to satisfying (1.11).

Theorem 1.4.1. For $X = UZ^T \in \mathscr{M}_r$, $U \in \mathcal{U}_{m,r}$, and $Z \in \mathbb{R}^{n\times r}_*$, (1.11) is equivalent

to

$$\dot{X} = \dot{U}Z^T + U\dot{Z}^T,$$

where

$$\dot{U} = \mathscr{P}_U^{\perp}\mathscr{L}Z(Z^TZ)^{-1}$$
$$\dot{Z} = \mathscr{L}^TU$$

with $\mathscr{P}_U^{\perp} = I - \mathscr{P}_U$, $\mathscr{P}_U = UU^T$, $\mathscr{P}_Z^{\perp} = I - \mathscr{P}_Z$, and $\mathscr{P}_Z = Z(Z^TZ)^{-1}Z^T$. For a matrix A, we'll define $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}^{\perp}(A) = \mathscr{P}_U^{\perp}A\mathscr{P}_Z^{\perp}$, and, naturally, $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}(A) = I - \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}^{\perp}(A)$. Then, we can express $\dot{X}$ as follows.

$$\dot{X} = \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}(\mathscr{L})$$

Proof. Since we require $\dot{X}, \delta_X \in \mathcal{T}_X\mathscr{M}_r$, by (1.6), we let $\dot{X} = \dot{U}Z^T + U\dot{Z}^T$ and $\delta_X = \delta_U Z^T + U\delta_Z^T$ for some $\dot{U}, \delta_U \in \mathcal{U}_{m,r}$ and $\dot{Z}, \delta_Z \in \mathbb{R}^{n\times r}$.

$$\langle \dot{U}Z^T + U\dot{Z}^T - \mathscr{L}, \delta_U Z^T + U\delta_Z^T\rangle = 0 \tag{1.16}$$

In the Frobenius norm with real matrices A and B, $\langle A, B\rangle = \mathrm{Tr}(A^TB) = \mathrm{Tr}(B^TA)$ where $\mathrm{Tr}(\cdot)$ denotes the trace. Note that the trace is a linear mapping. Since this must be true for all δ_U, we may set it to zero and we'll have the following.

$$\mathrm{Tr}(Z\cancelto{0}{\dot{U}^TU}\delta_Z^T) + \mathrm{Tr}(\dot{Z}\cancelto{I}{U^TU}\delta_Z^T) = \mathrm{Tr}(\mathscr{L}^TU\delta_Z^T)$$
$$\Rightarrow \mathrm{Tr}(\dot{Z}\delta_Z^T) = \mathrm{Tr}(\mathscr{L}^TU\delta_Z^T)$$

Now, we'll specify δ_Z to take the following form.

$$\delta_Z = \mathbb{1}_{ij}$$
$$\Rightarrow (\delta_Z)_{kl} = \begin{cases} 1, & k = i, l = j \\ 0, & \text{otherwise} \end{cases}$$

Lemma 1.4.1. For a real matrix A,

$$\mathrm{Tr}(A^T \mathbb{1}_{ij}) = A_{ij}$$

Proof. The proof follows directly from the definition of the trace operation.

$$\mathrm{Tr}(A^T \mathbb{1}_{ij}) = \sum_{kl} A_{kl} (\mathbb{1}_{ij})_{k,l} = A_{ij}$$

□

With lemma 1.4.1, we have the following.

$$\dot{Z}_{ij} = (\mathscr{L}^T U)_{ij}$$

This is true for all i, j, so we have shown that $\dot{Z} = \mathscr{L}^T U$. What remains is to find the expression for $\dot{U}$. This time, we'll set $\delta_Z = 0$ and solve for δ_U.

We require $\delta_U \in \mathcal{U}_{m,r}$, and from (1.5), we know this means $U^T \delta_U = 0$. Let $\delta_Y \in \mathbb{R}^{m \times n}$ be some matrix. Then, we'll let $\delta_U = \mathscr{P}_U^\perp \delta_Y$. Note that immediately, we see $\delta_U \in \mathcal{U}_{m,r}$ since $U^T \delta_U = U^T (I - UU^T) \delta_Y = U^T \delta_Y - U^T \delta_Y = 0$. We'll also let $\delta_Z = 0$.

$$\langle \dot{U} Z^T + U \dot{Z}^T - \mathscr{L}, \mathscr{P}_U^\perp \delta_Y Z^T \rangle = 0$$

Using the definition of the Frobenius product, we obtain the following.

$$\mathrm{Tr}(Z \dot{U}^T \mathscr{P}_U^\perp \delta_Y Z^T) + \mathrm{Tr}(\dot{Z} \cancelto{0}{U^T \mathscr{P}_U^\perp \delta_Y} Z^T) = \mathrm{Tr}(\mathscr{L}^T \mathscr{P}_U^\perp \delta_Y Z^T)$$

Now, we'll use the cyclic permutation property of the trace operator, where $\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB) = \mathrm{Tr}(BCA)$.

$$\mathrm{Tr}(Z^T Z \dot{U}^T \mathscr{P}_U^\perp \delta_Y) = \mathrm{Tr}(Z^T \mathscr{L}^T \mathscr{P}_U^\perp \delta_Y)$$

Again, since $\delta_Y \in \mathbb{R}^{n \times r}$ is arbitrary, we can set it to $\mathbb{1}_{ij}$. Applying lemma 1.4.1, we obtain the following.

$$(Z^T Z \dot{U}^T \mathscr{P}_U^\perp)_{ji} = (Z^T \mathscr{L}^T \mathscr{P}_U^\perp)_{ji}$$

Again, this is true for all i, j. Also note that since $\dot{U} \in \mathcal{U}_{m,r}$, $\dot{U}^T \mathscr{P}_U^\perp = \dot{U}^T - \cancelto{0}{\dot{U}^T U} U^T = \dot{U}^T$. Upon inverting the square matrix $Z^T Z$, we have shown the following result.

$$\dot{U} = \mathscr{P}_U^\perp \mathscr{L} Z (Z^T Z)^{-1}$$

Simply multiplying out $\dot{X} = \dot{U} Z^T + U \dot{Z}^T$, we now prove the last result.

$$\begin{aligned}
\dot{X} &= \dot{U} Z^T + U \dot{Z}^T \\
&= \mathscr{P}_U^\perp \mathscr{L} \mathscr{P}_Z + \mathscr{P}_U \mathscr{L} \\
&= \mathscr{P}_U^\perp \mathscr{L} \mathscr{P}_Z + \mathscr{L} - \mathscr{P}_U^\perp \mathscr{L} \\
&= \mathscr{L} - \mathscr{P}_U^\perp \mathscr{L} \mathscr{P}_Z^\perp \\
&= (I - \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r}^\perp) \mathscr{L} \\
&= \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L}
\end{aligned}$$

□

Another way of completing this proof is using matrix calculus, differentiating $\frac{1}{2}||\dot{U} Z^T + U \dot{Z}^T - \mathscr{L}||^2$ with respect to $\dot{U}$ and $\dot{Z}$ subject to $\dot{U}^T U = I$, and then setting the derivatives equal to zero (see appendix C). Nevertheless, we have defined operators that project onto the tangent and normal spaces of the low-rank manifold. Hence, as a bonus, we may define the normal space to the low-rank manifold at X as the set of matrices whose normal projection is equal to the matrix itself (i.e. has no tangent component).

$$\mathcal{N}_X \mathscr{M}_r = \left\{ N \in \mathbb{R}^{m \times n} : \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r}^\perp N = N \right\}$$

Another way to view the normal space is to consider tangent vectors in the form (1.6). As stated in [30], tangent vectors are in the sum of two overlapping linear spaces: the row space of Z^T and the column space of U. So, for a matrix N to be in the normal space, we must have that N is not in the row space of Z^T nor in the column space of U. So, from [20], we have an equivalent definition of the normal space.

$$\mathcal{N}_X \mathscr{M}_r = \left\{ N \in \mathbb{R}^{m \times n} : U^T N = 0, N Z = 0 \right\}$$

To conclude, we'll write out a formulation to solve the initial motivating problem

in approximating (1.7, 1.8).

$$\dot{X} = \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L}(t, B(t)) \tag{1.17}$$

$$X(0) = \mathscr{P}_{\mathscr{M}_r} A_0 \tag{1.18}$$

Above, we've introduced the term $B(t)$ to allow for two cases. When we know $A(t)$ (e.g. in data compression, see section 3.2), we set $B(t) = A(t)$. When we do not know $B(t)$ (e.g. in solving stochastic partial differentials, see section 3.4), we set $B(t) = X(t)$. We can expand (1.17) using the equations just derived for the tangent space, letting $X = UZ^T$.

$$\dot{U} = (I - UU^T)\mathscr{L} Z (Z^T Z)^{-1}, \quad \dot{Z} = \mathscr{L}^T U \tag{1.19}$$

$$U(0)Z(0)^T = \mathscr{P}_{\mathscr{M}_r} A_0 \tag{1.20}$$

Equation (1.20) can be computed by computing the singular value decomposition of A_0, letting $U(0)$ be the left singular vectors, and letting $Z(0)$ equal to the right singular vectors right multiplied by the singular values. In the next section, we will address how to solve (1.19).

Chapter 2

Retractions onto the low-rank manifold

2.1 Motivation and preliminaries

With projections onto the tangent space defined, retractions (which will be defined soon) come into play. To motivate this discussion, let's consider a naive first attempt at solving (1.19). This is a nonlinear coupled system of matrix ordinary differential equations. Not only is there nonlinearity in U and Z directly, but $\mathscr{L}$ may (and almost always does) depend on U and Z. For now, though, let's focus on a simple example from [18]: updating U and Z independently by a forward Euler discretization. Henceforth, we'll use a subscript to denote a discrete time index such that $U_n = U(t_n)$ for $t_n = n\Delta t$ (for some $\Delta t \in \mathbb{R}$) and similarly for Z, where, without loss of generality, we've assumed our time interval of interest is $[0, T_f]$. We define $\dot{U}$ and $\dot{Z}$ as follows.

$$\Delta t\dot{U} \equiv U_{n+1} - U_n \approx \int_{t_n}^{t_{n+1}} (I - UU^T)\mathscr{L}Z(Z^TZ)^{-1}dt \tag{2.1}$$

$$\Delta t\dot{Z} \equiv Z_{n+1} - Z_n \approx \int_{t_n}^{t_{n+1}} \mathscr{L}^T U dt \tag{2.2}$$

Now, in the continuous limit where $\Delta t \to 0$, it suffices to use (1.19) in defining $\dot{U}$ and $\dot{Z}$ where $\mathscr{L}$ is evaluated at time t_n. For non-infinitesimal Δt, from (1.7) we have the following.

$$A(t_{n+1}) - A(t_n) = \int_{t_n}^{t_{n+1}} \mathscr{L}(t)dt$$

Of course, it is seldom possible to exactly evaluate that integral, so we may use any number of classic time integration schemes (e.g. Euler, leapfrog, Runge-Kutta, et cetera) to approximate that integral; we will denote the approximation of that integral divided by Δt as $\overline{\mathscr{L}}$.

$$\overline{\mathscr{L}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathscr{L}(t)dt + \mathcal{O}(\Delta t^k)$$

Hence, $\overline{\mathscr{L}}$ may be obtained via a kth-order time integration scheme, and we may also assume that any error due to spatial discretization is built into $\overline{\mathscr{L}}$. So henceforth, when dealing with discrete time intervals, we will use $\overline{\mathscr{L}}$ instead of $\mathscr{L}$ in our analysis, which gives the following slightly modified definitions for our retractions.

$$\dot{U} = (I - UU^T)\overline{\mathscr{L}}Z(Z^TZ)^{-1}, \quad \dot{Z} = \overline{\mathscr{L}}^T U \tag{2.3}$$

In our Euler forward retraction, we are assuming that $\dot{U}$ and $\dot{Z}$ are functions of information known at time t_n and that $\overline{\mathscr{L}}$ is given. That is, the time stepping is explicit. With this, consider the next point X_{n+1}.

$$\begin{aligned} X_{n+1} = U_{n+1}Z_{n+1}^T &= (U_n + \Delta t\dot{U})(Z_n + \Delta t\dot{Z})^T \\ &= U_nZ_n^T + \Delta t\left[U_n\dot{Z}^T + \dot{U}Z_n^T\right] + \Delta t^2\dot{U}\dot{Z}^T \\ &= \underbrace{X_n + \Delta t\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}}_{\text{consistent integrator}} + \underbrace{\Delta t^2\dot{U}\dot{Z}^T}_{\text{retraction error}} \end{aligned}$$

The first two components form a consistent integrator. However, we have an $\mathcal{O}(\Delta t^2)$ error. One may ask why not just exclude the last term in the scheme? But, this is tricky to do – the term arose naturally from the integration scheme. Simply removing that term would not allow us to write $X_{n+1} = U_{n+1}Z_{n+1}^T$ in a factored form, and, in general, removing the error term would cause X_{n+1} to depart from the low-rank manifold (though there are exceptions). This is intuitive, as taking a finite-sized (by which we mean not infinitesimally small) step along the affine tangent space will cause the solution to depart the low-rank manifold since the low-rank manifold has higher-order curvature. In some sense, this reveals the "purpose" of what we have called the retraction error: it keeps the solution on the low-rank manifold.

At this point, it is important to recognize that there are four main errors when integrating along the low-rank manifold. There is time-integration error, spatial discretization error, model closure error, and manifold curvature error. We are lumping

our first two errors into $\overline{\mathscr{L}}$, and the work we present is agnostic to whether or not $\overline{\mathscr{L}}$ is exact or approximate. Model closure error is somewhat inevitable and arises due to the truncation of the solution; some of the closure error may be described via the Dirac-Frenkel time-dependent variational principle, which will be expounded upon in section 3.3. The manifold curvature error is what we seek to minimize in this thesis. Because the low-rank manifold has high-order curvature, projecting onto the affine tangent space ignores high-order curvature, and the methods discussed in this chapter take the high-order curvature into account. We also re-emphasize that in the continuous time limit as $\Delta t \to 0$, the manifold curvature error is exactly zero. That is, the DO equations (1.19) are exact. But when dealing with discrete time, we need a richer formulation to better advance our low-rank solutions through time.

Using an alternative parameterization of the low-rank manifold elucidates how the retraction error changes with the singular values of X. We can always switch between $X = UZ^T, U \in \mathcal{V}_{m,r}, Z \in \mathbb{R}_*^{n\times r}$ and $X = USV^T, U \in \mathcal{V}_{m,r}, S \in \mathbb{R}_*^{r\times r}, V \in \mathcal{V}_{n,r}$ by keeping U the same and setting $Z = VS^T$ or $[V, S^T] = \texttt{qr}(Z)$. With this, we'll write out the retraction error and switch parameterizations to prove the following theorem.

Theorem 2.1.1. Consider a point $X = UZ^T \in \mathscr{M}_r$ with the retraction error defined as $\varepsilon_{\text{ret}}(X) \equiv \Delta t^2 \dot{U}\dot{Z}^T$, where $\dot{U}$ and $\dot{Z}$ are defined by (2.3). The retraction error is bound by the following inequality

$$\frac{||\varepsilon_{\text{ret}}(X)||_2}{||\overline{\mathscr{L}}||_2^2} \leq \frac{\Delta t^2}{\sigma_{\min}(X)}, \tag{2.4}$$

where $\sigma_{\min}(X)$ denotes the smallest singular value of X.

Proof.

$$\begin{aligned}
\dot{U}\dot{Z}^T &= (I - UU^T)\overline{\mathscr{L}}Z(Z^TZ)^{-1}U^T\overline{\mathscr{L}} \\
&= (I - UU^T)\overline{\mathscr{L}}VS^T(SV^TVS^T)^{-1}U^T\overline{\mathscr{L}} \\
&= (I - UU^T)\overline{\mathscr{L}}VS^{-1}U^T\overline{\mathscr{L}}
\end{aligned}$$

Now, we'll note that the Frobenius and L^2 norms are invariant under unitary operations such as multiplication by U and V. Also, for a matrix A, note that the triangle

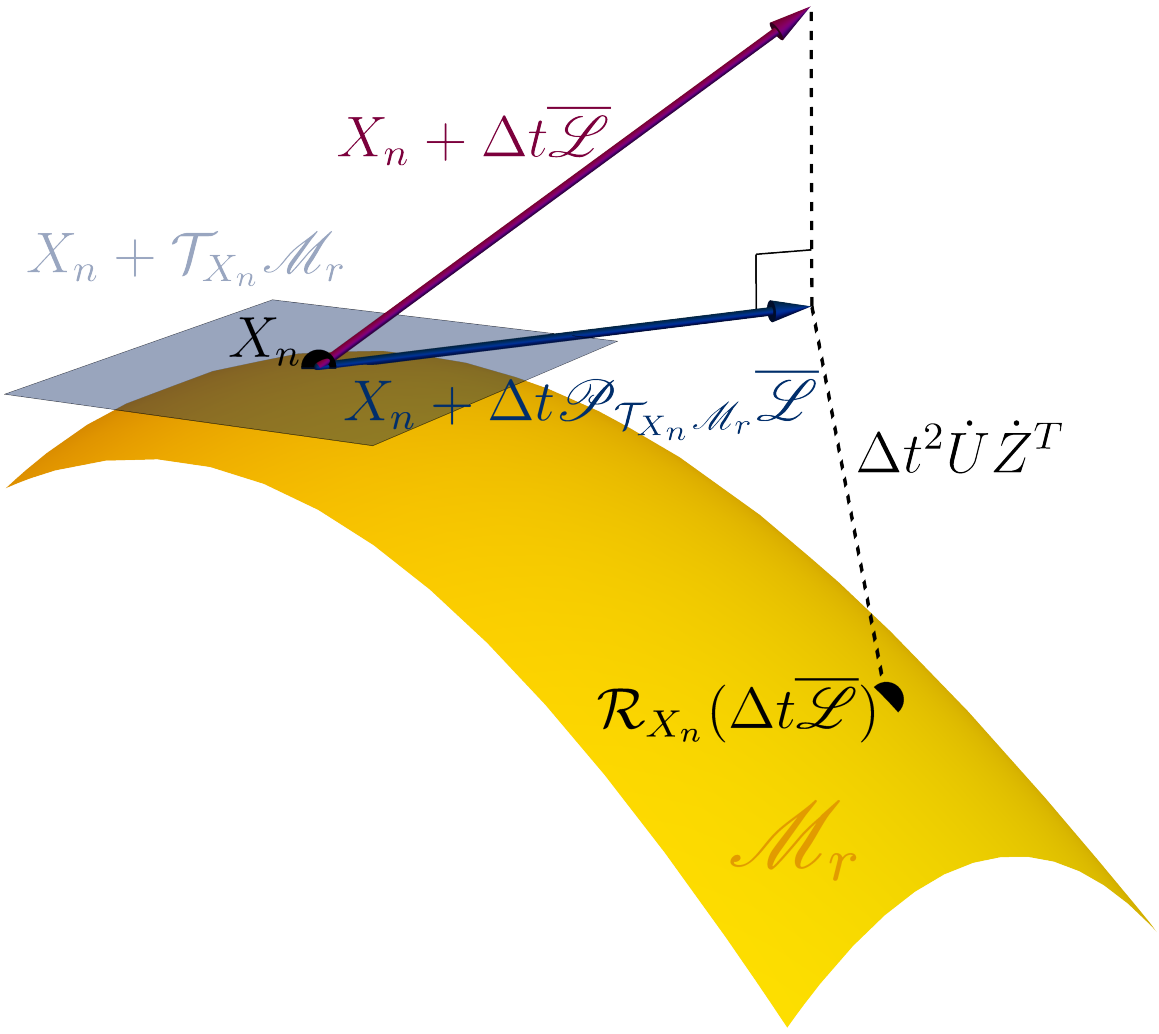


Figure 2-1: This figure illustrates how a retraction maps a vector, projected onto the affine tangent space, to the low-rank manifold. This incurs a finite-sized error, which in the case of the Euler forward retraction, is of order $\mathcal{O}(\Delta t^2)$.

inequality imply the following.

$$\begin{aligned} A &= \mathscr{P}_U^{\perp} A + \mathscr{P}_U A \\ \Rightarrow ||\mathscr{P}_U^{\perp} A|| &= ||A - \mathscr{P}_U A|| \\ &\leq ||A|| + ||\mathscr{P}_U A|| \\ &\leq ||A|| \end{aligned}$$

Together with the Cauchy-Schwartz inequality, we have the following

$$\begin{aligned} ||\dot{U}\dot{Z}^T|| &= (I - UU^T)\overline{\mathscr{L}}VS^{-1}U^T\overline{\mathscr{L}} \\ &\leq ||\overline{\mathscr{L}}||^2||S^{-1}|| \end{aligned}$$

In the L^2 norm, $||A||_2 = \sigma_{\max}(A)$. That is, the L^2 norm of the matrix is equal to the max singular value of that matrix. Hence, $||A^{-1}||_2 = \frac{1}{\sigma_{\min(A)}}$. Also note that $\sigma(S) = \sigma(X)$: the singular values of X are the singular values of the current point on the manifold. □

This theorem concurs with the fact that the curvature of the manifold scales with the smallest singular value of X [20]. The retraction error is essentially just the difference between a flat approximation of the manifold and the curved manifold itself, i.e. its curvature.

With that, we dive into the formal definition of a retraction. First defined in [38], a retraction maps a tangent vector/matrix back to the manifold. From [30], we have that a retraction $\mathcal{R}$ at point $X \in \mathscr{M}_r$ of an element $\xi \in \mathcal{T}_X\mathscr{M}_r$ can be expressed as follows.

$$\mathcal{R}_X(\xi) = X + \xi + o(||\xi||) \tag{2.5}$$

What distinguishes one retraction from another is the $o(||\xi||)$ term, which determines how to get back to $\mathscr{M}_r$. Below, we more rigorously define a retraction from [39].

Definition 2.1.1. A retraction $\mathcal{R} : \mathcal{T}\mathscr{M}_r \to \mathscr{M}_r$ on $\mathscr{M}_r$ is a smooth mapping from the tangent bundle to the manifold such that

1. $\mathcal{R}$ is defined and smooth on a neighborhood of the zero section in $\mathcal{T}\mathscr{M}_r$,
2. $\mathcal{R}_X(0) = X \quad \forall X \in \mathscr{M}_r$,
3. $\frac{d}{dt}\mathcal{R}_X(t\xi)\big|_{t=0} = \xi \quad \forall X \in \mathscr{M}_r$ and $\xi \in \mathcal{T}_X\mathscr{M}_r$.

The zero section of the tangent bundle is the submanifold of that bundle that consists of all the zero matrices. This guarantees that for any point $X \in \mathscr{M}_r$, there is a neighborhood of points in $\mathcal{T}_X\mathscr{M}_r$ around the zero matrix where the retraction is smooth and exists. The second condition simply states that if our tangent vector is zero, the retraction should map back onto the original point. And the third condition is essentially the same as equation (2.5).

Definition 2.1.2. Adopting the terminology of [39], an extended retraction $\mathcal{R} : \mathcal{E} \to \mathscr{M}_r$ is a mapping such that

1. $\mathcal{R}$ is defined and smooth on a neighborhood of the zero section in $\mathcal{T}\mathscr{M}_r$,
2. $\mathcal{R}_X(0) = X \quad \forall X \in \mathscr{M}_r$,
3. $\frac{d}{dt}\mathcal{R}_X(t\xi)\big|_{t=0} = \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\xi \quad \forall X \in \mathscr{M}_r$ and $\xi \in \mathcal{E}$.

Note that the first two conditions are exactly the same. The extended retraction simply allows for ξ to be any vector/matrix rather than being restricted to the affine tangent space of the manifold at X. The derivative condition is similar, just that now we must project the matrix ξ onto the tangent space at X. We use the same notation for retractions and extended retractions since they accomplish the same feat, and whether the retraction itself is extended or not may be determined implicitly by the argument.

For a survey of retractions onto the low-rank manifold, see [39]. One retraction is mapping onto a geodesic of $\mathscr{M}_r$. A geodesic is the generalization of a straight line onto curved spaces. Imagine driving along a curved road where you (and the car) are much, much smaller than the curvature of the road such that, locally, the road appears flat. If you were to drive in (what appeared to you as) as straight line, this would trace out a geodesic on the manifold. It is commonly said that using geodesics as a retraction would be "theoretically ideal." In some ways, it is the most agnostic choice; considering our driving example, if we are told to drive ten miles and only provided with an initial velocity and a direction in which to drive, driving in that direction at the same speed for the entire ten miles would be the "natural," most obvious thing to do without inferring extra information. Unfortunately, moving along geodesics is computationally expensive [40]. Although there is a closed-form expressions for a particular Riemannian metric on $\mathscr{M}_r$ [41], most other metrics do not admit such expressions and the geodesics often must be solved numerically from differential equations.

Fortunately, retractions abstract away from geodesics by insisting only that they match geodesics to the first order. This allows for a cheaper approximation to the geodesic, and the higher-order terms are typically unimportant for small step sizes anyways. That said, the higher-order terms are a new source of error we must consider. This brings us to another definition of a new type of retraction.

Definition 2.1.3. A second-order retraction $\mathcal{R} : \mathcal{T}\mathscr{M} \to \mathscr{M}_r$ is a retraction whose second-order error belongs to the normal space of $\mathscr{M}_r$ at X. More precisely, the following holds.

$$\frac{d^2}{dt^2}\mathcal{R}_X(t\xi) \in \mathcal{N}_X\mathscr{M}_r \tag{2.6}$$

We note that the idea of a second-order extended retraction is naturally defined by just extending the domain of $\mathcal{R}_X$ to the embedding Euclidean space. Also, (2.6) may be thought of in another way. Denoting the $o(||\xi||)$ term in (2.5) as ε, we have the following.

$$\begin{aligned} &\mathcal{R}_X(t\xi) = X + t\xi + \varepsilon(t) \\ \Rightarrow &\frac{d}{dt}\mathcal{R}_X(t\xi) = \xi + \frac{d}{dt}\varepsilon(t) \\ \Rightarrow &\frac{d^2}{dt^2}\mathcal{R}_X(t\xi) = \frac{d^2}{dt^2}\varepsilon(t) \end{aligned}$$

If we assume ε is $\mathcal{O}(t^2)$, which is consistent with the $o(||t\xi||)$ condition from (2.5), then we can express it as $\varepsilon = \frac{1}{2}St^2 + o(t^2)$ for some S which may depend on X and ξ. Then, for $\mathcal{R}$ to be a second-order retraction, we must have that $S \in \mathcal{N}_X\mathscr{M}_r$. This condition is also motivated by matching geodesics, as the second derivative of a parameterized geodesic $\gamma(t)$ lies in the normal space. Intuitively, we may return to our driving example. If, locally, we drive in a straight line, certainly we are not accelerating in the plane of our reference frame – we are not turning or increasing our speed. But, we may still be accelerating due to the curvature of the manifold upon which we are driving. Recall from physics that the acceleration of a particle with velocity v and path curvature R is v^2/R in the direction normal to the velocity. Hence, the only acceleration will be normal to the car's current velocity, and extending this idea more generally to geodesics, we would expect a geodesic path to have "acceleration" only in the direction normal to the tangent space at a given point. With these ideas, we will proceed in defining a few simple retractions followed by new retractions derived using perturbation theory.

2.2 Projective retractions

We've already seen what we call the Euler forward retraction from equation (2.5). Below, we'll quickly show a lemma as to why it is not ideal.

Lemma 2.2.1. The Euler forward retraction is not a second-order retraction.

Proof. It is equivalent to show that $\dot{U}\dot{Z}^T \notin \mathcal{N}_X\mathscr{M}_r \Leftrightarrow \mathscr{P}^{\perp}_{\mathcal{T}_X\mathscr{M}_r}\dot{U}\dot{Z}^T \neq \dot{U}\dot{Z}^T$.

$$\begin{aligned}\mathscr{P}^{\perp}_{\mathcal{T}_X\mathscr{M}_r}\dot{U}\dot{Z}^T &= \mathscr{P}^{\perp}_{\mathcal{T}_X\mathscr{M}_r}(I-UU^T)\overline{\mathscr{L}}Z(Z^TZ)^{-1}U^T\overline{\mathscr{L}} \\ &= (I-UU^T)(I-UU^T)\overline{\mathscr{L}}Z(Z^TZ)^{-1}U^T\overline{\mathscr{L}}(I-Z(Z^TZ)^{-1}Z^T) \\ &= (I-UU^T)\overline{\mathscr{L}}Z(Z^TZ)^{-1}U^T\overline{\mathscr{L}}(I-Z(Z^TZ)^{-1}Z^T)\end{aligned}$$

Hence, $\mathscr{P}^{\perp}_{\mathcal{T}_X\mathscr{M}_r}\dot{U}\dot{Z}^T \neq \dot{U}\dot{Z}^T$ in general. □

From this, it is clear that the Euler forward integration scheme induces a retraction error that can be reduced with a different choice retraction. The natural next step may be to think about an Euler backward, or implicit Euler, scheme. The Euler backward retraction could be defined in writing $\dot{U}$ and $\dot{Z}$ as functions of U^{n+1} and Z^{n+1}. But, recall this would yield nonlinear systems of equations which may prove costly to solve.

While the Euler forward scheme is perhaps the simplest, cheapest scheme one can implement, we consider the other end of the spectrum: the extended projective retraction, or the projection onto the manifold. The extended projective retraction is as it sounds – the retraction is defined by taking the projection of $X + \Delta t\overline{\mathscr{L}}$ onto the low-rank manifold. In some sense, this can be thought of the ideal retraction (instead of the "theoretically ideal" geodesic retraction) because it minimizes the total retraction error by definition. It may be implemented via a standard truncation of the singular value decomposition (as similarly done in principal component analysis). A cheaper implementation would be the (non-extended) projective retraction, where we project $X + \Delta t\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ onto the low-rank manifold.

$$\mathcal{R}_X^{\text{ext proj}}(\xi) = \mathscr{P}_{\mathscr{M}_r}(X+\xi) \tag{2.7}$$

$$\mathcal{R}_X^{\text{proj}}(\xi) = \mathscr{P}_{\mathscr{M}_r}(X+\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\xi) \tag{2.8}$$

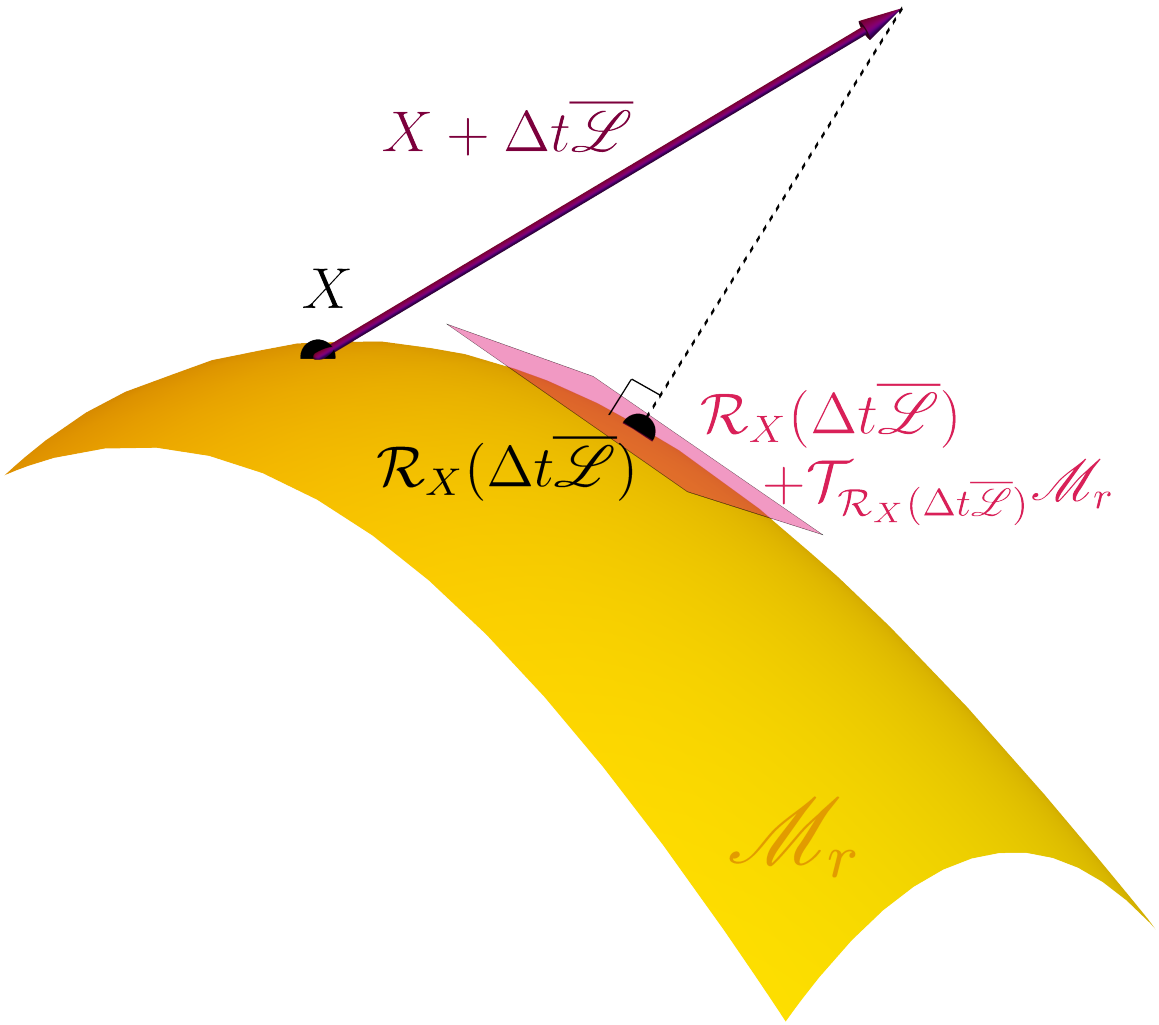


Figure 2-2: This figure shows an extended projective retraction. $X+\Delta t\overline{\mathscr{L}}$ is projected onto the low-rank manifold, and hence the line connecting $X+\Delta t\overline{\mathscr{L}}$ to the retraction is orthogonal to the affine tangent space at $\mathcal{R}_X(\Delta t\overline{\mathscr{L}})$.

As good measure, we'll show property three of the extended projective retraction; the first property is true (see [42, Prop. 6]) and the second is obviously true. Consider the singular value decompositions $X = \sum_{i=1}^{r} \sigma_i u_i v_i^T = U\Sigma V^T$ and $\mathscr{L} = \sum_{i=1}^{r_L} \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^T = \tilde{U}\tilde{\Sigma}\tilde{V}^T$. It suffices to show that $\mathcal{R}_X(X + \Delta t\mathscr{L}) = X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L} + \mathcal{O}(\Delta t^2)$. Now, we can take Δt small enough such that the singular values of $\Delta t\mathscr{L}$, $\Delta t\tilde{\sigma}_i$ are all smaller than the singular values of X, σ_i. Then, the contribution from $\Delta t\mathscr{L}$ only comes from the projection of $\tilde{U}$ onto U and $\tilde{V}$ onto V.

$$\begin{aligned} X + \Delta t\mathscr{L} &= \sum_{i=1}^{r} \sigma_i u_i v_i^T + \Delta t \sum_{i=1}^{r} u_i u_i^T \left(\sum_{j=1}^{r_L} \tilde{\sigma}_j \tilde{u}_j \tilde{v}_j^T \right) v_i v_i^T \\ &= \sum_{i=1}^{r} \left(\sigma_i + \Delta t u_i^T \mathscr{L} v_i \right) u_i v_i^T \\ &= U \left(\Sigma + \Delta t U^T \mathscr{L} V \right) V^T \\ &= U\Sigma V + \Delta t U U^T \mathscr{L} V V^T \end{aligned}$$

Indeed, this is equivalent to $X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L}$, just in a different parameterization of the low-rank manifold.

It is shown in [42] that the projective retraction is a second-order retraction. However, the extended projective retraction is not second-order. To be second-order, we would need the Δt^2 component of the retraction to be in the normal space of the low-rank manifold at X. Because we know the (non-extended) protective retraction is second-order, we have that $\mathscr{P}_{\mathscr{M}_r} \left(X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} \right) = X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} + d$, where $d \in \mathcal{N}_X \mathscr{M}_r$ encompasses all of the $\mathcal{O}(\Delta t^2)$ terms. For the extended projective retraction to be second-order, we would need

$$\mathscr{P}_{\mathscr{M}_r} \left(X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} + \Delta t \mathscr{P}^{\perp}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} \right) = X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} + \tilde{d}$$

with $\tilde{d} \in \mathcal{N}_X \mathscr{M}_r$ encompassing all $\mathcal{O}(\Delta t^2)$ terms. This would indeed be the case if $\mathscr{P}_{\mathscr{M}_r}$ were a linear operator; however, $\mathscr{P}_{\mathscr{M}_r}$ is highly nonlinear – if it were linear, our life would be much, much easier and this thesis would be quite short. Due to this nonlinearity, $\tilde{d} \notin \mathcal{N}_X \mathscr{M}_r$. To illustrate this further, consider figure 2-4. To generate this test case, random $U \in \mathbb{R}^{10 \times 9}$, $Z \in \mathbb{R}^{12 \times 9}$, and $\overline{\mathscr{L}} \in \mathbb{R}^{10 \times 12}$ were generated. U was orthonormalized, and we set $X = UZ^T$. Then, for various Δt, we considered $X + \Delta t \overline{\mathscr{L}}$ and performed both projective and extended projective retractions via the truncated SVD. To obtain the $\mathcal{O}(\Delta t^2)$ component of the retraction, which we'll denote d, we set $d = \mathcal{R}_X(X + \Delta t \overline{\mathscr{L}}) - (X + \Delta t \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}})$. Then for each Δt, we

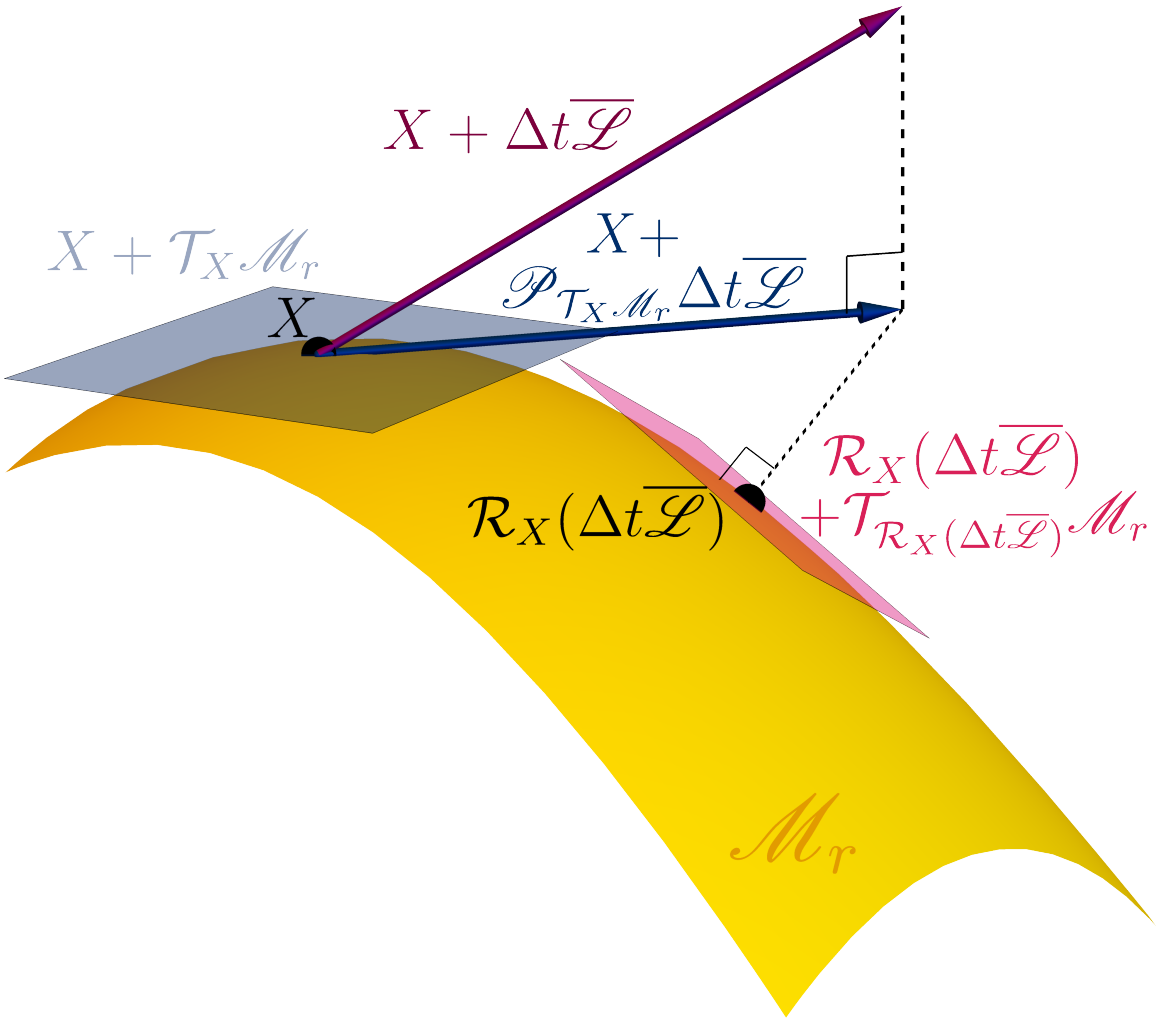


Figure 2-3: This figure shows a projective retraction. After projecting onto the affine tangent space, $X + \Delta t\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ is projected onto the low-rank manifold. Consequently, the line connecting $X + \Delta t\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ and $\mathcal{R}_X(\Delta t\overline{\mathscr{L}})$ is orthogonal to the affine tangent space at $\mathcal{R}_X(\Delta t\overline{\mathscr{L}})$.

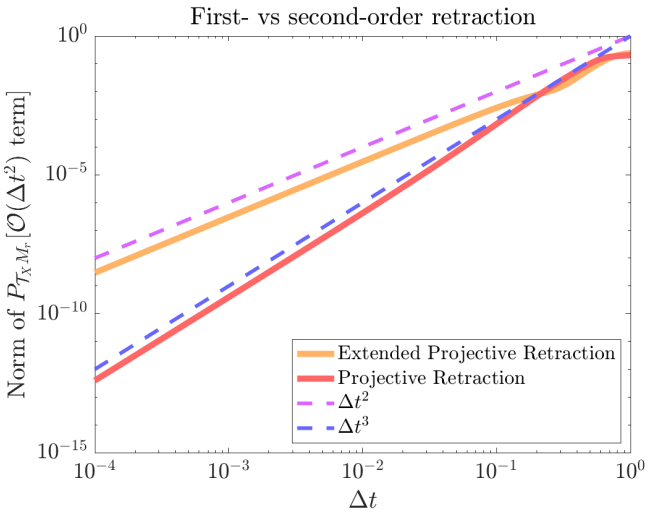


Figure 2-4: The Frobenius norm of the projection of the $\mathcal{O}(\Delta t^2)$ terms of the projective retraction onto the affine tangent space at X decay as $\mathcal{O}(\Delta t^3)$ because it is a second-order retraction. However, the extended projective retraction exhibits $\mathcal{O}(\Delta t^2)$ convergence because the $\mathcal{O}(\Delta t^2)$ has some components in the affine tangent space at X.

projected d onto the tangent space and took the norm of that projection. Hence, we have numerically shown an example where the extended projective retraction is not second-order. So while a second-order retraction matches a geodesic quite well, if we would rather approximate the projection operator, it is not a criterion of great interest.

The downside of the projective retractions is that computing the singular value decomposition can be quite expensive for large problems. Recall we would be taking the SVD of $X_0 + \Delta t \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$, which is assumed to be an $m \times n$ matrix. An efficient implementation of taking this SVD is addressed in [43, p. 1222] and [42, p. 21] using a USV^T matrix parameterization, which reduces the computation to computing the SVD of a $2r \times 2r$ matrix. This is feasible since X_0 and $\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\overline{\mathscr{L}}$ are both rank r, so their sum must be have at most rank $2r$. To see why, consider the following implementation of adding to matrices A and B, both of which we'll assume have r linearly independent columns.

$$A + B = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} I \\ I \end{bmatrix}$$

Above, I is an $r \times r$ identity matrix. Clearly, $\text{rank}\left(\begin{bmatrix} A & B \end{bmatrix}\right) \leq \text{rank}\left(\begin{bmatrix} I & I \end{bmatrix}^T\right) \leq 2r$.

A similar implementation is addressed in [18, p. 610] for a UZ^T matrix parameterization, which reduces the computation to taking the SVD of an $m \times r$ matrix. An iterative approach is also proposed in [18]. Below, we derive a similar technique for the UZ^T matrix parameterization using only the SVD of an $2r \times 2r$ matrix given the current point X_0 and U_0, Z_0 as defined in (2.3).

Algorithm 1: Projective retraction

Input: $U_0 \in \mathcal{V}_{m,r}$, $Z_0 \in \mathbb{R}_*^{n\times r}$, $\dot{U} \in \mathcal{U}_{m,r}$, $\dot{Z} \in \mathbb{R}^{n\times r}$, $\Delta t \in \mathbb{R}$

Output: $U_1 \in \mathcal{V}_{m,r}$, $Z_1 \in \mathbb{R}^{n\times r}$

1 $\begin{bmatrix} Q_1 & R_1 \end{bmatrix} = \texttt{qr}(\Delta t \dot{U}), \qquad \begin{bmatrix} Q_2 & R_2 \end{bmatrix} = \texttt{qr}\left(\begin{bmatrix} Z_0 & \Delta t \dot{Z} \end{bmatrix}\right)$

2 $K = \begin{bmatrix} R_2(:, 1:r)^T + R_2(:, r+1:2r)^T \\ R_1 R_2(:, 1:r)^T \end{bmatrix}$

3 $\tilde{U}\tilde{S}\tilde{V}^T = \texttt{svd}(K)$

4 $\tilde{U} \leftarrow \tilde{U}(:, 1:r), \qquad \tilde{S} \leftarrow \tilde{S}(1:r, 1:r), \qquad \tilde{V} \leftarrow \tilde{V}(:, 1:r)$

5 $U_1 = \begin{bmatrix} U_0 & Q_1 \end{bmatrix} \tilde{U}, \qquad Z_1 = Q_2 \tilde{V} \tilde{S}^T$

Note above that $\tilde{S}^T = \tilde{S}$, so in reality the transpose need not be computed. Furthermore, R_2 is a $2r \times 2r$ matrix, making K a $2r \times 2r$ matrix. We say $Z_1 \in \mathbb{R}^{n\times r}$ instead of $Z_1 \in \mathbb{R}_*^{n\times r}$ because if K has rank less than r, then we cannot guarantee that Z_1 has rank r. However, in practice, this would be a rarity, and $Z_1 \in \mathbb{R}_*^{n\times r}$ in essentially all realistic, non-pathological examples.

From [44, p. 75], the qr decomposition can be computed using the Householder algorithm in $2ab^2 - \frac{2}{3}b^3$ flops for an $a \times b$ matrix. For us, $b = r \ll m, n$, and so the qr decomposition is very cheap. From [44, p. 237], the SVD may be computed via Golub-Kahan bidiagonalization in about $4ab^2 - \frac{4}{3}b^3$ flops, twice that of qr. Note that faster algorithms are available if $a \gg b$ (e.g. Lawson-Hanson-Chan bidiagonalization); the two algorithms can in fact be combined in what's called three-step bidiagonalization. But since we are operating on a $2r \times 2r$ matrix, Golub-Kahan bidiagonalization is suitable, and it's cheap since r is (relatively) small!

Lemma 2.2.2. Algorithm 1 returns $U_1 Z_1^T = \mathscr{P}_{\mathscr{M}_r}(U_0 Z_0^T + \Delta t U_0 \dot{Z}^T + \Delta t \dot{U} Z_0^T)$ with $U_1 \in \mathcal{V}_{m,r}$.

Proof. First, we'll show $U_1 \in \mathcal{V}_{m,r}$.

$$\begin{aligned} U_1 &= \begin{bmatrix} U_0 & Q_1 \end{bmatrix} \tilde{U} \\ \Rightarrow U_1^T U_1 &= \tilde{U}^T \begin{bmatrix} U_0^T \\ Q_1^T \end{bmatrix} \begin{bmatrix} U_0 & Q_1 \end{bmatrix} \tilde{U} \\ &= \tilde{U}^T \begin{bmatrix} U_0^T U_0 & U_0^T Q_1 \\ Q_1^T U_0 & Q_1^T Q_1 \end{bmatrix} \tilde{U} \end{aligned}$$

Now, we have that $U_0^T Q_1 = 0$ since Q_1 is a basis for $\dot{U} \in \mathcal{U}_{m,r} \rightarrow U_0^T \dot{U} = 0$.

$$\begin{aligned} U_1^T U_1 &= \tilde{U}^T \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \tilde{U} \\ &= \tilde{U}^T \tilde{U} = I \end{aligned}$$

Now, we'll show that $U_1 Z_1^T = \mathscr{P}_{\mathscr{M}_r}(U_0 Z_0^T + \Delta t U_0 \dot{Z}^T + \Delta t \dot{U} Z_0^T)$.

$$U_1 Z_1^T = \begin{bmatrix} U_0 & Q_1 \end{bmatrix} \tilde{U} \tilde{S} \tilde{V}^T Q_2^T$$

We've already shown that U_0 and Q_1 are orthogonal. Plus, they're (semi-)orthonormal, so $\begin{bmatrix} U_0 & Q_1 \end{bmatrix}$ is semi-orthonormal. Q_2 is also orthonormal. Then, $\begin{bmatrix} U_0 & Q_1 \end{bmatrix} \tilde{U}$, $\tilde{S}$, and $\tilde{V}^T Q_2^T$ form a valid singular value decomposition, which indeed orthogonally projects onto the low-rank manifold.

$$\begin{aligned} U_1 Z_1^T &= \begin{bmatrix} U_0 & Q_1 \end{bmatrix} \mathscr{P}_{\mathscr{M}_r}(K) Q_2^T \\ &= \mathscr{P}_{\mathscr{M}_r}\left(\begin{bmatrix} U_0 & Q_1 \end{bmatrix} K Q_2^T \right) \\ &= \mathscr{P}_{\mathscr{M}_r}\left(\begin{bmatrix} U_0 & Q_1 \end{bmatrix} \begin{bmatrix} R_2(:, 1:r)^T + R_2(:, r+1:2r)^T \\ R_1 R_2(:, 1:r)^T \end{bmatrix} Q_2^T \right) \\ &= \mathscr{P}_{\mathscr{M}_r}\left(U_0 R_2(:, 1:r)^T Q_2^T + U_0 R_2(:, r+1:2r)^T Q_2^T + Q_1 R_1 R_2(:, 1:r)^T Q_2^T \right) \end{aligned}$$

Note that $Q_2 R_2(:, 1:r) = Z$ and $Q_2 R_2(:, r+1:2r) = \Delta t \dot{Z}$ by construction.

$$U_1 Z_1^T = \mathscr{P}_{\mathscr{M}_r}\left(U_0 Z_0^T + \Delta t U_0 \dot{Z}^T + \Delta t \dot{U} \dot{Z}^T \right)$$

□

Below, we describe a novel retraction by generalizing the retraction above to be an extended retraction. We assume that $\overline{\mathscr{L}}$ can be factored into a form $\overline{\mathscr{L}}_U \overline{\mathscr{L}}_Z^T$ with rank r_L. For differential equations, this is often feasible as the rank of the derivative term may stay low. For example, consider a simple example $\frac{d}{dt}A(t) = A(t)$. We would let $\overline{\mathscr{L}} = A$, and if $A = UZ^T$, then $\overline{\mathscr{L}}_U = U$ and $\overline{\mathscr{L}}_Z = Z$. This example is admittedly trivial, but in more realistic examples the principle holds even if $r_L \gg r$, and we hope that $r_L \ll m, n$.

Algorithm 2: Extended projective retraction

Input: $U_0 \in \mathcal{V}_{m,r}$, $Z_0 \in \mathbb{R}_*^{n\times r}$, $\overline{\mathscr{L}}_U \in \mathbb{R}^{m\times r_L}$, $\overline{\mathscr{L}}_Z \in \mathbb{R}^{n\times r_L}, \Delta t \in \mathbb{R}$

Output: $U_1 \in \mathcal{V}_{m,r}$, $Z_1 \in \mathbb{R}^{n\times r}$

1 $\begin{bmatrix} Q_1 & R_1 \end{bmatrix} = \texttt{qr}\left(\begin{bmatrix} U_0 & \overline{\mathscr{L}}_U \end{bmatrix}\right), \qquad \begin{bmatrix} Q_2 & R_2 \end{bmatrix} = \texttt{qr}\left(\begin{bmatrix} Z_0 & \Delta t\overline{\mathscr{L}}_Z \end{bmatrix}\right)$

2 $K = R_1 R_2^T$

3 $\tilde{U}\tilde{S}\tilde{V}^T = \texttt{svd}(K)$

4 $\tilde{U} \leftarrow \tilde{U}(:, 1:r), \qquad \tilde{S} \leftarrow \tilde{S}(1:r, 1:r), \qquad \tilde{V} \leftarrow \tilde{V}(:, 1:r)$

5 $U_1 = Q_1\tilde{U}, \qquad Z_1 = Q_2\tilde{V}\tilde{S}^T$

Here, K is $(r + r_L) \times (r + r_L)$. So, if $r_L \ll m, n$, the SVD is still cheap.

Lemma 2.2.3. Algorithm 2 returns $U_1 Z_1^T = \mathscr{P}_{\mathscr{M}_r}(U_0 Z_0^T + \Delta t \overline{\mathscr{L}}_U \overline{\mathscr{L}}_Z^T)$ with $U_1 \in \mathcal{V}_{m,r}$.

Proof. First, we'll show $U_1 \in \mathcal{V}_{m,r}$.

$$\begin{aligned} U_1 &= Q_1\tilde{U} \\ \Rightarrow U_1^T U_1 &= \tilde{U}^T Q_1^T Q_1 \tilde{U} \\ &= \tilde{U}^T\tilde{U} = I \end{aligned}$$

Now, we'll show that $U_1 Z_1^T = \mathscr{P}_{\mathscr{M}_r}(U_0 Z_0^T + \Delta t \overline{\mathscr{L}}_U \overline{\mathscr{L}}_Z^T)$.

$$U_1 Z_1^T = Q_1 \tilde{U}\tilde{S}\tilde{V}^T Q_2^T$$

Q_1 and Q_2 are orthonormal, so $Q_1\tilde{U}$, $\tilde{S}$, and $\tilde{V}^T Q_2^T$ form a valid singular value de-

composition, which orthogonally projects onto the low-rank manifold.

$$\begin{aligned} U_1 Z_1^T &= Q_1 \tilde{U} \mathscr{P}_{\mathscr{M}_r}(K) Q_2^T \\ &= \mathscr{P}_{\mathscr{M}_r}\left(Q_1 K Q_2^T\right) \\ &= \mathscr{P}_{\mathscr{M}_r}\left(Q_1 R_1 R_2^T Q_2^T\right) \\ &= \mathscr{P}_{\mathscr{M}_r}\left(\begin{bmatrix} U_0 & \overline{\mathscr{L}}_U \end{bmatrix} \begin{bmatrix} Z_0^T \\ \Delta t \overline{\mathscr{L}}_Z^T \end{bmatrix}\right) \\ &= \mathscr{P}_{\mathscr{M}_r}\left(U_0 Z_0^T + \Delta t \overline{\mathscr{L}}_U \overline{\mathscr{L}}_Z^T\right) \end{aligned}$$

□

Though this retraction is essentially ideal, it can be quite expensive if r_L is large. So, in the next section, we discuss a perturbative method that will approximate the extended projective method.

2.3 Perturbative retractions

In this section, we develop new retractions that exhibit high-order convergence in Δt. But to what are we converging? We cannot hope to converge to the full-rank solution; at each time step, there is a local $\mathcal{O}(\Delta t)$ error (and hence a global $\mathcal{O}(1)$ error) in the normal space which we cannot reduce unless we increase the rank of our solution. But if the full-rank solution stays close to the low-rank manifold, the error between the best low-rank approximation and the full-rank solution may be bounded [19, 20]. The goal, then, is to approximate and to converge to the best low-rank approximation at each time step given by $\mathscr{P}_{\mathscr{M}_r}(X_0 + \Delta t \overline{\mathscr{L}})$; the Euler forward method discussed in previous sections is a first-order approximation by projecting onto the affine tangent space. Indeed, [45] shows that projection onto the tangent space is the first-order perturbation expansion of the truncated SVD. Here we develop higher-order approximations, which we can interpret as projecting onto a higher-order polynomial approximation of the low-rank manifold.

To proceed, recall (1.16). Geometrically, this says that the residual $\mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \overline{\mathscr{L}} - \overline{\mathscr{L}}$ must be orthogonal to the affine tangent space at $X_0 = U Z^T$. This is actually not what we would like to enforce for two reasons. First, this condition says nothing about the retraction error $\dot{U}\dot{Z}^T$; it's completely ignored. Second, we want our residual to be orthogonal to the affine tangent space at X_1, not at X_0. Incorporating these new ideas into the original formulation, we seek to find $\dot{U} \in \mathcal{U}_{m,r}, \dot{Z} \in \mathbb{R}^{n \times r}$ such that the

following holds for all $\delta_U \in \mathcal{U}_{m,r}, \delta_Z \in \mathbb{R}^{n\times r}$.

$$\begin{aligned}&\langle (UZ^T + \Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T) - (UZ^T + \Delta t\overline{\mathscr{L}}),\\ &\qquad\qquad (U + \Delta t\dot{U})\delta_Z^T + \delta_U(Z + \Delta t\dot{Z})^T\rangle = 0\\ \Leftrightarrow\ &\langle \dot{U}Z^T + U\dot{Z}^T + \Delta t\dot{U}\dot{Z}^T - \overline{\mathscr{L}}, (U + \Delta t\dot{U})\delta_Z^T + \delta_U(Z + \Delta t\dot{Z})^T\rangle = 0 \qquad (2.9)\end{aligned}$$

For graphical intuition, contrast figures 2-1 and 2-2. We really want the residual orthogonal to the affine tangent space in red rather than blue. So, the condition can be read as minimizing the residual between the ideal new point $UZ^T + \Delta t\overline{\mathscr{L}}$ and the retracted point $(UZ^T + \Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T)$ at the affine tangent space defined at the new retracted point.

Before we proceed, we must address that we have defined $\delta_U \in \mathcal{U}_{m,r}$, which is in the space of matrices that are orthogonal to U. But, we are using δ_U to define an affine tangent space at $U + \dot{U}$, so do we need to insist that $\delta_U^T(U + \dot{U}) = 0$ instead of $\delta_U^T U = 0$? The answer is no because we are able to paramaterize the affine tangent space at $(U + \dot{U})(Z + \dot{Z})^T$ with matrices orthogonal to U.

Lemma 2.3.1. Let $\tilde{X} = (U + \Delta t\dot{U})(Z + \Delta t\dot{Z})^T$ with $\dot{U} \in \mathcal{U}_{m,r}, \dot{Z} \in \mathbb{R}^{n\times r}$. Any $\delta_X \in \mathcal{T}_{\tilde{X}}\mathscr{M}_r$ can be expressed as

$$\delta_X = \delta_U(Z + \Delta t\dot{Z})^T + (U + \Delta t\dot{U})\delta_Z^T,$$

where

$$\delta_U = (I - (U + \Delta t\dot{U})U^T)\Delta_U, \qquad \delta_Z = \Delta_Z + (Z + \Delta t\dot{Z})\Delta_U^T U$$

for some $\Delta_U \in \mathcal{V}_{m,r}$ and $\Delta_Z \in \mathbb{R}_*^{n\times r}$. As such, $\delta_U \in \mathcal{U}_{m,r}$.

Proof. Note that the first statement that any $\delta_X \in \mathcal{T}_{\tilde{X}}\mathscr{M}_r$ can be expressed as $\delta_U(Z + \Delta t\dot{Z})^T + (U + \Delta t\dot{U})\delta_Z^T$ would be nothing new if we insisted that $\delta_U^T(U + \dot{U}) = 0$. The fact that a tangent vector may be written as above is addressed in previous sections of this thesis. We are just using a new point $(U + \Delta t\dot{U})(Z + \Delta t\dot{Z})^T$ instead of UZ^T. What is new is insisting that $\delta_U^T U = 0$ instead of $\delta_U^T(U + \Delta t\dot{U}) = 0$. In an argument similar to that of [20, p. 517], we will substitute in our expressions for δ_U and δ_Z into

our expression for δ_X.

$$\begin{aligned}
\delta_X &= \delta_U(Z + \Delta t\dot{Z})^T + (U + \Delta t\dot{U})\delta_Z^T \\
&= (I - (U + \Delta t\dot{U})U^T)\Delta_U(Z + \Delta t\dot{Z})^T + (U + \Delta t\dot{U})(\Delta_Z^T + U^T\Delta_U(Z + \Delta t\dot{Z})^T) \\
&= \Delta_U(Z + \Delta t\dot{Z})^T - \cancel{(U + \Delta t\dot{U})U^T\Delta_U(Z + \Delta t\dot{Z})^T} + \\
&\qquad (U + \Delta t\dot{U})\Delta_Z^T + \cancel{(U + \Delta t\dot{U})U^T\Delta_U(Z + \Delta t\dot{Z})^T} \\
&= \Delta_U(Z + \Delta t\dot{Z})^T + (U + \Delta t\dot{U})\Delta_Z^T
\end{aligned}$$

Since we have allowed Δ_U and Δ_Z to be free, we have shown that δ_X can represent any tangent vector in the above parameterization. What remains to show is that $\delta_U \in \mathcal{U}_{m,r}$.

$$\begin{aligned}
U^T\delta_U &= U^T(I - (U + \Delta t\dot{U})U^T)\Delta_U \\
&= (U^T - (I + 0)U^T)\Delta_U = 0
\end{aligned}$$

□

Now, we have justified our parameterization of the affine tangent space at $(U + \Delta t\dot{U})(Z + \Delta t\dot{Z})^T$. For the purpose of completeness, we'll go a step further and define the mapping from $\delta_X \to (\delta_U, \delta_Z)$ since we already have the mapping $(\delta_U, \delta_Z) \to \delta_X$.

Lemma 2.3.2. Given $\tilde{X} = UZ^T \in \mathscr{M}_r$ with $U \in \mathcal{V}_{m,r}$, $Z \in \mathbb{R}_*^{n\times r}$, $\dot{U} \in \mathcal{U}_{m,r}$, and $\dot{Z} \in \mathbb{R}^{n\times r}$ such that $Z + \Delta t\dot{Z} \in \mathbb{R}_*^{n\times r}$, and $\delta_X \in \mathcal{T}_{\tilde{X}}\mathscr{M}_r$, $(\delta_U, \delta_Z) \in \mathcal{U}_{m,r} \times \mathbb{R}^{n\times r}$ are given as

$$\delta_U = (I - (U + \Delta t\dot{U})U^T)\delta_X(Z + \Delta t\dot{Z})\left[(Z + \Delta t\dot{Z})^T(Z + \Delta t\dot{Z})\right]^{-1}, \qquad \delta_Z = \delta_X^T U$$

Proof. First, we'll find δ_Z by taking the left matrix product of δ_X with U^T.

$$U^T\delta_X = U^T(U + \Delta t\dot{U})\delta_Z^T + U^T\delta_U(Z + \Delta t\dot{Z})^T = \delta_Z^T$$

Next, to find δ_U, we'll take the the left matrix of product of δ_X with $(I-(U+\Delta t\dot{U})U^T)$.

$$\begin{aligned}(I-(U+\Delta t\dot{U})U^T)\delta_X &= (I-(U+\Delta t\dot{U})U^T)(U+\Delta t\dot{U})\delta_Z^T \\ &\quad + (I-(U+\Delta t\dot{U})U^T)\delta_U(Z+\Delta t\dot{Z})^T \\ &= (U+\Delta t\dot{U})\delta_Z^T - (U+\Delta t\dot{U})U^T\cancelto{I}{(U+\Delta t\dot{U})}\delta_Z^T + \delta_U(Z+\Delta t\dot{Z})^T \\ &\quad - (U+\Delta t\dot{U})\cancelto{0}{U^T\delta_U}(Z+\Delta t\dot{Z})^T \\ &= \delta_U(Z+\Delta t\dot{Z})^T\end{aligned}$$

After taking the Moore-Penrose inverse, we have the following.

$$\delta_U = (I-(U+\Delta t\dot{U})U^T)\delta_X(Z+\Delta t\dot{Z})\left[(Z+\Delta t\dot{Z})^T(Z+\Delta t\dot{Z})\right]^{-1}$$

□

Before proceeding, one further step of justification is needed. Can we represent any point $Y \in \mathscr{M}_r$ in the form $X_1 = (U+\Delta t\dot{U})(Z+\Delta t\dot{Z})^T$ for $\dot{U} \in \mathcal{U}_{m,r}$, $\dot{Z} \in \mathbb{R}^{n\times r}$? If we cannot, then perhaps this retraction will not be accurate for some Y. Essentially, we would like our retraction to be surjective, or onto, in $\mathscr{M}_r$ so that we can cover the whole manifold. To answer this question, consider $Y = \tilde{U}\tilde{Z}^T$ for some $\tilde{U} \in \mathcal{V}_{m,r}$, $\tilde{Z} \in \mathbb{R}_*^{n\times r}$. If $\dot{U}$ were not restricted to $\mathcal{U}_{m,r}$, we could simply define $\dot{U} = (\tilde{U}-U)/\Delta t$ and $\dot{Z} = (\tilde{Z}-Z)/\Delta t$, and then $X_1 = Y$. But, of course, life is not that easy. We'll try to construct a bijective mapping (therefore implying a surjective mapping) between $(\dot{U},\dot{Z})$ and $(\tilde{U},\tilde{Z})$ given (U,Z). If such a mapping exists, then there will always be one X_1 (which is fully determined by $(\dot{U},\dot{Z})$ and (U,Z)) for one Y (which is fully determined by (U_1,Z_1)), and vice versa.

First, we'll postulate a mapping $(\dot{U},\dot{Z}) \to (\tilde{U},\tilde{Z})$ defined below.

$$(U+\Delta t\dot{U})(Z+\Delta t\dot{Z})^T = UZ^T + \Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T = \tilde{U}\tilde{Z}^T$$

The question then, is can we find a mapping $(\tilde{U},\tilde{Z}) \to (\dot{U},\dot{Z})$? First consider left-multiplying the equation above by U^T.

$$\begin{aligned}Z^T + \Delta t\dot{Z}^T &= U^T\tilde{U}\tilde{Z}^T \\ \Rightarrow \dot{Z} &= \frac{1}{\Delta t}\left(\tilde{Z}\tilde{U}^TU - Z\right)\end{aligned} \tag{2.10}$$

Now, left-multiply the same equation by $\mathscr{P}_U^\perp$.

$$\Delta t\dot{U}(Z + \Delta t\dot{Z})^T = \mathscr{P}_U^\perp \tilde{U}\tilde{Z}^T$$

Now, we substitute in (2.10).

$$\begin{aligned} \Delta t\dot{U}U^T\tilde{U}\tilde{Z}^T &= \mathscr{P}_U^\perp \tilde{U}\tilde{Z}^T \\ \Rightarrow \dot{U} &= \frac{1}{\Delta t}\mathscr{P}_U^\perp \tilde{U}\tilde{Z}^T\tilde{Z}\tilde{U}^T U\left(U^T\tilde{U}\tilde{Z}^T\tilde{Z}\tilde{U}^T U\right)^{-1} \end{aligned} \tag{2.11}$$

In (2.11), we have assumed that $U^T\tilde{U}\tilde{Z}^T\tilde{Z}\tilde{U}^T U$ is invertible. We can dig deeper into this assumption: we've already assumed $\tilde{Z} \in \mathbb{R}_*^{n\times r}$, which implies that $\tilde{Z}^T\tilde{Z}$ is invertible. Since the product of two square invertible matrices is also invertible, we only need to assume that $U^T\tilde{U}$ is invertible. If this is the case, we can simplify (2.11).

$$\dot{U} = \frac{1}{\Delta t}\mathscr{P}_U^\perp \tilde{U}\left(U^T\tilde{U}\right)^{-1} \tag{2.12}$$

Unfortunately, this is not always true; just pick columns of $\tilde{U}$ that are orthogonal to the columns of U for a counterexample. So, we cannot create a bijective mapping for all Y. However, we can create a bijective mapping for a set $\tilde{\mathscr{M}}_r(X = UZ^T) \equiv \left\{Y = \tilde{U}\tilde{Z}^T \in \mathscr{M}_r : \text{rank}(U^T\tilde{U}) = r\right\}$.

Lemma 2.3.3. For any $X = UZ^T \in \mathscr{M}_r$ with $m \geq 2r$, $\tilde{\mathscr{M}}_r(X)$ is dense in $\mathscr{M}_r$.

Proof. Consider a point $Y = \tilde{U}\tilde{Z}^T$ in $\mathscr{M}_r$. If $U^T\tilde{U}$ is invertible, then it is in $\tilde{\mathscr{M}}_r(X)$. Otherwise, $\text{rank}(U^T\tilde{U}) < r$ and $Y \notin \tilde{\mathscr{M}}_r(X)$. But, consider a point $\hat{Y} = \sqrt{1-\varepsilon_0^2}Y + \varepsilon_0 UQ\tilde{Z}^T$ for some $0 < \varepsilon_0 < 1$ and some orthonormal $r \times r$ Q with columns orthogonal to the columns of $U^T\tilde{U}$. Defining such a Q is possible assuming $m \geq 2r$ by the rank-nullity theorem, which, in practice is always the case since we've assumed $m \gg r$. Note the following.

$$\begin{aligned} ||\hat{Y} - Y||^2 &= \varepsilon_0^2||UQ\tilde{Z}^T||^2 + \left(1 - \sqrt{1-\varepsilon_0^2}\right)^2 ||\tilde{U}\tilde{Z}^T|| \\ &= 2\left(1 - \sqrt{1-\varepsilon_0^2}\right)||\tilde{Z}||^2 \end{aligned}$$

The last equality is because U, Q, and $\tilde{U}$ are semi-orthonormal, and we've chosen a norm such as the L^2 or Frobenius norm that is unitarily invariant. In the limit where

$\varepsilon_0 \to 0$, the expression simplifies.

$$\lim_{\varepsilon_0 \to 0} ||\hat{Y} - Y||^2 = \varepsilon_0^2 ||\tilde{Z}||^2$$

Note that $2\left(1 - \sqrt{1-\varepsilon_0^2}\right) \leq 2\varepsilon_0^2$ for $|\varepsilon_0| \leq 1$, hence we can use that upper bound without the limit. So, we can make the norm as small as we want by shrinking ε_0 to be arbitrarily small. If we define an epsilon-ball $\mathcal{B}_\varepsilon(Y) = \{A : ||A - Y|| < \varepsilon\}$, we have that $\hat{Y} \in \mathcal{B}_\varepsilon(Y) \quad \forall \varepsilon > 0$ so long as we choose $\varepsilon_0 < \frac{\sqrt{2}\varepsilon}{2||\tilde{Z}||}$. We can write write $\hat{Y}$ in the following form.

$$\hat{Y} = \left(\sqrt{1-\varepsilon_0^2}\tilde{U} + \varepsilon_0 UQ\right)\tilde{Z}^T$$

Define $\hat{U} = \sqrt{1-\varepsilon_0^2}\tilde{U} + \varepsilon_0 UQ$. Is $\hat{Y} \in \tilde{\mathscr{M}}_r(X)$? First, we'll ensure it's in $\mathscr{M}_r$. $\hat{U}$ and $\tilde{Z}$ are dimensionally correct, but is $\hat{U}$ semi-orthonormal?

$$\begin{aligned}\hat{U}^T\hat{U} &= \left(1-\varepsilon_0^2\right)\cancelto{I}{\tilde{U}^T\tilde{U}} + \varepsilon_0^2\cancelto{I}{Q^TU^TUQ} + \varepsilon_0\sqrt{1-\varepsilon_0^2}\cancelto{0}{\tilde{U}^TUQ} + \varepsilon_0\sqrt{1-\varepsilon_0^2}\cancelto{0}{Q^TU^T\tilde{U}} \\ &= I\end{aligned}$$

So, $\hat{Y} \in \mathscr{M}_r$. What remains to check is if $\text{rank}(U^T\hat{U}) = r$.

$$U^T\hat{U} = \sqrt{1-\varepsilon_0^2}U^T\tilde{U} + \varepsilon_0 Q$$

Let $\lambda_j(\bullet)$ denote the jth eigenvalue of $\bullet$. Recall that the eigenvalues of an orthonormal matrix are on the unit circle in the complex plane. By the spectral mapping theorem, we have the following

$$\lambda_j(U^T\hat{U}) = \varepsilon_0 e^{i\theta_j} + \sqrt{1-\varepsilon_0^2}\lambda_j(U^T\tilde{U})$$

for some $\theta_j \in \mathbb{R}$. Since we've assumed that $Y \notin \tilde{\mathscr{M}}_r(X)$, we know there exists at least one $\lambda_j(U^T\tilde{U}) = 0$. But, by adding some small ε_0 to the zero eigenvalue, we will have that for all j $\lambda_j(U^T\hat{U}) \neq 0$. One may ask, how do we know that we aren't setting another eigenvalue to zero by adding ε_0? Well if we are, just take $\varepsilon_0 = \varepsilon_0/2$, and this will rectify the situation. Since there are a finite number of eigenvalues, we can ensure that there exists some $\varepsilon_0 > 0$ such that all of the eigenvalues of $U^T\hat{U}$ are nonzero, and hence $U^T\hat{U}$ is invertible. Therefore, $\hat{Y} \in \tilde{\mathscr{M}}_r(X)$, and we have shown that $\forall Y \in \mathscr{M}_r$, there exists a $\hat{Y}$ in any neighborhood of Y such that $\hat{Y} \in \tilde{\mathscr{M}}_r(X)$. $\square$

With that lemma, we can say that there is either a bijective mapping for any point $Y \in \mathscr{M}_r$ or some point arbitrarily close to Y. Numerically, this is good enough – as long as we can get arbitrarily close to Y, we are happy. Putting this all together, we have the following theorem.

Theorem 2.3.1. Given a point $X = UZ^T \in \mathscr{M}_r$ with $U \in \mathcal{V}_{m,r}$, $Z \in \mathbb{R}_*^{n\times r}$, $m \geq 2r$, and a point $Y \in \mathscr{M}_r$, for any $\varepsilon > 0$, one can always write $\hat{Y} = \tilde{U}\tilde{Z}^T = (U + \Delta t\dot{U})(Z + \Delta t\dot{Z})^T$ with $\tilde{U} \in \mathcal{V}_{m,r}$ and $\tilde{Z} \in \mathbb{R}_*^{n\times r}$ for some $\dot{U} \in \mathcal{U}_{m,r}$ and $\dot{Z} \in \mathbb{R}^{n\times r}$ such that $||\hat{Y} - Y|| < \varepsilon$.

Proof. From lemma 2.3.3, we know that we can choose $\hat{Y} \in \tilde{\mathscr{M}}_r(X)$ arbitrarily close to $Y \in \mathscr{M}_r$. Let the following expressions hold.

$$\dot{U} = \frac{1}{\Delta t}\mathscr{P}_U^{\perp}\tilde{U}\left(U^T\tilde{U}\right)^{-1}$$
$$\dot{Z} = \frac{1}{\Delta t}(\tilde{Z}\tilde{U}^TU - Z)$$

Clearly, $\dot{U} \in \mathcal{U}_{m,r}$. Now, evaluate $\hat{Y}$.

$$\begin{aligned}
\hat{Y} &= \left(U + \mathscr{P}_U^{\perp}\tilde{U}\left(U^T\tilde{U}\right)^{-1}\right)\left(Z + \tilde{Z}\tilde{U}^TU - Z\right)^T \\
&= UU^T\tilde{U}\tilde{Z}^T + (I - UU^T)\tilde{U}\cancel{\left(U^T\tilde{U}\right)^{-1}}\cancel{U^T\tilde{U}}\tilde{Z}^T \quad (I) \\
&= UU^T\tilde{U}\tilde{Z}^T + (I - UU^T)\tilde{U}\tilde{Z}^T \\
&= \tilde{U}\tilde{Z}^T
\end{aligned}$$

□

So, we have shown that our retraction may perform arbitrarily well in that it can access a dense set of the whole manifold. At this point, we return to the main problem of finding what $\dot{U}$ and $\dot{Z}$ should be to minimize a residual. The following theorem digs deeper into that idea.

Theorem 2.3.2. Given $U \in \mathcal{V}_{m,r}$, $Z \in \mathbb{R}_*^{n\times r}$, and $\mathscr{L} \in \mathbb{R}^{m\times n}$, the solution to (2.9) is given by $\dot{U} \in \mathcal{U}_{m,r}$ and $\dot{Z} \in \mathbb{R}^{n\times r}$ that satisfy the following nonlinear equations.

$$(U + \Delta t\dot{U})^T\Delta t\overline{\mathscr{L}} = \Delta t^2\dot{U}^T\dot{U}(Z + \Delta t\dot{Z})^T + \Delta t\dot{Z}^T \tag{2.13}$$
$$\mathscr{P}_U^{\perp}\Delta t\overline{\mathscr{L}}(Z + \Delta t\dot{Z}) = \Delta t\dot{U}(Z + \Delta t\dot{Z})^T(Z + \Delta t\dot{Z}) \tag{2.14}$$

Proof. Because (2.9) is valid $\forall \delta_U$, we set δ_U equal to zero first. We also scale the first argument by Δt to more clearly show the order of each term.

$$\begin{aligned}
&\langle \Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\overline{\mathscr{L}}, (U + \Delta t\dot{U})\delta_Z^T\rangle = 0 \\
\Leftrightarrow \operatorname{Tr}&\left(\delta_Z(U + \Delta t\dot{U})^T\left[\Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\overline{\mathscr{L}}\right]\right) = 0
\end{aligned}$$

We use a similar argument as in the proof of theorem 1.4.1, which is that because δ_Z can be anything, we can remove the argument from the trace and set it equal to zero.

$$(U + \Delta t\dot{U})^T\left[\Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\overline{\mathscr{L}}\right] = 0$$

So, the first final equation is as follows.

$$(U + \Delta t\dot{U})^T\Delta t\overline{\mathscr{L}} = \Delta t^2\dot{U}^T\dot{U}(Z + \Delta t\dot{Z})^T + \Delta t\dot{Z}^T$$

This equation is nonlinear in $\dot{U}, \dot{Z}$. In principle, it can be solved iteratively, but, as we shall see, perturbation theory also works.

Now, we return to (2.9) and let $\delta_Z = 0$. Furthermore, since $\delta_U \in \mathcal{U}_{m,r}$, let $\delta_U = \mathscr{P}_U^\perp\delta_Y$ for $\delta_Y \in \mathbb{R}^{n\times r}$.

$$\begin{aligned}
&\langle \Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\mathscr{L}, \delta_U(Z + \Delta t\dot{Z})^T\rangle = 0 \\
\Leftrightarrow \operatorname{Tr}&\left(\delta_Y^T\mathscr{P}_U^\perp\left[\Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\overline{\mathscr{L}}\right](Z + \Delta t\dot{Z})\right) = 0 \\
&\Leftrightarrow \mathscr{P}_U^\perp\left[\Delta t\dot{U}Z^T + \Delta tU\dot{Z}^T + \Delta t^2\dot{U}\dot{Z}^T - \Delta t\overline{\mathscr{L}}\right](Z + \Delta t\dot{Z}) = 0
\end{aligned}$$

Our second final equation is as follows.

$$\mathscr{P}_U^\perp\Delta t\overline{\mathscr{L}}(Z + \Delta t\dot{Z}) = \Delta t\dot{U}(Z + \Delta t\dot{Z})^T(Z + \Delta t\dot{Z})$$

□

The goal, henceforth, is to solve (2.13) and (2.14). We'll assume that $\overline{\mathscr{L}}, U, Z \sim \mathcal{O}(1)$, and we'll let $\dot{U}$ and $\dot{Z}$ take the following forms.

$$\Delta t\dot{U} = \sum_{i=0}^{\infty}\Delta t^i\dot{\mathcal{U}}_i, \qquad \Delta t\dot{Z} = \sum_{i=0}^{\infty}\Delta t^i\dot{\mathcal{Z}}_i$$

By including $i = 0$ in our summation, we allow for there to be a zeroth-order correction to U and Z if necessary. We also assume that $\dot{\mathcal{U}}_i, \dot{\mathcal{Z}}_i \sim \mathcal{O}(1)\ \forall i$. Now, we'll plug these expressions into (2.13) and (2.14).

$$\begin{aligned}\left(U + \sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{U}}_i\right)^T \Delta t\overline{\mathscr{L}} = \\ \left(\sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{U}}_i\right)^T \left(\sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{U}}_i\right) \left(Z + \sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{Z}}_i\right)^T + \left(\sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{Z}}_i\right)^T\end{aligned} \tag{2.15}$$

$$\begin{aligned}\mathscr{P}_U^{\perp} \Delta t\overline{\mathscr{L}} \left(Z + \sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{Z}}_i\right) = \\ \left(\sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{U}}_i\right) \left(Z + \sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{Z}}_i\right)^T \left(Z + \sum_{i=0}^{\infty} \Delta t^i \dot{\mathcal{Z}}_i\right)\end{aligned} \tag{2.16}$$

Theorem 2.3.3. Given $U \in \mathcal{V}_{m,r}$, $Z \in \mathbb{R}_*^{n \times r}$, $\mathscr{L} \in \mathbb{R}^{m \times n}$, and $\Delta t \in \mathbb{R}$ the zeroth through fourth order solutions to (2.15) and (2.16) are given by the following linear equations.

$$\dot{\mathcal{U}}_0 = 0 \tag{2.17}$$

$$\dot{\mathcal{Z}}_0 = 0 \tag{2.18}$$

$$\dot{\mathcal{U}}_1 = \mathscr{P}_U^{\perp} \overline{\mathscr{L}} Z \left(Z^T Z\right)^{-1} \tag{2.19}$$

$$\dot{\mathcal{Z}}_1 = \overline{\mathscr{L}}^T U \tag{2.20}$$

$$\dot{\mathcal{U}}_2 = \left[\mathscr{P}_U^{\perp} \overline{\mathscr{L}} \dot{\mathcal{Z}}_1 - \dot{\mathcal{U}}_1 (Z^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z)\right] \left(Z^T Z\right)^{-1} \tag{2.21}$$

$$\dot{\mathcal{Z}}_2 = \left(\overline{\mathscr{L}}^T - Z \dot{\mathcal{U}}_1^T\right) \dot{\mathcal{U}}_1 \tag{2.22}$$

$$\begin{aligned}\dot{\mathcal{U}}_3 = \Big[\mathscr{P}_U^{\perp} \overline{\mathscr{L}} \dot{\mathcal{Z}}_2 - \dot{\mathcal{U}}_2 (Z^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z) - \\ \dot{\mathcal{U}}_1 (Z^T \dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{Z}}_1^T \dot{\mathcal{Z}}_1)\Big] \left(Z^T Z\right)^{-1}\end{aligned} \tag{2.23}$$

$$\dot{\mathcal{Z}}_3 = \overline{\mathscr{L}}^T \dot{\mathcal{U}}_2 - Z(\dot{\mathcal{U}}_1^T \dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_2^T \dot{\mathcal{U}}_1) - \dot{\mathcal{Z}}_1 \dot{\mathcal{U}}_1^T \dot{\mathcal{U}}_1 \tag{2.24}$$

$$\begin{aligned}\dot{\mathcal{U}}_4 = \Big[\mathscr{P}_U^{\perp} \overline{\mathscr{L}} \dot{\mathcal{Z}}_3 - \dot{\mathcal{U}}_3 \left(Z^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z\right) - \dot{\mathcal{U}}_2 \left(Z^T \dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{Z}}_1^T \dot{\mathcal{Z}}_1\right) \\ - \dot{\mathcal{U}}_1 \left(Z^T \dot{\mathcal{Z}}_3 + \dot{\mathcal{Z}}_3^T Z + \dot{\mathcal{Z}}_2^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T \dot{\mathcal{Z}}_2\right)\Big] \left(Z^T Z\right)^{-1}\end{aligned} \tag{2.25}$$

$$\begin{aligned}\dot{\mathcal{Z}}_4 = \overline{\mathscr{L}}^T \dot{\mathcal{U}}_3 - Z \left(\dot{\mathcal{U}}_1^T \dot{\mathcal{U}}_3 + \dot{\mathcal{U}}_2^T \dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_3^T \dot{\mathcal{U}}_1\right) - \\ \dot{\mathcal{Z}}_1 \left(\dot{\mathcal{U}}_2^T \dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1 \dot{\mathcal{U}}_2\right) - \dot{\mathcal{Z}}_2 \dot{\mathcal{U}}_1^T \dot{\mathcal{U}}_1\end{aligned} \tag{2.26}$$

Proof. First, we'll only consider the zeroth order terms of Δt. The left-hand sides of (2.15) and (2.16) go to zero.

$$\begin{aligned} 0 &= \dot{\mathcal{U}}_0^T \dot{\mathcal{U}}_0 Z^T + \dot{\mathcal{Z}}_0^T + \dot{\mathcal{U}}_0^T \dot{\mathcal{U}}_0 \dot{\mathcal{Z}}_0^T \\ 0 &= \dot{\mathcal{U}}_0 (Z + \dot{\mathcal{Z}})^T (Z + \dot{\mathcal{Z}}_0) \end{aligned}$$

The solution to which is $\dot{\mathcal{U}}_0 = \dot{\mathcal{Z}}_0 = 0$.

Now, we proceed to the first-order correction. We start with (2.15).

$$\begin{aligned} U^T \overline{\mathscr{L}} &= \dot{\mathcal{Z}}_1^T \\ \Rightarrow \dot{\mathcal{Z}}_1 &= \overline{\mathscr{L}}^T U \end{aligned}$$

Now, we deal with (2.16).

$$\begin{aligned} \mathscr{P}_U^\perp \overline{\mathscr{L}} Z &= \dot{\mathcal{U}}_1 Z^T Z \\ \Rightarrow \dot{\mathcal{U}}_1 &= \mathscr{P}_U^\perp \overline{\mathscr{L}} Z \left(Z^T Z \right)^{-1} \end{aligned}$$

With that, we have recovered the original DO projection onto the tangent space from theorem 1.4.1! It is now clear that this is a first-order correction to the nonlinear solution.

We now proceed to a second-order correction with the same procedure.

$$\begin{aligned} \dot{\mathcal{U}}_1^T \overline{\mathscr{L}} &= \dot{\mathcal{U}}_1^T \dot{\mathcal{U}}_1 Z^T + \dot{\mathcal{Z}}_2^T \\ \Rightarrow \dot{\mathcal{Z}}_2 &= \left(\overline{\mathscr{L}}^T - Z \dot{\mathcal{U}}_1^T \right) \dot{\mathcal{U}}_1 \end{aligned}$$

$$\begin{aligned} \mathscr{P}_U^\perp \overline{\mathscr{L}} \dot{\mathcal{Z}}_1 &= \dot{\mathcal{U}}_2 Z^T Z + \dot{\mathcal{U}}_1 Z^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T Z \\ \Rightarrow \dot{\mathcal{U}}_2 &= \left[\mathscr{P}_U^\perp \overline{\mathscr{L}} \dot{\mathcal{Z}}_1 - \dot{\mathcal{U}}_1 (Z^T \dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z) \right] \left(Z^T Z \right)^{-1} \end{aligned}$$

We now have a second-order correction, which goes beyond the current methodology. Note that $\dot{\mathcal{U}}_1$ and $\dot{\mathcal{Z}}_1$ both are functions of $\mathscr{L}$, so $\dot{\mathcal{U}}_2$ is a second-order polynomial of $\mathscr{L}$, and the same is true of $\dot{\mathcal{Z}}_2$. This allows for the interpretation that we projecting onto a quadratic approximation of $\mathscr{M}_r$. It's also worth it to take a moment to appreciate how auspicious it is that $\dot{\mathcal{U}}_2$ and $\dot{\mathcal{Z}}_2$ can be solved for explicitly; the only nonlinear terms appear as lower degrees of $\dot{U}$ and $\dot{Z}$. Furthermore, $\dot{\mathcal{U}}_2$ is orthogonal to U since the first term in (2.21) is multiplied by $\mathscr{P}_U^\perp$ and the second term is mul-

tiplied by $\dot{\mathcal{U}}_1$, which is also orthogonal to U. These observations will continue to be true for arbitrarily high orders of approximation. Below, we'll derive the third and fourth order corrections, but we note that this process can easily be done indefinitely.

$$\begin{aligned}\dot{\mathcal{U}}_2^T\overline{\mathscr{L}} &= \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1^T Z^T + \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_2 Z^T + \dot{\mathcal{Z}}_3^T = \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1\dot{\mathcal{Z}}_1^T \\ \Rightarrow \dot{\mathcal{Z}}_3 &= \overline{\mathscr{L}}^T\dot{\mathcal{U}}_2 - Z(\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1) - \dot{\mathcal{Z}}_1\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1\end{aligned}$$

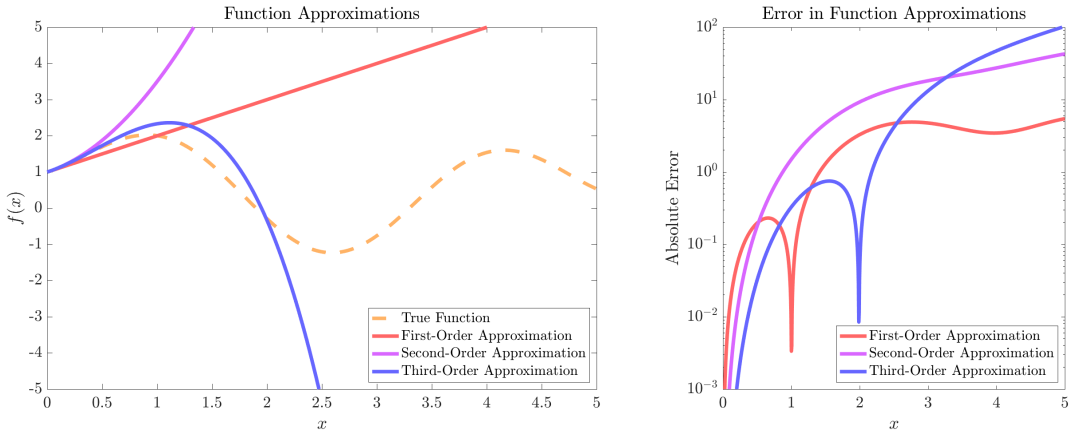
$$\begin{aligned}\mathscr{P}_U^\perp\overline{\mathscr{L}}\dot{\mathcal{Z}}_2 &= \dot{\mathcal{U}}_3 Z^T Z + \dot{\mathcal{U}}_2 Z^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{U}}_1 Z^t\dot{\mathcal{Z}}_2 + \dot{\mathcal{U}}_1\dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{U}}_2\dot{\mathcal{Z}}_1^T + \dot{\mathcal{U}}_1\dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1 \\ \Rightarrow \dot{\mathcal{U}}_3 &= \Big[\mathscr{P}_U^\perp\overline{\mathscr{L}}\dot{\mathcal{Z}}_2 - \dot{\mathcal{U}}_2(Z^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z) - \\ &\qquad \dot{\mathcal{U}}_1(Z^T\dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1)\Big]\left(Z^T Z\right)^{-1}\end{aligned}$$

$$\begin{aligned}\dot{\mathcal{U}}_3^T\overline{\mathscr{L}} &= \left(\dot{\mathcal{U}}_3^T\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_3\right)Z^T + \dot{\mathcal{Z}}_4 + \left(\dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1\dot{\mathcal{U}}_2\right)\dot{\mathcal{Z}}_1^T + \dot{\mathcal{U}}_1\dot{\mathcal{U}}_1\dot{\mathcal{Z}}_2^T \\ \Rightarrow \dot{\mathcal{Z}}_4 &= \overline{\mathscr{L}}^T\dot{\mathcal{U}}_3 - Z\left(\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_3 + \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_3^T\dot{\mathcal{U}}_1\right) - \\ &\qquad \dot{\mathcal{Z}}_1\left(\dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1\dot{\mathcal{U}}_2\right) - \dot{\mathcal{Z}}_2\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1\end{aligned}$$

$$\begin{aligned}\mathscr{P}_U^\perp\overline{\mathscr{L}}\dot{\mathcal{Z}}_3 &= \dot{\mathcal{U}}_4 Z^T Z + \dot{\mathcal{U}}_3 Z^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{U}}_2 Z^T\dot{\mathcal{Z}}_2 + \dot{\mathcal{U}}_1 Z^T\dot{\mathcal{Z}}_3 + \\ &\quad \dot{\mathcal{U}}_3\dot{\mathcal{Z}}_1^T Z + \dot{\mathcal{U}}_2\dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{U}}_1\dot{\mathcal{Z}}_3^T Z + \dot{\mathcal{U}}_2\dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{U}}_1\dot{\mathcal{Z}}_2^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{U}}_1\dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_2 \\ \Rightarrow \dot{\mathcal{U}}_4 &= \Big[\mathscr{P}_U^\perp\overline{\mathscr{L}}\dot{\mathcal{Z}}_3 - \dot{\mathcal{U}}_3\left(Z^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T Z\right) - \dot{\mathcal{U}}_2\left(Z^T\dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^T Z + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1\right) \\ &\qquad -\dot{\mathcal{U}}_1\left(Z^T\dot{\mathcal{Z}}_3 + \dot{\mathcal{Z}}_3^T Z + \dot{\mathcal{Z}}_2^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_2\right)\Big]\left(Z^T Z\right)^{-1}\end{aligned}$$

□

The power of these retractions is that we can get an arbitrarily high order of convergence at relatively low cost. In fact, for each additional element of the perturbation series we compute, we obtain an additional order of accuracy. It is simple to show that this method converges linearly. The rate of convergence for a sequence $\{a_k\}$ that converges to a is defined as $q^* = \sup_q\left\{q : \lim_{k\to\infty}\frac{|a_{k+1}-a|}{|a_k-a|^q} = 0\right\}$ [46, 47]. For the perturbative retractions, we have that $\left.\frac{|a_{k+1}-a|}{|a_k-a|^q}\right|_{q=1} = \mathcal{O}(\Delta x)$; increasing q would cause the fraction to blow up as $\Delta x \to 0$. One may ask: why not use a second-order iterative method like Newton-Raphson? Recall that this would require a large matrix inversion at each step, whereas each perturbative retraction is quite cheap (only



(a) Taylor series approximations

(b) Error in approximations

Figure 2-5: For a function $f(x) = 1 + \log(x+1)\left[\cos(x) + \sin(2x)\right]$, we plot first-, second-, and third-order approximations centered at $x = 0$. We also plot their absolute error. For $|x| \ll 1$, the third-order approximation is best, followed by the second-order approximation, and finally the first-order approximation. But, as x grows, the higher-order approximations overshoot, and the approximation with the least error of the three is actually the first-order approximation.

requiring the inversion of an $r \times r$ matrix), especially for low-order corrections. Once we fix the number of terms (p) in the perturbation series to compute, the given perturbative retraction will be a p-th order time integrator when considering the global error.

Another benefit of these retractions is that there is no qr decomposition nor any SVD to compute; there are only matrix multiplications. The caveat to this efficiency, however, is in the assumptions. Namely, we've assumed $\mathscr{L} = \mathcal{O}(1)$, and the series (2.15, 2.16) may only be valid for $\Delta t \ll 1$. We have not shown that the series will converge; instead, we assume that at finite truncation of i, the series becomes more and more accurate. More precisely, we require that $\Delta t^i ||\dot{\mathcal{U}}_i|| \ll ||U||$ and $\Delta t^i ||\dot{\mathcal{Z}}_i|| \ll ||Z||$ for all i. Otherwise, the asymptotic series may overshoot, and we could be better off taking a lower-order approximation.

Hence, we propose an *adaptive* perturbative retraction which adjusts the order of the retraction depending on the norms of the terms above (see algorithm 3). To determine where to truncate the series, the values $\frac{\Delta t^i}{||U||}||\dot{\mathcal{U}}_i||$ and $\frac{\Delta t^i}{||Z||}||\dot{\mathcal{Z}}_i||$ are computed; we want these values to be much less than 1. Practically, this requires a hyperparameter $\varepsilon \ll 1$ to indicate whether or not the aforementioned value is much less than 1. Heuristically, we have found that if $\max\left(\frac{\Delta t^i}{||U||}||\dot{\mathcal{U}}_i||, \frac{\Delta t^i}{||Z||}||\dot{\mathcal{Z}}_i||\right) < \varepsilon \in [0.025, 0.1]$ works well. If it is deemed that the terms of the asymptotic series are appropriately scaled,

the next perturbation is computed; otherwise, a lower-order truncation is used.

Algorithm 3: General adaptive perturbative retraction

Input: $U_0 \in \mathcal{V}_{m,r}$, $Z_0 \in \mathbb{R}_*^{n\times r}$, $\overline{\mathscr{L}} \in \mathbb{R}^{m\times n}$, $\Delta t \in \mathbb{R}$, $\varepsilon \in \mathbb{R}$

Output: $U_1 \in \mathcal{V}_{m,r}$, $Z_1 \in \mathbb{R}^{n\times r}$

$U_1 = U_0, \quad Z_1 = Z_0, \quad \dot{\mathcal{U}}_0 = 0, \quad \dot{\mathcal{Z}}_0 = 0,$
$\alpha_0 = ||Z_0||, \quad \alpha = 0, \quad n = 0$

while $\alpha < \varepsilon$ **do**

  $U_1 \leftarrow U_1 + \Delta t^n \dot{\mathcal{U}}_n, \quad Z_1 \leftarrow Z_1 + \Delta t^n \dot{\mathcal{Z}}_n$

  $n \leftarrow n + 1$

  Compute $\dot{\mathcal{U}}_n$ and $\dot{\mathcal{Z}}_n$

  $\alpha = \frac{\Delta t^n}{\alpha_0} \max\left(||\dot{\mathcal{U}}_n||, ||\dot{\mathcal{Z}}_n||\right)$

$U_1, Z_1 \leftarrow$ re-orthonormalization procedure[a] on U_1, Z_1

[a]See Appendix D.1 for details as to why re-orthonormalization is necessary and for implementation details.

In practice, however, we essentially always want to keep the first-order retraction. Furthermore, we don't want to indefinitely compute higher-order terms even if they are increasingly small and converge – if they are getting increasingly small, then the computational cost is not worth the marginal increased accuracy. Hence, algorithm 10 in appendix D.2 is proposed as a more realistic implementation.

Finally, we will show that the higher-order corrections are not second-order retractions. Here we note that the terminology of "second-order retraction" and "second-order correction" is unfortunate and confusing. A second-order retraction refers to (2.6), and a retraction with a second-order correction refers to theorem 2.3.3. Our higher-order retractions (now referring to retractions with higher-order corrections) have $\mathcal{O}(\Delta t^2)$ components that are not in the normal space of $\mathscr{M}_r$ at X. This is not a surprise as we showed that the extended projective retraction (which we are asymptotically approximating) is not a second-order retraction. It suffices to show that the retraction with a second-order correction is not a second-order retraction since the third- and higher-order corrections only add $\mathcal{O}(\Delta t^3)$ terms. We refer back to (2.19, 2.20, 2.21, 2.22).

$$
\begin{aligned}
&\mathscr{P}^{\perp}_{\mathcal{T}_X\mathscr{M}_r}\left(\dot{\mathcal{U}}_2 Z^T + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T + U\dot{\mathcal{Z}}_2^T\right)\\
&= \mathscr{P}^{\perp}_{U}\left(\dot{\mathcal{U}}_2 Z^T + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T + U\dot{\mathcal{Z}}_2\right)\mathscr{P}^{\perp}_{Z}\\
&= \dot{\mathcal{U}}_2 Z^T \cancelto{0}{\mathscr{P}^{\perp}_{Z}} + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1 \mathscr{P}^{\perp}_{Z} + \cancelto{0}{\mathscr{P}^{\perp}_{U} U} \dot{\mathcal{Z}}_2 \mathscr{P}^{\perp}_{Z}
\end{aligned}
$$

Recall that all $\dot{\mathcal{U}}_i$ are invariant under $\mathscr{P}_U^\perp$. Now, we'll expand the last remaining term.

$$\dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1 \mathscr{P}_Z^\perp = \mathscr{P}_U^\perp \overline{\mathscr{L}} Z \left(Z^T Z\right)^{-1} U^T \overline{\mathscr{L}} \mathscr{P}_Z^\perp \tag{2.27}$$

With that, we must show that $\dot{\mathcal{U}}_2 Z^T + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T + U \dot{\mathcal{Z}}_2^T$ differs from the expression above. We'll go term-by-term and then add them.

$$\begin{aligned}
\dot{\mathcal{U}}_2 Z^T = & \Big[\mathscr{P}_U^\perp \overline{\mathscr{L}} \overline{\mathscr{L}}^T U - \mathscr{P}_U^\perp \overline{\mathscr{L}} Z (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T U - \\
& \qquad \mathscr{P}_U^\perp \overline{\mathscr{L}} Z (Z^T Z)^{-1} U^T \overline{\mathscr{L}} Z \Big] (Z^T Z)^{-1} Z^T \\
= & \, \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp \overline{\mathscr{L}}^T U (Z^T Z)^{-1} Z^T - \mathscr{P}_U^\perp \overline{\mathscr{L}} Z (Z^T Z)^{-1} U^T \overline{\mathscr{L}} \mathscr{P}_Z
\end{aligned} \tag{2.28}$$

$$\dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T = \mathscr{P}_U^\perp \overline{\mathscr{L}} Z (Z^T Z)^{-1} U^T \overline{\mathscr{L}}$$

$$\begin{aligned}
U \dot{\mathcal{Z}}_2^T &= U (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T \mathscr{P}_U^\perp \overline{\mathscr{L}} - U (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z \\
&= U (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp
\end{aligned} \tag{2.29}$$

Now, the term in (2.29) and the second term in (2.28) combine such that the sum is as follows.

$$\begin{aligned}
\dot{\mathcal{U}}_2 Z^T + \dot{\mathcal{U}}_1 \dot{\mathcal{Z}}_1^T + U \dot{\mathcal{Z}}_2^T = & \, \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp \overline{\mathscr{L}}^T U (Z^T Z)^{-1} Z^T + \\
& U (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp + \mathscr{P}_U^\perp \overline{\mathscr{L}} Z (Z^T Z)^{-1} U^T \overline{\mathscr{L}} \mathscr{P}_Z^\perp
\end{aligned} \tag{2.30}$$

The last term in (2.30) matches (2.27). So in order for the perturbative methods to second-order retractions, we would need $\mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp \overline{\mathscr{L}}^T U (Z^T Z)^{-1} Z^T + U (Z^T Z)^{-1} Z^T \overline{\mathscr{L}}^T \mathscr{P}_U^\perp \overline{\mathscr{L}} \mathscr{P}_Z^\perp = 0$. And while there is symmetry between those two terms, the sum is not zero in general. Because of this, in the next chapter where the higher-order retractions are investigated with numerical examples, the term second-order or higher-order retraction will not refer to (2.6) but to theorem 2.3.3.

To close off this chapter, we will re-emphasize that in the case of solving differential equations, we have broken the time integration schemes into two steps. The first step is to use a classical time integration scheme to compute $\overline{\mathscr{L}} \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \mathscr{L} dt$ with any numerical scheme. This step essentially tells the scheme in what direction in

the embedding Euclidean space we would like to go. The second step is to apply a retraction in order to stay on the low-rank manifold. By using a high-order retraction, we can preserve the order of accuracy of the time integration scheme used in step one. In the following chapter, we will show that these retractions do indeed preserve high-order convergence to the best low-rank approximation in several test cases.

Chapter 3

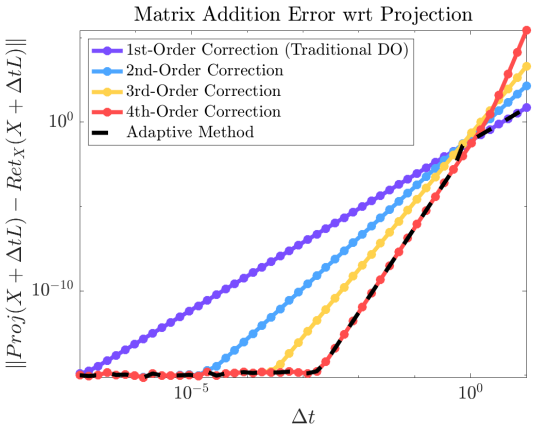
Results and applications

3.1 Matrix addition

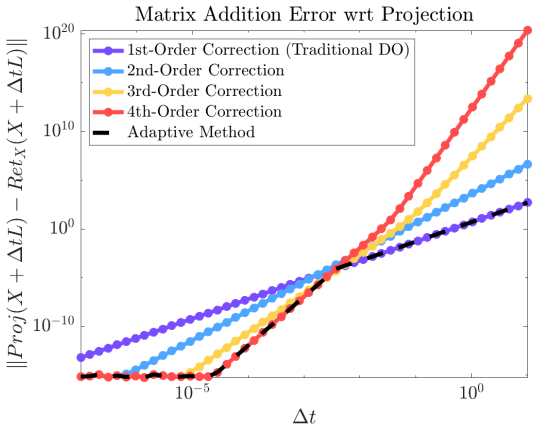
In this example, we consider the simple case of adding two matrices, X and $\Delta t L$. We normalize X and L such that they have Frobenius norm one; hence Δt controls the relative scaling between the norms of X and L. In particular, $X, L \in \mathbb{R}^{50\times 100}$, and X has rank 10. For $X = UZ^T$, uniformly random U and Z are chosen, and then U is orthonormalized. This typically yields a condition number of approximately seven. To see how the retraction performs in the near-singular case, we took the singular value decomposition of Z. Then, we set the last five singular values of Z equal to $\sigma_1/1000$, where σ_1 denotes the max singular value of Z. This yields an L^2 condition number of 1000.

We see that the n-th order perturbative retraction exhibits Δt^{n+1} convergence locally. This is as expected since the n-th order correction has Δt^{n+1} error unaccounted for. Note that for time-dependent problems, the global error will be Δt^n since the number of time steps in a fixed time interval scales as $1/\Delta t$. Furthermore, when $\Delta t \gg 1$, all the retractions tend to overshoot as discussed in the previous section, which shows why the adaptive method is necessary. The adaptive method roughly has the minimum retraction error at all Δt by analyzing when the asymptotic series converges, and for this example we choose hyperparameter $\varepsilon = 0.1$.

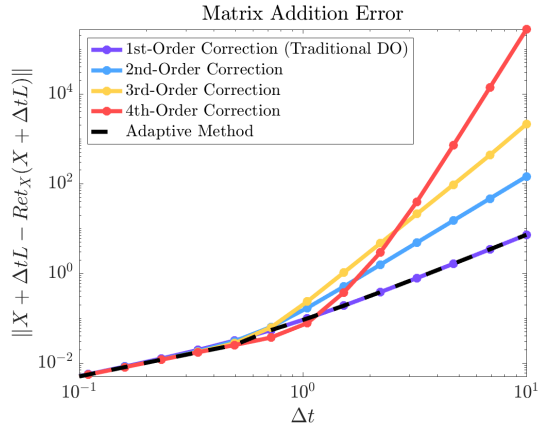
In figures 3-1a and 3-1b, we see high-order convergence all the way to machine precision. These plots measure the error between the projection of $X + \Delta t L$ and the retraction onto the low-rank manifold. Because the error between $X + \Delta t L$ and its projection lie in the normal space to the manifold at the projection, figures 3-1a and 3-1b roughly correspond to error in the tangent space; the error in the normal space is



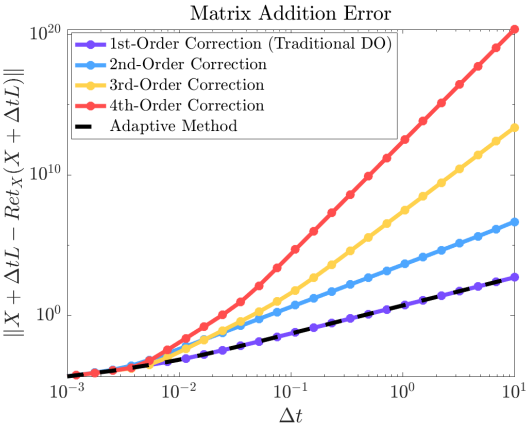
(a) Error with respect to projection, $\text{cond}(X) \approx 7$



(b) Error with respect to projection, $\text{cond}(X) = 1000$



(c) Total error, $\text{cond}(X) \approx 7$



(d) Total error, $\text{cond}(X) = 1000$

Figure 3-1: We compare the convergence of first-, second-, third-, and fourth-order perturbative retractions along with the adaptive method (see algorithm 10) for two X's with different condition numbers in the L^2 norm.

inevitable unless we increase the rank of our solution. As such, we have accomplished our goal in asymptotically approximating the projection onto the manifold.

Figures 3-1c and 3-1d measure the total error between $X+\Delta tL$ and the retraction. For Δt relatively large, we observe high-order convergence, but then all the methods only converge at first-order rates. This is due to the error in the normal space to the low-rank manifold at the projected point. For Δt small, the normal error dominates – that is, we have already done a decent job of approximating the tangent error. As Δt is reduced, the normal error is also scaled down, which is why we see first-order convergence at all.

In both the tangent error and total error cases, by increasing the condition number of X, we see the retractions require a smaller Δt for the same accuracy. This is because the low-rank manifold has larger curvature. So, these low-order polynomial approximations to the low-rank manifold are less accurate. Furthermore, the higher-order corrections more easily overshoot. This further elucidates the utility of the adaptive method since the lower-order corrections are less sensitive to small singular values in X.

One may ask why we even attempt using these higher-order corrections if the total error is dominated by the normal error. There are three potential answers to this. First, in a situation such as matrix differential equations, there could exist an operator that ignores the error normal to the low-rank manifold. As such, approximating the error in the tangent space alone is enough to obtain an accurate dynamical low-rank approximation. Second, consider a matrix differential equation $\frac{dX}{dt} = A_1(X) + A_2(X)+\cdots+A_n(X) \equiv L(X)$, where A_i are some functions that operate on the current state X. It's reasonable to choose a scheme that projects L onto the tangent space at every step and then retracts back onto the tangent space. But, computing and/or storing L in memory may be difficult as the rank of L could be $\sum_i \operatorname{rank}(A_i(X))$, which may be very large. As such, it's reasonable to first project each addition onto the tangent space, i.e. $L \approx \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}(A_1(X) + \mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}(A_2(X) + \cdots))$. Because the operator we have chosen is linear, in this case, the approximation is actually an equality; however, more complicated choices of $L(X)$ will yield some approximation error. In this case, because we would already be projecting L onto the tangent space at the end anyways, we don't care about the normal error so much; we are really trying to approximate L in the tangent space, and so our perturbative method would be highly effective. Third, this whole methodology assumes that $X + \Delta tL$ stays close to the low-rank manifold, which requires that L is well-represented by its projection onto the affine tangent space at X. In the examples described already,

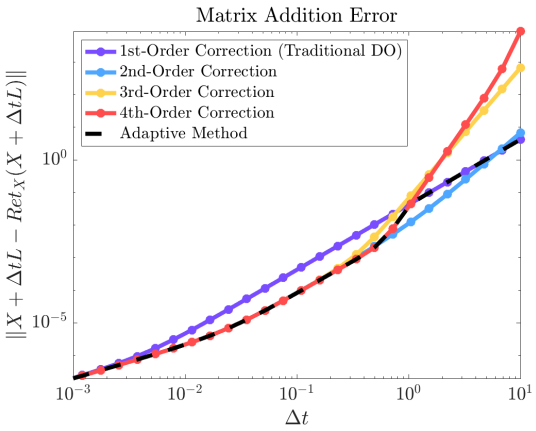
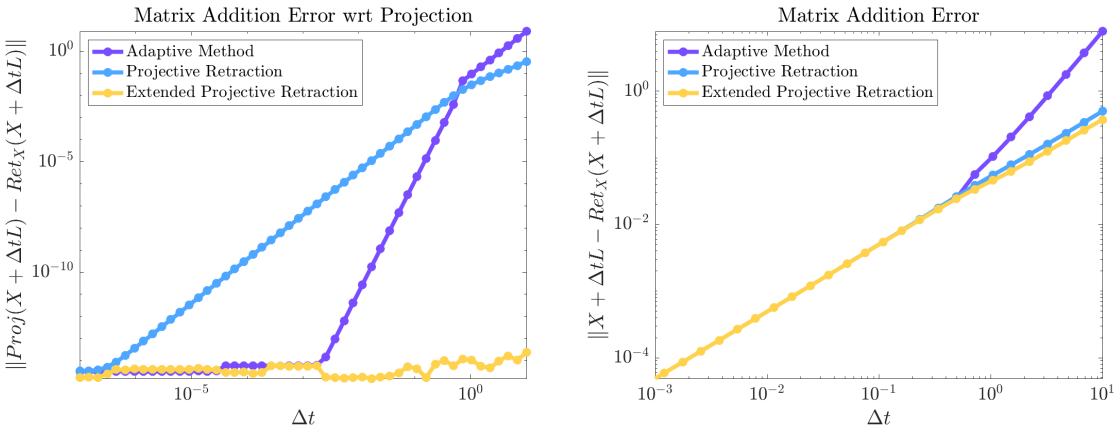


Figure 3-2: Total error when $L = \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} L + \delta$

L is chosen randomly, and so it has a large component in the normal space. This suggests adaptively increasing the rank of the approximation, but this goes beyond the scope of this thesis. Instead, consider another example where L does not have a large component in the normal space.

Figure 3-2 depicts the total error between $X + \Delta tL$ and the retraction when L has a small normal component. To construct this example, we projected a random L onto the tangent space at X and then added a uniformly random matrix times $1/100$, denoted δ. We see that there is a significant range of Δt for which higher-order retractions significantly reduce the error from the first-order correction. It appears that the second-order correction captures most of the total error. So in some applications, the second-order retraction may be most attractive without considering higher-order retractions.

Here we plot how the projective retractions compare with the adaptive perturbative retraction. Of course, the extended projective retraction has machine zero error with respect to the projection – the singular value decomposition gives the projection onto the low-rank manifold. Recall that the extended projective retraction may be thought of as the ideal retraction in some ways. Perhaps surprisingly, the (non-extended or "vanilla") projective retraction does not do as well as the adaptive perturbative retraction in reducing the error with respect to the projection (see figure 3-3a). Even though it projects from the tangent space to the low-rank manifold, it still incurs error from the initial projection onto the tangent space in which information about the normal space of $X + \Delta tL$ is lost. Figure 3-3b tells a different story. The vanilla projective retraction is near ideal and outperforms the perturbative retraction for large Δt. This is likely because it avoids the overshoot problem that



(a) Error with respect to projection

(b) Total error

Figure 3-3: We compare the convergence of the adaptive retraction with the projective retractions.

the perturbative approximations are subject to. So, the vanilla projective retraction seems to do well in avoiding error in the normal space.

3.2 Real-time data compression

Another application of higher-order retractions is real-time data compression. Imagine we have high-definition video (e.g. 4k or 8k) recording in real-time, and we want to broadcast it to many viewers; however, we don't have the bandwidth to send the full data stream, at least not in an uncompressed format. Now, one may consider taking the truncated SVD at each frame of the video, but this is extremely costly to do for such high resolution video, especially in real time. We may not have the computational power to compress 30 or even 60 frames per second. So, we need a cheaper way to compress the data.

Consider this solution: compute the truncated SVD on the first-frame (maybe inducing a small, acceptable time delay/lag at the beginning of the stream), and then for each subsequent frame, use higher-order retractions to stay in the compressed format. This would require only looking at the changes from frame-to-frame, and then retracting these differences onto the low-rank manifold. In other words, we would treat the difference between frame i and frame $i+1$ as the analogue to the time derivative in a differential equation.

To test this, we took a roughly four-second video recorded at 4k (3840×2160) 60 frame per second of a peacock walking across a road in Split, Croatia. Though

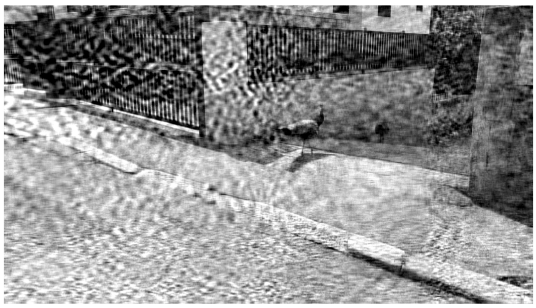
(a) True image

we could apply the compression methodology to each of the red, green, and blue data streams, for simplicity we converted the video to grayscale. Starting from a truncated SVD of ranks 100 and 500, we applied retractions on the frame-to-frame differences and compared the error with respect to the true video and with respect to the best truncation given by taking the truncated SVD of each frame. Before analyzing the error, we'll show the last frame of the video (figure 3-4a) along with the reconstructed last frames from the best approximation (figures 3-4b and 3-4c), the extended projective retraction (figures 3-4d and 3-4e), the adaptive perturbative retraction with $\varepsilon = 0.025$ (figures 3-4f and 3-4g), and the first-order retraction (figures 3-4h and 3-4i), each for ranks of 100 and 500.

We can see that clearly a rank of 100 is sufficient for an acceptable image. At a rank of 500, more details are present in the best approximation (see the cobblestone in the image). The extended projective retraction has a fair amount of error with $r = 100$ but significantly less with $r = 500$. This is because errors compound over time; so small errors at the beginning of the video grow over time and affect the last frame. Nevertheless, even at $r = 100$, the image still captures the main feature of the scene. The extended projective retraction can be viewed as our baseline – it's computationally expensive but in some ways ideal in that it minimizes the Frobenius error at each step. The adaptive perturbative retraction does worse than the extended projective retraction, but, again, it captures the main features of the video despite

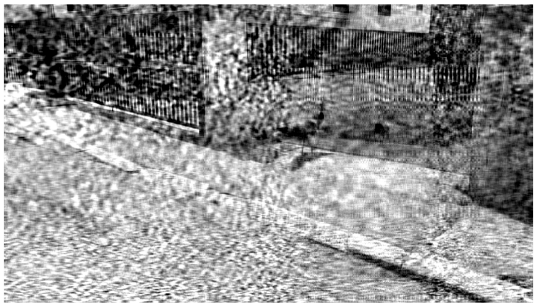
(b) Best approximation, $r = 100$

(c) Best approximation, $r = 500$

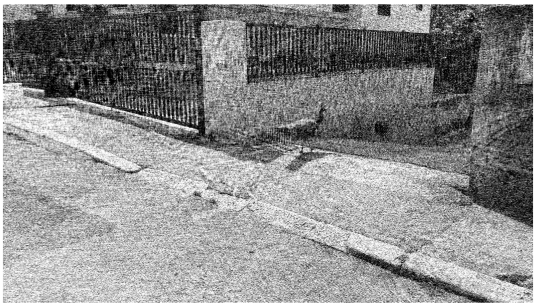


(d) Extended projective retraction, $r = 100$

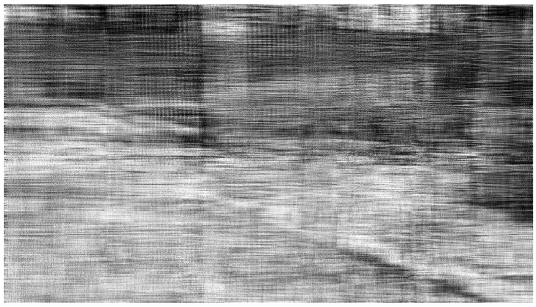
(e) Extended projective retraction, $r = 500$



(f) Adaptive perturbative retraction, $r = 100$



(g) Adaptive perturbative retraction, $r = 500$



(h) First-order retraction, $r = 100$

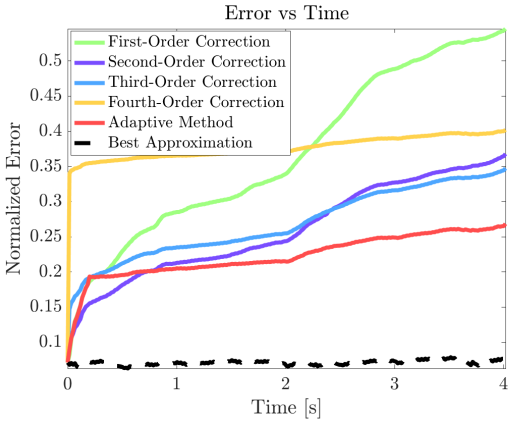
(i) First-order retraction, $r = 500$

Figure 3-4: Comparison of retractions at the last frame of a 4k 60 Hz video

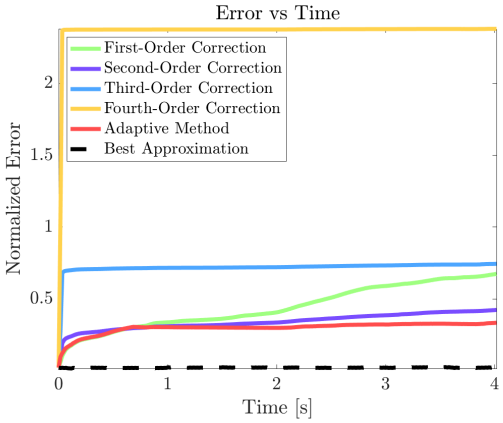
the strong noise. What's interesting is in the $r = 500$ case, the noise seems to be more granular, and the image is definitely more interpretable. Lastly, the first-order retraction loses almost all of the information after four seconds at $r = 100$. For $r = 500$, we can make out the general shape of the fence if we squint, but the peacock is gone. This is a great demonstration of how much more effective the adaptive perturbative retraction is than the first-order retraction. The higher-order corrections make a big difference after 240 frames.

For further analysis, we look to plots of error vs time. In figure 3-5, we plot the Frobenius norm of the error at each frame normalized by the Frobenius norm of the first frame of the video. Notably, the fourth-order retraction does quite poorly in the $r = 500$ case. It seems to overstep very early on and cannot recover. Meanwhile, the first-order retraction does well at first but then the error continues to grow with time. This highlights the efficacy of the adaptive scheme; it filters out the oversteps of the higher-order methods but still captures the slow error growth as time continues. The adaptive perturbative method is also plotted with the projective retractions for reference. Surprsingly, the adaptive method is more effective than the projective retraction, though the extended projective retraction still beats it. Unlike the matrix addition case, the error with respect to the best approximation tells the same story as the error with respect to the true video. This suggests that there is error in both the normal and tangent spaces to the low-rank manifold at the point of best approximation.

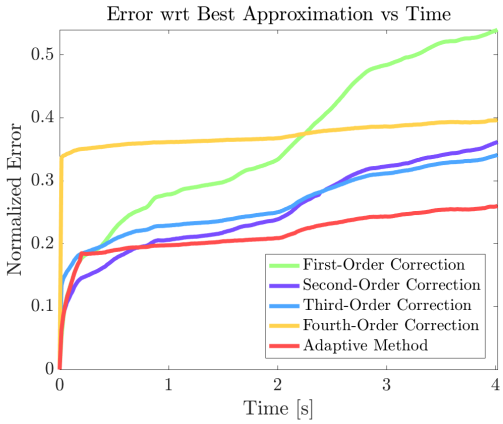
Realistically, the retractions cannot be used straight out of the box for real-time video compression – they just don't perform well enough. Video codecs like MPEG are very complex and are specialized to compress videos, whereas these retractions are far more general. Perhaps a combination of the methodologies could be used, specializing the retractions for the video compression setting. Another suggestion would be to take the truncated SVD every few frames or however often is affordable in a real-time environment. This would prevent the compressed video from departing too far from the true video. In any case, the adaptive perturbation has proven itself to be far more effective than the first-order retraction at a much cheaper cost than the extended perturbative retraction.



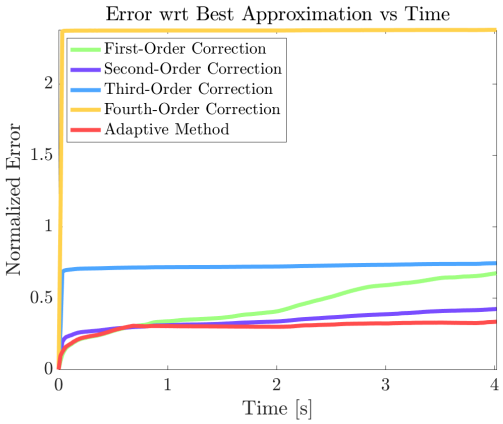
(a) Perturbative retractions, $r = 100$



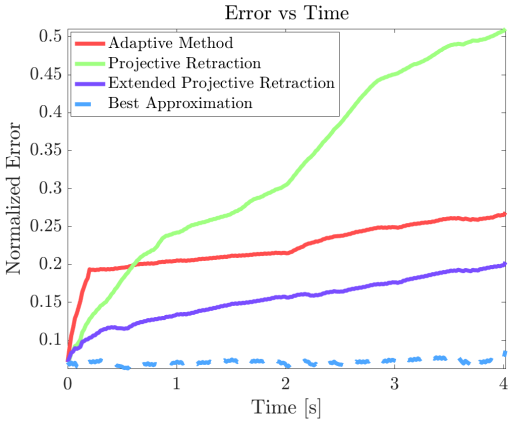
(b) Perturbative retractions, $r = 500$



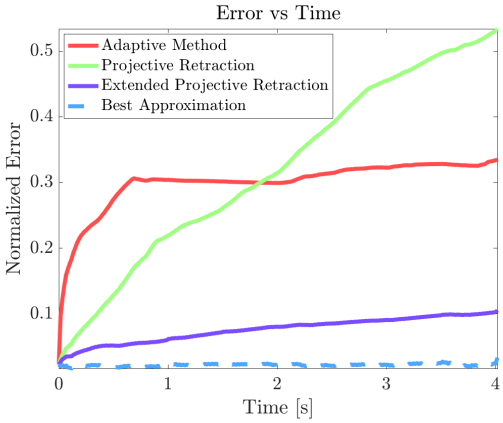
(c) Perturbative retractions wrt best approx., $r = 100$



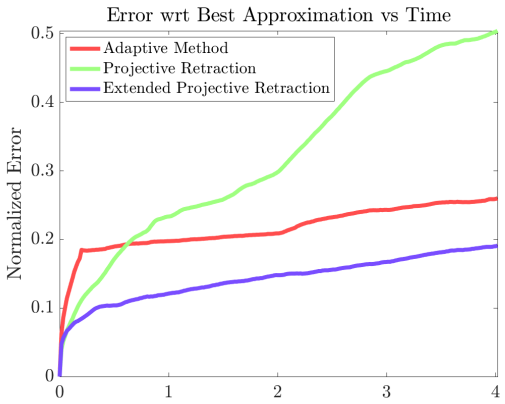
(d) Perturbative retractions wrt best approx., $r = 500$



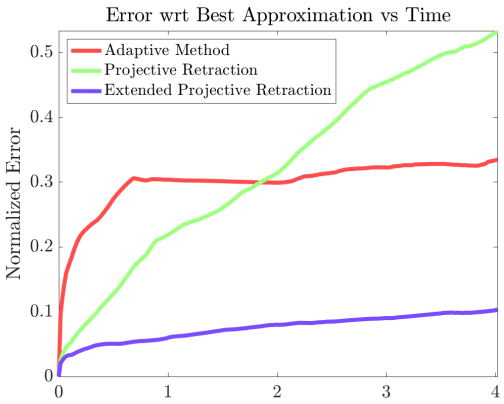
(e) Projective retractions, $r = 100$



(f) Projective retractions, $r = 500$



(g) Projective retractions wrt best approx., $r = 100$



(h) Projective retractions wrt best approx., $r = 500$

Figure 3-5: Error due to different retractions as a function of time for movie compression

3.3 Matrix differential equations

Next, we turn our attention to matrix differential equations. A motivating example is $2m$ systems of coupled of linear oscillators $A \in \mathbb{R}^{2m \times 2m}$ with the following differential equation governing their motion.

$$\ddot{A} = -\Lambda A, \qquad A(0) = A_0, \qquad \dot{A}(0) = V_0 \tag{3.1}$$

Consider a solution given by

$$A(t) = R(t)QS \tag{3.2}$$

where Q is an orthonormal $2m \times 2m$ matrix and S is an $2m \times 2m$ diagonal matrix with non-increasing entries $\{s_i\}_{i=1}^m$. $R(t)$ is a rotation (and hence orthogonal) matrix given as follows.

$$R(t) = \begin{bmatrix} \cos(\omega_1 t) & -\sin(\omega_1 t) & & & & & \\ \sin(\omega_1 t) & \cos(\omega_1 t) & & & & 0 & \\ & & \cos(\omega_2 t) & -\sin(\omega_2 t) & & & \\ & & \sin(\omega_2 t) & \cos(\omega_2 t) & & & \\ & & & & \ddots & & \\ & 0 & & & & \cos(\omega_m t) & -\sin(\omega_m t) \\ & & & & & \sin(\omega_m t) & \cos(\omega_m t) \end{bmatrix}$$

Note we can write out the derivatives of R explicitly.

$$\begin{aligned} \dot{R}(t) &= \begin{bmatrix} -\omega_1 \sin(\omega_1 t) & -\omega_1 \cos(\omega_1 t) & & & & \\ \omega_1 \cos(\omega_1 t) & -\omega_1 \sin(\omega_1 t) & & & 0 & \\ & & \ddots & & & \\ & 0 & & & -\omega_m \sin(\omega_m t) & -\omega_m \cos(\omega_m t) \\ & & & & \omega_m \cos(\omega_m t) & -\omega_m \sin(\omega_m t) \end{bmatrix} \\ &= \Omega \begin{bmatrix} -\sin(\omega_1 t) & -\cos(\omega_1 t) & & & & \\ \cos(\omega_1 t) & -\sin(\omega_1 t) & & & 0 & \\ & & \ddots & & & \\ & 0 & & & -\sin(\omega_m t) & -\cos(\omega_m t) \\ & & & & \cos(\omega_m t) & -\sin(\omega_m t) \end{bmatrix} \end{aligned} \tag{3.3}$$

We define $\Omega = \text{diag}(\omega_1, \omega_1, \omega_2, \omega_2, \ldots, \omega_m, \omega_m)$. With this, we have the following.

$$\ddot{R}(t) = -\Omega^2 R(t)$$

To ensure (3.2) is a solution to (3.1), we differentiate with respect to time.

$$\ddot{A} = \ddot{R}QS = -\Omega^2 RQS = -\Omega^2 Y$$

Hence, setting $\Lambda = \Omega^2$, $A_0 = QS$, and $V_0 = \dot{R}(0)QS$ with $\dot{R}$ defined by (3.3), we have found a closed-form solution to the matrix differential equation.

By construction, the solution has particular properties. Because R and Q are orthonormal, the singular values of Y do not vary in time. In fact, the singular values are simply the diagonal entries of S. This will allow for a clear analysis of the retractions.

For the test case, we take $m = 13$ (so A is 26×26) and construct S with 16 large entries and 10 small entries defined below

$$S_{ii} = \begin{cases} 100 + 10z_i & i \leq 16 \\ 10^{-3+\frac{i-17}{9}} & i > 16 \end{cases}$$

where z_i are iid standard normal random variables. After construction, the diagonal entries of S are sorted from greatest to least for simple analysis. Q is constructed by orthonormalizing (via MATLAB's `orth` function) a 26×26 matrix of uniformly distributed random numbers. For our numerical experiment, we consider the case of truncating Y to a rank of 16. This allows us to capture almost all of the variability in Y, and because the singular values of $Y(t)$ do not change, the whole solution trajectory will stay very close to the low-rank manifold. In order to solve this system of ODEs, we use a sixth-order symplectic Yoshida integrator [48], which is a generalization of the common leapfrog integrator (see algorithm 4. We'll compare the true solution as well as the best approximation to the solution to each of the retractions from times $t = 0$ to 10. Note that the best approximation in this case is given by $R(t)Q(:, 1:16)S(1:16, :)$. We'll denote a low-rank approximation to Y.

Figure 3-6 shows the error from perturbative and projective retractions as a function of time. In appendix B, extra figures (see figure B-1) are given of the error with respect to the best approximation. They are not given here since they tell the same story as figure 3-6. Again, the y-axis corresponds to the normalized Frobenius error, where the normalization constant is equal to the norm of $A(0)$. In figure 3-6a, the

Algorithm 4: Sixth-order integrator

Input: $X \in \mathbb{R}^{m\times n}, \dot{X} \in \mathbb{R}^{m\times n}, \Delta t \in \mathbb{R}, F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Output: $X, \dot{X}$

$w_1 = -1.17767998417887$

$w_2 = 0.235573213359357$

$w_3 = 0.784513610477560$

$w_0 = 1 - 2(w_1 + w_2 + w_3)$

$b = \begin{bmatrix} w_3 & w_2 & w_1 & w_0 & w_1 & w_2 & w_3 \end{bmatrix}$

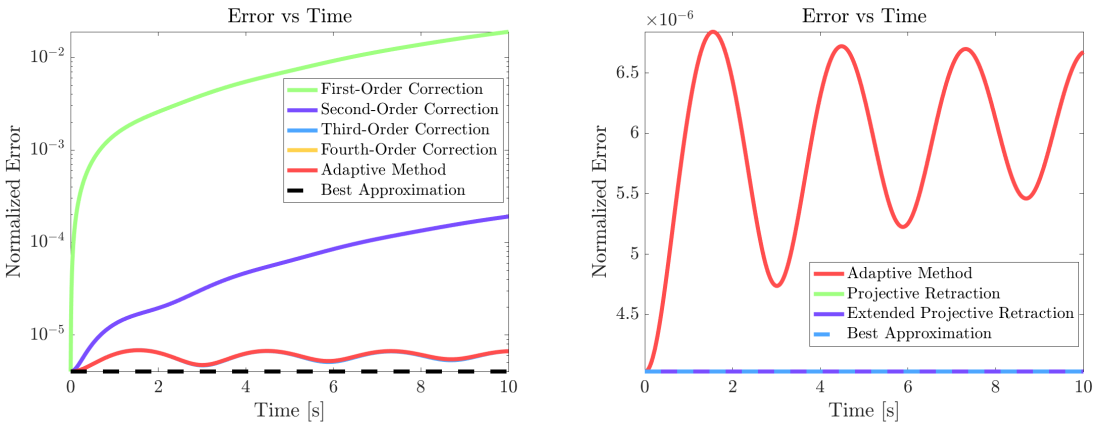
$a = \frac{1}{2}\begin{bmatrix} w_3 & w_3 + w_2 & w_2 + w_1 & w_1 + w_0 & w_1 + w_0 & w_2 + w_1 & w_3 + w_2 & w_3 \end{bmatrix}$

for $i = 1 : 7$ **do**

  $X = X + \Delta t a_i \dot{X}$

  $\dot{X} = \dot{X} + \Delta t b_i F(X)$

$X = X + \Delta t a_8 \dot{X}$



(a) Perturbative retractions

(b) Projective retractions

Figure 3-6: Error due to different retractions as a function of time for matrix differential equations, $\Delta t = 0.01$

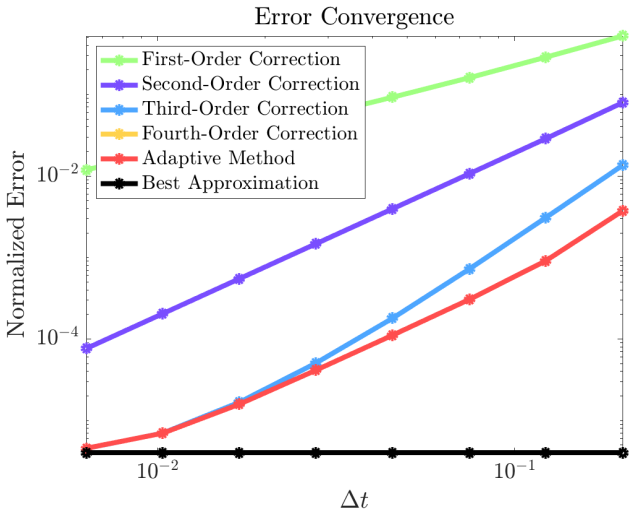


Figure 3-7: Convergence plots of perturbative retractions for matrix differential equations

adaptive (with $\varepsilon = 0.025$), third-, and fourth-order retractions are all stacked on top of each other. We can see that the third-order correction is very effective in reducing the error relative to the first- and second-order corrections. The oscillatory nature of the adaptive error is likely related to the oscillatory nature of the solution itself. The projective retractions seem to have almost no error whatsoever. In fact, their error is very slightly greater than the sum of the singular values not included in X. So in contrast to the movie compression example where the adpative method outperformed the projective retraction, the projective retraction easily beats the adaptive retraction in this system of ODEs. The error with respect to the best approximation tells mostly the same story as error with respect to the true solution.

In addition, we do a convergence study of each for each of the retractions by taking smaller and smaller time steps. For additional figures, see figure B-2. In figure 3-7, we see that the first-order correction exhibits convergence of order $\mathcal{O}(\Delta t)$, the second-order corrections exhibits convergence of order $\mathcal{O}(\Delta t^2)$, and the adaptive, third-, and fourth-order corrections exhibit convergence of order $\mathcal{O}(\Delta t^3)$. As the adaptive, third-, and fourth-order methods approach the best approximation, the convergence slows slightly, which is not a concern since we are seeing the effects of the model error from truncation.

One may wonder why we don't see fourth-order convergence in the adaptive and fourth-order retractions. But recall there is, in general, an $\mathcal{O}(\Delta t)$ error (locally) induced by the Dirac-Frenkel time-dependent variational principle. That is, the derivative information we obtain at each step is from the truncated solution on the low-rank

manifold instead of the derivative from the full solution. In other words, denoting the low-rank approximation to A as X, we are solving $\ddot{X} = -\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\Omega^2 X(t)$ instead of $\ddot{X} = -\mathscr{P}_{\mathcal{T}_X\mathscr{M}_r}\Omega^2 Y(t)$. It is this information gap that ruins our convergence at higher orders.

Another way to view this error is in the discrete sense. Suppose we start on the low-rank manifold with X_0 and march the solution one step. Before retracting, we'll call the point $\tilde{X}_1 = X_0 + \Delta t\overline{\mathscr{L}}(t_0, X_0)$, where $\overline{\mathscr{L}}$ is an approximation of a time average of our differential operator. After retraction, we obtain $X_1 = \mathcal{R}_{X_0}(\Delta t\overline{\mathscr{L}}(t_0, X_0))$. Now we take another step from each of these points without retraction.

$$\text{Stepping from } \tilde{X}_1 : \tilde{X}_1 + \Delta t\overline{\mathscr{L}}(t_1, \tilde{X}_1)$$
$$\text{Stepping from } X_1 : X_1 + \Delta t\overline{\mathscr{L}}(t_1, X_1)$$

Even before retracting the second step back onto the low-rank manifold, we see the problematic difference. Of course we expect to start from different points $\tilde{X}_1$ and X_1. The problem is in calculating our differential operator $\overline{\mathscr{L}}$, we are evaluating at different points $\tilde{X}_1$ and X_1, which is shown in figure 3-8. We have shown that using our n-th order perturbative retractions, we have that $X_1 = \mathscr{P}_{\mathscr{M}_r}\tilde{X}_1 + \mathcal{O}(\Delta t^n)$. If we assume that the full-rank solution stays close to the low-rank manifold such that $||\tilde{X}_1 - \mathscr{P}_{\mathscr{M}_r}\tilde{X}_1|| < \varepsilon$, then $||X_1 - \tilde{X}_1|| = \mathcal{O}(\varepsilon, \Delta t^n)$. Finally, if we assume that $\mathscr{L}$ is Lipschitz continuous (and so $\overline{\mathscr{L}}$ is also Lipschitz continuous), then we have that $\Delta t||\overline{\mathscr{L}}(t_1, \tilde{X}_1) - \overline{\mathscr{L}}(t_1, X_1)|| = \mathcal{O}(\varepsilon\Delta t, \Delta t^{n+1})$. Hence, we have shown that there is a local first-order error using the Dirac-Frenkel time-dependent variational principle.

One may ask why the aforementioned error doesn't ruin second- or third-order convergence. Well because we've constructed S such that the truncated singular values of Y are very small, ε in the analysis above is also very small. Hence, the first-order error is not observable until the error becomes very small, which is only achieved in the higher-order retractions for these values of Δt.

The next logical question would then be to ask what would happen if we did not use the Dirac-Frenkel time-dependent variational principle; that is, what if we used the full-rank derivative information from Y in our calculation of X. We would then expect to see higher-order convergence in the fourth-order and adaptive methods. Indeed we do see higher-order convergence, but before discussing the results, we need to describe the modified integrator. The sixth-order Yoshida integrator can be thought of as having eight semi-implicit steps; we alternate integrating the velocity and the position. So the question is where do we provide the full-rank derivative

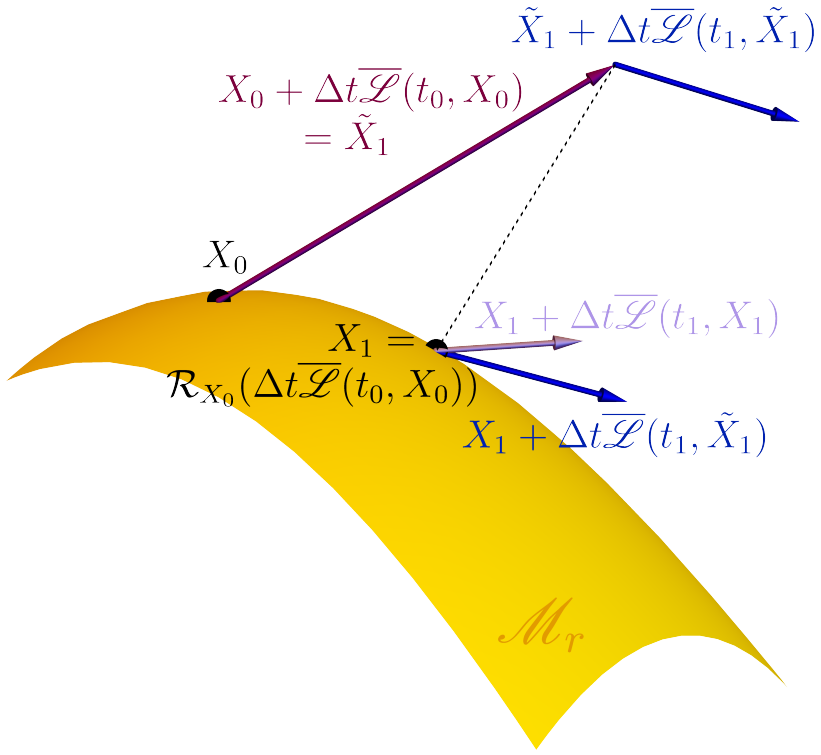


Figure 3-8: Comparison of two steps with and without the Dirac-Frenkel time-dependent variational principle. Note that $\overline{\mathscr{L}}(t_1, \tilde{X}_1) \neq \overline{\mathscr{L}}(t_1, X_1)$.

information? It is sufficient to only change the first step of integrating both X and $\dot{X}$. The integration algorithm is written out in algorithm 5, and it should be contrasted with algorithm 4. Henceforth, we will not include the coefficients w_i, a, or b in the algorithm; they are the same as in algorithm 4. The key note here is to use $F(A+\Delta t a_1 \dot{A})$ instead of $F(X)$ and to use $\dot{A}$ instead of $\dot{X}$ in the first integration of $\dot{X}$; otherwise, the error from X and $\dot{X}$ pollutes the higher-order convergence. Hence, we must effectively integrate $\dot{A}$ one step using the full-rank Y and $\dot{A}$. As a consequence, we don't really need to return $\dot{X}$ since it will be calculated from the full-rank solution in the next step. Contrasting figure 3-9a with figure 3-6a, we see that providing the full-rank derivative information reduces the error for the first- and second-order retraction and gets rid of the oscillations in the error for the higher-order retractions. But what's more interesting is looking at the convergence plots. Figure 3-9b shows "cleaner" convergence than figure 3-7. We recover fourth-order convergence for both the fourth-order and adaptive methods. However, the error does not decrease all the way down to the best approximation when the full-rank derivative is given.The reason for this error is subtle.

Recall we have shown that for $X = \mathscr{P}_{\mathscr{M}_r} A$, given $\dot{A} = \mathscr{L}$, the reduced-order

Algorithm 5: Sixth-order integrator with full-rank derivative information

Input: $X \in \mathbb{R}^{m\times n}, \dot{X} \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{m\times n}, \dot{A} \in \mathbb{R}^{m\times n}, \Delta t \in \mathbb{R}, F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Output: $X, \dot{X}$

$X = X + \Delta t a_1 \dot{A}$

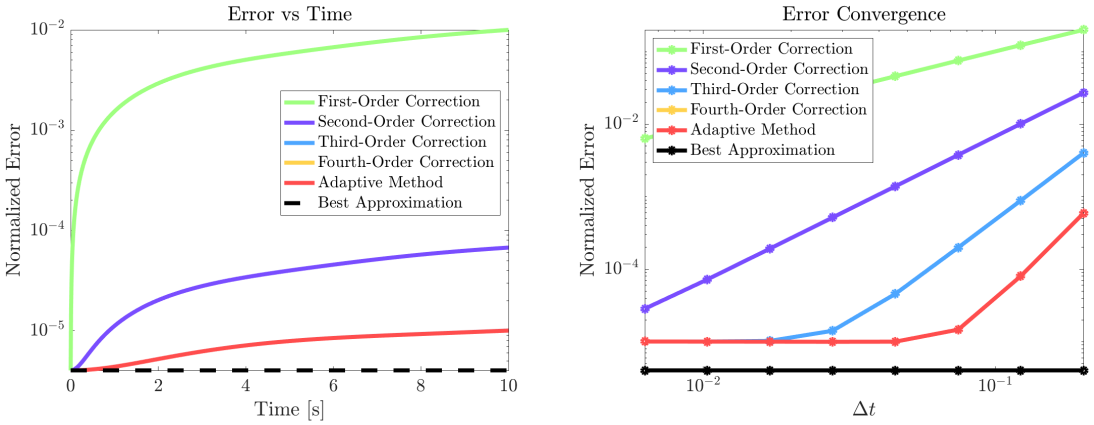
$\dot{X} = \dot{A} + \Delta t b_1 F(A + \Delta t a_1 \dot{A})$

for $i = 2 : 7$ **do**

  $X = X + \Delta t a_i \dot{X}$

  $\dot{X} = \dot{X} + \Delta t b_i F(X)$

$X = X + \Delta t a_8 \dot{X}$



(a) Error as a function of time with full-rank derivative information, $\Delta t = 0.01$

(b) Convergence plots for perturbative retractions given full-rank derivative information

Figure 3-9: Error plots for perturbative retractions when using algorithm 5

differential equation we seek to solve is $\dot{X} = \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L}(t, A(t))$. That is, we project the full-rank dynamics onto the tangent space at X. Let $\tilde{X}$ denote our approximation to $\mathscr{P}_{\mathscr{M}_r} A$; as we integrate $\tilde{X}$, we accumulate error, so when we are projecting $\mathscr{L}(t, A(t))$ onto the tangent space, we are really computing $\mathscr{P}_{\mathcal{T}_{\tilde{X}} \mathscr{M}_r} \mathscr{L}(t, A(t)) \neq \mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \mathscr{L}(t, A(t))$. How might one fix this error? The first thought may be to use X instead of $\tilde{X}$ to project the dynamics and then add the projected dynamics to the previous value of $\tilde{X}$. For the moment, ignore the computational feasibility of such an algorithm (since if we could compute $\mathscr{P}_{\mathscr{M}_r} A$ at every step, we would have no need to use these retractions in the first place). However, this does not work – we would be projecting the dynamics onto a mismatched tangent space. In other words, the geometry of the manifold is different at X and $\tilde{X}$, so we cannot swap their tangent spaces without incurring an additional error.

An alternative approach is to correct the full-rank derivative to reflect the difference between X and $\tilde{X}$. By definition, $\overline{\mathscr{L}}(t_n, A(t_n)) \approx \frac{A(t_{n+1}) - A(t_n)}{\Delta t}$. In defining $X = \mathscr{P}_{\mathscr{M}_r} A$, the difference between X_n and $A(t_n)$ exists only in the normal space to the affine tangent space at X_n. Hence, we have the following.

$$\mathscr{P}_{\mathcal{T}_X \mathscr{M}_r} \left[\frac{A(t_{n+1}) - A(t_n)}{\Delta t} \right] = \frac{X_{n+1} - X_n}{\Delta t}$$

A simple numerical scheme would then be to take $\tilde{X}_{n+1} = \mathcal{R}_{\tilde{X}_n} \left(\Delta t \mathscr{P}_{\mathcal{T}_{\tilde{X}} \mathscr{M}_r} \overline{\mathscr{L}}(t_n, A(t_n)) \right)$. To first-order, we have $\tilde{X}_{n+1} \approx \tilde{X}_n + (X_{n+1} - X_n)$. If $\tilde{X} = X$, this scheme would work well, but instead we have a $\tilde{X}_n - X_n$ error. So, we correct for this with the following scheme.

$$\tilde{X}_{n+1} = \mathcal{R}_{\tilde{X}_n} \left(\mathscr{P}_{\mathcal{T}_{\tilde{X}} \mathscr{M}_r} \left[\overline{\mathscr{L}}(t, A(t)) + \frac{X_n - \tilde{X}_n}{\Delta t} \right] \right)$$

This corrects the direction of $\overline{\mathscr{L}}$ for the point which we are actually at. Of course, the full-rank derivative and X_n are typically not known, but for the pedagogical purposes, we show that the error indeed vanishes with this scheme, written out in algorithm 6. Figure 3-10 indeed shows that the perturbative retractions converge to the best approximation.

Algorithm 6: Sixth-order integrator with full-rank derivative information corrected for the current point

Input: $X \in \mathbb{R}^{m\times n}, \dot{X} \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{m\times n}, \dot{A} \in \mathbb{R}^{m\times n}, \Delta t \in \mathbb{R}, F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Output: $X, \dot{X}$

$\dot{A} \leftarrow \dot{A} + \frac{X - \mathscr{P}_{\mathscr{M}_r} A}{\Delta t}$

$X = X + \Delta t a_1 \dot{A}$

$\dot{X} = \dot{A} + \Delta t b_1 F(A + \Delta t a_1 \dot{A})$

for $i = 2 : 7$ **do**

  $X = X + \Delta t a_i \dot{X}$

  $\dot{X} = \dot{X} + \Delta t b_i F(X)$

$X = X + \Delta t a_8 \dot{X}$

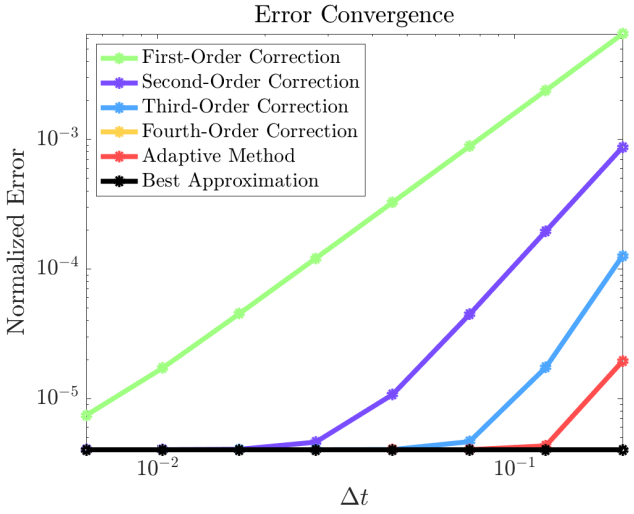


Figure 3-10: Convergence plot for perturbative retractions given corrected full-rank derivative information

Algorithm 7: Sixth-order integrator with internal retractions

Input: $X \in \mathbb{R}^{m\times n}, \dot{X} \in \mathbb{R}^{m\times n}, \Delta t \in \mathbb{R}, F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Output: $X, \dot{X}$

for $i = 1 : 7$ **do**

  $X = \texttt{reOrth}^a(\mathcal{R}_X(\Delta t a_i \dot{X}))$

  $\dot{X} = \texttt{reOrth}(\mathcal{R}_{\dot{X}}(\Delta t b_i F(X)))$

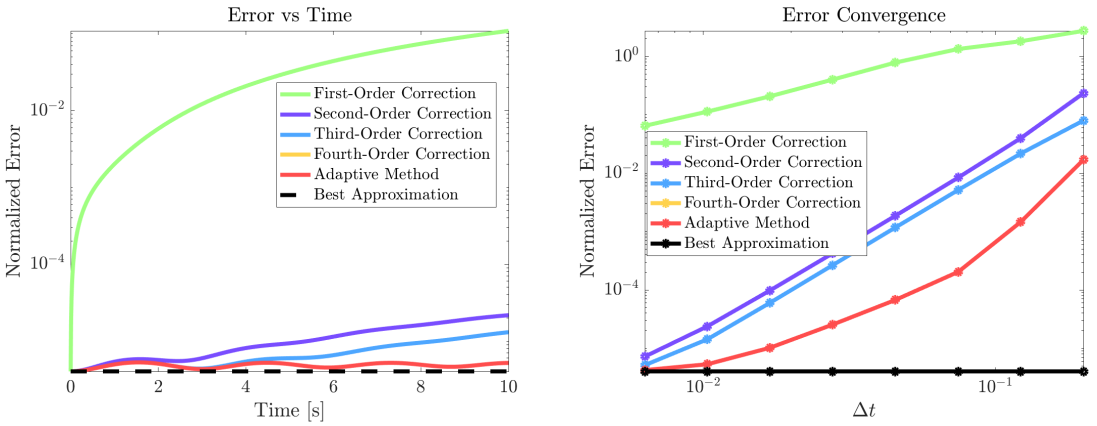
$X = \mathscr{P}_{\mathscr{M}_r}(X + \Delta t a_8 \dot{X})$

[a] See Appendix D.1 for the re-orthonormalization algorithm.

Now we consider another use case for retractions: using them internally in the calculation of $\overline{\mathscr{L}}$. This is desirable because if $\overline{\mathscr{L}}$ contains many additions (where the ranks of the summands add) and/or Hadamard products (where the ranks of the product terms multiply), retracting $\overline{\mathscr{L}}$ back to the manifold may be extremely expensive since the rank will be huge. In the Yoshida integrator, we have 15 integration steps, all of which have additions. What's more, the ranks will compound since each integration step depends on the last. Needless to say, retracting back to the manifold internally in the calculation of $\overline{\mathscr{L}}$ is well-motivated and may in some cases be necessary. This idea is investigated in Runge-Kutta methods in [49]. So, we consider algorithm 7 as an integrator. For a fair comparison, for the last retraction of X we use the extended projective retraction (i.e. the projection onto the low-rank manifold via truncated SVD) no matter the internal retraction chosen.

Figure 3-11b depicts the convergence of using the internal retractions. The first-order correction exhibits first-order convergence. The second- and third-order corrections exhibit third-order convergence, and the fourth-order and adaptive methods exhibit fifth-order convergence, at least at first. Recall that just because a numerical method is n-th order does not mean it cannot exhibit higher-order convergence – the coefficient on the $\mathcal{O}(\Delta t^n)$ term may be very small and not affect the overall convergence order for relatively large Δt. Interestingly, figure 3-11a shows the error from the third- and fourth-order methods separate (in contrast to figure 3-9a). So, using a higher-order method in internal retractions may be more important than when just using them once per time step; this could be because we are applying the retraction so many times in each step of integration.

Lastly, we consider a method with internal retractions and full-rank derivative information provided. There is an important, non-obvious distinction between algorithms 8 and 7. We must project $\dot{A}$ onto the low-rank manifold so that we can use it to retract in the first integration step of $\dot{X}$, letting $\hat{\dot{A}} = \mathscr{P}_{\mathscr{M}_r}\dot{A}$ This may prove



(a) Error as function of time with internal retractions, $\Delta t = 0.01$

(b) Convergence of perturbative retractions used internally (see algorithm 7) and the extended projective retraction as the final step

Figure 3-11: Error plots when using algorithm 7

too expensive to implement in some examples. An approximate method would be to use $\mathcal{R}_{\dot{X}}(\Delta t b_1 F(Y + \Delta t a_1 \dot{A}))$ instead of $\mathcal{R}_{\hat{\dot{A}}}(\Delta t b_1 F(A + \Delta t a_1 \dot{A}))$; however, this may induce error that ruins higher-order convergence.

Algorithm 8: Sixth-order integrator with internal retractions and full-rank derivative information

Input: $X \in \mathbb{R}^{m\times n}, \dot{X} \in \mathbb{R}^{m\times n}, A \in \mathbb{R}^{m\times n}, \dot{A} \in \mathbb{R}^{m\times n}, \Delta t \in \mathbb{R}, F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$

Output: $X, \dot{X}$

$\hat{\dot{A}} = \mathscr{P}_{\mathscr{M}_r} \dot{A}$

$X = \texttt{reOrth}^a(\mathcal{R}_X(\Delta t a_1 \dot{A}))$

$\dot{X} = \texttt{reOrth}(\mathcal{R}_{\hat{\dot{A}}}(\Delta t b_1 F(A + \Delta t a_1 \dot{A})))$

for $i = 2 : 7$ **do**

  $X = \texttt{reOrth}(\mathcal{R}_X(\Delta t a_i \dot{X}))$

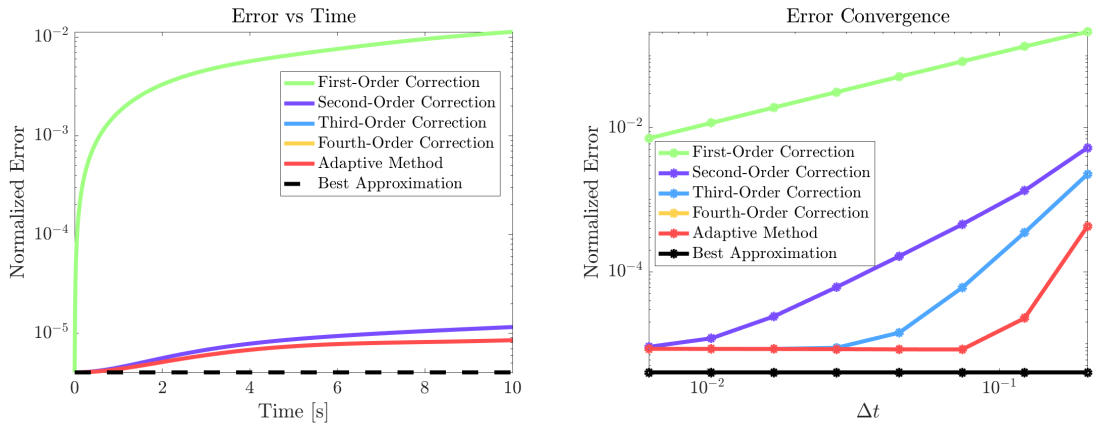
  $\dot{X} = \texttt{reOrth}(\mathcal{R}_{\dot{X}}(\Delta t b_i F(X)))$

$X = \mathscr{P}_{\mathscr{M}_r}(X + \Delta t a_8 \dot{X})$

[a] See Appendix D.1 for the re-orthonormalization algorithm.

Figure 3-12a again shows the error oscillations in time are gone, and figure 3-12b shows that the error does not decay all the way down to the best approximation for the same reasons as before. One may employ the same full-rank derivative correction scheme for better results (not depicted). For the first-order retraction, we see first-order convergence. For the second-order retraction, we have convergence some-

where between second- and third-order, and for the third-order retraction, we have convergence somewhere between third- and fourth-order. The fourth-order and adaptive methods exhibit sixth-order convergence, at least for relatively large Δt before it quickly converges to its minimum error value.



(a) Error as function of time with internal retractions, $\Delta t = 0.01$

(b) Convergence of perturbative retractions used internally (see algorithm 8) and the extended projective retraction as the final step

Figure 3-12: Error plots when using algorithm 8

To calculate the orders of convergence, we use three metrics different measures of the slope of the log-log plots. We calculate the slope between the first and second points, the second-to-last and last points, and the average slope over the whole Δt interval. These, in general, give different values since in some circumstances, the error converges to a fixed value for small Δt; at the other end, the asymptotic analysis may not hold for the larger values of Δt. So, each value must be taken in context. For each of the aforementioned algorithms and retractions, the orders of convergence are tabulated in tables 3.1, 3.2, and 3.3.

Table 3.1: Convergence order calculated from the errors at the largest two Δt values

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.2119	2.0524	2.9879	2.8496	2.8496
Full-rank derivative	0.98083	1.9987	3.0508	4.0366	4.0366
Corrected derivative	2.0303	3.0162	4.0079	3.0395	3.0395
Internal retractions	0.82845	3.5909	2.7764	4.961	4.961
Internal + full-rank	0.94297	2.7632	3.6729	5.9292	5.9292

Table 3.2: Convergence order calculated from the errors at the smallest two Δt values

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.0159	1.9977	0.8736	0.86432	0.86432
Full-rank derivative	1.0015	1.8862	-6.7847E-4	-3.2331E-3	-3.2331E-3
Corrected derivative	1.7054	7.9466E-4	4.2583E-08	2.4139E-11	2.4139E-11
Internal retractions	1.1372	2.4182	2.0712	0.46944	0.46944
Internal + full-rank	0.99995	0.55581	-3.2837E-3	-3.5073E-3	-3.5073E-3

Table 3.3: Convergence order calculated from the errors over the whole Δt interval

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.0941	2.0097	2.3137	1.9403	1.9403
Full-rank derivative	0.9984	1.985	1.7309	1.1777	1.1777
Corrected derivative	1.9605	1.555	0.99643	0.455	0.455
Internal retractions	1.0839	3.0058	2.7925	2.4003	2.4003
Internal + full-rank	0.98458	1.8407	1.6145	1.1331	1.1331

3.4 Stochastic partial differential equations

Here, we apply the new retractions to a stochastic partial differential equation. We consider a variant of Burgers' equation [50, 51] with periodic boundary conditions.

$$\frac{\partial u}{\partial t} + \beta(\omega) u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} + f(x,t;\omega), \quad x \in \mathcal{D} = [-1,1], \quad t \geq 0, \quad \omega \in \Omega$$
$$u(-1) = u(1), \quad \left.\frac{\partial u}{\partial x}\right|_{x=-1} = \left.\frac{\partial u}{\partial x}\right|_{x=1}, \quad u(x,0;\omega) = u_0(x;\omega)$$

Above, we've introduced a new stochastic parameter β that scales the nonlinear advection speed where ω denotes a simple event in the event space Ω. The forcing function is also stochastic, as are the initial conditions. The kinematic viscosity ν is deterministic and taken to be 0.01 – allowing ν to vary stochastically could potentially lead to numerical instabilities due to shock waves. For our example, we let β be uniformly distributed from $-\frac{3}{4}$ to $\frac{3}{4}$, i.e. $\beta \sim U\left(-\frac{3}{4}, \frac{3}{4}\right)$. It's important to not let β become too large, otherwise the numerical scheme, again, becomes unstable due to shocks. The forcing function is described as two stochastic Gaussian sources/sinks that oscillate in time. That is,

$$f(x,t;\omega) = \frac{1}{100}\gamma_1(\omega)\sin(t)e^{-100\left(x+\frac{1}{2}\right)^2} + \frac{1}{100}\gamma_2(\omega)\cos(t)e^{-64\left(x-\frac{1}{2}\right)^2}$$

where $\gamma_1, \gamma_2 \sim U\left(-\frac{3}{4}, \frac{3}{4}\right)$. Lastly, we'll define our initial conditions as follows. First define a function

$$g(x) = \frac{1}{2}\text{sinc}(4\pi x) + \frac{3}{4}\sin(15\pi x),$$

where $\text{sinc}(x) = \frac{\sin(x)}{x}$. It is clear that at $g(\pm 1) = 0$, so g can be represented with a shifted Fourier sine series on the interval $[-1, 1]$. Note that it in the continuous sense, it may seem more natural to represent g with a Fourier cosine series (that is not shifted) since $\text{sinc}(x)$ is even about zero. However, when we go to the discrete problem, we have a discrete set points of $\{g_k\}_{k=1}^N$ (with N equal to the number of discretized points in x) which start and end at zero, and the most natural representation is given by the discrete sine transform. Let $\hat{g}_q = \text{DST}(g_k)$ denote the coefficients of the discrete sine transform of g_k. Then, we define u_0 as follows.

$$u_0 = \text{IDST}\left(\hat{g}_q\left(1 + \frac{1}{\sqrt{q}}\varepsilon_q(\omega)\right)\right)$$

IDST stands for the inverse discrete sine transform, and $\varepsilon_q \sim \mathcal{N}(0, 1)$. $\beta, \gamma_1, \gamma_2$, and $\{\varepsilon_q\}_q$ are all independent. In defining u_0 like so, we ensure that the initial conditions are smooth, and the high frequencies are not as random as the low frequencies due to the $1/\sqrt{q}$ scaling. This scaling will become relevant later in this section when looking at random realizations of the solution.

First, we solve the PDE in a Monte Carlo (MC) sense, taking $10,000$ random samples and solving them in parallel until $t = T = 10$. We use the following semi-implicit numerical scheme from [52], denoting $u(x_i, t_n) = u_i^n$ for points $(x_i, t_n) \in \mathcal{D} \times [0, T]$.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\beta\frac{u_{i-1}^n + 4u_i^n + u_{i+1}^n}{6}\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} + \nu\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2} + f_i^{n+1} \quad (3.4)$$

This scheme is unconditionally stable. And because the only implicit term is linear, each time step only requires a tridiagonal matrix inversion. Unlike previous cases, the numerical scheme is not in a conducive form for retractions. We typically set $u^{n+1} = \mathcal{R}_{u^n}(\xi)$ for some value ξ, requiring that in our numerical scheme $u^{n+1} = u^n + \xi$.

We may rewrite (3.4) in matrix/vector form by defining the following matrices.

$$D_1 = I + \nu\frac{\Delta t}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & -1 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

$$D_2 = \frac{1}{6}\begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ 1 & 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

$$D_3 = \frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & -1 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Then, our scheme is as follows, letting $\odot$ denote the Hadamard (element-wise) product.

$$u^{n+1} = D_1^{-1}\left(u^n - \beta(D_2u^n)\odot(D_3u^n) + f^{n+1}\right) \tag{3.5}$$

To use the dynamical low-rank approximation, we first consider how to use retractions at each step of the numerical scheme. Let χ denote the right-hand side of (3.5), and let $\tilde{\chi}$ equal the right-hands side of (3.5) without D_1^{-1}. That is, $\tilde{\chi} = -\beta(D_2u^n)\odot(D_3u^n) + f^{n+1}$. We have two options for the retractions. Either $u^{n+1} = \mathcal{R}_{u^n}(\chi - u^n)$, or $u^{n+1} = D_1^{-1}\mathcal{R}_{u^n}(\tilde{\chi})$. The former choice is quite general and may be used for essentially any implicit or nonlinear numerical scheme. The latter choice may be more natural in that the current state is not subtracted from $\tilde{\chi}$; we essentially retract before applying the inverse differential operator D_1. Note, however, that the latter choice would require re-orthonormalization after left-multiplying by D_1^{-1}. The numerical

error from each method of retracting was nearly the same. So for the rest of this example, we will only consider the former retractive method as it is the most general procedure. We will, however, include the numerical errors from the latter retractive method in appendix A.

In this problem, the higher-order retractions became unstable. Although the time integration scheme is unconditionally unstable, that stability analysis does not fully apply to the coupled system that arises from the dynamical low-rank approximation. In evolving the low-rank solutions, the realizations of $\frac{\partial u}{\partial t} = \mathscr{L}$ are, in some sense, mixed by the right-multiplication of $Z(Z^TZ)^{-1}$ in (2.19). When the PDE is nonlinear, the superposition principle does not apply, and we cannot guarantee that linear combinations of solutions will remain stable. Furthermore, when the solution becomes singular, taking the matrix inverse introduces significant numerical noise into the system. This appears as high-frequency noise in the solution, which numerical schemes are often sensitive to; when the scheme blows up, it's typically these high-frequency oscillations that are the root cause of the instability. Others have noted that often with standard integrators, a CFL-like condition arises with the dynamical low-rank approximation that becomes excruciatingly restrictive [53, 54]. This appears to be the situation here – if we take Δt small enough, the higher-order retractions should stabilize because the manifold will locally represent a flat. But, doing so would be extremely computationally expensive. One may ask why the higher-order retractions are more sensitive to large time steps than the first-order retraction. The answer lies in their inherent construction. Because the higher-order retractions build off of the lower-order retractions, any numerical error and/or sensitivity in the lower-order retractions is compounded as the higher-order retractions are computed, especially since each higher-order retraction has another matrix inverse. Both the third- and fourth-order retractions were unstable even for $r = 5$ with 1001 time steps. However, the adaptive method remained stable just as the first- and second-order retractions did. This shows the true strength of the adaptive retraction. In previous examples, the adaptive retraction seemed to always line up with the fourth-order retraction as it was always best; but in this example, it jumps around, often employing the first-order retraction in order to maintain stability. In this example, we again set the hyperparameter of the adaptive method $\varepsilon = 0.025$.

Table 3.4: Normalized error with respect to Monte Carlo run with $r = 5$

	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L^2 Error	0.17615	0.18027	0.1757	0.17685	0.17783
L^2 Mean Error	0.024533	0.023788	0.022944	0.023855	0.023671
L^2 Variance Error	0.01708	0.018528	0.018078	0.017503	0.017781

Table 3.5: Normalized error with respect to Monte Carlo run with $r = 10$

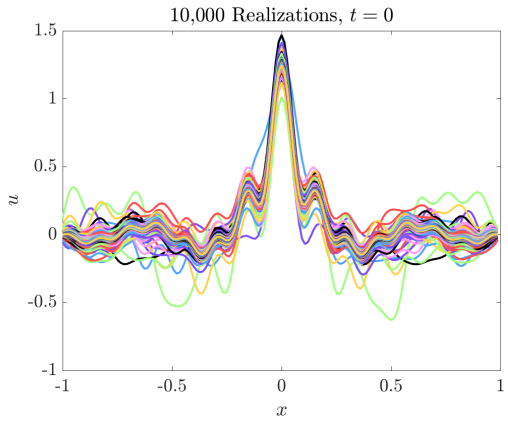
	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L^2 Error	0.016666	0.015258	0.016139	0.016667	0.016595
L^2 Mean Error	1.6508E-3	1.6838E-3	1.3404E-3	1.5466E-3	1.2666E-3
L^2 Variance Error	8.9853E-4	8.0744E-4	8.5267E-4	9.5329E-4	8.3406E-4

Table 3.6: Normalized error with respect to Monte Carlo run with $r = 15$

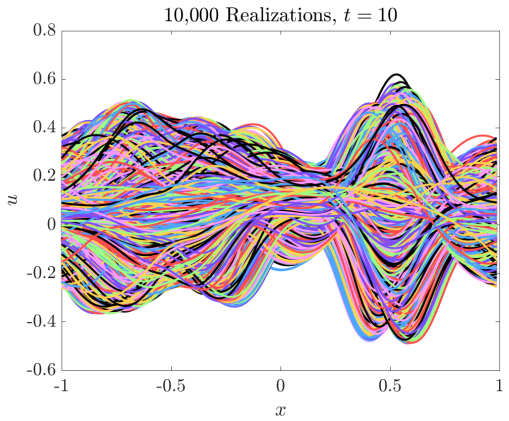
	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L^2 Error	3.5369E-3	4.0246E-3	3.3733E-3	3.5055E-3	1.9144E-3
L^2 Mean Error	8.1585E-4	8.3887E-4	3.2831E-4	4.2303E-4	2.0323E-4
L^2 Variance Error	1.8548E-4	1.2507E-4	1.3786E-4	1.7472E-4	3.4425E-5

In tables 3.4, 3.5, and 3.6, we have three different error metrics. Each error is normalized by the L^2 norm of $g(x)$ defined previously. The mean L^2 error is defined by taking the expected value of the L^2 error between the low-rank approximation and the Monte Carlo simulations. This measures the error realization-by-realization, and L^2 convergence implies convergence in probability. The L^2 mean error is defined as the spatially averaged (in the L^2 sense) error in the sample mean. Similarly, the L^2 variance error is defined as the spatially averaged in the sample variance. These two measures look at the error between the statistical moments of the different simulations and are related to convergence in distribution, or weak convergence. Note that convergence in probability implies convergence in distribution. We can see that the second-order retraction does not universally perform better than the first-order retraction in this case, especially for low rank. This is because it is on the verge of becoming unstable, which introduces numerical error due to overshoot. Similarly, the adaptive, projective, and the extended projective retractions are not universally

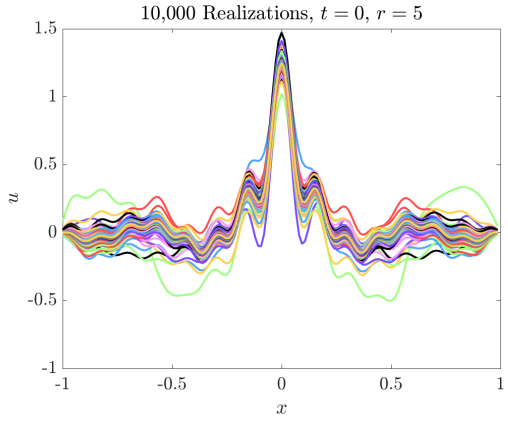
better than the first-order retraction. If we take small enough time steps to stabilize the higher-order methods, there will not be a computational advantage to using the higher-order retractions since the low-rank manifold will locally resemble a flat. And if we take relatively large time steps, then the higher-order retractions become unstable. So in this example where we are restricted by a CFL-like condition, the first-order retraction performs competitively with the others. But in general, without knowledge of a particular system, it seems that the adaptive retraction should be the go-to since it maintains stability and also reaps the benefits of higher-order retractions when possible.



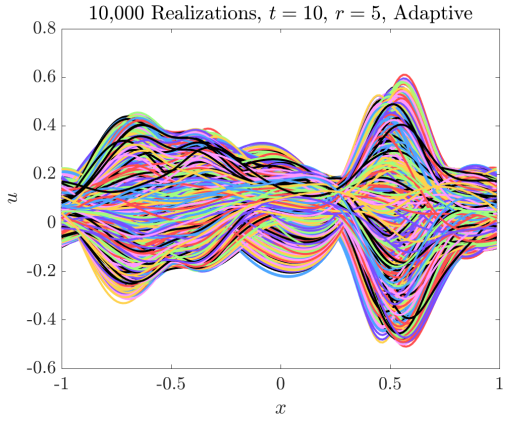
(a) MC at $t = 0$



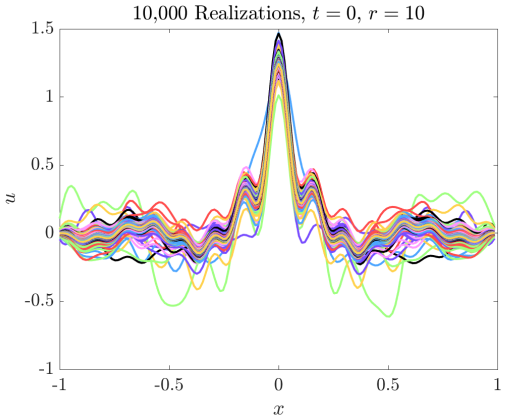
(b) MC at $t = 10$



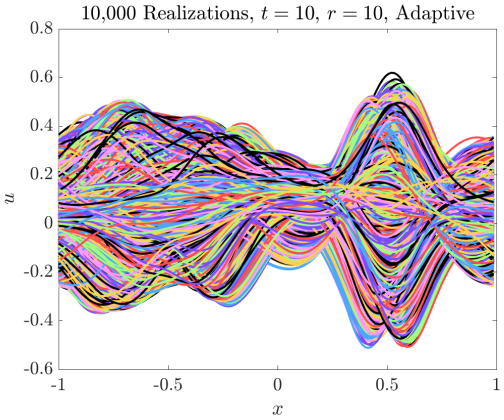
(c) $r = 5$ at $t = 0$



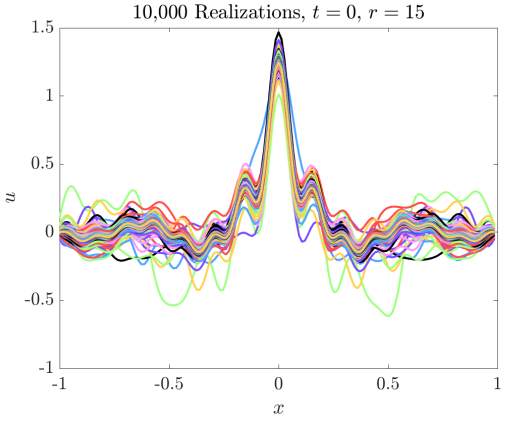
(d) $r = 5$ at $t = 10$



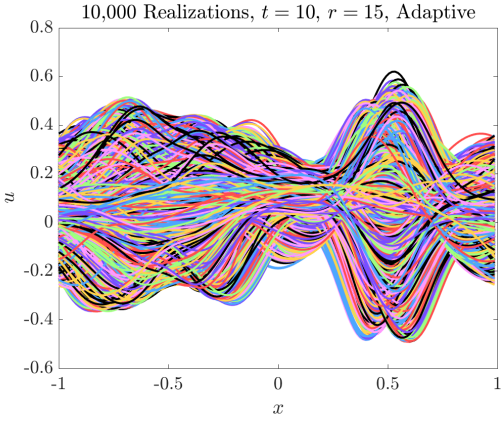
(e) $r = 10$ at $t = 0$



(f) $r = 10$ at $t = 10$



(g) $r = 15$ at $t = 0$

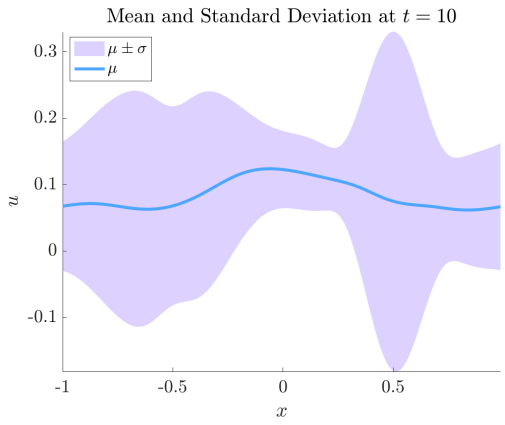


(h) $r = 15$ at $t = 10$

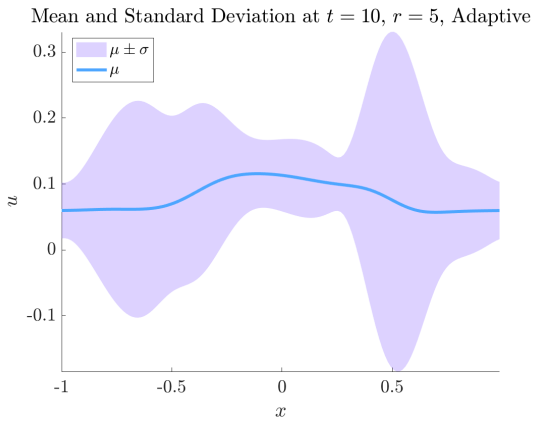
Figure 3-13: Realizations of MC and adaptive retraction solutions

Since no retraction was particularly better, we'll show figures from the adaptive retraction in this section and add the sames figures for the first-order retraction in appendix B. Figure 3-13 shows the Monte Carlo and adaptive retraction realizations at the initial conditions and at $t = 10$. We can see that what started off as just a little stochastic variability in the initial conditions led to a large amount of variability at $t = 10$. While the low-rank ($r = 5$) case captures the general shape of the initial conditions, the high frequencies are not well represented. We can see that the $r = 5$ realizations do not oscillate in the same way as the MC realizations. This is expected since the high-frequency noise is of lower magnitude by construction, so we lose that variability after truncation. It is, however, recovered at $r = 10$ and $r = 15$. At $t = 10$, the $r = 5$ realizations seems to have lost some variability, especially near $x \pm 1$. Again, this is recovered at higher ranks.

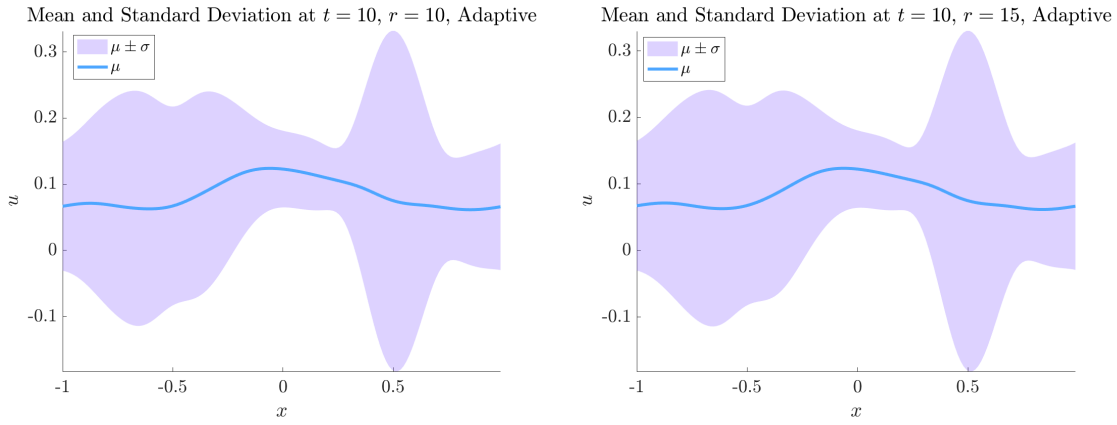
Figure 3-14 plots the marginal mean and standard deviation of the realizations at $t = 10$. Clearly, the stochastic field is non-stationary with respect to x and t. Again, we see that the $r = 5$ simulation underestimates the variance at $x \pm 1$ which is recovered at higher ranks. It also seems that, at least visually, after $r = 10$, there is not much information to gain by increasing the rank.



(a) MC



(b) $r = 5$

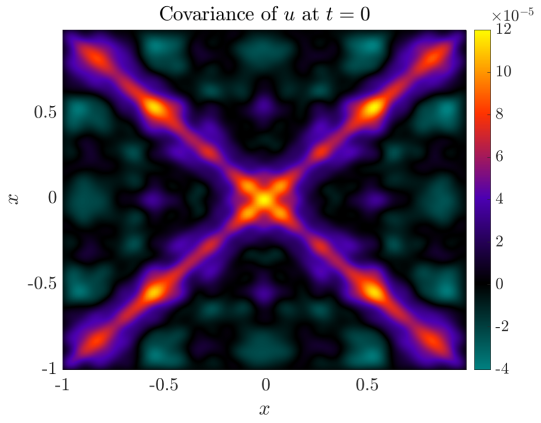


(c) $r = 10$ (d) $r = 15$

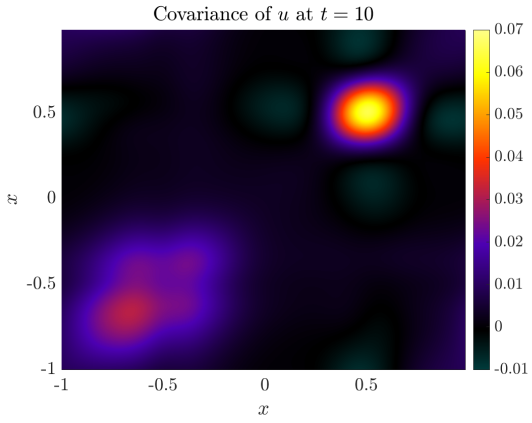
Figure 3-14: Marginal mean and standard deviation of Monte Carlo and adaptive retraction solutions at $t = 10$

Figure 3-15 shows how the spatial covariance changes between $t = 0$ and $t = 10$. The spatial covariance of the initial conditions is a little smeared for $r = 5$, and some details are recovered as the rank is increased. The same is true at $t = 10$, but it is nice to see that the covariance plots do not completely change at low ranks; we just get a coarser picture.

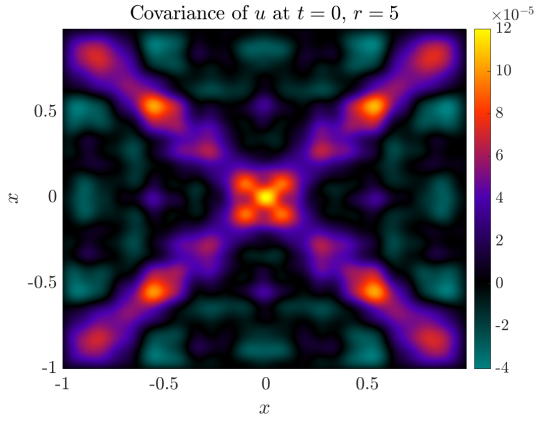
Finally 3-16 shows histograms of the MC runs and the adaptive retraction dynamical low-rank approximation of u at $x = 0$, $t = 10$. At low-rank, we see that the histogram is certainly different from the MC runs – the low-rank approximation has a narrower distribution. However, at $r = 10$, the distributions match quite well, and they are almost the same at $r = 15$. As such, increasing the rank improves the accuracy of the dynamical low-rank approximation, but even a cheap, very low-rank model captures the main features of a stochastic simulation.



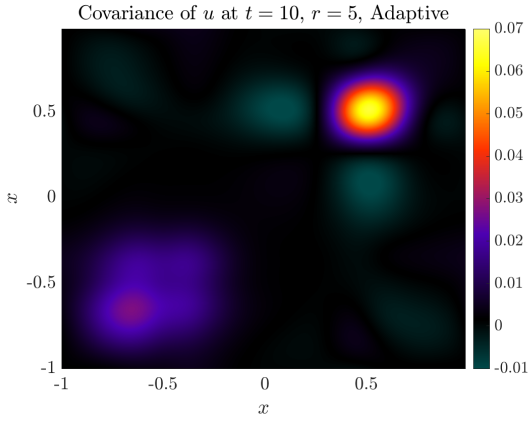
(a) MC at $t = 0$



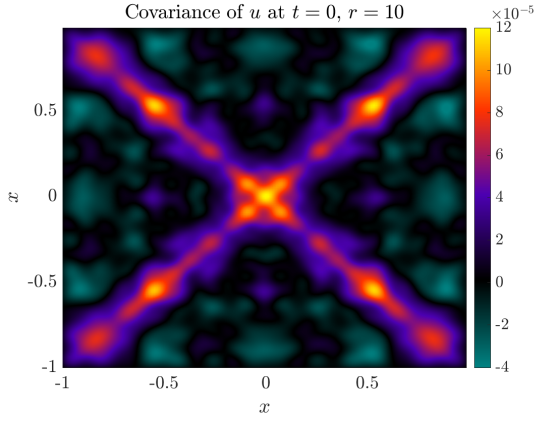
(b) MC at $t = 10$



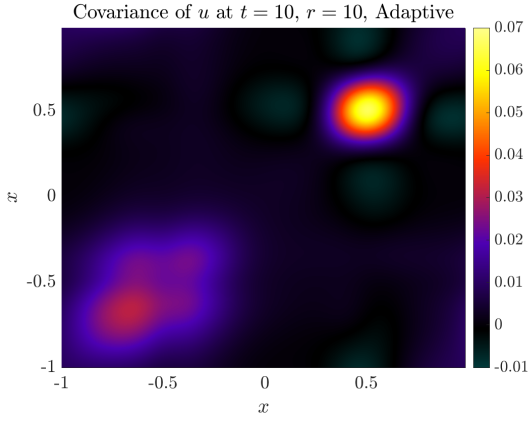
(c) $r = 5$ at $t = 0$



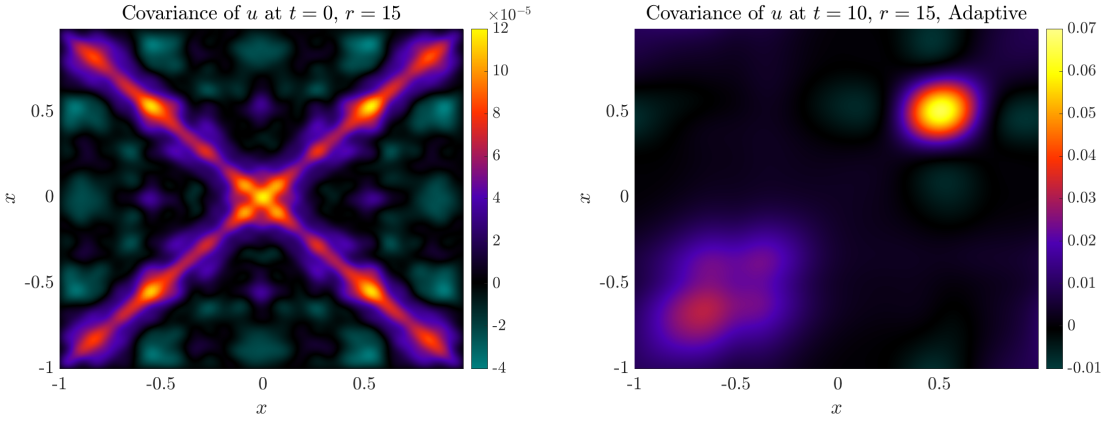
(d) $r = 5$ at $t = 10$



(e) $r = 10$ at $t = 0$

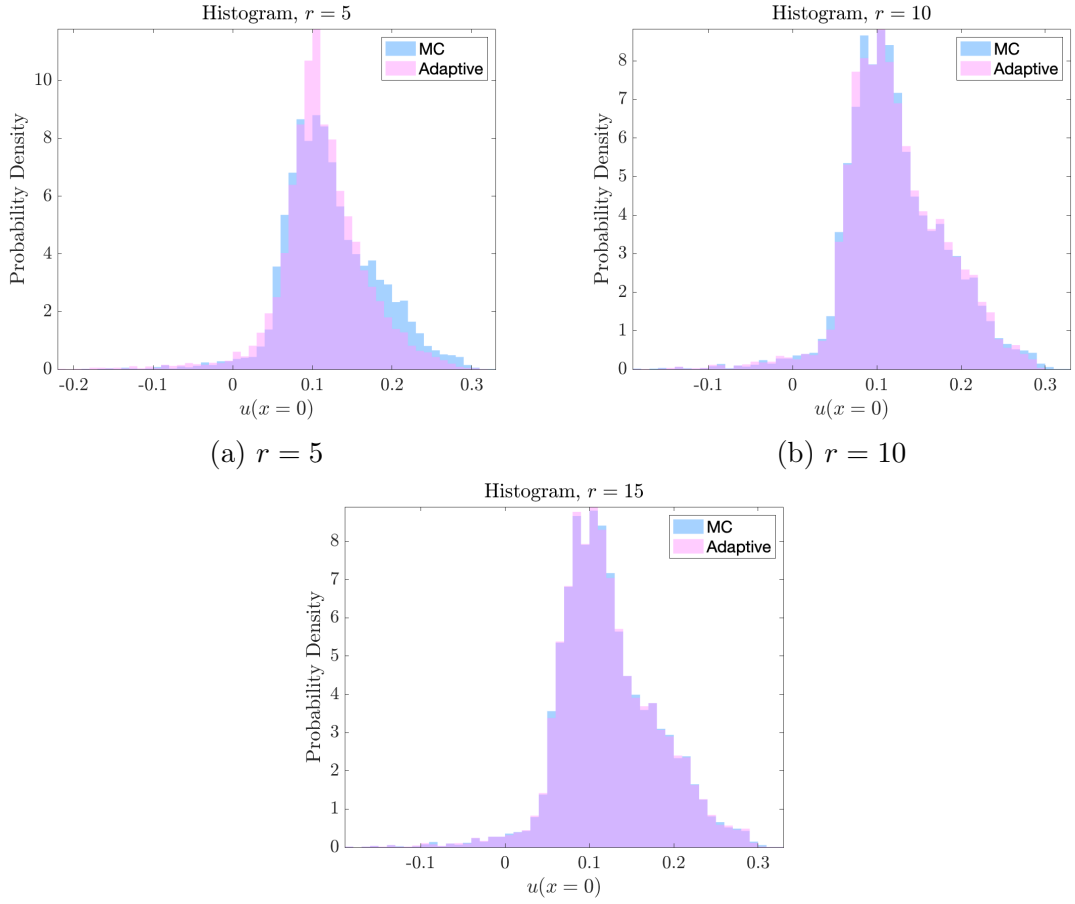


(f) $r = 10$ at $t = 10$



(g) $r = 15$ at $t = 0$ (h) $r = 15$ at $t = 10$

Figure 3-15: Spatial covariance of Monte Carlo and adaptive retraction solutions



(a) $r = 5$ (b) $r = 10$

(c) $r = 15$

Figure 3-16: Histograms of MC and adaptive retraction solutions at $x = 0$, $t = 10$

3.5 Two-dimensional partial differential equations

In this example, we examine a diffusion equation with imaginary diffusivity in two dimensions.

$$\frac{\partial u}{\partial t} = \frac{i}{2k}\nabla^2 u, \quad (x,y) \in \mathcal{D} = [0,250] \times [0,300], \quad t \geq 0$$
$$u|_{\partial\mathcal{D}} = 0, \quad u(x,y,0) = u_0(x,y),$$

We let k vary spatially. On different length and mass scales than solved in this example, this may correspond to the Schrödinger equation in an infinite potential well with spatially-varying mass. This is physically relevant in crystal impurities, semiconductor heterostructure, and more [55, 56, 57, 58, 59, 60]. Alternatively, if we let k be constant and switch t to a range variable such as z, this would correspond to the paraxial (or parabolic) wave equation used extensively in optics [61, 62, 63, 64] and acoustics [65, 66, 67]. Note that in the parabolic wave equation interpretation, we would then be solving a three-dimensional (time-independent) partial differential equation.

We consider times $t \in [0, T = 500]$, and let u_0 be the normalized sum of two-dimensional Gaussians as follows.

$$\begin{aligned}\tilde{u}_0(x,y) = {} & e^{-\left(\frac{x-50}{25}\right)^2} e^{-\left(\frac{y-100}{25}\right)^2} + \frac{1}{2} e^{-\left(\frac{x-75}{25}\right)^2} e^{-\left(\frac{y-100}{25}\right)^2} \\ & - \frac{3}{4} e^{-\left(\frac{x-50}{25}\right)^2} e^{-\left(\frac{y-150}{25}\right)^2} - \frac{2}{3} e^{-\left(\frac{x-175}{15}\right)^2} e^{-\left(\frac{y-200}{15}\right)^2} + \frac{2}{3} e^{-\left(\frac{x-175}{15}\right)^2} e^{-\left(\frac{y-100}{15}\right)^2} \\ & + \frac{1}{4} e^{-\left(\frac{x-75}{15}\right)^2} e^{-\left(\frac{y-100}{15}\right)^2} - \frac{5}{3} e^{-\left(\frac{x-60}{10}\right)^2} e^{-\left(\frac{y-80}{10}\right)^2} + \frac{1}{3} e^{-\left(\frac{x-160}{10}\right)^2} e^{-\left(\frac{y-180}{10}\right)^2} \\ & - \frac{5}{3} e^{-\left(\frac{x-160}{40}\right)^2} e^{-\left(\frac{y-180}{40}\right)^2} + \frac{1}{5} e^{-\left(\frac{x-130}{32}\right)^2} e^{-\left(\frac{y-140}{32}\right)^2}\end{aligned}$$
$$u_0 = \frac{\tilde{u}_0}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\tilde{u}_0(x,y)|^2 dx dy}$$

Furthermore, define k as follows.

$$k(x,y) = \frac{240\pi}{1500 + 500\exp\left(-\left(\frac{x-125}{62.5}\right)^2\right)\exp\left(-\left(\frac{x-150}{75}\right)^2\right)}$$

To solve this problem, we employ the Dufort-Frankel finite difference scheme [68] since it is explicit and unconditionally stable, and we extend it to two dimensions. We denote $u(x_i, y_j, t_n) = u_{ij}^n$ for points $(x_i, y_j, t_n) \in \mathcal{D} \times [0,T]$ and similarly for

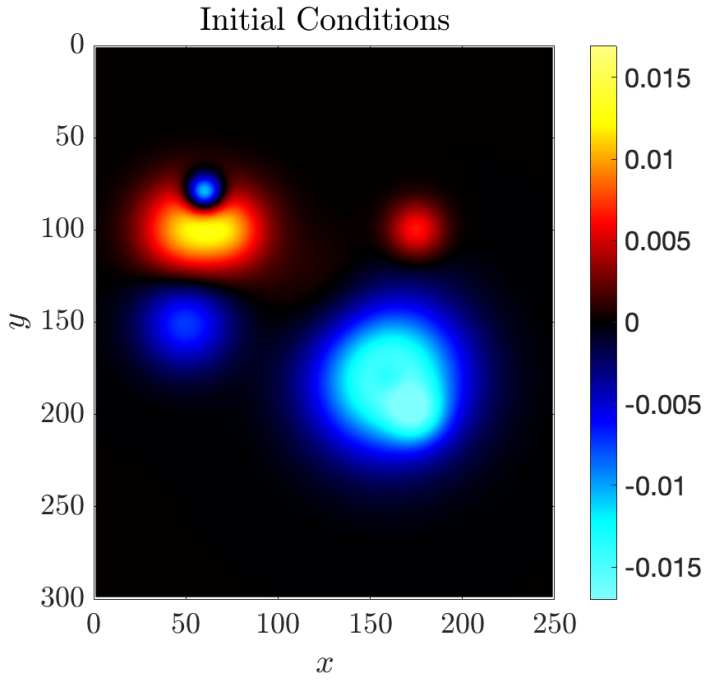


Figure 3-17: Initial conditions for diffusion equation

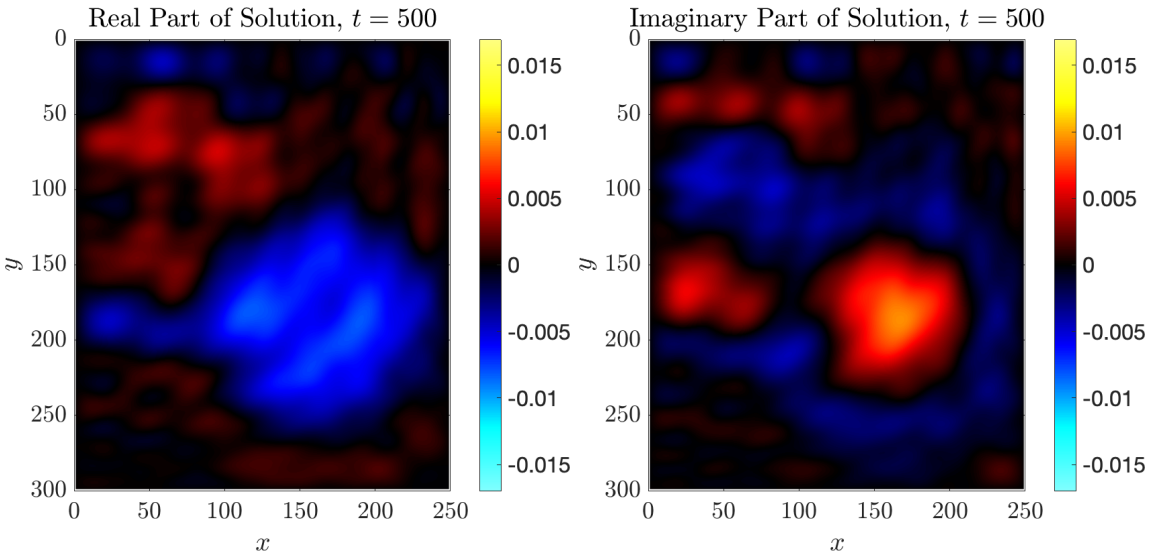
$k(x_i, y_j) = k_{ij}$. Furthermore, let $\lambda_x = \frac{\Delta t}{\Delta x^2}$ and $\lambda_y = \frac{\Delta t}{\Delta y^2}$.

$$u_{ij}^{n+1} = \frac{1}{1 + \frac{i}{k_{ij}}(\lambda_x + \lambda_y)} \left[\frac{i}{k_{ij}} \lambda_x (u_{i-1,j}^n + u_{i+1,j}^n) + \frac{i}{k_{ij}} \lambda_y (u_{i,j-1}^n + u_{i,j+1}^n) + \left(1 - \frac{i}{k_{ij}}(\lambda_x + \lambda_y)\right) u_{ij}^{n-1} \right] \quad (3.6)$$

We remark here that when solving this PDE numerically, the usual reshaping of the 2D solution into a vector is not necessary. Instead, we keep the solution as a matrix u_{ij}. Then to apply a finite difference operator D, e.g.

$$D = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

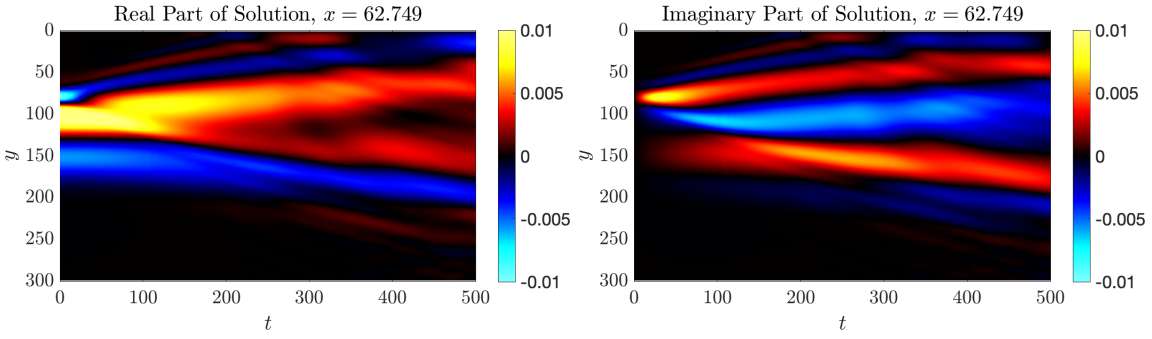
to u_{ij}, one would simply left-multiply D by u_{ij} to compute operations on the index i corresponding to x (i.e. taking linear combinations of the rows of u_{ij}) and right



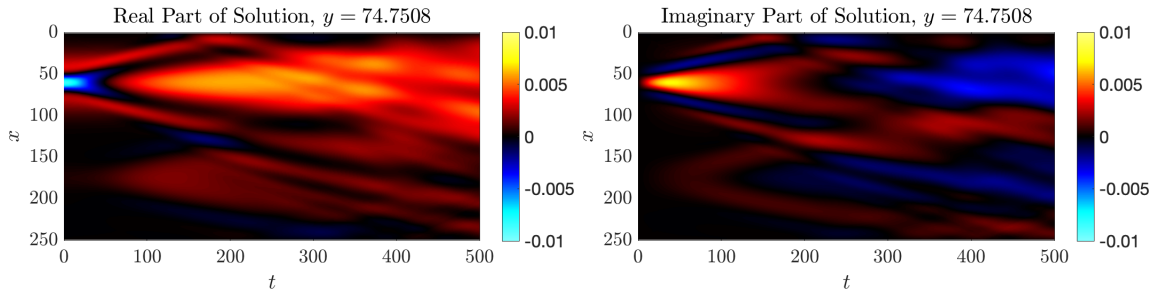
(a) Real part of solution (b) Imaginary part of solution

Figure 3-18: Numerical solution to diffusion equation at final time $t = T$

multiply by D^T to compute operations on the index j corresponding to y (taking linear combinations of the columns of u_{ij}). Lastly, notice that the solution is complex-valued, so this example demonstrates that these retractions work just as well on the Hermitian manifold as the Riemannian manifold. In short, the only way the algorithms change is that the transpose operator becomes the conjugate transpose operator. Figure 3-18 depicts the full-rank solution at $t = 500$. This will be a reference as to how low-rank solutions compare using different retractions. We may also look at different slices of the solution, holding x or y constant, depicted in figure 3-19.



(a) Real slice of solution, x constant (b) Imaginary slice of solution, x constant



(c) Real slice of solution, y constant (d) Imaginary slice of solution, y constant

Figure 3-19: Slices of solution to diffusion equation holding x or y constant

Now, we investigate the low-rank solutions given by various retractions at ranks $r = 1, 2, 4, 8$ with the adaptive hyperparameter $\varepsilon = 0.025$. Note that as in the stochastic PDE example, the scheme (3.6) is not directly in a form conducive to a retraction. Letting χ denote the right-hand side of (3.6), for this scheme, we let $u^{n+1} = \mathcal{R}_{u^{n-1}}(\chi - u^{n-1})$.

In figure 3-20, we show how the time-averaged error converges as we increase the rank of the solutions. Figure 3-21 shows how the error evolves over time for different ranks. The normalized error plotted is measured by the Frobenius norm of the difference between the full-rank and low-rank solutions normalized by the norm of the initial conditions (which is one in this case). At very low ranks of $r = 1, 2$, the retractions are indistinguishable. This is because the modeling error overwhelms any retraction error. But, at $r = 4$ and 8, we see the modeling error from the low-rank approximation becomes small enough for us to distinguish the first and second-order retractions from the rest.

From this, it is clear that past the second-order retraction, it will not be visually distinguishable which retraction we choose. So for the next figures, we'll only show examples from the adaptive scheme, and we'll include examples from the first-order scheme in appendix B. While the adaptive scheme is more accurate than the first-order scheme, the visual differences in the plots are marginal – essentially only the interference patterns are slightly different. The goal here is more to show how the solution becomes more accurate with larger rank than to compare the retractions.

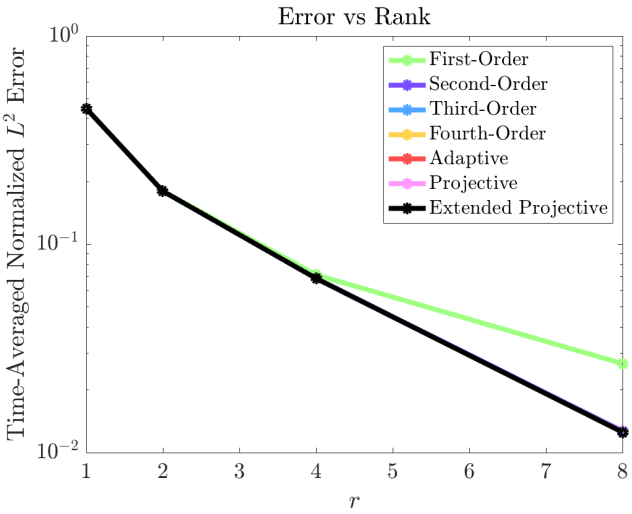
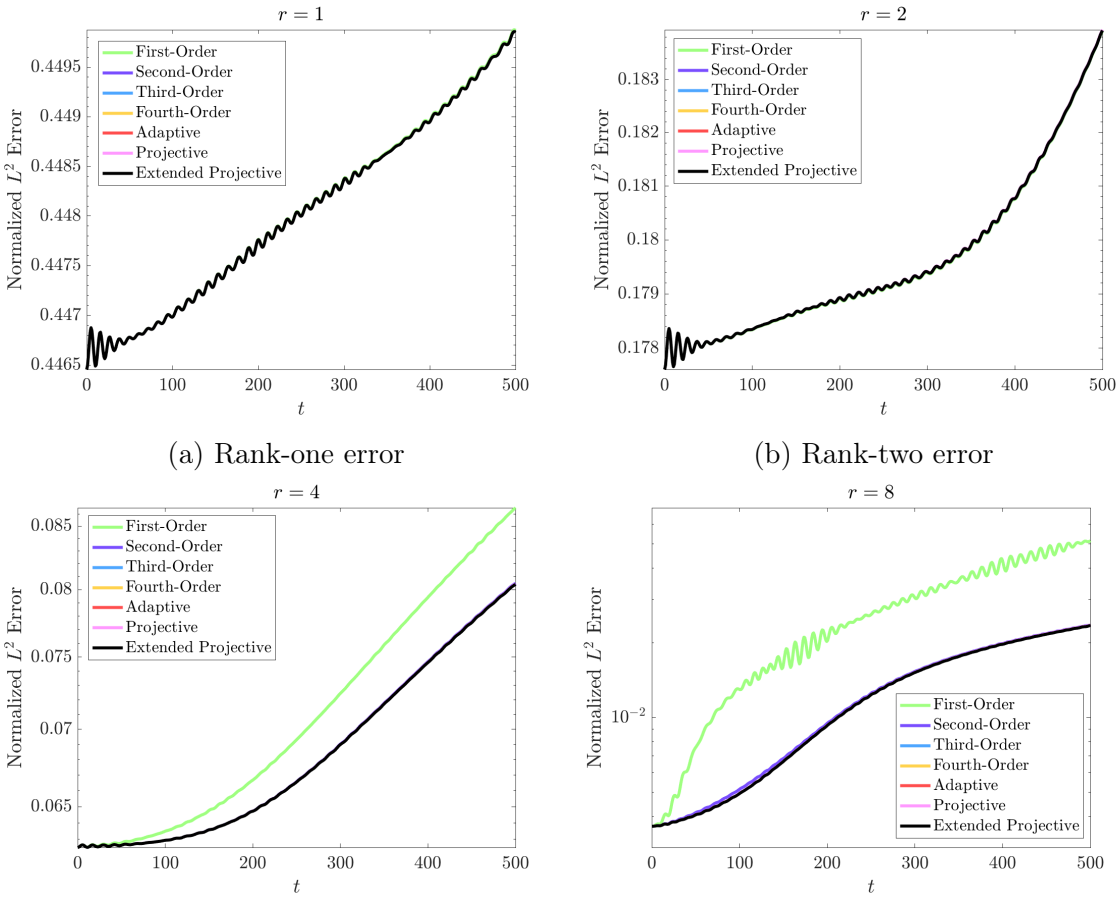


Figure 3-20: Time-averaged error for retractions at different rank



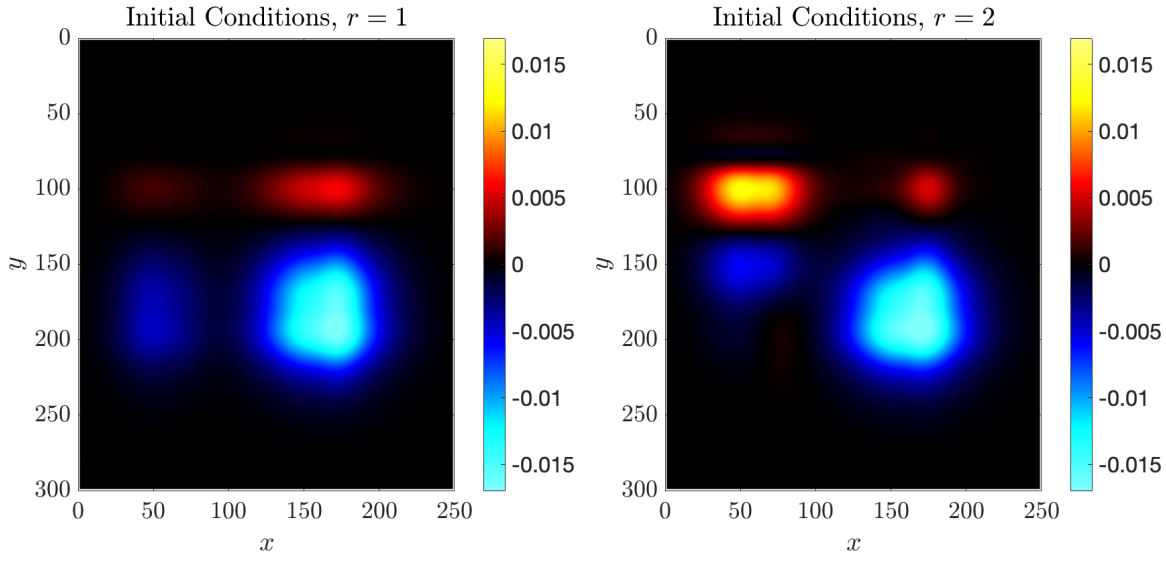
(a) Rank-one error

(b) Rank-two error

(c) Rank-four error

(d) Rank-eight error

Figure 3-21: Error vs time for retractions at $r = 1, 2, 4, 8$



(a) $r = 1$

(b) $r = 2$

(c) $r = 4$

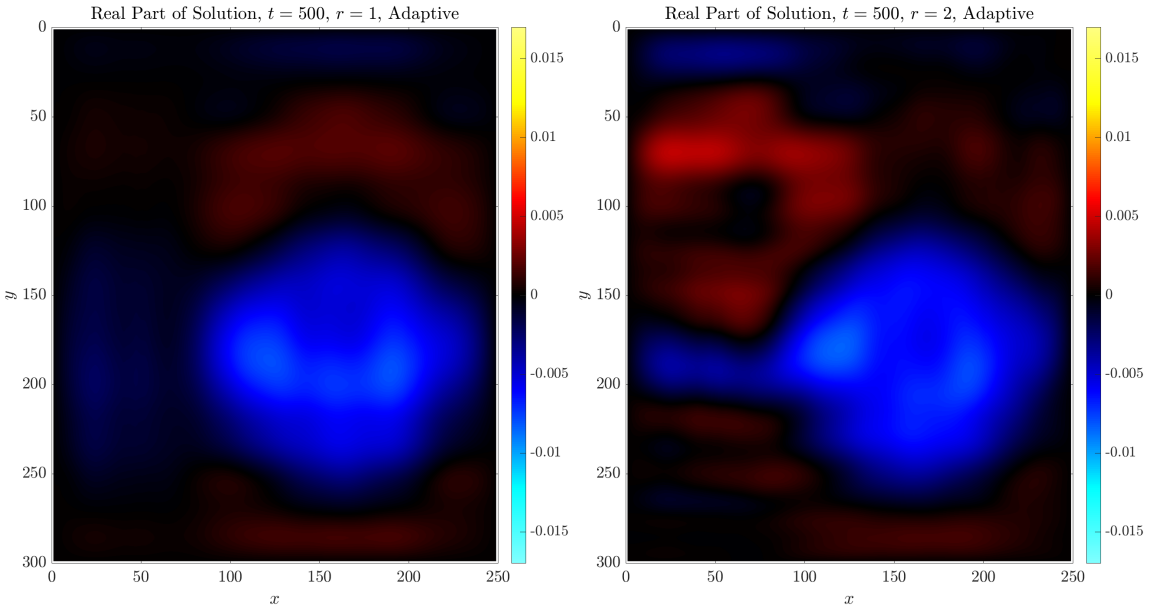
(d) $r = 8$

Figure 3-22: Initial conditions at $r = 1, 2, 4, 8$

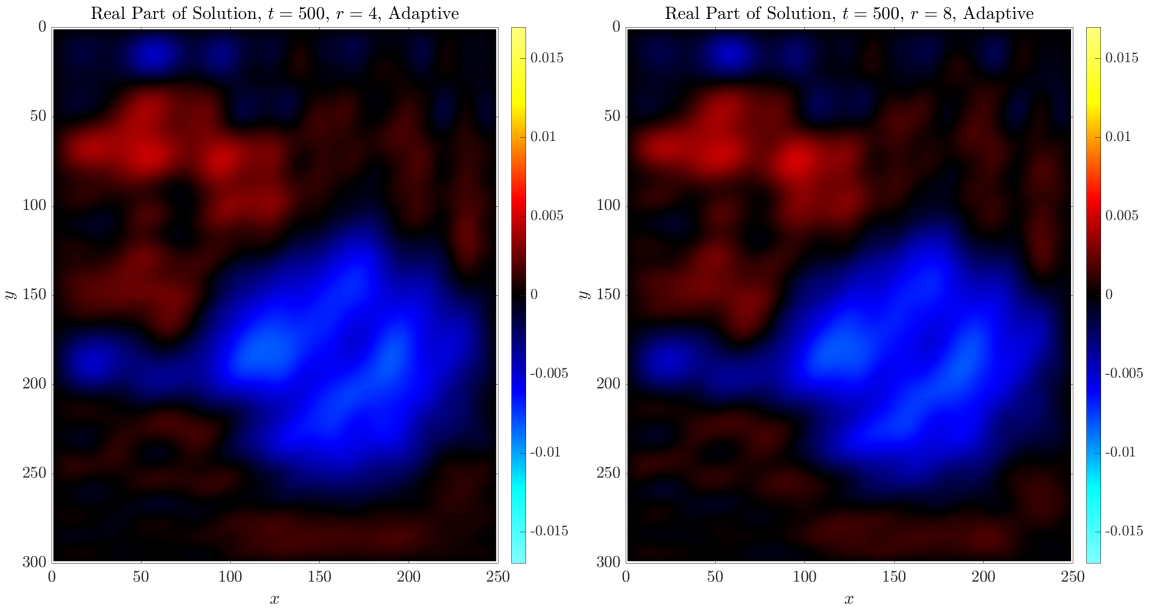
Figure 3-22 shows how the truncated rank affects the initial conditions. We can see that by $r = 8$, we have recovered essentially the full solution. The initial conditions with $r = 4$ is close to correct, but there are some visible aberrations in the blue splotch at the bottom right.

Figures 3-23 and 3-24 show how rank affects the final solution. The low-rank solutions are not able to capture the interference pattern from wave interactions. But it is nice to see that the solution integrity seems to decay smoothly. That is, there is not a large jump in solution correctness with a small change in r. And even at $r = 1$, it is promising that the main solution features are retained.

Finally, we plot slices holding x and y constant in figures 3-25, 3-26, 3-27, 3-28. In these, we see that we don't capture a lot of the energy in the problem at low-rank. One partial remedy would be to renormalize the initial conditions after truncation, but because this is a linear PDE, we would just be rescaling the solution. As the rank grows, the solution slices become more and more detailed. But again, at low rank, the solution still makes sense and seems to preserve the physics.

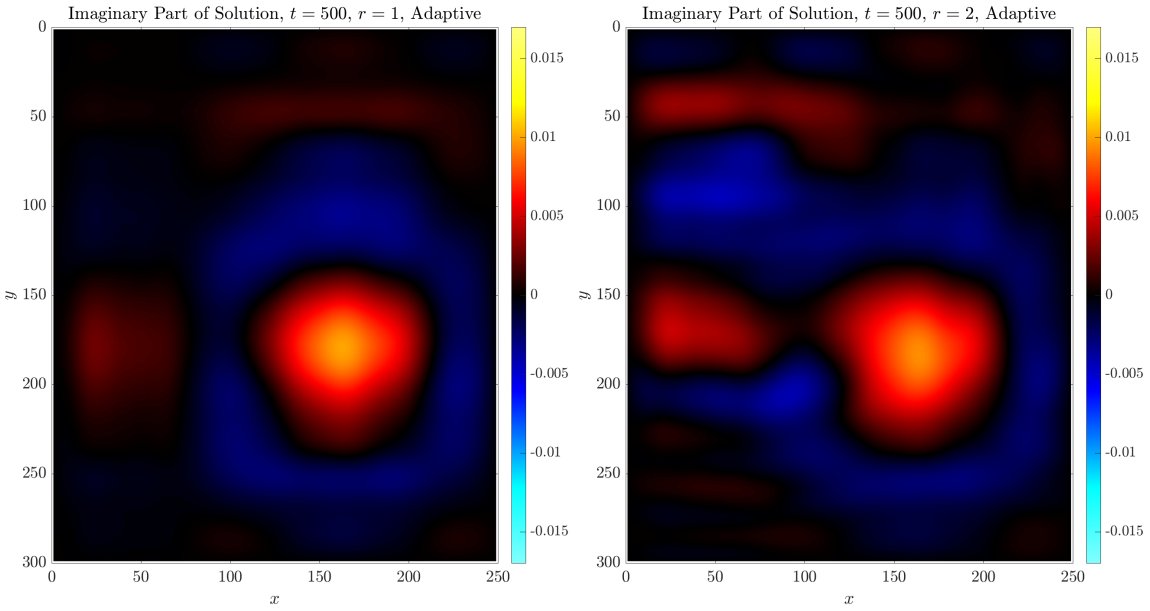


(a) $r = 1$ (b) $r = 2$

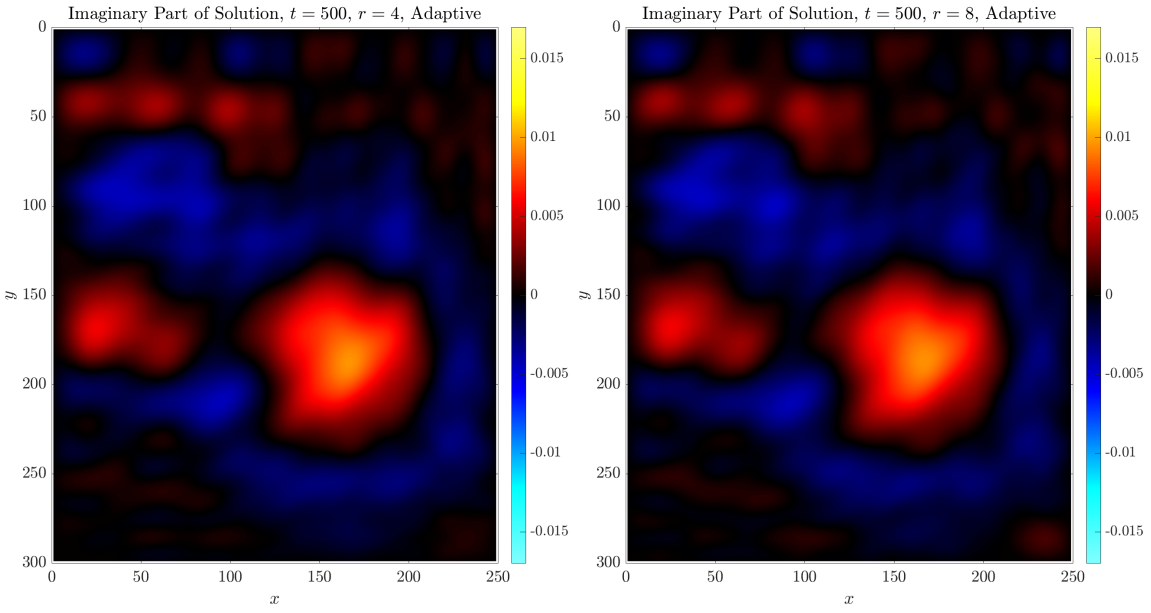


(c) $r = 4$ (d) $r = 8$

Figure 3-23: Real part of solution using adaptive retraction at $r = 1, 2, 4, 8$

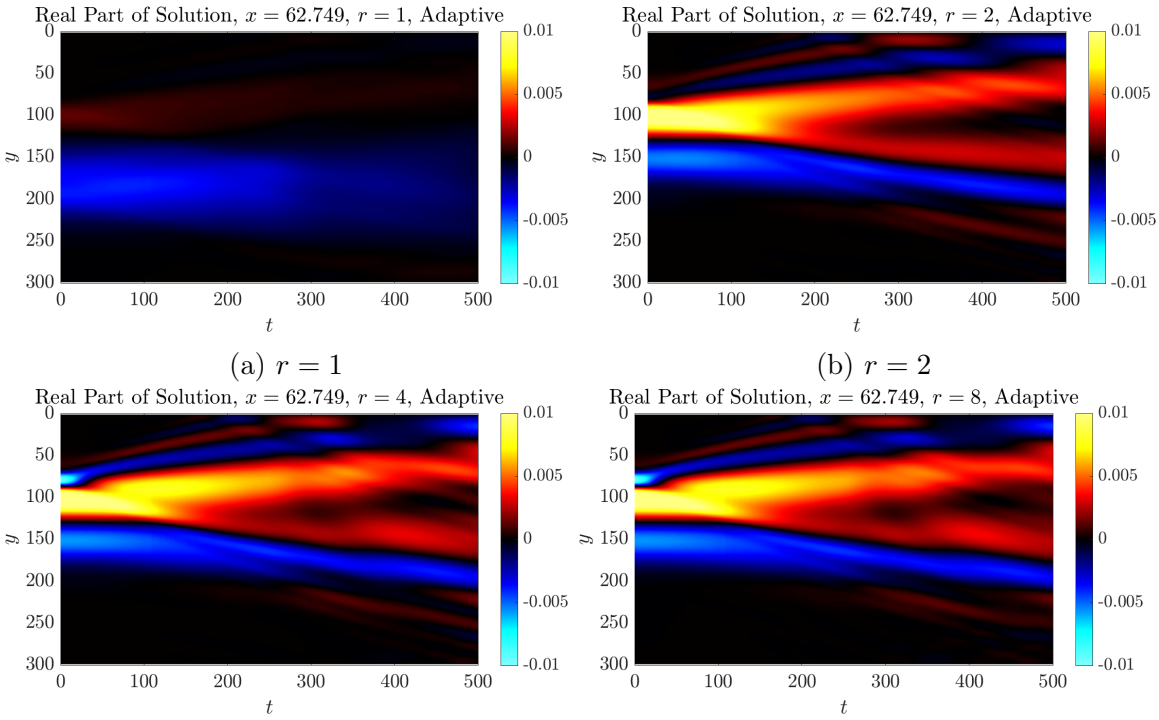


(a) $r = 1$ (b) $r = 2$



(c) $r = 4$ (d) $r = 8$

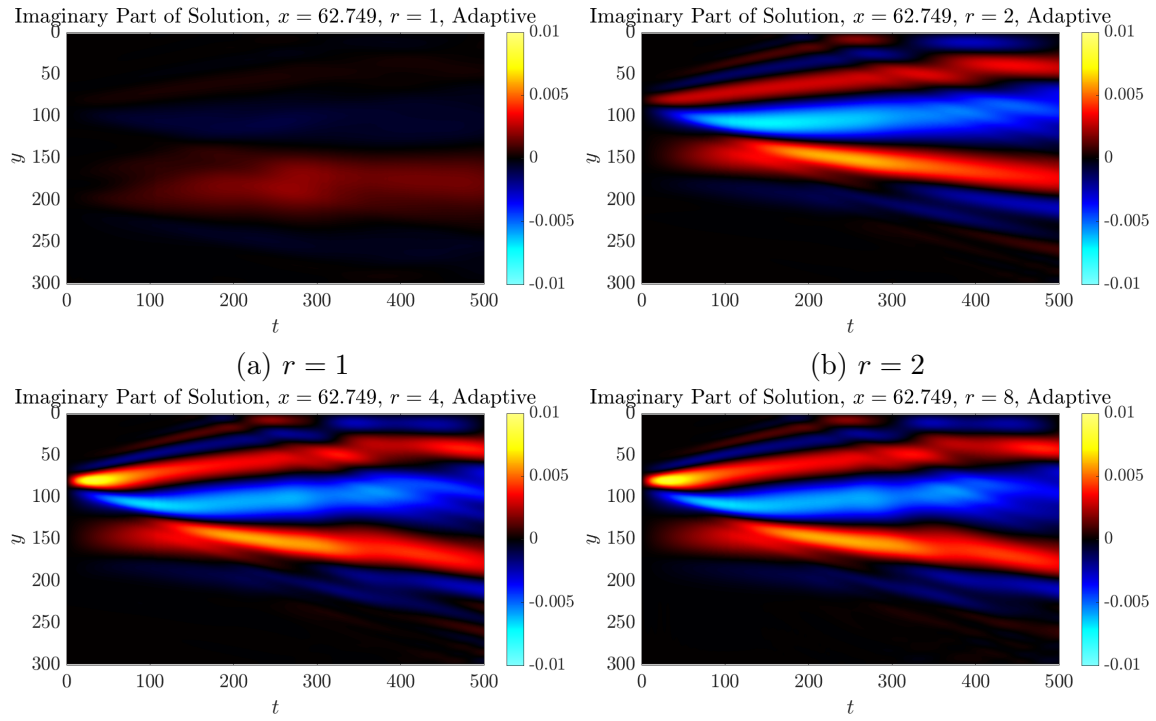
Figure 3-24: Imaginary part of solution using adaptive retraction at $r = 1, 2, 4, 8$



(a) $r = 1$ (b) $r = 2$

(c) $r = 4$ (d) $r = 8$

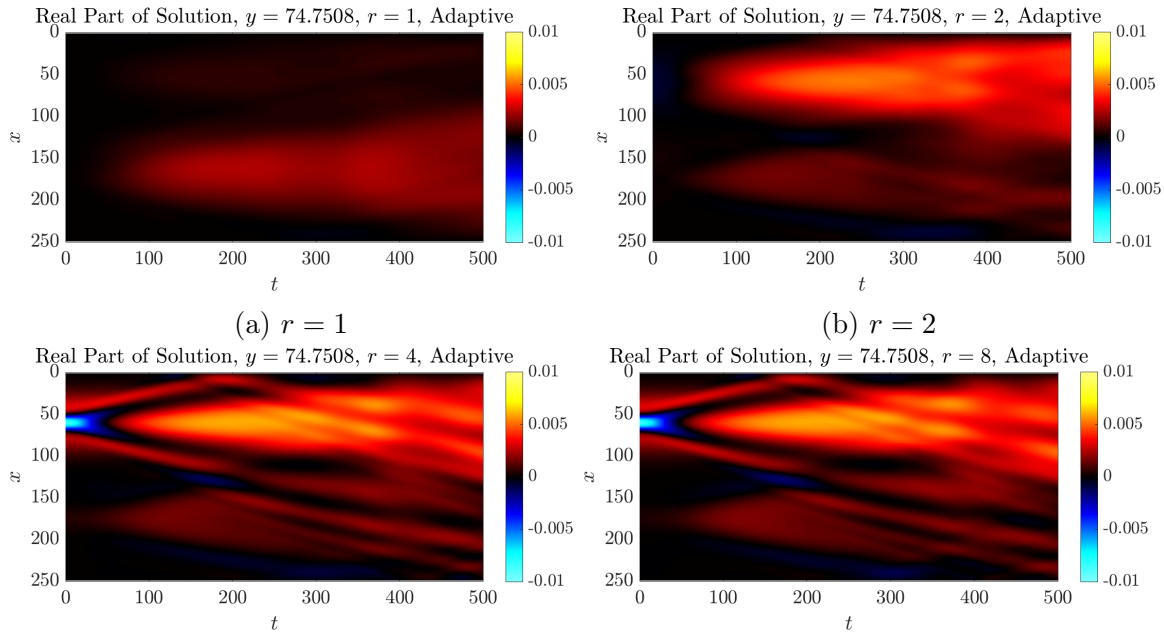
Figure 3-25: Real slices of solution using adaptive retraction at $r = 1, 2, 4, 8$, x constant



(a) $r = 1$ (b) $r = 2$

(c) $r = 4$ (d) $r = 8$

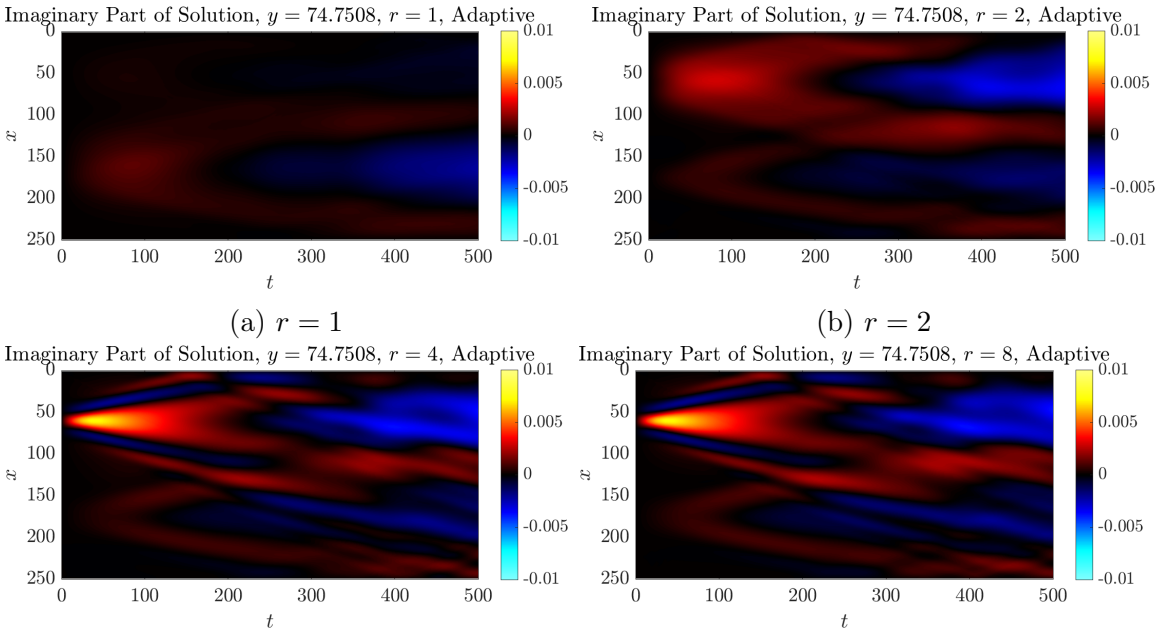
Figure 3-26: Imaginary slices of solution using adaptive retraction at $r = 1, 2, 4, 8$, x constant



(a) $r = 1$ (b) $r = 2$

(c) $r = 4$ (d) $r = 8$

Figure 3-27: Real slices of solution using adaptive retraction at $r = 1, 2, 4, 8$, y constant



(a) $r = 1$

(b) $r = 2$

(c) $r = 4$

(d) $r = 8$

Figure 3-28: Imaginary slices of solution using adaptive retraction at $r = 1, 2, 4, 8$, y constant

Conclusion

In this thesis, chapter one provided an overview of ideas and terminology from differential geometry to familiarize the reader with concepts relevant to retractions. Most importantly, the tangent space of the low-rank manifold was parameterized, and projections onto the low-rank manifold were explicitly characterized. In chapter two, we adapted a projective retraction from the literature to the UZ^T parameterization (rather than the USV^T paramaterization) of a low-rank matrix allowing for a direct implementation without changing parameterizations or diving into complex numerical integration schemes. For partial differential equations of the form $\frac{\partial u}{\partial t} = \mathscr{L}$, where $\mathscr{L}$ may be a stochastic or deterministic operator on u, if $\overline{\mathscr{L}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathscr{L}(t)dt$ factors into a low-rank form $\mathscr{L}_U\mathscr{L}_Z^T$, we derived an efficient extended projective retraction that exactly projects a matrix (up to numerical precision) onto the low-rank manifold. Next, we showed that the perturbative retractions form a dense set on the low-rank manifold implying that any point may be reached arbitrarily closely by these new methods. The perturbative retractions are derived, and in doing so, shown to exhibit high-order convergence to the projection operator without the need for a singular value decomposition; only typical matrix operations are required. In chapter three, we showed that the new projective and perturbative retractions are highly effective for matrix addition and real-time data compression as well as for deterministic and stochastic differential equations. We show higher-order convergence in local error for matrix addition as well as in global error for matrix differential equations. And for the stochastic partial differential equation and the two-dimensional deterministic partial differential equation, we show how even at low-rank, the simulations capture most of the variability and respect the physics of the full-rank solution.

The dynamical low-rank approximation has the potential to vastly speed up simulation run-time. Fewer operations need to be computed, and even writing the low-rank solution at each time step to RAM is much faster due to the reduced solution size. Furthermore, in the case of even moderately large simulations, we cannot store the solution at each time step in an uncompressed form, especially not in RAM. Even

small Monte Carlo simulations can end up being hundreds of gigabytes, and a low-rank representation is necessary if we want to know the solution at intermediate time values. Orthonormality and other properties can also be maintained numerically with appropriate methods [69]. So if we need a low-rank solution at the end, it certainly makes sense to use the dynamical low-rank approximation from the start.

There are, however, some weaknesses of the perturbative retractions. In particular, for uncertainty quantification and stochastic differential equations, the higher Karhunen-Loève modes have high spatial frequency. These can induce instabilities in numerical schemes, especially explicit schemes, and the problem is worsened when the coefficient matrix Z is ill-conditioned. Of course, the time discretization may be refined to avoid instabilities, but this sometimes makes the simulation extremely computationally expensive and can nullify any computational gain from the low-rank representation. These issues are not insurmountable. With regards to ill-conditioned matrices, the rank of the solution may be adaptively increased and decreased using metrics given in [70, 20, 71]. A more rigorous stability analysis of the retractions is necessary to develop sufficient stability criteria similar to the work done on projector-splitting methods [53, 54] in [72]. Implicit time marching schemes would resolve numerical stability issues, but implementing these naively would necessitate reconstructing the big matrix $X = UZ^T$ in order to implement matrix inversion. As such, we have restricted the test cases in this thesis to explicit schemes, which is quite prohibitive, especially for nonlinear problems. In order to implement implicit methods cheaply, an avenue of future research will be how to efficiently invert matrices of the form $I + A$, where A is low-rank. There has already been some work on iterative methods for implicit time marching (see, e.g., [49]).

There are also two clear generalizations to be made from this work. First, higher-order time derivatives may be evaluated using a phase space representation. For example, the wave equation of the form $\frac{\partial^2 u}{\partial t^2} = \nabla^2 u$ may be implemented by letting $v = \frac{\partial u}{\partial t}$. Then we'll have a coupled system of first-order PDEs, and the framework developed may be directly applied. In fact, a symplectic low-rank integrator was proposed to solve the stochastic wave equation in [73]. Second, low-rank tensors are a direction of future research. This would allow us to solve PDEs of arbitrary dimension in low-rank form without vectorizing the physical and stochastic dimensions. Not only would we gain efficiency in not forming huge matrices as a result of vectorization, oftentimes the symmetry of solutions is destroyed when flattening tensors into matrices, and so a low-rank tensor may admit a more efficient representation. These ideas are explored in [74, 75, 54]. In addition to these generalizations, retractions

may be applied to other fields such as constrained optimization problems [76, 77, 78].

Another challenge of these reduced-order methods is when spatially-dependent or stochastic coefficients have large ranks. Even if we reduce the rank of the solution, spatially-dependent or stochastic coefficients numerically express themselves as Hadamard (element-wise) products; the rank of the product is equal to the product of the ranks of factors. As such, the rank of the solution may grow exceptionally quickly when computing $\overline{\mathscr{L}}$ making retractions expensive. Often, the rank of a coefficient however may be reduced via a change of variables. For instance, if we have a spatially-dependent coefficient with a gradient that is slanted with respect to the axes of the problem, a linear transformation aligning the slanted gradient with the axes will greatly reduce the rank. Such decompositions are employed, for example, to represent sound-speed in stochastic underwater sound propagation [79, 80, 81]. An area of future interest is to find the variable transform that will yield the lowest rank for a given spatially-dependent or stochastic coefficient, or perhaps there is a way do a change of variables non-intrusively for reduced-order models. A similar problem is when stochastic coefficients are non-separable. That is, they do not exhibit a low-rank form expressed as the sum of the products of deterministic and stochastic functions. Such is the case when dealing with stochastic bathymetry in ocean acoustics problems (e.g. [82]). As such, perhaps there is another representation that efficiently encodes non-separability.

We emphasize that the discrete dynamically orthogonal equations admit a non-intrusive method, and these new retractions allow for fast and accurate time integration. To implement the examples in this thesis, a new class in MATLAB was written where the basic mathematical operations (e.g. addition, matrix and element-wise multiplication, matrix norms) were overloaded. In addition, the retractions were coded as methods of the class. This allowed for great ease in solving new and different problems. Instead of deriving and discretizing a new set of differential equations for every problem we seek to solve, which can take days of careful calculations and is very error-prone, we can code new problems in a low-rank form as easily as the full-rank form while still reaping the benefits of reduced-order models. Admittedly, there are some problems where deriving the dynamically orthogonal equations intrusively may yield improved efficiency. This can occur through cancellation of certain terms in the differential equation due to orthogonality of the modes and uncorrelatedness of the coefficients. However, any problem with nonlinearity, spatially-dependent coefficients, or stochastic coefficients will likely not have significant cancellations, and these are often the problems of interest. What's more, by intrusively projecting the dynamics of

the PDE onto the low-rank manifold, we lose information about the dynamics in the normal space. This means that, without reformulation and future research to include higher-order approximations to the low-rank manifold, the intrusive methodology is limited to the first-order perturbative retraction and cannot make use of extended retractions. So, not only is the non-intrusive method easier to code and requires much less of a human time investment, there is high-order convergence to be gained. It is our hope that these methods will not only improve reduced-order modeling but will also increase the accessibility of the dynamically orthogonal equations and their variants.

Appendix A

Extra Tables

A.1 Convergence of matrix differential equations

Table A.1: Convergence order calculated from the errors (with respect to the best approximation) at the largest two Δt values

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.2119	2.0524	2.9879	2.8496	2.8496
Full-rank derivative	0.98083	1.9987	3.0509	4.0391	4.0391
Corrected derivative	2.0303	3.0166	4.0627	5.0659	5.0659
Internal retractions	0.82845	3.5909	2.6375	4.961	4.961
Internal + full-rank	0.94297	2.7632	3.7635	5.961	5.961

Table A.2: Convergence order calculated from the errors (with respect to the best approximation) at the smallest two Δt values

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.0159	2.0002	2.0302	2.0015	2.0015
Full-rank derivative	1.0015	1.9035	-8.0789E-4	-3.8509E-3	-3.8509E-3
Corrected derivative	2.0016	3.0025	3.9994	2.0945	2.0945
Internal retractions	1.1372	2.7667	2.9512	2.0012	2.0012
Internal + full-rank	0.99995	0.6551	-4.2241E-3	-4.512E-3	-4.512E-3

Table A.3: Convergence order calculated from the errors (with respect to the best approximation) over the whole Δt interval

	1^{st}-Order	2^{nd}-Order	3^{rd}-Order	4^{th}-Order	Adaptive
Vanilla	1.0941	2.0101	2.5371	2.1614	2.1614
Full-rank derivative	0.9984	1.9879	1.7562	1.203	1.203
Corrected derivative	2.0109	3.0102	4.0212	4.2664	4.2664
Internal retractions	1.0839	3.0597	2.9301	2.7408	2.7408
Internal + full-rank	0.98458	1.8722	1.6508	1.1695	1.1695

A.2 Error in stochastic partial differential equations

Table A.4: Normalized error with respect to Monte Carlo run with $r = 5$ and $u^{n+1} = D_1^{-1}\mathcal{R}_{u^n}(\tilde{\chi})$

	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L_2 Error	0.17734	0.18176	0.17678	0.17853	0.18042
L_2 Mean Error	0.024548	0.022689	0.024476	0.023654	0.023386
L_2 Variance Error	0.017682	0.019197	0.01814	0.018073	0.018764

Note that the second-order retraction performs worse for $r = 5$ since it is on the verge of becoming unstable.

Table A.5: Normalized error with respect to Monte Carlo run with $r = 10$ and $u^{n+1} = D_1^{-1}\mathcal{R}_{u^n}(\tilde{\chi})$

	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L_2 Error	0.016339	0.016271	0.016175	0.01663	0.01597
L_2 Mean Error	1.314E-3	1.3063E-3	1.1613E-3	1.4789E-3	1.0882E-3
L_2 Variance Error	9.2795E-4	8.5754E-4	8.5514E-4	9.5298E-4	8.1713E-4

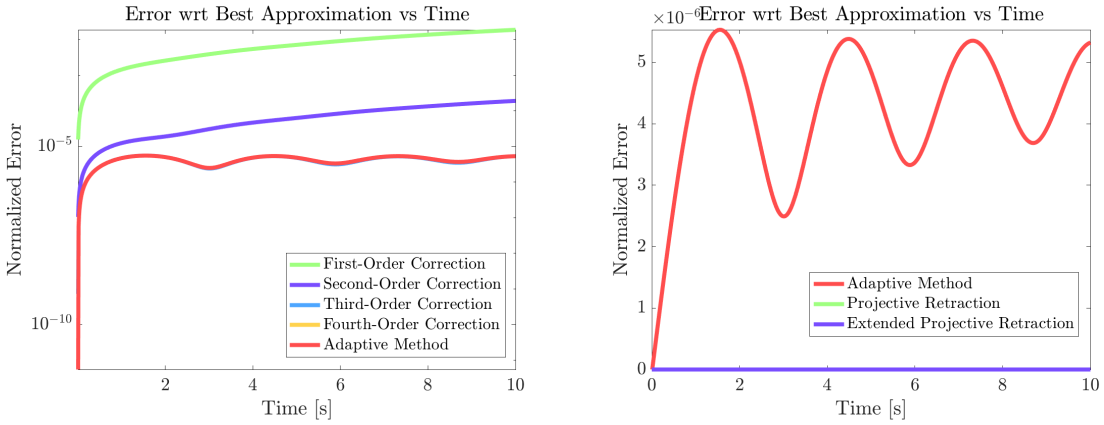
Table A.6: Normalized error with respect to Monte Carlo run with $r = 10$ and $u^{n+1} = D_1^{-1}\mathcal{R}_{u^n}(\tilde{\chi})$

	1^{st}-Order	2^{nd}-Order	Adaptive	Proj.	Ext. Proj.
Mean L_2 Error	3.0323E-3	3.3386E-3	5.0294E-3	3.5006E-3	2.1391E-3
L_2 Mean Error	3.8022E-4	1.2605E-3	5.9611E-4	4.1723E-4	2.0320E-4
L_2 Variance Error	1.6455E-4	1.6092E-4	1.0698E-4	1.7951E-4	4.8319E-5

Appendix B

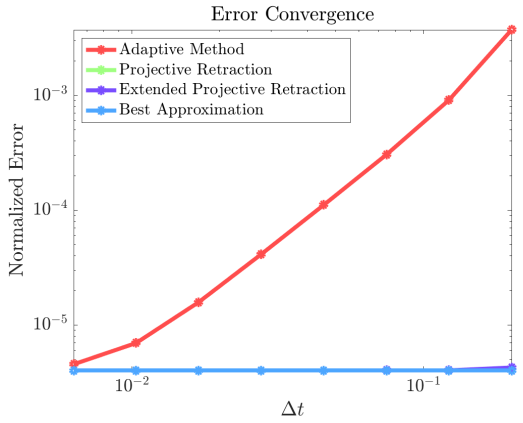
Extra Figures

B.1 Matrix differential equations

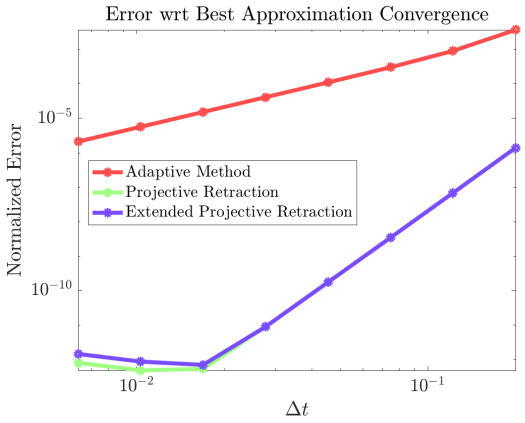


(a) Perturbative retractions wrt best approx. (b) Projective retractions wrt best approx.

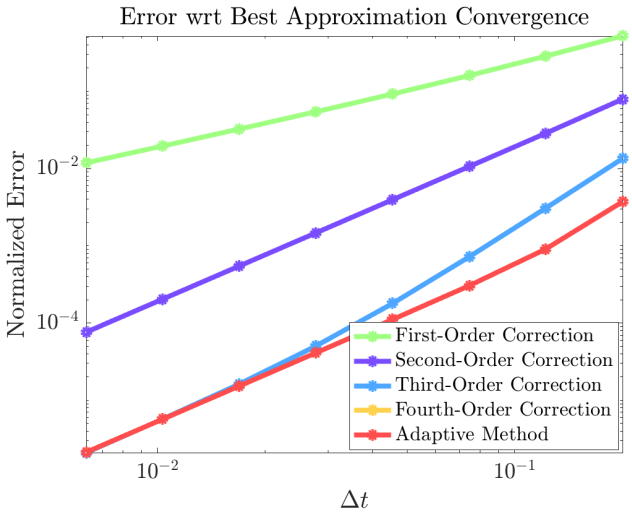
Figure B-1: Error with respect to best approximation due to different retractions as a function of time for matrix differential equations, $\Delta t = 0.01$



(a) Projective retractions

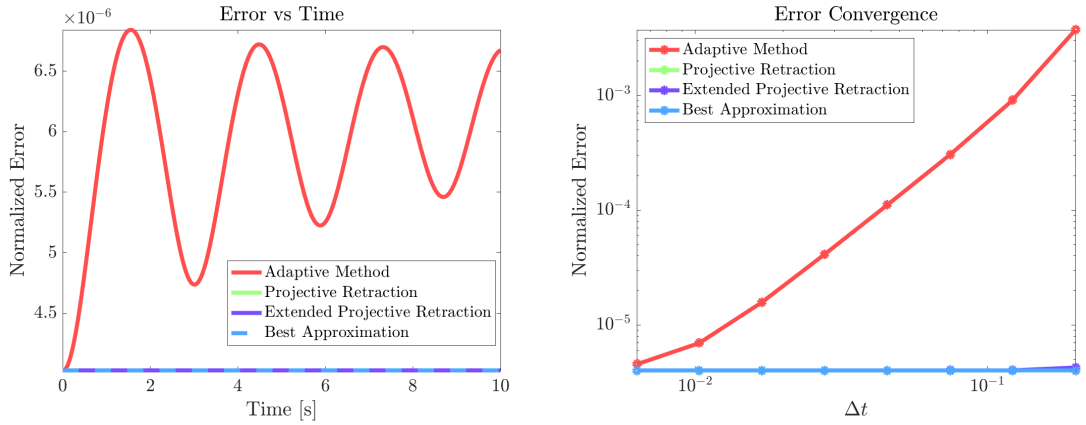


(b) Projective retractions wrt best approx.



(c) Perturbative retractions wrt best approx.

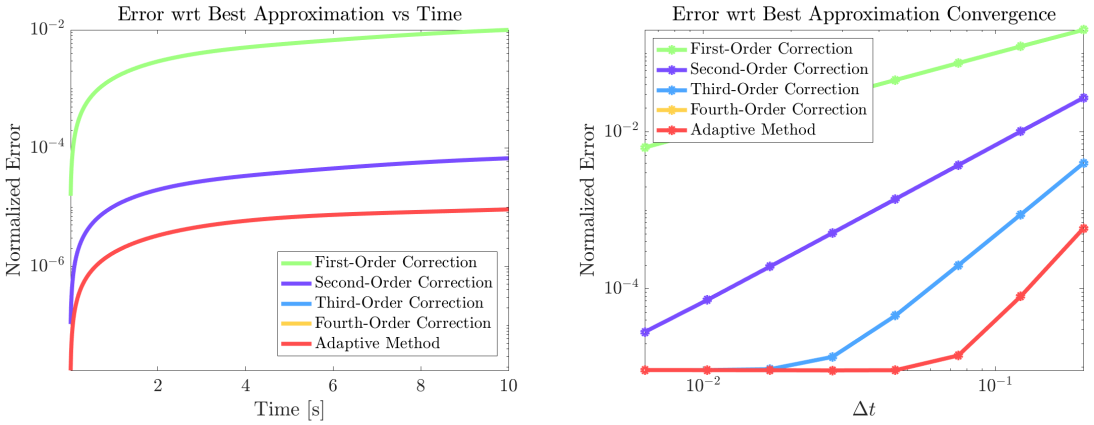
Figure B-2: Convergence plots of perturbative and projective retractions for matrix differential equations



(a) Error with respect to best approximation as a function of time with full-rank derivative information, $\Delta t = 0.01$

(b) Convergence plots of error given full-rank derivative information

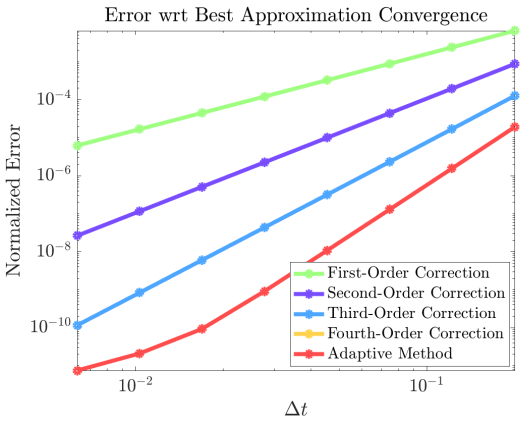
Figure B-3: Plots of error with respect to the best approximation for projective retractions when using algorithm 5



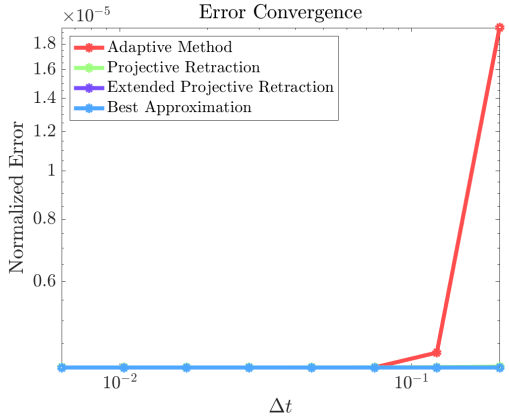
(a) Error with respect to best approximation as a function of time with full-rank derivative information, $\Delta t = 0.01$

(b) Convergence plots of error with respect to best approximation given full-rank derivative information

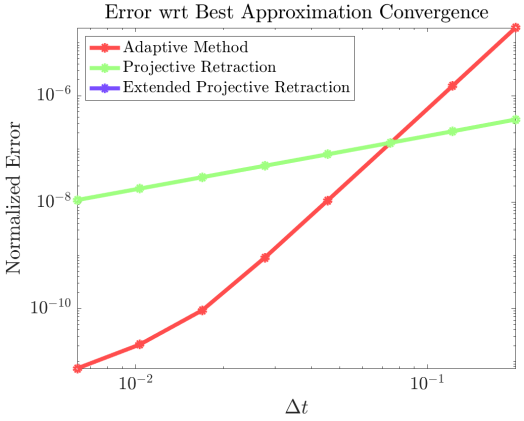
Figure B-4: Plots of error with respect to the best approximation for perturbative retractions when using algorithm 5



(a) Error with respect to best approximation from perturbative retractions given corrected full-rank derivative information

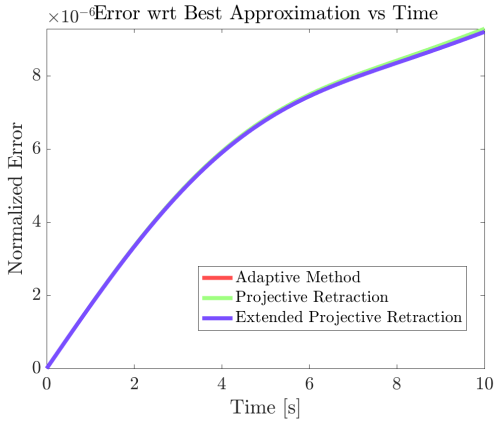


(b) Convergence plots of error from projective retractions given corrected full-rank derivative information

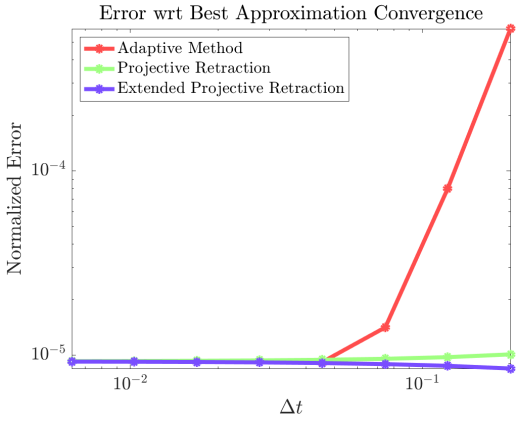


(c) Convergence plots of error with respect to best approximation from projective retractions given full-rank derivative information

Figure B-5: Convergence plots of error when using algorithm 6

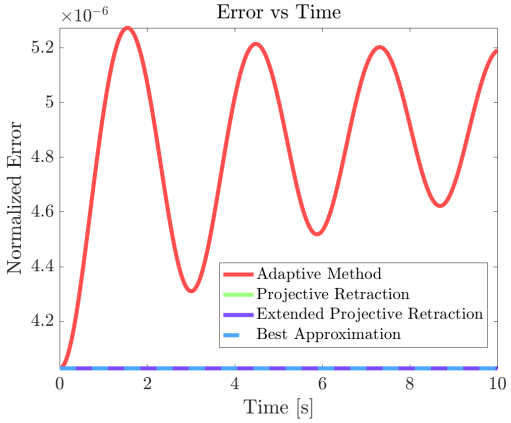


(a) Error with respect to best approximation as a function of time with full-rank derivative information, $\Delta t = 0.01$

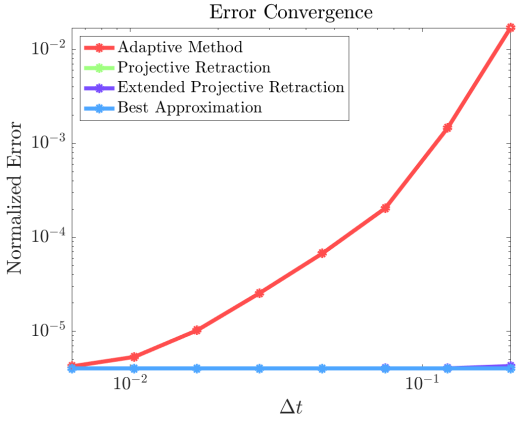


(b) Convergence plots of error with respect to best approximation given full-rank derivative information

Figure B-6: Plots of error with respect to the best approximation for projective retractions when using algorithm 5

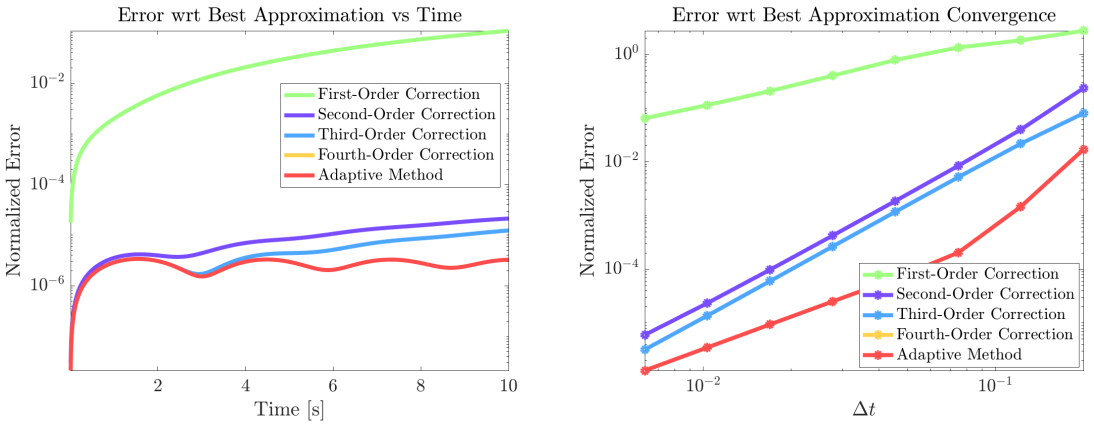


(a) Error as function of time with internal retractions, $\Delta t = 0.01$



(b) Convergence plots with retractions used internally (see algorithm 7) and the extended projective retraction as the final step

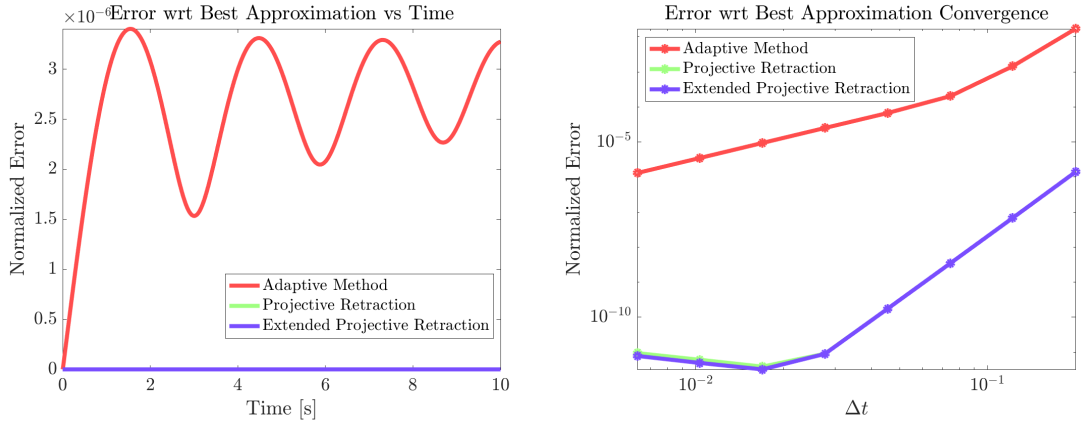
Figure B-7: Plots of error projective retractions when using algorithm 7



(a) Error with respect to best approximation as function of time with internal retractions, $\Delta t = 0.01$

(b) Convergence plots of error with respect to best approximation with retractions used internally (see algorithm 7) and the extended projective retraction as the final step

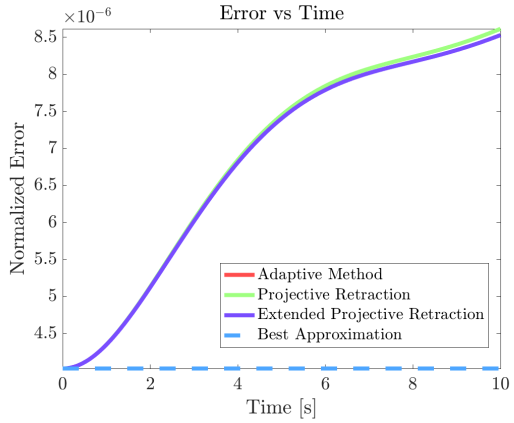
Figure B-8: Plots of error with respect to best approximation for perturbative retractions when using algorithm 7



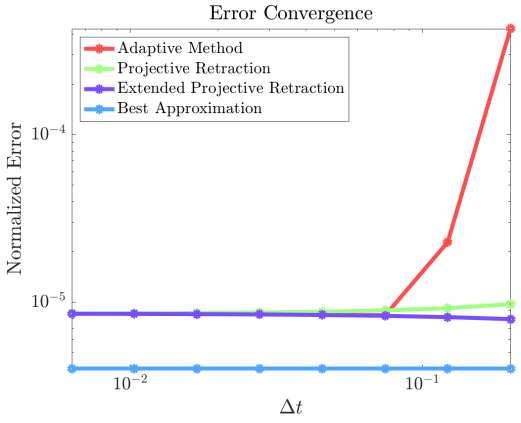
(a) Error with respect to best approximation as function of time with internal retractions, $\Delta t = 0.01$

(b) Convergence plots of error with respect to best approximation with retractions used internally (see algorithm 7) and the extended projective retraction as the final step

Figure B-9: Plots of error with respect to best approximation for projective retractions when using algorithm 7

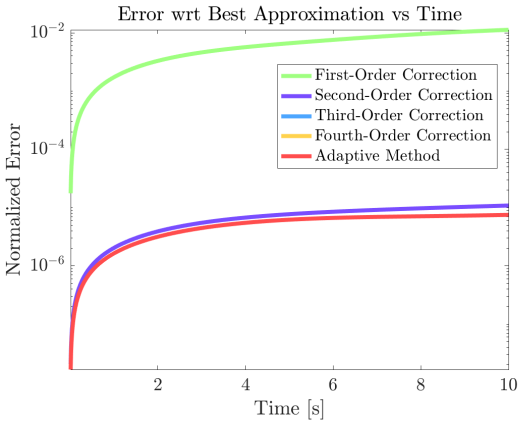


(a) Error as function of time with internal retractions, $\Delta t = 0.01$

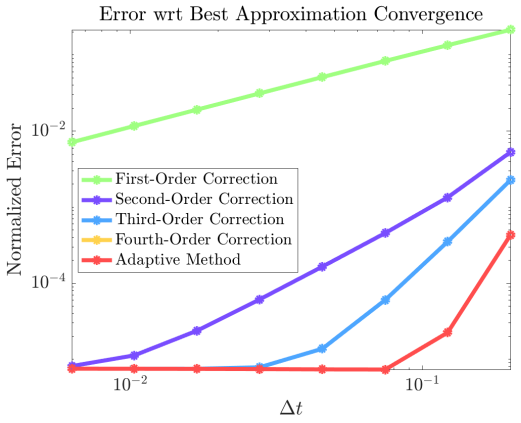


(b) Convergence plots with retractions used internally (see algorithm 7) and the extended projective retraction as the final step

Figure B-10: Plots of error projective retractions when using algorithm 8

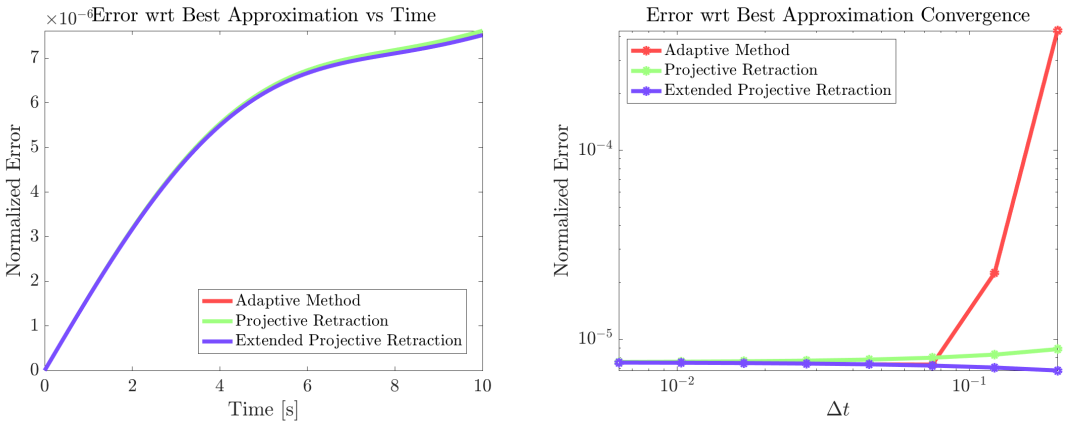


(a) Error with respect to best approximation as function of time with internal retractions, $\Delta t = 0.01$



(b) Convergence plots of error with respect to best approximation with retractions used internally (see algorithm 8) and the extended projective retraction as the final step

Figure B-11: Plots of error with respect to best approximation for perturbative retractions when using algorithm 8

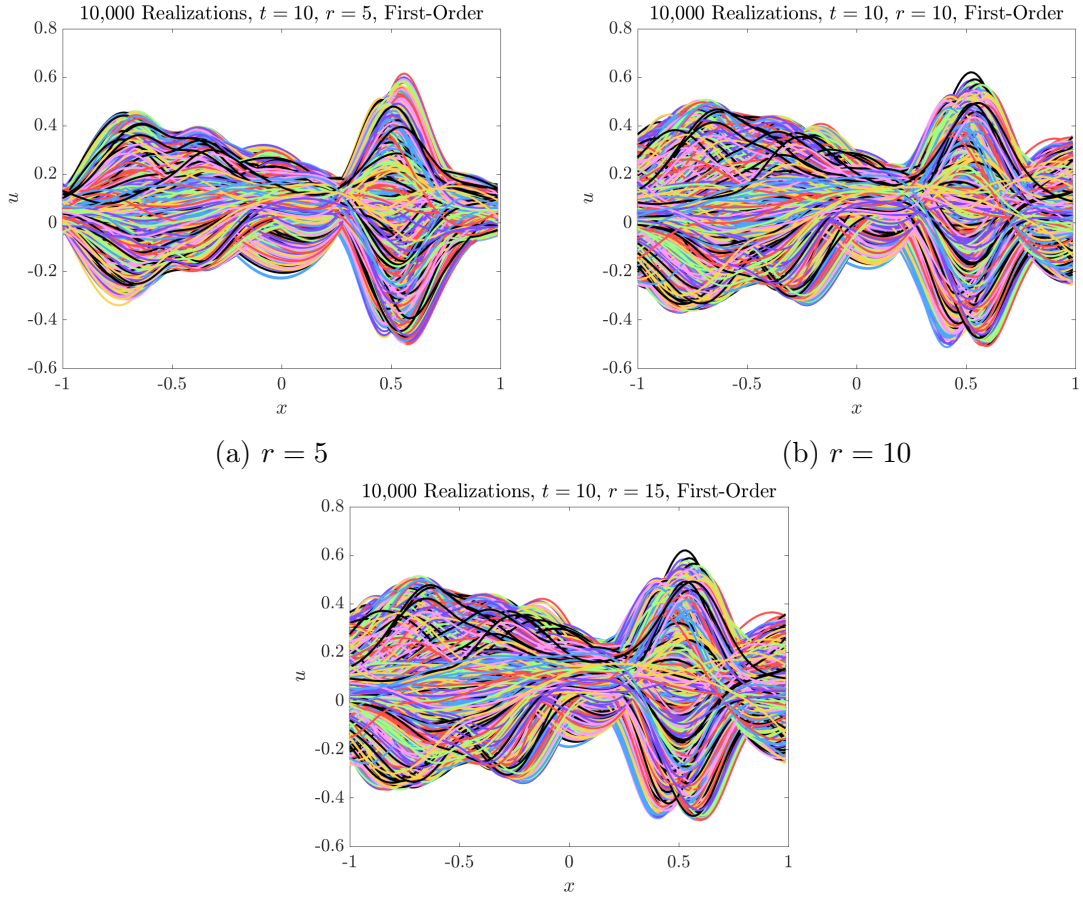


(a) Error with respect to best approximation as function of time with internal retractions, $\Delta t = 0.01$

(b) Convergence plots of error with respect to best approximation with retractions used internally (see algorithm 8) and the extended projective retraction as the final step

Figure B-12: Plots of error with respect to best approximation for projective retractions when using algorithm 8

B.2 Stochastic partial differential equations

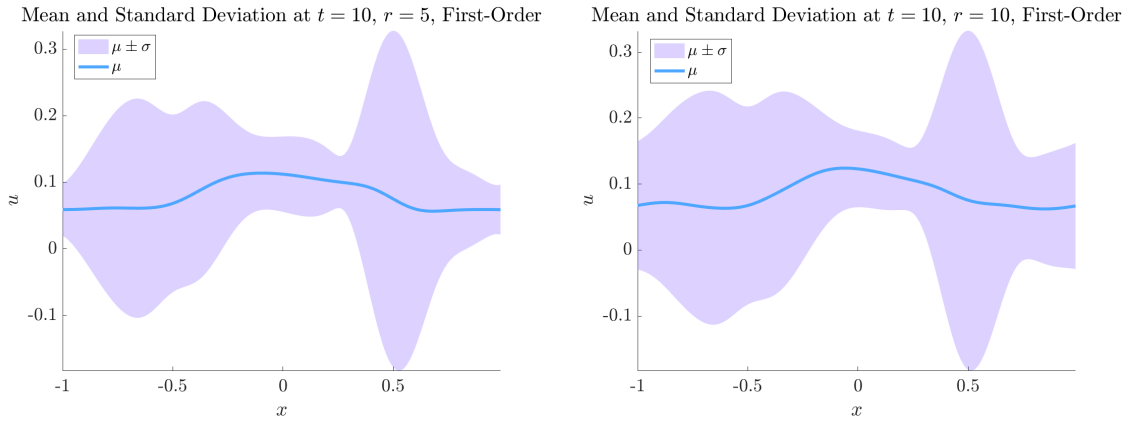


(a) $r = 5$

(b) $r = 10$

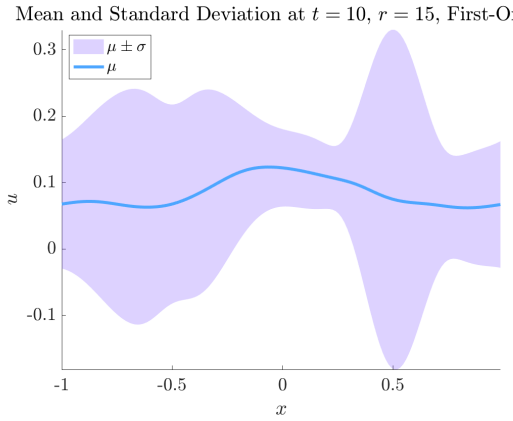
(c) $r = 15$

Figure B-13: Realizations of first-order retraction solutions at $t = 10$



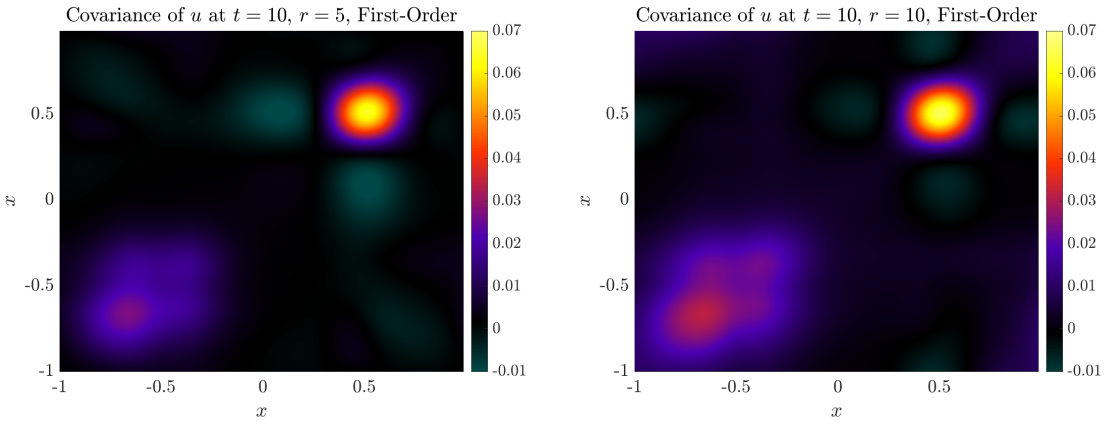
(a) $r = 5$

(b) $r = 10$



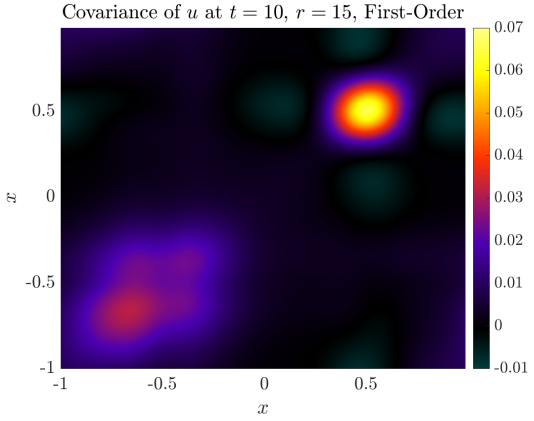
(c) $r = 15$

Figure B-14: Marginal mean and standard deviation of first-order retraction solutions at $t = 10$



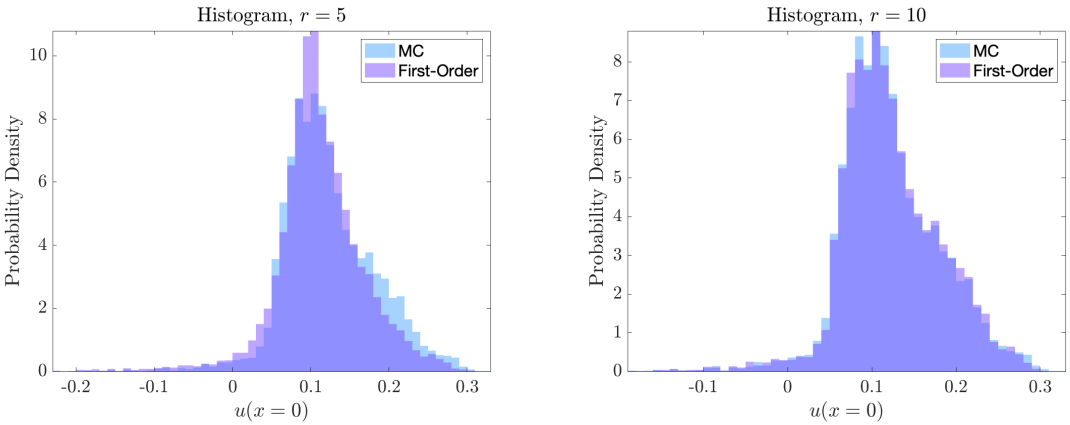
(a) $r = 5$

(b) $r = 10$



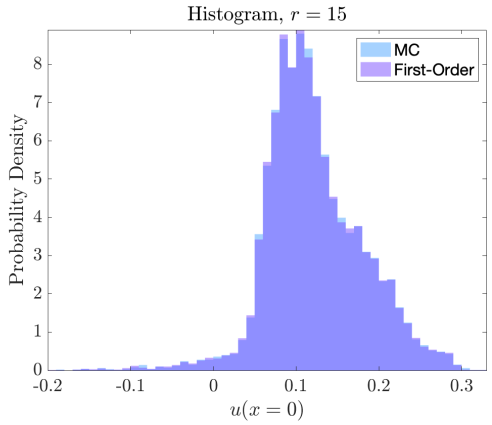
(c) $r = 15$

Figure B-15: Spatial covariance of first-order retraction solutions at $t = 10$



(a) $r = 5$

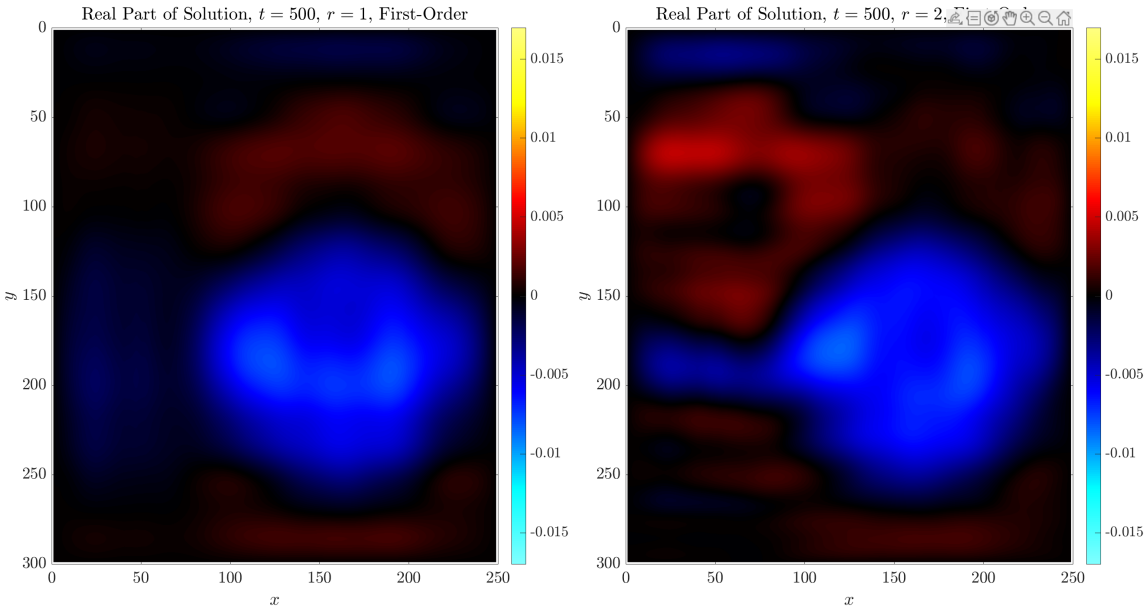
(b) $r = 10$



(c) $r = 15$

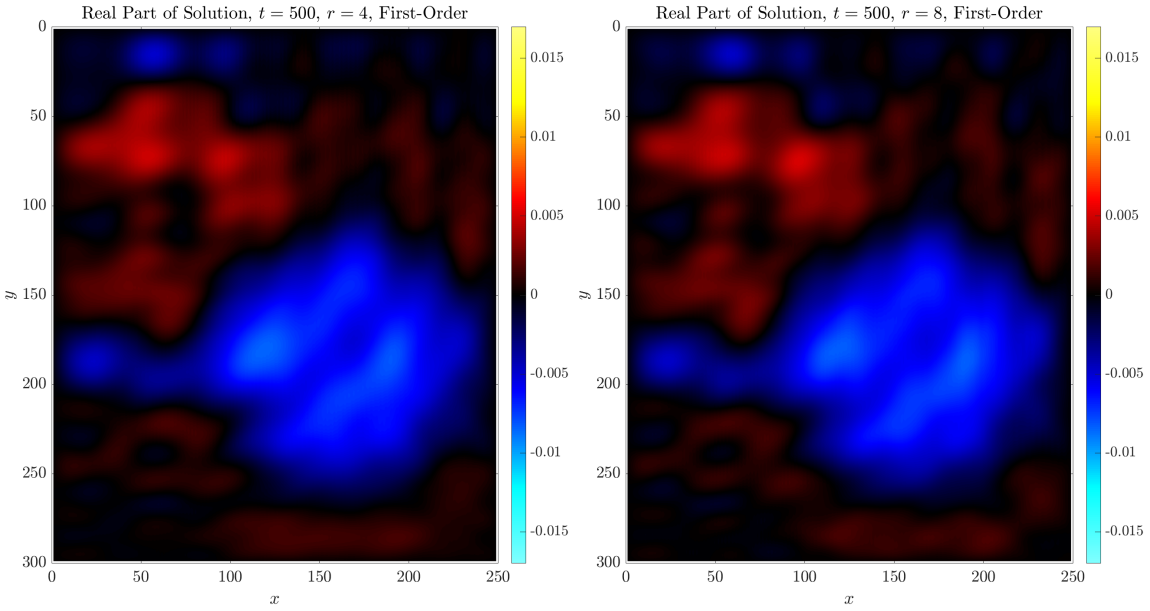
Figure B-16: Histograms of first-order retraction solutions at $x = 0$, $t = 10$

B.3 Two-dimensional partial differential equations



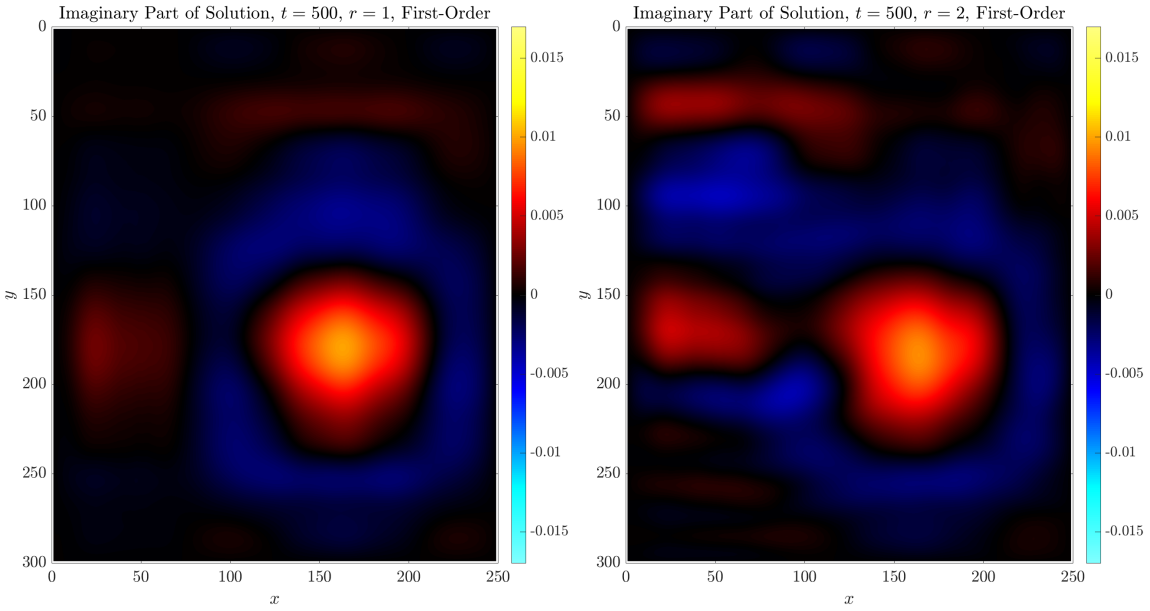
(a) $r = 1$

(b) $r = 2$

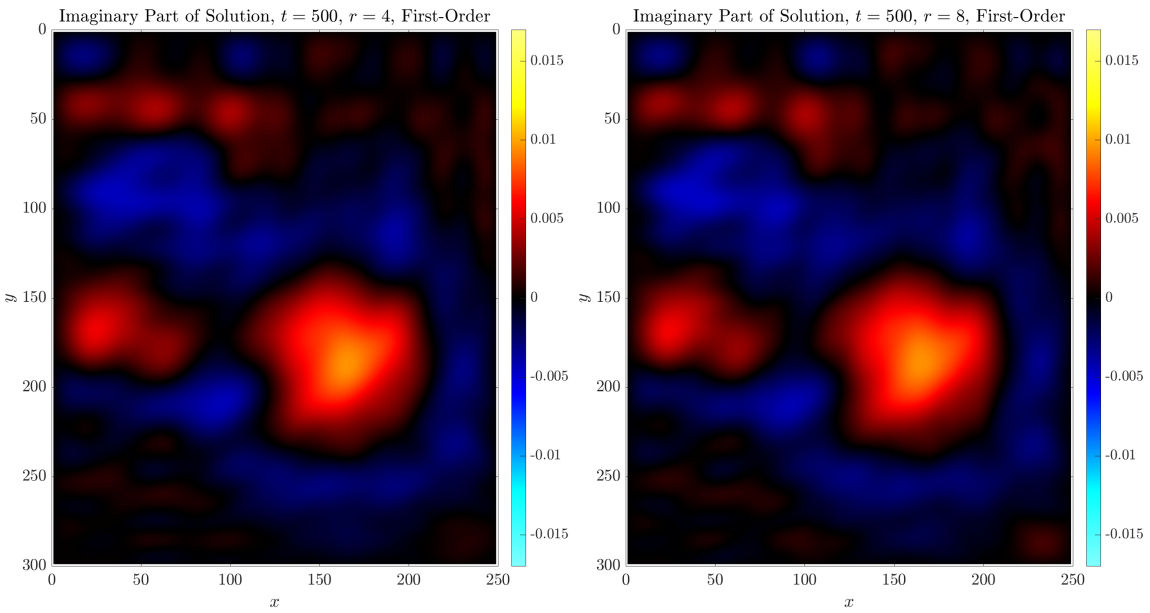


(c) $r = 4$ (d) $r = 8$

Figure B-17: Real part of solution using first-order retraction at $r = 1, 2, 4, 8$

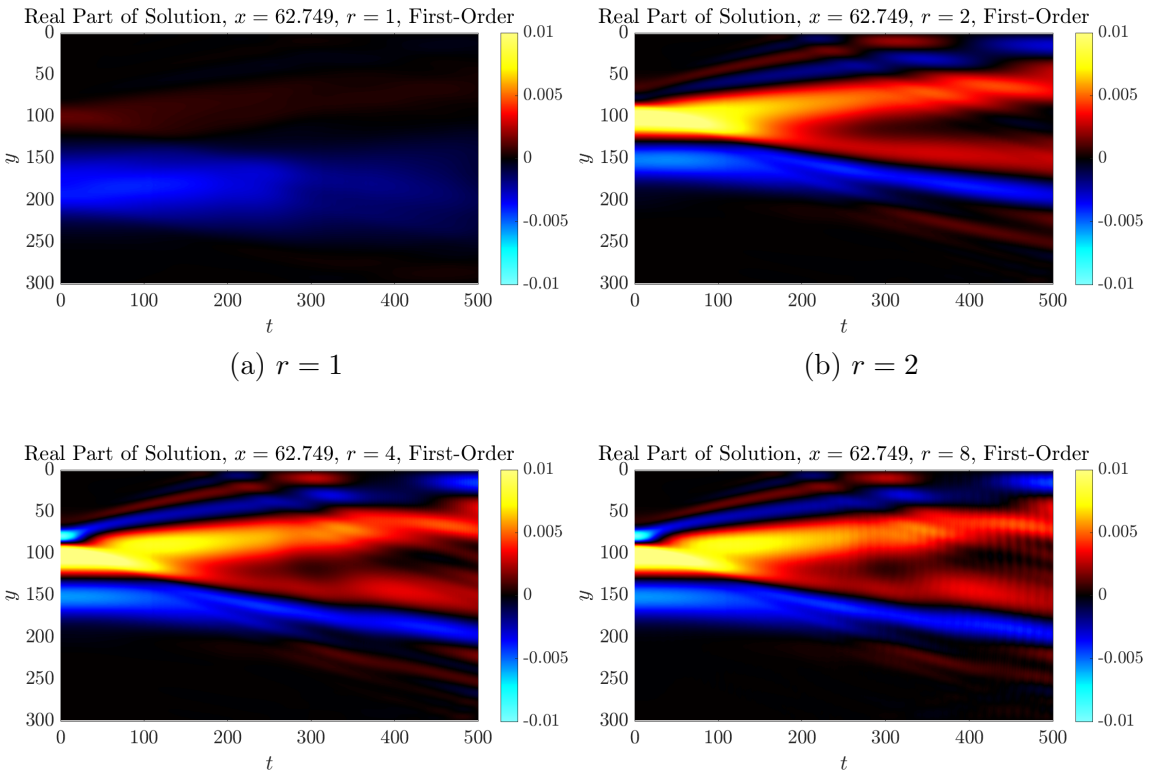


(a) $r = 1$ (b) $r = 2$



(c) $r = 4$ (d) $r = 8$

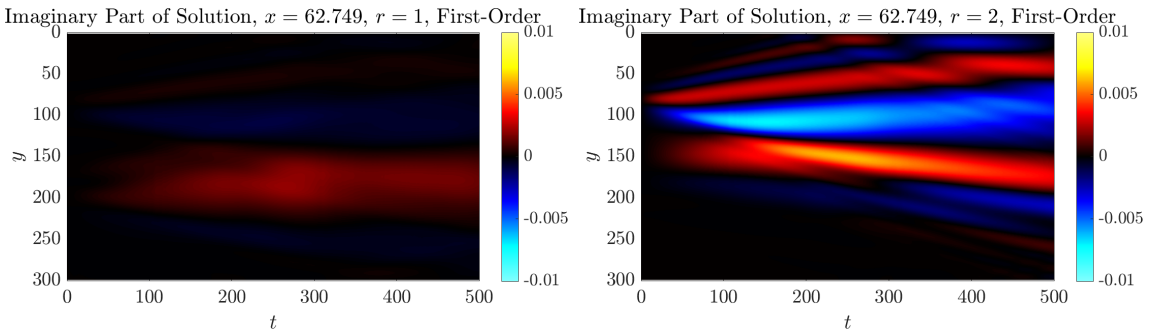
Figure B-18: Imaginary part of solution using first-order retraction at $r = 1, 2, 4, 8$



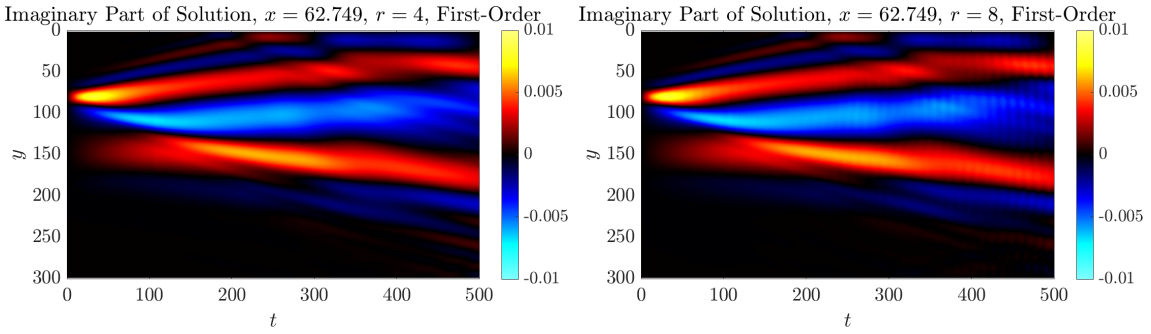
(a) $r = 1$ (b) $r = 2$

(c) $r = 4$ (d) $r = 8$

Figure B-19: Real slices of solution using first-order retraction at $r = 1, 2, 4, 8$, x constant

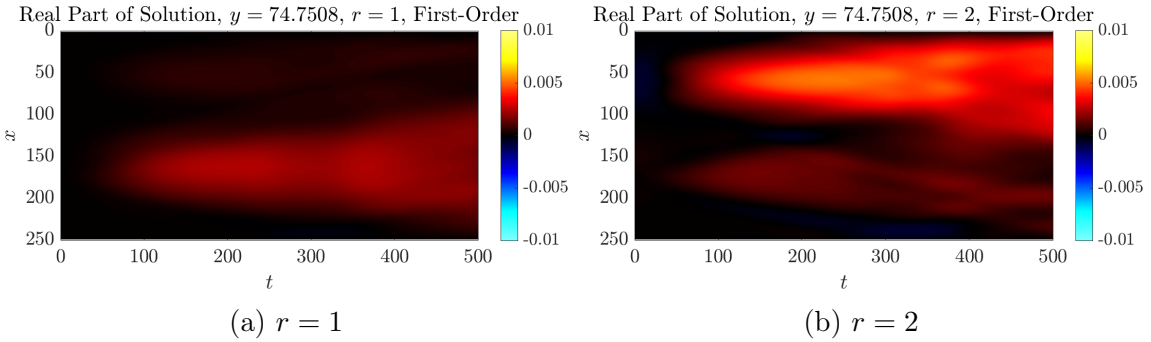


(a) $r = 1$ (b) $r = 2$

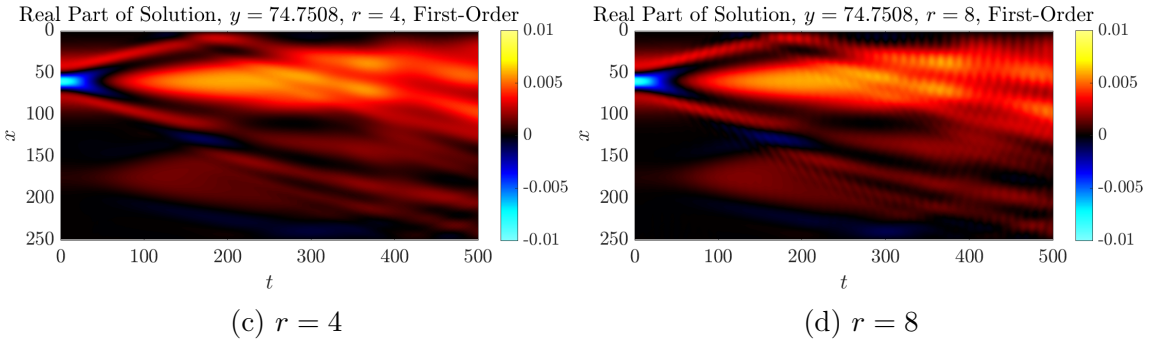


(c) $r = 4$ (d) $r = 8$

Figure B-20: Imaginary slices of solution using first-order retraction at $r = 1, 2, 4, 8$, x constant

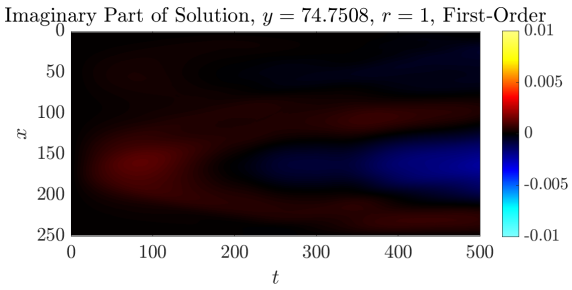


(a) $r = 1$ (b) $r = 2$

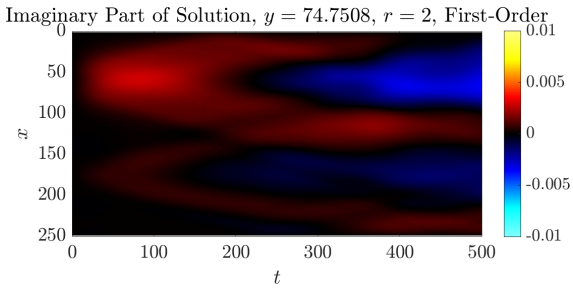


(c) $r = 4$ (d) $r = 8$

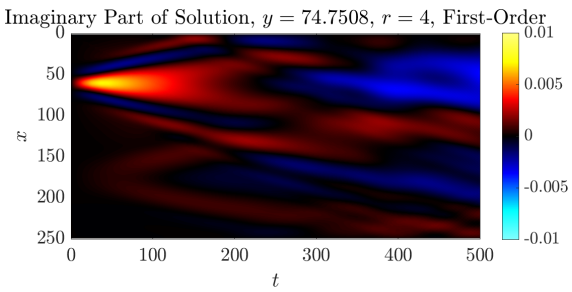
Figure B-21: Real slices of solution using first-order retraction at $r = 1, 2, 4, 8$, y constant



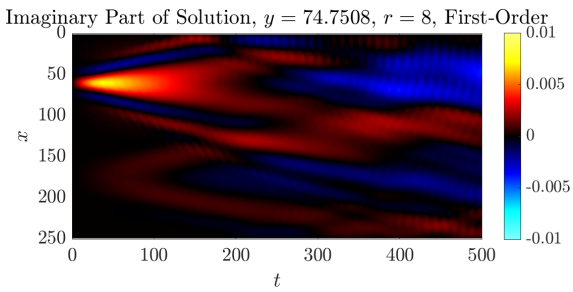
(a) $r = 1$



(b) $r = 2$



(c) $r = 4$



(d) $r = 8$

Figure B-22: Imaginary slices of solution using first-order retraction at $r = 1, 2, 4, 8$, y constant

Appendix C

Alternate Proofs

C.1 Alternate Proof to Theorem 1.4.1

Proof. We seek to minimize $||\dot{X} - \mathscr{L}||$. This is equivalent to minimizing $J \equiv \frac{1}{2}||\dot{U}Z^T + U\dot{Z}^T - \mathscr{L}||^2$. Recall we choose the Frobenius norm. First, we differentiate with respect $\dot{Z}$.

$$\begin{aligned}\left(\frac{\partial J}{\partial \dot{Z}}\right)^T &= U^T(\dot{U}Z^T + U\dot{Z}^T - \mathscr{L}) \\ &= U\dot{Z}^T - \mathscr{L} = 0\end{aligned}$$

Using the orthonormality of U, we have the following by left multiplying by U^T and taking the transpose.

$$\dot{Z} = \mathscr{L}^T U \tag{C.1}$$

Now, we differentiate with respect to $\dot{U}$.

$$\frac{\partial J}{\partial \dot{U}} = (\dot{U}Z^T + U\dot{Z}^T - \mathscr{L})Z = 0$$

Multiplying on the left by $(I - UU^T)$ and using orthogonality of U and $\dot{U}$, we obtain the following.

$$\begin{aligned}\dot{U}Z^TZ &= (I - UU^T)\mathscr{L}Z \\ \Rightarrow \dot{U} &= (I - UU^T)\mathscr{L}Z(Z^TZ)^{-1}\end{aligned}$$

The rest of the proof is the same as the original. □

Appendix D

Additional Algorithms

D.1 Re-orthonormalization procedure

For a retraction of the form $U_1 Z_1^T = (U + \Delta t \dot{U})(Z + \Delta t \dot{Z})^T$, a re-orthonormalization procedure is necessary to ensure that U_1 is orthonormal. As in the rest of this thesis, we'll assume $U \in \mathcal{V}_{m,r}$ and $Z \in \mathbb{R}_*^{n \times r}$ and that $\dot{U} \in \mathcal{U}_{m,r}$ and $\dot{Z} \in \mathbb{R}^{n \times r}$. Consider the following.

$$\begin{aligned} U_1^T U_1 &= (U + \Delta t \dot{U})^T (U + \Delta t \dot{U}) \\ &= \cancelto{I}{U^T U} + \cancelto{0}{U^T \dot{U}} + \cancelto{0}{\dot{U}^T U} + \Delta t^2 \dot{U}^T \dot{U} \\ &= I + \Delta t^2 \dot{U}^T \dot{U} \end{aligned}$$

So, there is a $\mathcal{O}(\Delta t^2)$ term that ruins orthonormality. This is an artifact of discrete time stepping. In the continuous limit where $\Delta t \to 0$, this is not a problem. Even when Δt is extremely small and that $\mathcal{O}(\Delta t^2)$ term is negligible, we still accumulate floating-point error as time-stepping continues, and so a re-orthonormalization is necessary to avoid numerical error overwhelming the solution.

As presented in [69], the main idea is to find a matrix $\tilde{U}_1$ that is as close as possible to U_1 while maintaining that $\tilde{U}_1^T \tilde{U}_1 = I$. We seek some matrix some matrix such that $\tilde{U}_1 = UA$, $\tilde{Z}_1 = ZA^{-T}$. Then, we will have the following.

$$\tilde{U}_1 \tilde{Z}_1^T = UAA^{-1} Z^T = UZ^T$$

Hence, we are not modifying the solution; we are only modifying the low-rank decomposition. An algorithm is presented below to accomplish this feat. Note that

Algorithm 9: Re-orthonormalization procedure

Input: $U \in \mathbb{R}^{m\times r}$, $Z \in \mathbb{R}_*^{n\times r}$
Output: $U \in \mathcal{V}_{m,r}$, $Z \in \mathbb{R}_*^{n\times r}$

1 $G = U^T U$
2 $G = \frac{1}{2}(G + G^T)$ `// ensure` G `is symmetric`
3 $V, \Lambda = \mathrm{eig}(G)$ `// compute eigendecomposition`
4 $\Lambda \leftarrow \sqrt{\Lambda}$
5 $U \leftarrow UV\Lambda^{-1}V^T$
6 $Z \leftarrow ZV\Lambda V^T$

the square root and reciprocal of Λ may be computed component-wise because Λ is diagonal.

D.2 Fourth-order adaptive perturbative retraction

Here, we have an algorithm written out explicitly for a fourth-order adaptive perturbative retraction. For details on the re-orthonormalization procedure, see appendix D.1. For a more general algorithm for an nth-order adaptive pertrubative retraction, see algorithm 3.

Algorithm 10: Fourth-order adaptive perturbative retraction

Input: $U_0 \in \mathcal{V}_{m,r}$, $Z_0 \in \mathbb{R}_*^{n\times r}$, $\mathscr{L} \in \mathbb{R}^{m\times n}$, $\Delta t \in \mathbb{R}$, $\varepsilon \in \mathbb{R}$

Output: $U_1 \in \mathcal{V}_{m,r}$, $Z_1 \in \mathbb{R}^{n\times r}$

$\alpha_0 = ||Z_0||$

$\dot{\mathcal{U}}_1 = (I - U_0U_0^T)\mathscr{L}Z_0\left(Z_0^TZ_0\right)^{-1}, \qquad \dot{\mathcal{Z}}_1 = \mathscr{L}^TU_0$

$\dot{\mathcal{U}}_2 = \left[(I - U_0U_0^T)\mathscr{L}\dot{\mathcal{Z}}_1 - \dot{\mathcal{U}}_1(Z_0^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^TZ_0)\right]\left(Z_0^TZ_0\right)^{-1},$
$\dot{\mathcal{Z}}_2 = \left(\mathscr{L}^T - Z_0\dot{\mathcal{U}}_1^T\right)\dot{\mathcal{U}}_1$

$\alpha_2 = \frac{\Delta t^2}{\alpha_0}\max\left(||\dot{\mathcal{U}}_2||, ||\dot{\mathcal{Z}}_2||\right)$

if $\alpha_2 < \varepsilon$ **then**

  $\dot{\mathcal{U}}_3 = \Big[(I - U_0U_0^T)\mathscr{L}\dot{\mathcal{Z}}_2 - \dot{\mathcal{U}}_2(Z_0^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^TZ_0) -$
  $\dot{\mathcal{U}}_1(Z_0^T\dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^TZ + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1)\Big]\left(Z_0^TZ_0\right)^{-1},$
  $\dot{\mathcal{Z}}_3 = \mathscr{L}^T\dot{\mathcal{U}}_2 - Z_0(\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1) - \dot{\mathcal{Z}}_1\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1$

  $\alpha_3 = \frac{\Delta t^3}{\alpha_0}\max\left(||\dot{\mathcal{U}}_3||, ||\dot{\mathcal{Z}}_3||\right)$

  if $\alpha_3 < \varepsilon$ **then**

    $\dot{\mathcal{U}}_4 = \Big[(I - U_0U_0^T)\mathscr{L}\dot{\mathcal{Z}}_3 - \dot{\mathcal{U}}_3\left(Z_0^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^TZ_0\right) -$
    $\dot{\mathcal{U}}_2\left(Z_0^T\dot{\mathcal{Z}}_2 + \dot{\mathcal{Z}}_2^TZ_0 + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_1\right) -$
    $\dot{\mathcal{U}}_1\left(Z_0^T\dot{\mathcal{Z}}_3 + \dot{\mathcal{Z}}_3^TZ_0 + \dot{\mathcal{Z}}_2^T\dot{\mathcal{Z}}_1 + \dot{\mathcal{Z}}_1^T\dot{\mathcal{Z}}_2\right)\Big]\left(Z_0^TZ_0\right)^{-1}$
    $\dot{\mathcal{Z}}_4 = \mathscr{L}^T\dot{\mathcal{U}}_3 - Z_0\left(\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_3 + \dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_2 + \dot{\mathcal{U}}_3^T\dot{\mathcal{U}}_1\right) -$
    $\dot{\mathcal{Z}}_1\left(\dot{\mathcal{U}}_2^T\dot{\mathcal{U}}_1 + \dot{\mathcal{U}}_1\dot{\mathcal{U}}_2\right) - \dot{\mathcal{Z}}_2\dot{\mathcal{U}}_1^T\dot{\mathcal{U}}_1$

    $\alpha_4 = \frac{\Delta t^4}{\alpha_0}\max\left(||\dot{\mathcal{U}}_4||, ||\dot{\mathcal{Z}}_4||\right)$

    if $\alpha_4 < \varepsilon$ **then**

      $U_1 = U_0 + \Delta t\dot{\mathcal{U}}_1 + \Delta t^2\dot{\mathcal{U}}_2 + \Delta t^3\dot{\mathcal{U}}_3 + \Delta t^4\dot{\mathcal{U}}_4,$
      $Z_1 = Z_0 + \Delta t\dot{\mathcal{Z}}_1 + \Delta t^2\dot{\mathcal{Z}}_2 + \Delta t^3\dot{\mathcal{Z}}_3 + \Delta t^4\dot{\mathcal{Z}}_4$

    else

      $U_1 = U_0 + \Delta t\dot{\mathcal{U}}_1 + \Delta t^2\dot{\mathcal{U}}_2 + \Delta t^3\dot{\mathcal{U}}_3,$
      $Z_1 = Z_0 + \Delta t\dot{\mathcal{Z}}_1 + \Delta t^2\dot{\mathcal{Z}}_2 + \Delta t^3\dot{\mathcal{Z}}_3$

  else

    $U_1 = U_0 + \Delta t\dot{\mathcal{U}}_1 + \Delta t^2\dot{\mathcal{U}}_2, \qquad Z_1 = Z_0 + \Delta t\dot{\mathcal{Z}}_1 + \Delta t^2\dot{\mathcal{Z}}_2$

else

  $U_1 = U_0 + \Delta t\dot{\mathcal{U}}_1, \qquad Z_1 = Z_0 + \Delta t\dot{\mathcal{Z}}_1$

$U_1, Z_1 \leftarrow$ re-orthonormalization procedure on U_1, Z_1

References

[1] Paul Fieguth. *An Introduction to Complex Systems*. Springer International Publishing, 2017. ISBN: 978-3-319-44606-6.

[2] Norbert Wiener. "The Homogeneous Chaos". In: *American Journal of Mathematics* 60.4 (1938), pp. 897–936. ISSN: 00029327, 10806377.

[3] N. Wiener. *Nonlinear Problems in Random Theory*. M.I.T. paperback series. Technology Press of Massachusetts Institute of Technology, 1958. ISBN: 9780262230049.

[4] Robert H Cameron and William T Martin. "The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals". In: *Annals of Mathematics* (1947), pp. 385–392.

[5] Dongbin Xiu and George Em Karniadakis. "The Wiener–Askey polynomial chaos for stochastic differential equations". In: *SIAM journal on scientific computing* 24.2 (2002), pp. 619–644.

[6] Dongbin Xiu and George Em Karniadakis. "Modeling uncertainty in flow simulations via generalized polynomial chaos". In: *Journal of computational physics* 187.1 (2003), pp. 137–167.

[7] Zhendong Luo and Goong Chen. *Proper orthogonal decomposition methods for partial differential equations*. Academic Press, 2018.

[8] Gal Berkooz, Philip Holmes, and John L Lumley. "The proper orthogonal decomposition in the analysis of turbulent flows". In: *Annual review of fluid mechanics* 25.1 (1993), pp. 539–575.

[9] DD Kosambi. "Statistics in function space". In: *DD Kosambi*. Springer, 2016, pp. 115–123.

[10] Kari Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Vol. 37. Sana, 1947.

[11] M Loéve. *Probability Theory II*. 4th ed. Vol. 2. Graduate Texts in Mathematics. Springer-Verlag New York, 1978.

[12] J David Logan. *Applied partial differential equations*. Springer, 2014.

[13] Jaromír Šimša. "The bestL 2-approximation by finite sums of functions with separable variables". In: *Aequationes mathematicae* 43.2-3 (1992), pp. 248–263.

[14] Themistoklis P. Sapsis and Pierre F. J. Lermusiaux. "Dynamically orthogonal field equations for continuous stochastic dynamical systems". In: *Physica D: Nonlinear Phenomena* 238.23–24 (Dec. 2009), pp. 2347–2360.

[15] P. A. M. Dirac. "Note on Exchange Phenomena in the Thomas Atom". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 26.3 (1930), pp. 376–385.

[16] Christian Lubich. *From quantum to classical molecular dynamics: reduced models and numerical analysis*. European Mathematical Society, 2008.

[17] M. P. Ueckermann, P. F. J. Lermusiaux, and T. P. Sapsis. "Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows". In: *Journal of Computational Physics* 233 (Jan. 2013), pp. 272–294.

[18] Florian Feppon and Pierre F. J. Lermusiaux. "Dynamically Orthogonal numerical schemes for efficient stochastic advection and Lagrangian transport". In: *SIAM Review* 60.3 (2018), pp. 595–625.

[19] Othmar Koch and Christian Lubich. "Dynamical Low-Rank Approximation". In: *SIAM Journal on Matrix Analysis and Applications* 29.2 (2007), pp. 434–454.

[20] Florian Feppon and Pierre F. J. Lermusiaux. "A Geometric Approach to Dynamical Model-Order Reduction". In: *SIAM Journal on Matrix Analysis and Applications* 39.1 (2018), pp. 510–538.

[21] John H. Hubbard and Barbara Burke Hubbard. "Manifolds, Taylor polynomials, quadratic forms, and curvature". In: *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach*. 4th ed. Ithaca, NY: Matrix Editions, 2009. Chap. 3, pp. 284–392.

[22] John Nash. "The Imbedding Problem for Riemannian Manifolds". In: *Annals of Mathematics* 63.1 (1956), pp. 20–63. ISSN: 0003486X.

[23] Andrew N Pressley. *Elementary differential geometry*. Springer Science & Business Media, 2010.

[24] Wikimedia Commons. *File:Tangent bundle.svg — Wikimedia Commons, the free media repository*. [Online; accessed 8-November-2020]. 2020.

[25] Isaac Chavel. "Riemannian Manifolds". In: *Riemannian Geometry : A Modern Introduction*. Vol. 2nd ed. Cambridge Studies in Advanced Mathematics Vol. 98. Cambridge University Press, 2006. Chap. 1, pp. 1–55. ISBN: 9780521853682.

[26] Alan Edelman, Tomás A. Arias, and Steven T. Smith. "The Geometry of Algorithms with Orthogonality Constraints". In: *SIAM J. Matrix Anal. Appl.* 20.2 (Apr. 1999), pp. 303–353. ISSN: 0895-4798.

[27] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[28] Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. "Guaranteed Rank Minimization via Singular Value Projection". In: *Neural Information Processing Systems (NIPS)*. Dec. 2010.

[29] Michael Steinlechner. "Riemannian Optimization for High-Dimensional Tensor Completion". In: *SIAM Journal on Scientific Computing* 38.5 (2016), S461–S484.

[30] André Uschmajew and Bart Vandereycken. "Geometric Methods on Low-Rank Matrix and Tensor Manifolds". In: *Handbook of Variational Methods for Nonlinear Geometric Data*. Ed. by Philipp Grohs, Martin Holler, and Andreas Weinmann. Cham: Springer International Publishing, 2020, pp. 261–313.

[31] Bamdev Mishra et al. "Fixed-rank matrix factorizations and Riemannian low-rank optimization". In: *Computational Statistics* 29 (2014), pp. 591–621.

[32] Robert Piziak and P.L. Odell. "Full Rank Factorization of Matrices". In: *Mathematics Magazine* 72 (June 1999), pp. 193–201.

[33] Erhard Schmidt. "Zur Theorie der linearen und nichtlinearen Integralgleichungen". In: *Mathematische Annalen* 63.4 (1907), pp. 433–476.

[34] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218.

[35] L. Mirsky. "Symmetric Gauge Functions and Unitarily Invariant Norms". In: *The Quarterly Journal of Mathematics* 11.1 (Jan. 1960), pp. 50–59. ISSN: 0033-5606.

[36] Kevin Long. *Gateaux differentials and Frechet derivatives*. Jan. 2009.

[37] P.-A. Absil and Jérôme Malick. "Projection-like Retractions on Matrix Manifolds". In: *SIAM Journal on Optimization* 22.1 (2012), pp. 135–158.

[38] Roy L Adler et al. "Newton's method on Riemannian manifolds and a geometric model for the human spine". In: *IMA Journal of Numerical Analysis* 22.3 (2002), pp. 359–390.

[39] P-A Absil and Ivan V Oseledets. "Low-rank retractions: a survey and new results". In: *Computational Optimization and Applications* 62.1 (2015), pp. 5–29.

[40] David G Luenberger. "The gradient projection method along geodesics". In: *Management Science* 18.11 (1972), pp. 620–631.

[41] P-A Absil, Luca Amodei, and Gilles Meyer. "Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries". In: *Computational Statistics* 29.3-4 (2014), pp. 569–590.

[42] P.-A. Absil and Jérôme Malick. "Projection-like Retractions on Matrix Manifolds". In: *SIAM Journal on Optimization* 22.1 (2012), pp. 135–158.

[43] Bart Vandereycken. "Low-Rank Matrix Completion by Riemannian Optimization". In: *SIAM Journal on Optimization* 23.2 (2013), pp. 1214–1236.

[44] L.N. Trefethen and D. Bau. *Numerical Linear Algebra*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1997. ISBN: 9780898719574.

[45] Trung Vu, Evgenia Chunikhina, and Raviv Raich. *Perturbation expansions and error bounds for the truncated singular value decomposition*. 2020.

[46] FA Potra. "On Q-order and R-order of convergence". In: *Journal of Optimization Theory and Applications* 63.3 (1989), pp. 415–431.

[47] Laurent O Jay. "A note on Q-order of convergence". In: *BIT Numerical Mathematics* 41.2 (2001), pp. 422–429.

[48] Haruo Yoshida. "Construction of higher order symplectic integrators". In: *Physics letters A* 150.5-7 (1990), pp. 262–268.

[49] Emil Kieri and Bart Vandereycken. "Projection methods for dynamical low-rank approximation of high-dimensional problems". In: *Computational Methods in Applied Mathematics* 19.1 (2019), pp. 73–92.

[50] Harry Bateman. "Some Recent Resarches On the Motion Of Fluids". In: *Monthly Weather Review* 43.4 (Apr. 1915), pp. 163–170.

[51] J.M. Burgers. “A Mathematical Model Illustrating the Theory of Turbulence”. In: ed. by Richard Von Mises and Theodore Von Kármán. Vol. 1. Advances in Applied Mechanics. Elsevier, 1948, pp. 171–199.

[52] Kaysar Rahman, Nurmamat Helil, and Rahmatjan Yimin. “Some new semi-implicit finite difference schemes for numerical solution of Burgers equation”. In: *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 14. IEEE. 2010, pp. V14–451.

[53] Christian Lubich and Ivan V Oseledets. “A projector-splitting integrator for dynamical low-rank approximation”. In: *BIT Numerical Mathematics* 54.1 (2014), pp. 171–188.

[54] Emil Kieri, Christian Lubich, and Hanna Walach. “Discretized dynamical low-rank approximation in the presence of small singular values”. In: *SIAM Journal on Numerical Analysis* 54.2 (2016), pp. 1020–1038.

[55] A. de Souza Dutra and C.A.S. Almeida. “Exact solvability of potentials with spatially dependent effective masses”. In: *Physics Letters A* 275.1 (2000), pp. 25–30. ISSN: 0375-9601.

[56] Jiang Yu and Shi-Hai Dong. “Exactly solvable potentials for the Schrödinger equation with spatially dependent mass”. In: *Physics Letters A* 325.3 (2004), pp. 194–198. ISSN: 0375-9601.

[57] Gang Chen and Zi-dong Chen. “Exact solutions of the position-dependent mass Schrödinger equation in D dimensions”. In: *Physics Letters A* 331.5 (2004), pp. 312–315. ISSN: 0375-9601.

[58] J.M. Luttinger and W. Kohn. “Motion of Electrons and Holes in Perturbed Periodic Fields”. In: *Physical Review* 97.4 (1955). cited By 2224, pp. 869–883.

[59] O. Rojo and J.S. Levinger. “Integrated cross section for a velocity-dependent potential”. In: *Physical Review* 123.6 (1961). cited By 16, pp. 2177–2179.

[60] Gerald Bastard et al. “Wave mechanics applied to semiconductor heterostructures”. In: (1988).

[61] Melvin Lax, William H Louisell, and William B McKnight. “From Maxwell to paraxial wave optics”. In: *Physical Review A* 11.4 (1975), p. 1365.

[62] Rebecca H Jordan and Dennis G Hall. “Free-space azimuthal paraxial wave equation: the azimuthal Bessel–Gauss beam solution”. In: *Optics letters* 19.7 (1994), pp. 427–429.

[63] G Nienhuis and L Allen. "Paraxial wave optics and harmonic oscillators". In: *Physical Review A* 48.1 (1993), p. 656.

[64] G Daniel Dockery. "Modeling electromagnetic wave propagation in the troposphere using the parabolic equation". In: *IEEE Transactions on Antennas and Propagation* 36.10 (1988), pp. 1464–1470.

[65] Denis J Donohue and JR Kuttler. "Propagation modeling over terrain using the parabolic wave equation". In: *IEEE Transactions on Antennas and Propagation* 48.2 (2000), pp. 260–277.

[66] Fred D Tappert. "The parabolic approximation method". In: *Wave propagation and underwater acoustics*. Springer, 1977, pp. 224–287.

[67] Finn B Jensen et al. *Computational ocean acoustics*. Springer Science & Business Media, 2011.

[68] "Model Equations". In: *Time-Dependent Problems and Difference Methods*. John Wiley & Sons, Ltd, 2013. Chap. 1, pp. 1–46. ISBN: 9781118548448.

[69] Jing Lin and Pierre F. J. Lermusiaux. "Minimum-Correction Second-Moment Matching: Theory, Algorithms and Applications". In: *Numerische Mathematik* (2020). Sub-judice.

[70] Themistoklis P. Sapsis and Pierre F. J. Lermusiaux. "Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty". In: *Physica D: Nonlinear Phenomena* 241.1 (2012), pp. 60–76.

[71] Jing Lin. "Bayesian Learning for High-Dimensional Nonlinear Dynamical Systems: Methodologies, Numerics and Applications to Fluid Flows". PhD thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology, Department of Mechanical Engineering, Sept. 2020.

[72] Yoshihito Kazashi, Fabio Nobile, and Eva Vidličková. *Stability properties of a projector-splitting scheme for dynamical low rank approximation of random parabolic equations*. 2020.

[73] Eleonora Musharbash. *Dynamical Low Rank approximation of PDEs with random parameters*. Tech. rep. EPFL, 2017.

[74] Boris N Khoromskij, Ivan V Oseledets, and Reinhold Schneider. "Efficient time-stepping scheme for dynamics on TT-manifolds". In: (2012).

[75] Gianluca Ceruti and Christian Lubich. "Time integration of symmetric and anti-symmetric low-rank matrices and Tucker tensors". In: *BIT Numerical Mathematics* (2020), pp. 1–24.

[76] Junyu Zhang and Shuzhong Zhang. "A Cubic Regularized Newton's Method over Riemannian Manifolds". In: *arXiv preprint arXiv:1805.05565* (2018).

[77] D. A. Kolesnikov and I. V. Oseledets. "Convergence analysis of projected fixed-point iteration on a low-rank matrix manifold". In: *Numerical Linear Algebra with Applications* 25.5 (2018). e2140 nla.2140, e2140.

[78] P-A Absil, Luca Amodei, and Gilles Meyer. "Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries". In: *Computational Statistics* 29.3-4 (2014), pp. 569–590.

[79] Wael Hajj Ali. "Dynamically Orthogonal Equations for Stochastic Underwater Sound Propagation". MA thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology, Computation for Design and Optimization Program, Sept. 2019.

[80] Wael H. Ali and Pierre F. J. Lermusiaux. "Dynamically Orthogonal Equations for Stochastic Underwater Sound Propagation: Theory, Schemes and Applications". In: (2020). In preparation.

[81] Wael H. Ali and Pierre F. J. Lermusiaux. "Acoustics Bayesian Inversion with Gaussian Mixture Models using the Dynamically Orthogonal Field Equations". In: (2020). In preparation.

[82] Wael Hajj Ali et al. "Stochastic Oceanographic-Acoustic Prediction and Bayesian Inversion for Wide Area Ocean Floor Mapping". In: *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE. Seattle, Oct. 2019, pp. 1–10.