Atlas Simulation: A Numerical Scheme for Approximating Multiscale Diffusions Embedded in High Dimensions

by

Miles Crosskey

Department of Mathematics
Duke University

Date: _____

Approved:

_____
Mauro Maggioni, Supervisor

_____
Jonathan Mattingly

_____
James Nolen

_____
Anita Layton

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Mathematics
in the Graduate School of Duke University
2014

Abstract

Atlas Simulation: A Numerical Scheme for Approximating Multiscale Diffusions Embedded in High Dimensions

by

Miles Crosskey

Department of Mathematics
Duke University

Date: _____

Approved:

_____
Mauro Maggioni, Supervisor

_____
Jonathan Mattingly

_____
James Nolen

_____
Anita Layton

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the Graduate School of Duke University
2014

Abstract

When simulating multiscale stochastic differential equations (SDEs) in high-dimensions, separation of timescales and high-dimensionality can make simulations expensive. The computational cost is dictated by microscale properties and interactions of many variables, while interesting behavior often occurs on the macroscale with few important degrees of freedom. For many problems bridging the gap between the microscale and macroscale by direct simulation is computationally infeasible, and one would like to learn a fast macroscale simulator. In this paper we present an unsupervised learning algorithm that uses short parallelizable microscale simulations to learn provably accurate macroscale SDE models. The learning algorithm takes as input: the microscale simulator, a local distance function, and a homogenization scale. The learned macroscale model can then be used for fast computation and storage of long simulations. I will discuss various examples, both low- and high-dimensional, as well as results about the accuracy of the fast simulators we construct, and its dependency on the number of short paths requested from the microscale simulator.

This thesis is dedicated to all my friends and family who have given me support over the years.

Contents

List of Figures

List of Abbreviations and Symbols

Symbols

C^k	Hölder space of order k.
Y_t	Original system of interest.
$\mathcal{M}$	Manifold in which Y_t lives.
ρ	Distance function on $\mathcal{M}$.
δ	Homogenization scale (spatial)
t_0	Homogenization scale (temporal)
$\Gamma = \{y_k\}$	Set of points in the δ-net.
$i \sim j$	Neighbor connections on net indices.
Φ_i	Mapping created by LMDS associated with chart i.
$\mathcal{A}$	Atlas denoted by the collection of $\{(x, i) \vert x \in \mathbb{R}^d, \vert x\vert < 2\delta, i \in \Gamma\}$.
$S_{i,j}$	Chart transition map $S_{i,j}(x) = (x - \mu_{i,j})T_{i,j} + \mu_{j,i}$.
$(x, i) \sim (y, j)$	equivalence class operator on points in the atlas.
$\mathcal{N}$	Smooth manifold created by $\mathcal{A}/\sim$.
Φ	Diffeomorphism from $\mathcal{N}$ to $\mathcal{M}$ which agrees with Φ_i locally.
i'	Shorthand denoting the new chart index chosen starting from $(x, i) \in \mathcal{A}$.
W	Wall function which keeps the Atlas within 2δ of the chart center.
I	Mapping from $\mathcal{A}$ to $\mathcal{N}$ using the equivalence class $\sim$.
X_t	Y_t mapped to $\mathcal{A}$.

$\widehat{X}_t$	Simulation scheme on the $\mathcal{A}$.
q	stationary distribution for X_t.
$\widehat{q}$	'almost stationary' distribution for $\widehat{X}_t$ (see (4.5)).

Notation

- For f, g functions on the same domain, we say that $f = O(g)$ if there is a constant $C > 0$ such that for all x in the domain of f and g we have $f(x) \leqslant Cg(x)$.

- If μ is a measure on a set Ω and f is a meas. function on Ω, then

$$f\mu = \int_\Omega f d\mu.$$

- If P is a measure preserving transformation on Ω, and f is a measurable function on Ω then

$$fP(x) = \int_\Omega f(y)P(x, y).$$

- If P is a measure preserving transformation on Ω, f is a measurable function on Ω, and μ is a measure on Ω then

$$fP\mu = f(P\mu) = (fP)\mu = \int_\Omega \int_\Omega f(y)P(x, dy)d\mu(x).$$

- If $F : M \to N$ is a measurable function and μ is a measure on M then we define the push forward measure of μ through F for any $A \subset N$ by

$$F_*\mu(A) = \int_M \mathbb{1}_A(F(x))d\mu(x).$$

Abbreviations

SDE	Stochastic Differential Equation
LMDS	Landmark Multi-Dimensional Scaling (see section 3.2)
MD	Molecular Dynamics

Acknowledgements

We thank J. Mattingly for useful discussions, and Y. Kevrekidis for introducing us to the questions around model reduction and equation free methods, and many discussions on these topics over the years.

1

Introduction

High-dimensional dynamical systems arise in a wide variety of applications, from the study of macromolecules in biology to finance and climate modeling. In many cases these systems are stochastic by nature, or are well-approximated by stochastic processes, for example as a consequence of slow-fast scale phenomena in the system. Simulations typically require significant amounts of computation, for several reasons. Each time step of the numerical scheme is often expensive because of the large dimensionality of the space, and the large number of interactions that need to computed. Fast scales and/or stochasticity may force each time step to be extremely small in order to have the requested accuracy. Finally, large-time behavior of the system may be dominated by rare transition events between stable regions, requiring very long paths to understand large-time dynamics. A large amount of research spanning multiple fields tackles the problems above.

Suppose we are given a high dimensional stochastic simulator, and we are faced with the problem of prohibitively expensive costs to run long simulations; in this scenario one asks the question "what can be learned from ensembles of short paths?" Short sample paths can be performed in parallel, and with the advent of distributed

computing (see S.Larson et al. (2002)) and GPU processors, tasks which can be parallelized are becoming more valuable. However several crucial problems to be addressed include: where in state space such short paths should be started? how many paths should be run locally? for how long? how does the local accuracy depend on these parameters? and once these local paths are constructed, and perhaps local simulators constructed, how can they be stitched together reasonable way to produce a global simulation scheme? What can be guaranteed about the accuracy of such a global scheme? Some examples, among many, in this direction are Markov State Models rfrom Bowman et al. (2009) and milestoning from Faradjian and Elber (2004); these methods discretize the state space into bins, or regions, and ask about transitions between such bins. Our method can be seen as a higher order approach, fitting a linear model to each bin.

We build upon existing ideas and methods in model reduction and multiscale techniques, and combine these with statistical learning and high-dimensional probabilistic ideas. The philosophy of reducing a high-dimensional system to a low-dimensional surrogate is well-established as enabling the simulation of complex, large, high-dimensional systems. These include model reduction Moore (1981); Antoulas et al. (2001); Huisinga et al. (2003), homogenization of PDE's Hornung (1997); Gilbert (1998), coarse-grained dynamics of high-dimensional systems Gear et al. (2005); Kevrekidis et al. (2003a), multiscale modeling of A. J. Majda and Vanden-Eijnden (2001); Shardlow and Stuart (2000); Kevrekidis et al. (2003a); Vanden-Eijnden (2003). We refer the reader to Weinan et al. (2007) for a summary of the motivations and applications of several of these techniques, and to the references therein.

We take concepts from *manifold learning* G. Chen (2013); Brand (2002); Z. Zhang (2002); Saul and Roweis (2003) in order to learn an underlying low-dimensional manifold around which most trajectories concentrate with high probability. We

approximate the macroscale manifold with linear low dimensional subspaces locally, which we call charts. These charts enable us to learn local properties of the system in low dimensional Euclidean space. Geometric Multi-Resolution Analysis (GMRA G. Chen (2013)) uses this concept to approximate high dimensional distributions on manifolds. These techniques perform the *model reduction* step, mapping the high-dimensional system from $\mathbb{R}^D$ down to d-dimensions, yielding a small set of coordinates describing the effective small number of degrees of freedom of the system.

We combine the above with *homogenization theory* G. Pavliotis (2008); Gilbert (1998); Vanden-Eijnden (2003) and learn a local homogenized version of the simulator. Locally we fit a constant coefficient SDE to each chart. If the macroscale simulator is well approximated by a smooth SDE, then constant coefficient SDEs will approximate the system well locally. This smooth SDE is the homogenized version of the original simulator, and under appropriate conditions (see G. Pavliotis (2008)), the homogenized version will capture long term dynamics of the original system.

Last, we add an extra ingredient of approximating transition maps between charts, generating a numerical approximation to an *atlas*. Learning such transition maps between charts is necessary to allow us to smoothly combine simulators on distinct charts into one global simulator on the atlas. Once all these ingredients combine, we have an atlas equipped with a simulator, which we will call the Atlas Simulator, or just Atlas for short.

Obtaining fast, accurate samples from the stationary distribution is a valuable tool in studying dynamical systems. A common tool for studying complex high dimensional dynamical systems is to obtain so called "reaction coordinates" or a set of global low dimensional coordinates describing the important states of the system. In this scenario, one commonly uses diffusion maps (see Coifman and Lafon (2004); Coifman et al. (2008); Rohrdanz et al. (2011)) which requires many samples from the stationary distribution to guarantee accuracy. These reaction coordinates

allow further analysis of dynamical systems by easily identifying stable states, and transitions between such states. In fact, the original motivation for this work was observing that the slowest part of running diffusion maps on such complicated high dimensional systems was obtaining the samples from the stationary distribution.

The paper is organized as follows: in section 2 we describe at high level our construction, algorithm, and informally state the main result on the accuracy of the Atlas for large times; then we illustrate the algorithm on simple examples. In section 3 we discuss the algorithm in detail. In section 4 we state and prove our main result. In section 5 we present a wide range of examples. We conclude with a discussion in section 6.

1.1 Review of Related Work

This chapter is intended to give the reader a review of other similar methods which can be compared to the method presented in this thesis. The heart of the problem we discuss is having an available microscale simulator which is slow, where the goal is to create a faster macroscale simulator based on knowledge learned from the available microscale simulator. There are a few notable methods for attacking this problem in the literature that we discuss: the equation free methdos (EFM) of Kevrekidis et al. (2003b), the heterogeneous multiscale method (HMM) of Weinan et al. (2007), and a scheme for simulating an SDE with multiscale properties from Vanden-Eijnden (2003). Another motivating work we discuss is Pavliotis and Stuart (2007) on parameter estimation for multiscale diffusions. Last we discuss Markov State Models Pande et al. (2010); Bowman et al. (2009), and algorithm from molecular dynamics which uses short sample paths to approximate the dynamics.

Both EFM and HMM begin by assuming knowledge of a fine and coarse description of each state, and maps between them. So given a fine scale state x, one should be able to produce a coarse state $X = R(x)$ via a *restriction operator* R. Also given a

coarse state X, one must be able to produce one (or more) fine scale states $x = L(X)$ using a *lifting operator* L. These mappings R, L should be consistent in the sense that $R \circ L = I$.

The EFM runs short bursts of the microscale simulator at each step, then extracts macroscale information through the restriction operator R. This macroscale information is then used in an update rule to extrapolate macroscale data in time (called coarse-projective integration), space (called gap-tooth scheme), or both (called patch dynamics). Next the lifting operator can then be used to obtain new initial conditions for the microscale simulator at the next step. In EFM, they are mainly concerned with evolving multiscale PDEs, as this is where extrapolation in space comes in handy. They do not explicitly mention using this scheme in our setting of solving an SDE.

In HMM, again the main idea is to run short bursts of the microscale simulator at each step and then extract macroscale information. In Weinan et al. (2007) they explicitly detail their algorithmic approach to solving an SDE of the following form:

$$\begin{cases} \dot{X}_t^\varepsilon = a(X_t^\varepsilon, Y_t^\varepsilon, \varepsilon), & X_0^\varepsilon = x \\ \dot{Y}_t^\varepsilon = \frac{1}{\varepsilon} b(X_t^\varepsilon, Y_t^\varepsilon, \varepsilon) + \frac{1}{\sqrt{\varepsilon}} \sigma(X_t^\varepsilon, Y_t^\varepsilon, \varepsilon) \dot{W}_t, & Y_0^\varepsilon = y \end{cases} \tag{1.1}$$

In this case $R([X, Y]) = X$, and a consistent choice for L could be $L(X) = [X, \vec{0}]$. On a timescale of $O(1)$ in ε, the dynamics of X_t behave like

$$\dot{\overline{X}}_t = \overline{a}(\overline{X}_t). \tag{1.2}$$

If $\overline{a}$ was known, (1.2) could be solved numerically in $O(1)$ time. Thus, the strategy presented in HMM is to estimate $\overline{a}$ from samples, and then select a macroscale ODE solver. In their example they use forward Euler, although note that other schemes could be chosen. Thus their update equation looks like:

$$\widetilde{a}_n = \frac{1}{N} \sum_{m=n_T}^{n_T+N-1} a(X_n, Y_{n,m}, \varepsilon) \tag{1.3}$$

$$X_{n+1} = X_n + \widetilde{a}_n \Delta t \tag{1.4}$$

where $Y_{n,m}$ is the output of the m^{th} step of the simulation from the microscale simulator, and n_T is the number of initial steps to skip while waiting for Y_n to be sufficiently close to stationary. Provided that $N + n_T << \varepsilon^{-1}$, this scheme obtains an increase in the simulation speed over the original microscale simulator.

Next consider the multiscale ODE

$$\begin{cases} \dot{X}_t^\varepsilon = f(X_t^\varepsilon, Y_t^\varepsilon, \varepsilon), & X_0^\varepsilon = x \\ \dot{Y}_t^\varepsilon = \frac{1}{\varepsilon} g(X_t^\varepsilon, Y_t^\varepsilon, \varepsilon) & Y_0^\varepsilon = y \end{cases} \tag{1.5}$$

with f, g being $O(1)$ and ε a small scale parameter. We give here a short overview of the work in Vanden-Eijnden (2003), in which they examine the dynamics on the ε^{-1} time scale. A key assumption is that the dynamics of Y_t alone with $X_t = x$ fixed has a unique invariant measure $\mu_x(dy)$. Then the asymptotic behavior of X_t as $\varepsilon \to 0$ is a rescaled SDE with $s = \varepsilon t$ of the form

$$dX_s = b(X_s)dt + \sigma(X_s)dW_s. \tag{1.6}$$

See Melbourne and Stuart (2011) for details on the weak convergence of (1.5) to (1.6).

In order to compute solutions faster than the microscale simulator, one would like to solve (1.6) rather than (1.5). Thus, one would like to estimate the coefficients $b(x), \sigma(x)$ given a value of x. This turns out to be a difficult problem; consider estimating the drift using the long time average:

$$\mathbb{E}_x[f] = \int_{\mathbb{R}^n} f(x, y, \varepsilon)\mu_x^\varepsilon \approx \frac{1}{T} \int_0^T f(x, Y_{\varepsilon t}^\varepsilon, \varepsilon)dt \tag{1.7}$$

It turns out we must estimate $\mathbb{E}_x[f]$ up to $O(\varepsilon)$ in order to obtain accurate solutions, and $\mathrm{Var}_x[f] = O(1)$ or else the dynamics of Y_t are irrelevant. The variance of the RHS of (1.7) is $\mathrm{Var}_x[f]/T$, thus estimating the $\mathbb{E}_x[f]$ requires at least $T \approx \varepsilon^{-2}$; at this point one may as well run the microscale simulator rather than this estimation procedure. The scheme proposed by Vanden-Eijnden (2003) gets away from this difficulty using more extensive knowledge of the system (namely, the knowledge of the derivatives of f, g, and the ability to rescale the equations by ε).

The scheme presented in Vanden-Eijnden (2003) has an update rule of the form

$$X_{n+1} = X_n + \widehat{b}_n \Delta t + \widehat{\sigma}_n \Delta W_n \tag{1.8}$$

with $\widehat{b}_n, \widehat{\sigma}_n$ being estimated from a sample simulation of M steps of Y_t. For more details we refer the reader to the paper itself.

The first important distinction between the work of HMM and EFM and our work is that our algorithm runs the microscale simulator in a preprocessing stage rather than inside the inner loop. This allows us to completely decouple the cost of running microscale simulations from producing long paths from our simulator. This strategy of learning the unknown macroscale coefficients beforehand works well if one expects that simulations will return to similar states many times before reaching equilibrium; this is the case with many ergodic SDEs which we are interested in improving simulation speeds.

The other main way our work differs from EFM and HMM is that the mappings R, L to and from the coarse/fine variables are unknown. Generating R, L may require intimate knowledge of important "reaction coordinates" (or slow variables) of the system. Finding such coarse variables may require long runs of the system. The fact that we do not require knowledge of these functions a priori allows us to easily apply our methodology to new problems with very little supervision from the user. Of course we do not get something from nothing, and knowledge

about what are good coarse variables to use is learned automatically from a given distance metric ρ. An example of a good distance metric for the problems (1.1), (1.5) is $\rho([X_1, Y_1], [X_2, Y_2]) = |X_1 - X_2|$. An important distinction of these distance functions is that they are *local*, and thus easier to choose in a relevant way. A good distance need only be able to distinguish points well locally, not globally (for example, see the example presented in section 5.4).

Next we discuss Pavliotis and Stuart (2007) on parameter estimation for multiscale diffusions. In this work, they examine the multiscale SDE

$$dx^\varepsilon(t) = -\alpha \nabla V(x^\varepsilon(t))dt - \frac{1}{\varepsilon}\nabla p\left(\frac{x^\varepsilon(t)}{\varepsilon}\right)dt + \sqrt{2\sigma}dW_t, \tag{1.9}$$

a type of multiscale SDE examined with our algorithm in later examples. Here p is a periodic function, and ε again the scale parameter. The timesteps one is allowed to take in this system is ε^2 so that the process does not move a significant part of one oscillation of p. As $\varepsilon \to 0$, x^ε converges weakly to an SDE of the form

$$dX_t = -\alpha K \nabla V(X_t)dt + \sqrt{2\sigma K}dW_t \tag{1.10}$$

with K depending exponentially on the relationship between p, σ (see Bensoussan et al. (2011); Pardoux (1999)). In Pavliotis and Stuart (2007), they begin with long paths of (1.9), knowledge of $V(x)$ and ε, then try to estimate the parameters $A = \alpha K$, $\Sigma = 2\sigma K$. Knowledge of the potential $V(x)$ may seem unreasonable, however given long paths one could estimate $V(x)$. The main difficulty to be overcome in this problem is that the obvious estimators to use for A, Σ are biased when the whole paths are used. The solution to this problem is to first downsample the paths by a temporal parameter $\delta \approx \varepsilon$ in order to obtain unbiased estimators for A, Σ. Clearly once equation (1.10) is learned, one could run simulations faster than (1.9), however this is impractical for our purposes since estimating (1.10) directly requires long paths from (1.9) in the first place.

Last, we discuss some recent work in molecular dynamics called Markov State Models (MSM) Pande et al. (2010); Bowman et al. (2009). This method begins by starting with many configurations, and using $k-$means to cluster them into k clusters. Once the clusters are fixed, short sample paths can then be used to estimate transition probabilities to neighboring clusters. Once these probabilities are computed and compiled into a matrix, this generates a Markov chain. From there, one can run long paths quickly, or compute a singular value decomposition. The first way in which our approach differs from MSM is that we use a δ-net procedure to divide the state space into clusters. The second and more imporant way in which our approach differs from MSM is that within each cluster, we fit a first order model rather than a zeroth order model. Our method will run an estimated linear SDE within each cluster, and use linear transition maps between clusters.

2

Construction and Main Results

The geometric intuition driving our construction is that the dynamics of the stochastic dynamical system $(Y_t)_{t\geqslant 0}$ in $\mathbb{R}^D$ under consideration is concentrated on or near an intrinsically low-dimensional manifold $\mathcal{M}$ of dimension d, with $d \ll D$. We refer to $\mathcal{M}$ as the *effective state space* of the system, as opposed to the full state space $\mathbb{R}^D$. This type of model may be appropriate in a wide variety of situations:

(i) the system has d degrees of freedom, and is therefore constrained (under suitable smoothness assumptions) to a d-dimensional manifold $\mathcal{M}$;

(ii) as in (i), but possibly with small deterministic or stochastic violations of those constraints (perhaps at a fast scale), but such that the trajectories stay close to $\mathcal{M}$ at all times.

In these cases it makes sense to approximate $\mathcal{M}$ by an efficient low-dimensional approximation $\mathcal{A}$, such as a union of d-dimensional linear affine sets (charts)Allard et al. (2012); Maggioni et al. (2013), and the dynamics of Y_t by surrogate dynamics on the atlas $\mathcal{A}$. Learning dynamics on $\mathcal{A}$ reduces the problem from learning a high-dimensional global simulator to a low-dimensional local simulator, together with

appropriate transitions between local simulators in different charts. We also gain computational efficiency by using the structure $\mathcal{A}$: long paths may be more quickly stored and simulated in lower dimensions. While this approach is useful in a wide variety of situations, here we will make assumptions about the geometry of the effective state space, and some smoothness of the underlying macroscale simulator, in order to prove large time accuracy results for the Atlas.

While in this paper we consider a special class of stochastic dynamical systems, those well-approximated by low-dimensional SDEs such as those leading to advection-diffusion equations along a manifold, the framework can be significantly extended, as we briefly discuss later in section 6, and this will be subject of future work.

2.1 Main Ideas and Steps

Our construction takes as input:

- a simulator $\mathcal{S}$ for the stochastic dynamical system $(Y_t)_{t\geqslant 0}$, which may be started upon request at any specified initial condition and run for a specified amount of time;
- a distance function ρ which may be used to compare data returned by the simulator;
- a spatial homogenization parameter δ;
- the dimension d of the effective state space, and a confidence parameter τ (optional).

We note here that the homogenization scale δ can also be given as a temporal scale t_0, and the two are related by scalings in the underlying dynamical system. Given a time t_0, running paths of length t_0 and examining the average distance traveled by such paths reveals a corresponding natural spatial scale $\delta(t_0)$ (in fact this is done in

example 5.5). Inversely, given δ, one could choose t_0 so that the average distance traveled by paths is approximiately δ. We later discuss the accuracy of the simulator, which is a function of the parameter δ.

We remark that while d is here considered as a parameter for the algorithm, in fact there is a lot of work on estimating the intrinsic dimension of high-dimensional data sets that would be applicable here. In particular, the techniques of Little et al. (2012, 2009) have strong guarantees, are robust with respect to noise, and are computationally efficient. See also Maggioni et al. (2013) for finite sample guarantees on the approximation of manifolds by local affine approximate tangent spaces. We will mention again the problem of estimating d again when constructing the local charts in section 3.2.

The confidence parameter τ sets the probability of success of the algorithm ($e^{-\tau^2}$), and is related to the number of sample paths one must use to approximate the local parameters of the simulator. We have written this as an optional parameter since using a default $O(1)$ constant is reasonable.

Our construction then proceeds in a few steps:

(i) **net construction**: find a well-distributed set of of points $\Gamma = \{y_k\}$ in $\mathcal{M}$, having a granularity parameter δ;

(ii) **learning the atlas**: learn local charts C_k near y_k which well represent $\mathcal{M}$ locally, and learn transition maps for transport between these charts;

(iii) **learning the simulator**: run $p = p(\delta, \tau)$ paths for time $t_0 = t_0(\delta)$ from each y_k and map them to the coordinate chart C_k. Use these low dimensional representations to estimate a simple simulator on each chart C_k.

2.1.1 Net construction

The first stage is to produce a δ-net $\Gamma = \{y_k\}$, which is a set of points $\{y_k\}$ such that no two points are closer than δ, and every point in $\mathcal{M}$ is at least δ close to some y_k. With abuse of notation, the range of k will also be denoted by Γ, so we may also write the net as $\{y_k\}_{k\in\Gamma}$. We say that two points y_k and y_j are connected, or $k \sim j$, if y_k and y_j are within 2δ. The Atlas will traverse the atlas $\mathcal{A}$ through neighboring connections. See section 3.1 for the details.

In real world examples, the space $\mathcal{M}$ may be unknown. In this case, one must first generate many samples $\{x_i\} \subset \mathcal{M}$ such that balls of radius $r \ll \delta$ cover $\mathcal{M}$. This first round of sampling should ideally have the following properties: it can be generated by a fast exploration method (e.g. see the recent work Zheng et al. (2013) for molecular dynamics, and references therein - this problem by itself is subject of much research); its samples do not require a significant number of calls to the simulator, or long runs of the simulators; different points may be sampled independently so that the process may be parallelized. We can then downsample these $\{x_i\}$ to obtain the desired net Γ.

It is important to remark that the algorithm we present is easily modified to run in exploratory mode: whenever configurations outside the explored region of space are encountered (an event that is quickly detectable using the data structures we employ), new charts and local simulators may be added on-the-fly. This is subject of future work.

2.1.2 Learning the Charts

The first step in learning the charts is to generate a set of landmarks A_k for each y_k in the net Γ. In our setting where $\mathcal{M}$ is d-dimensional, we can sample $m \geqslant d$ paths from the simulator of Y_t, starting at y_k and run until time $t_0 = t_0(\delta)$. As long as the diffusion Y_t is nondegenerate on the tangent plane, their projections will span the

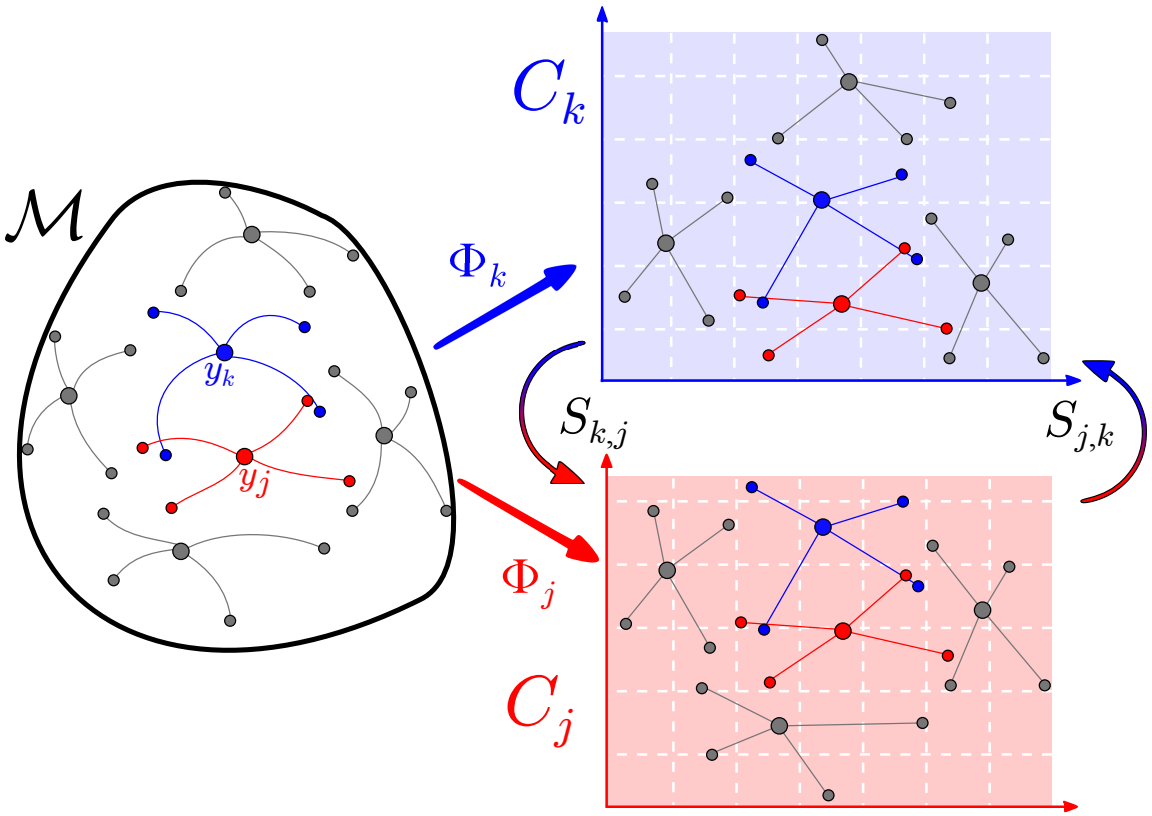

FIGURE 2.1: This figure depicts $m = 4$ samples per net point being used to learn the charts. Large circles represent net points (or projections of net points) and small circles represent path end points (or projections of the path endpoints). The LMDS mappings Φ_k, Φ_j use all the circles to learn the chart, while the transition maps $S_{k,j}, S_{j,k}$ use only the colored circles.

tangent space.

Next we learn a mapping Φ_k to a coordinate chart C_k for each y_k. In order that neighboring coordinate charts overlap on a region of size δ, we learn Φ_k from $L_k = \bigcup_{j \sim k} A_k$, the union of neighboring landmarks. This overlap will allow us to smoothly transition the simulator from one chart to the next. Each mapping Φ_k is constructed using LMDS on L_k, minimizing distortion of pairwise distances between the landmarks L_k (see section 3.2).

For any $k \sim j$, L_k and L_j have the landmarks $A_k \cup A_j$ in common; thus the charts C_k and C_j overlap on $A_k \cup A_j$. These landmarks span the local charts, and

are the points used to learn the transition maps between neighboring charts. The affine transition map $S_{k,j}$ is chosen as the "best" linear mapping from $\Phi_k(A_k \cup A_j)$ to $\Phi_j(A_k \cup A_j)$ described in section 3.3. Figure 2.1 shows a cartoon version of the points used to learn the atlas.

2.1.3 Learning the Simulator

Once the charts are known, we learn an approximation to the simulator on each chart. For each $y_k \in \Gamma$, we run $p = p(\delta, \tau)$ paths via the original simulator $\mathcal{S}$ up to time $t_0 = t_0(\delta)$ starting from y_k. Next we project the samples to C_k in order to estimate local simulation parameters. In this paper we use constant coefficient SDEs to model the simulator on each chart:

$$d\bar{X}_t = \bar{b}_k dt + \bar{\sigma}_k dB_t \,, \tag{2.1}$$

for some $\bar{b}_k \in \mathbb{R}^d$ and some positive definite $\bar{\sigma}_k \in \mathbb{R}^{d\times d}$. The solution to this constant coefficient SDE is a Gaussian with mean $\bar{b}_k\, t_0$ and covariance $\bar{\sigma}_k \bar{\sigma}_k^T\, t_0$. Therefore, we will simply choose $\bar{b}_k$ and $\bar{\sigma}_k$ in such a way that these statistics match the sample mean and sample covariance of the endpoints of the p paths we have run. Finite sample bounds for these empirical values are easily proved and determine how large p should be in order to achieve a desired accuracy (δ) with the requested confidence (τ). Note that this part may also be performed in parallel, both in k (the chart in which the learning takes place) and within each chart (each of the p paths may be run independently). At the end of this process we have obtained the family of parameters $(\bar{b}_k, \bar{\sigma}_k)_{k\in\Gamma}$ for a family of simulators $(\widehat{\mathcal{S}}_k)_{k\in\Gamma}$.

The local simulators $(\widehat{\mathcal{S}}_k)_{k\in\Gamma}$ are extended to a global simulator $\widehat{\mathcal{S}}$ on $\mathcal{A}$ using the transition maps between charts. This is done by alternating between steps from $(\widehat{\mathcal{S}}_k)_{k\in\Gamma}$, and transition map operations (see 3.5). Our main result guarantees that the local accuracy of the Atlas in fact yields long time accuracy; more precisely it

generates long paths with distribution which is $O(\delta \ln(1/\delta))$ close to the stationary distribution of the original system, see Theorem 2.

The choice of the local SDE's and the estimator of its parameters is quite simple, however we will see a collection of these simple simulators combine to reproduce much more complicated systems. Naturally the ideas may be extended to richer families of local SDE's, for which appropriate estimators based on the statistics of local trajectories may be constructed.

2.2 Theoretical Results

We present here a simplified version of the results contained in section 4. Suppose the given stochastic dynamical system Y_t is driven by an SDE on a d-dimensional manifold $\mathcal{M}$ of the form

$$dY_t = b(Y_t)dt + \sigma(Y_t)dB_t \tag{2.2}$$

with b, σ Lipschitz functions, and σ uniformly nondegenerate on the tangent bundle $T(\mathcal{M})$. Let q be the stationary distribution of Y_t on $\mathcal{M}$, and $\hat{q}$ be a measure on $\mathcal{A}$ defined later in equation (4.5) and computed by running the Atlas we construct. Let G be the inverse mapping from $\mathcal{A}$ to $\mathcal{M}$ defined in section 4.2. Then if the number of sample paths is at least $O((d + \tau^2)/\delta^4)$, with probability at least $1 - 2e^{-\tau^2}$,

$$||q - G(\widehat{q})||_{L^1(\mathcal{M})} < c\delta \ln(1/\delta) \tag{2.3}$$

for some constant c depending on geometric properties of $\mathcal{M}$, the Lipschitz constants of the drift b and diffusion σ, and the lower bound on singular values of σ along the tangent plane.

Remarks:

One can think of Y_t as the underlying homogenized system which we are trying to learn. Even if the microscale simulator does not satisfy these conditions, it is

possible the system is well-approximated by a macroscale simulator of the form (2.2) satisfying the conditions of the theorem on the timescale t_0; in this case the error in approximating the original simulator by Y_t is simply added to the right hand side of (2.3).

The condition that b, σ are Lipschitz is typically assumed for SDEs of the form (2.2) in order to guarantee existence and uniqueness Øksendal (2003), which is a sufficient but not necessary condition. In our case, Lipschitz coefficients b, σ play a key role in allowing us to approximate them locally by constants.

The condition the σ is uniformly nondegenerate is a condition which ensures that the noise propagates along all directions of the manifold $\mathcal{M}$. Another way of thinking about this condition is that starting at some y_0, one should be able to travel to any point within distance δ in time t_0 given an appropriate realization of the noise. This is not the case for ODEs for example, which have $\sigma = 0$, since starting at y_0 there is only one point that will be traveled to in time t_0.

2.3 Examples

Here we present some examples showcasing the usefulness of the Atlas. The examples shown here have Brownian motion in a potential well, although the theorem guarantees accuracy for any simulator of the form (2.2). Further examples will be discussed in section 5.

2.3.1 Brownian Motion on a Manifold

Given a d-dimensional smooth compact manifold $\mathcal{M}$, one may construct the potential

$$U_\varepsilon(x) = \frac{1}{\varepsilon}\text{dist}(x, \mathcal{M})^2$$

and consider the Itô diffusion in $\mathbb{R}^D$ given by

$$dY_t = \nabla U_\varepsilon dt + dB_t \tag{2.4}$$

If one simulates (2.4) numerically for small ε, the timesteps must be at least as small as $O(\varepsilon)$. We can view this thin potential around the manifold as our microscale interactions which forces our choice of timestep. What we are interested in is the macroscale behavior determined by the manifold $\mathcal{M}$.

For $\varepsilon \to 0$ this converges to the canonical Brownian motion on the manifold $\mathcal{M}$. For ε sufficiently small (compared to the curvature of $\mathcal{M}$) Y_t is well-approximated locally by (the low-dimensional) Brownian motion on $\mathcal{M}$, and the stationary distribution of Y_t is close to that of Brownian motion on $\mathcal{M}$. Our results apply to this setting, yielding an efficient d-dimensional simulator for Y_t, without a priori knowledge of $\mathcal{M}$.

2.3.2 One Dimensional Example

In this numerical example, we start with Brownian motion in a simple double well, and add a high frequency term to the potential to get $U(x)$.

$$U(x) = 16x^2(x-1)^2 + \frac{1}{6}\cos(100\pi x) \tag{2.5}$$

The high frequency term gives the Lipschitz constant $L \sim 10^2$, forcing the forward Euler scheme to use time steps on the order of $L^{-2} \sim 10^{-4}$ in order to just achieve stability (using a higher order method would not solve these problems as higher derivatives of U will be even larger). The first term in U is much smoother, and homogenization theory (see Pavliotis and Stuart (2007)) leads us to expect that the system is well-approximated by a smoother system with Lipschitz constant $l \sim 10$ or less; This is the target system we wish to approximate. Running Atlas with $\delta = 0.1 \sim l^{-1}$, we obtain a smoothed version of the potential homogenizing the high frequency term (see Figure 2.2).

The Atlas takes time steps which are over 10^2 times larger than the original system, and thus long paths can be simulated 10^2 times faster. Note that increasing the

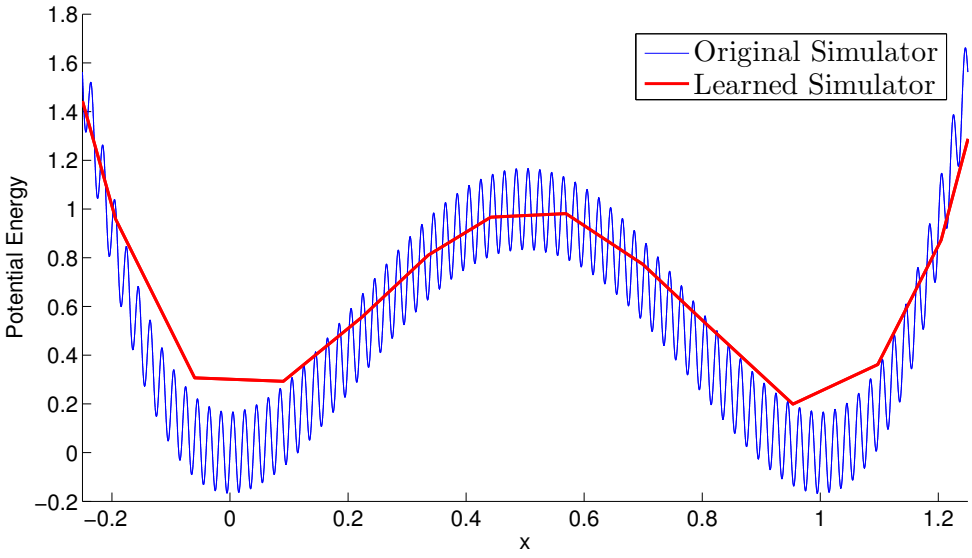

FIGURE 2.2: Original stiff potential U (shown in blue), and effective potential $\widehat{U}$ (shown in red) for the Atlas, learned from short trajectories of the original simulator.

frequency of the oscillating term (thereby increasing L) does not affect the speed of the Atlas, only the speed of constructing the Atlas. This means that our algorithm allows for a decoupling of the microscale complexity from the macroscale complexity. A histogram of the stationary distributions are shown in Figure 2.3 comparing the Atlas and the original system. See section 5.2.2 for more details about the experiment, and Figures 5.6, 5.7 for the errors in true effective potential vs. estimated effective potential, and the error in approximating the time evolution of the original system by the Atlas for a multiscale choice of times.

2.4 Algorithmic Complexity

Suppose that a single call to the original simulator of length t_0 is S. The total number of points in the net Γ contains $O(\delta^{-d})$ points with constant depending on the volume of the manifold $\mathcal{M}$. From each point we will see that we must choose $p = O(\delta^{-4})$ to estimate the parameters of the Atlas simulator to within accuracy δ. Assuming the

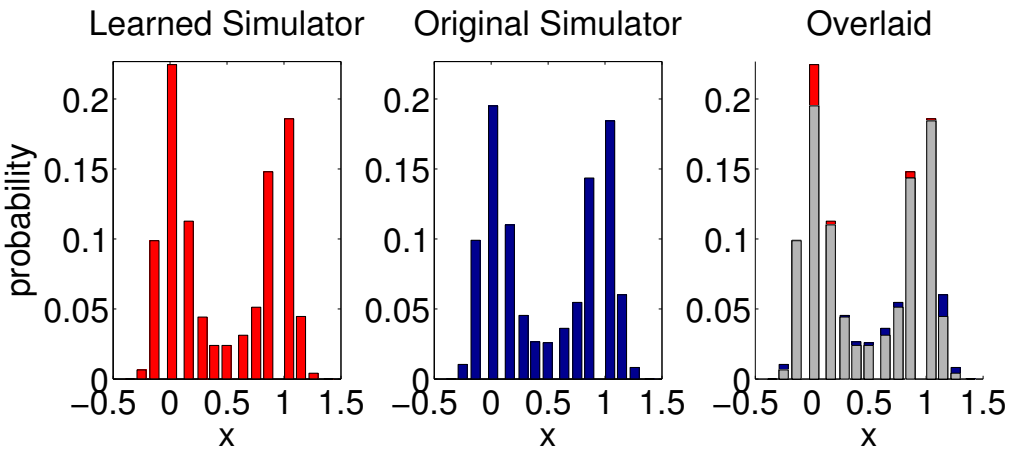

FIGURE 2.3: Stationary distribution comparison between the original simulator and the Atlas with 10^5 samples for example 2.3.2.

expensive part of the construction algorithm is running the simulations, the total cost of construction is $O(S\delta^{-d-4})$.

During each call to the Atlas, we will see from the proof that we must choose our timestep $\Delta t = t_0/\log(1/\delta)$. The $\log(1/\delta)$ term is negligible, so we will leave it from the discussion. At each call to the simulator, we must compute distance to each neighbor, of which there are $O(2^d)$. Each distance costs d flops, so the total cost of running the Atlas for time t_0 is $O(d2^d)$.

Comparing the running time of the original simulator S with the Atlas amounts to comparing the cost of S to $d2^d$. The benefit of the Atlas over the original one then clearly depends upon how expensive the original simulator was, which can depend on many factors: length of the timestep, cost of evaluating functions, ambient dimension, etc. One thing is clear - the cost of S depends on microscale properties, while $d2^d$ does not. Since the cost S varies for each problem, in our examples in section 5 we compare the Atlas cost to the original simulator cost by comparing the size of the timestep alone.

The running time is not the only benefit of the Atlas. One advantage is that

long paths of the simulator can be stored using only d dimensions rather than the ambient dimension where $\mathcal{M}$ lives. Another benefit is that some postprocessing has already been done; suppose you would like to ask the question "how long does Y_t spend near the state Y^*?" After running the original simulator one would have to compute a distance to Y^* for many data points. After running the Atlas answering this question requires only computing distances from the few net points near Y^* to Y^* to obtain the result with accuracy 2δ, and the Atlas already knows which parts of the paths are in charts near Y^*.

3

Algorithm

In this section we present the algorithm in detail, since the main result will state properties of the output of the algorithm; pseudo-code is presented in figure 3.1. First we discuss algorithms that are used during the learning phase. Then we discuss the full details of the simulator learning phase and the simulation phase. The algorithm uses several parameters:

δ: We will assume this parameter is given to us and represents the homogenization scale, and is related to the desired accuracy of the simulator via (4.7) in Theorem 2.

t_0: This represents the time short paths will be simulated for. In most examples in this paper, we choose $t_0 = \delta^2$. In practice, one should choose t_0 so that sample paths are an average distance δ from the starting location at time t_0.

m: The number of landmarks for each net point for learning the chart and transition maps. m should be $O(d)$.

p: The number of sample paths computed for each point in the net. p should

be $O(\delta^{-4})$.

Δt: Time step of the Atlas. The proofs lead us to believe Δt should be $O(\delta/\ln(1/\delta))$. In the examples we used $\delta/5$.

These choices of parameters are informed by the results and proofs in section 4. We will see that for these choices of parameters, the Atlas produces paths whose stationary distribution has error $O(\delta \ln(1/\delta))$.

3.1 Net construction

In a metric space $(\mathcal{M}, \rho)$ we define a δ-net of points as follows:

Definition 1 (δ-net)**.** *A δ-net for a metric space $(\mathcal{M}, \rho)$ is a set of points $\{y_k\}_{k\in\Gamma}$ such that*

1. $\forall k_1, k_2 \in \Gamma \ \rho(y_{k_1}, y_{k_2}) \geqslant \delta$
2. $\forall x \in \mathcal{M} \ \exists k \in \Gamma \, \rho(x, y_k) \leqslant \delta$

In view of our purposes, the first property ensures that the net points are not too close together: this is essential so we do not waste time exploring regions of the space that we have already explored, do not construct many more local simulators than needed, and do not switch between charts much more often than necessary. The second property ensures that $\{B_\delta(y_k)\}_{k\in\Gamma}$ is a cover for $\mathcal{M}$, guaranteeing that we explore the whole space $\mathcal{M}$. We will connect nearby net points: if $d(y_{k_1}, y_{k_2}) \leqslant 2\delta$ we say that y_{k_1} and y_{k_2} are neighbors and we write $k_1 \sim k_2$.

3.1.1 Computational cost

Algorithms for efficiently constructing δ-nets in metric spaces satisfying a doubling condition exist and are non-trivial, for example by constructing a data structure called cover trees (see Beygelzimer et al. (2006)): in $O(C^d n \log(n) D)$ time, where d

Main Algorithm

$\widehat{\mathcal{S}}$ = construction_phase($\{x_j\}, \rho, \mathcal{S}$)
  $\{y_k\}_{k\in\Gamma} \leftarrow \delta-\text{net}(\{x_j\})$
  for $k \in \Gamma$
    % *create* $m+1$ *landmarks for LMDS*
    $\{a_{k,l}\}_{l=1..m} = \mathcal{S}(y_k, m, t_0)$
    $A_k = y_k \cup \{a_{i,l}\}$
  end
  for $k \in \Gamma$
    % *simulate* p *paths for estimating drift and diffusion*
    $\{x_{k,l}\}_{l=1..p} = \mathcal{S}(y_k, p, t_0)$
    $L_k = \bigcup_{k\sim i} A_k$
    $[L'_k, \{x'_{k,l}\}_{l=1..p}] = \text{LMDS}(L_k, \{x_{k,l}\}_{l=1..p}, \rho)$
    $\{\widehat{\mathcal{S}}.c_{k,j}\} \leftarrow \bigcup_{j\sim k} y_j$ in L'_k
    shift coordinates so $c_{k,k} = 0$
    $\widehat{\mathcal{S}}.\bar{b}_k \leftarrow \sum_l x'_{k,l}/pt_0$
    $\widehat{\mathcal{S}}.\bar{\sigma}_k \leftarrow (\text{Cov}(\{x'_{k,l}\})/t_0)^{1/2}$
    % *compute switching maps*
    for $j \sim k$, $j < k$
      $L_{k,j} = A_k \bigcup A_j$
      $L'_{k,j} = L_{k,j}$ in L'_k coordinates
      $L'_{j,k} = L_{j,k}$ in L'_j coordinates
      $\widehat{\mathcal{S}}.\mu_{k,j} \leftarrow \mathbb{E}[L'_{k,j}]$
      $\widehat{\mathcal{S}}.\mu_{j,k} \leftarrow \mathbb{E}[L'_{j,k}]$
      $\widehat{\mathcal{S}}.T_{k,j} \leftarrow (L'_{k,j} - \mu_{k,j})^{\dagger}(L'_{j,k} - \mu_{j,k})$
    end
  end

FIGURE 3.1: Main algorithm for constructing the Atlas: it constructs the δ-net, computes chart embeddings, learns chart simulators from short sample paths, and transition maps.

is the intrinsic dimension (e.g. doubling dimension), n is the number of points in $\mathcal{M}$, and D is the cost of computing the distance between a pair of points in $\mathcal{M}$. These data structures are especially useful for both finding near points to any given point, and for constructing nets of points at multiple resolutions.

A slower, simpler algorithm for constructing a net is to add points one a time if they are farther than δ from any point already in the net; finish when no more

points can be added. For simplicity, this is the algorithm we have used in examples presented in this paper.

3.2 Landmark Multidimensional Scaling (LMDS)

LMDS takes as input a set of landmarks $L \subset \mathcal{M}$ and a set of other points $Z \subset \mathcal{M}$, and constructs a map $\Phi : L \cup Z \to \mathbb{R}^d$ embedding L, Z into $\mathbb{R}^d$. LMDS computes all pairwise distances between points L and points Z, and returns low dimensional coordinates which minimize the distortion given by

$$\sum_{i,j}(\rho(l_i, l_j)^2 - ||\Phi(l_i) - \Phi(l_j)||^2_{\mathbb{R}^d})^2 \tag{3.1}$$

over all such mappings Φ. Each new point $z \in Z$ has coordinates $\Phi(z)$ which minimize

$$\sum_{i}(\rho(l_i, z)^2 - ||\Phi(l_i) - \Phi(z)||^2_{\mathbb{R}^d})^2 \tag{3.2}$$

over all d dimensional vectors $\Phi(z)$. For a full description of the algorithm, see Silva and Tenenbaum (2004). If the distance ρ is euclidean, the algorithm reduces to principal component analysis (PCA).

If the dimension d is unknown, one could learn d at this stage from observing the eigenvalues of the squared distance matrix obtained during MDS. Eigenvalues which are of order δ^2 correspond to directions along the manifold, and eigenvalues which are of order δ^4 or lower correspond to curvature (or noise). Thus, one could learn d by choosing a cutoff threshold depending upon δ (in fact this is done in example 5.5).

3.2.1 Computational cost

The computational cost of this algorithm is $O((|L|^2 + |L| \cdot |Z|)D)$, where D is the cost of evaluating ρ.

3.3 Least-squares switching maps

We will use the pseudoinverse (see Penrose (1956)) to solve a least squares problem of finding the best linear mapping for the transition map. If X and Y are two mean zero $l \times d$ matrices, then the matrix $T = X^{\dagger}Y$ minimizes $||XT - Y||_2$ over all $d \times d$ matrices. In the construction algorithm that follows, for each connection $k \sim j$, we take a set of common landmarks $L_{k,j}$ and let $X = \Phi_k(L_{k,j})$ and $Y = \Phi_j(L_{k,j})$. Since these choices of X, Y are not mean zero, we must also shift by the corresponding mean. The charts C_k and C_j represent overlapping areas on $\mathcal{M}$, and so there will exist a matrix T which has small error.

3.3.1 Computational cost

The cost of computing the pseudoinverse is $O(ld^2)$ since we must compute the singular value decomposition of X.

3.4 Learning Phase

The first part of the Atlas algorithm is the learning phase, in which we use the sample paths to learn local chart coordinates, local simulators and transition maps. In this part of the algorithm, we store all the information necessary for the global simulator in $\widehat{\mathcal{S}}$. We will use the notation $\widehat{\mathcal{S}}.var$ to denote the variable var within the simulator $\widehat{\mathcal{S}}$. Recall that $\{x_j\}$ is a dense enough sample of $\mathcal{M}$ to produce a net at scale δ. Let $\mathcal{S}(y, p, t_0)$ denote running p paths of the simulator starting at y for time t_0. We treat all points on the charts (resulting from LMDS) as row vectors.

3.4.1 Computational cost

If $c \approx 2^d$ is the maximum number of connections each net point has, the computational cost of the construction phase, for each chart, is of order

$$\underbrace{mS}_{\text{landmark simulation}} + \underbrace{pS}_{\text{path simulation}} + \underbrace{2^d mpD}_{\text{LMDS}} = (m+p)S + 2^d mpD \tag{3.3}$$

We note that the term $2^d mpD$ can be decreased to dpD since since a d-dimensional plane may be estimated with only $O(d)$ points. Instead of using all cm points as landmarks, one could choose a (e.g. random) subset of these landmarks for the initial embedding (although all these landmarks will be needed later for computing T). All these steps are easily parallelized, so the per-chart cost above is also a per-processor cost if enough processors are available. Finally, observe that there at most $O(\delta^{-d})$ such charts (this follows from the property of the δ-net, which ensures that balls of radius $\delta/2$ centered at net points are disjoint).

3.5 The Atlas simulator

In order to define the Atlas, we must describe what a single step of time Δt looks like starting at a location x in chart i. Figure 3.2 contains the pseudocode for the algorithm implementing the strategy we now detail. The following is written assuming x is a row vector.

The Atlas runs in d dimensions, and does not require calls to the original simulator, so the running time now only depends on the local complexity of the homogenized problem. The number of simulation steps required to approach stationarity still depends upon the time it takes to converge to equilibrium, but so too did the original simulator.

If $c \approx 2^d$ is the maximum number of connections each net point has, the compu-

Atlas Simulator

$(x, i') = \text{simulator\_step}(x, i, \widehat{\mathcal{S}})$
  % *select new coordinate chart*
  $i' = \text{argmin}_j ||x - \widehat{\mathcal{S}}.c_{i,j}||^2_{\mathbb{R}^d}$
  if $i' \neq i$
    $x \leftarrow (x - \widehat{\mathcal{S}}.\mu_{i,i'})\widehat{\mathcal{S}}.T_{i,i'} + \widehat{\mathcal{S}}.\mu_{i',i}$
  end

  % *forward Euler step*
  $\eta \sim \mathcal{N}(0, I_d)$
  $x \leftarrow x + \widehat{\mathcal{S}}.\overline{b}_i \Delta t + \eta \widehat{\mathcal{S}}.\overline{\sigma}_i \sqrt{\Delta t}$

  % *prevent escape from local chart*
  if $|x| > 3\delta/2$
    $x \leftarrow \frac{x}{|x|}\left(2\delta - \frac{\delta}{2}\exp\left(3 - \frac{2}{\delta}|x|\right)\right)$
  end

FIGURE 3.2: Algorithm for running the Atlas, by combining local diffusions and linear transition map between charts.

tational cost of each time step of the Atlas is of order

$$\underbrace{d2^d}_{\text{distance computation}} + \underbrace{d^2}_{\text{forward step}} = (2^d + d)d\,. \tag{3.4}$$

4

Theoretical Results and Guarantees

In this section, we first introduce the minimum amount of material to precisely state the Theorem in section 4.1. Then we introduce the necessary mathematical objects to state the Lemmata used during the main proof in section 4.2. In section 4.3, we prove Theorem 2 and the Lemmata used.

4.1 Theorem Statement

Let $\{y_k\}_{k\in\Gamma}$ denote the set of net points. For each k we have the mapping Φ_k from $\mathcal{M}$ to $\mathbb{R}^d$ given by LMDS. If the mapping to the tangent plane at y_k is invertible on a ball of radius 2δ, then Φ_k will also be invertible on a ball of radius 2δ (see the proof of 6). Let $\mathcal{A} = B_{2\delta}(0) \times \Gamma$ equipped with the transition maps $\{S_{k,j}\}$ denote the atlas. We note that even though we call $\mathcal{A}$ an atlas, it does not quite satisfy the rigorous definition of an atlas since the transition maps are computed numerically and make small errors. Recall that the coordinates are shifted so that $\Phi_k(y_k) = 0$, so these are all points in the chart within 2δ of $\Phi_k(y_k)$. Define a mapping back to the original space by $G(x, i) = \Phi_i^{-1}(x)$ for each $x \in \mathcal{A}$.

Next we state some definitions in order to mathematically define a step of the Atlas. If $i \sim j$, let $S_{i,j}(x) = (x - \mu_{i,j})T_{i,j} + \mu_{j,i}$ be the transition map between charts i and j. Let $W(x)$ be the wall function (which keeps the simulator confined to $B_{2\delta}(0)$) defined by

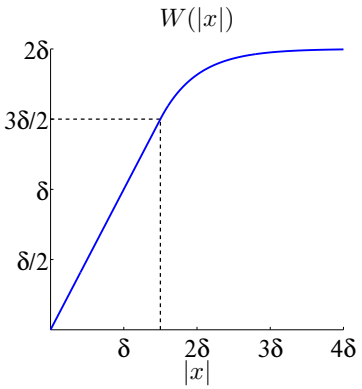

FIGURE 4.1: The wall function W.

$$W(x) = \begin{cases} x & |x| \leqslant \frac{3\delta}{2} \\ \frac{2\delta x}{|x|} - \frac{\delta x}{2|x|}\exp\left(3 - \frac{2}{\delta}|x|\right) & |x| > \frac{3\delta}{2} \end{cases}$$

There are other possible choices for W, but the main ingredients are: W is C^2, invertible, equal to the identity on a ball with radius in $3\delta/2$, and takes $\mathbb{R}^d \to B_{2\delta}(0)$.

Let B_t be a standard Brownian motion in $\mathbb{R}^d$. The update rule for the Atlas starting at (x_0, i_0) is

$$i_{k+1} = \text{argmin}_j \left|x_k - c_{i_k,j}\right| \tag{4.1}$$

$$x_{k+1} = W\left(S_{i_k,i_{k+1}}(x_k) + \bar{b}_{i_{k+1}}\Delta t + \bar{\sigma}_{i_{k+1}}B_{\Delta t}\right) \tag{4.2}$$

We show in Lemma 3 that under the conditions of Theorem 2, the Atlas has a unique stationary distribution μ on $\mathcal{A}$. Next define the continuous time process $\widehat{X}_t^z$ starting at $z = (x, i) \in \mathcal{A}$ by

$$i' = \text{argmin}_j \left|x - c_{i,j}\right| \tag{4.3}$$

$$\widehat{X}_t^z = W\left(S_{i,i'}(x) + \bar{b}_{i'}t + \bar{\sigma}_{i'}B_t\right) \tag{4.4}$$

Let $\widehat{P}_t$ denote the transition density for $\widehat{X}_t$ on $\mathcal{A}$. Then $\mu\widehat{P}_t$ is the distribution of $\widehat{X}_t$ with initial condition μ. Define a measure $\widehat{q}$ on $\mathcal{A}$ by

$$\widehat{q} = \int_0^{\Delta t} \mu\widehat{P}_t dt \tag{4.5}$$

Samples of $\widehat{q}$ can be approximated by the Atlas by running N steps of size Δt, and then one step of size $\tau \sim$ Unif$(0, \Delta t)$, where N is large enough to well approximate μ. Choosing N is not easy and depends on the problem, although this is a difficulty with the original simulator as well; in practice, one should choose N large enough that many simulations reach a large fraction of the charts in the simulator.

Theorem 2. *Let $(Y_t)_{t \geqslant 0}$ be an Itô diffusion on a smooth compact connected d-dimensional manifold $\mathcal{M}$ with no boundary:*

$$dY_t = b(Y_t)dt + \sigma(Y_t)dB_t\,, \tag{4.6}$$

with b, σ Lipschitz, and σ uniformly elliptic, i.e. there exists $\lambda > 0$ such that for all $x \in \mathcal{M}$ and $v \in T_x(\mathcal{M})$ $\sigma(x)\sigma(x)^T v \geqslant \lambda |v|$. Let δ be small enough so that for every x the orthogonal projection $\mathcal{M} \to T_x(\mathcal{M})$ is invertible on a ball of radius 2δ on $T_x(\mathcal{M})$. Let q be the unique stationary distribution of Y_t. Let $\widehat{q}$ be the measure on $\mathcal{A}$ generated by the Atlas, as defined above in (4.5)*. There exists constants c_1, c_2 such that if the number of sample paths satisfies $p > c_1(\tau, \mathcal{M})/\delta^4$ then with probability at least $1 - 2\exp(-\tau^2)$,*

$$||q - G_* \widehat{q}||_{L^1(\mathcal{M})} \leqslant c_2 \delta \ln(1/\delta) \tag{4.7}$$

Lipschitz coefficients guarantee existence and uniqueness; strong ellipticity (together with smoothness and connectivity of $\mathcal{M}$) guarantees that the process Y_t has a unique stationary distribution (this latter assumption may be weakened to include hypo-elliptic systems).

From a computational perspective, we note that as G appears in the statement of the Theorem, it would be very useful to compute this mapping G. This is in general hard in an arbitrary metric spaces, although one can use the approximation $\widehat{G}(x, i) = y_i$ which approximates G at scale δ.

4.2 Preliminaries for Proofs

Before diving into the proof of Theorem 2, we need several definitions. First we construct an intermediate space $\mathcal{N}$ on which to compare the stationary distributions of the processes to be considered. For this process, and what follows, we restrict each Φ_k to a ball of radius 2δ in the range (so the domain of Φ_k is the set of all points in $\mathcal{M}$ which the original Φ_k takes inside a ball of radius 2δ.) $\mathcal{N}$ is obtained by smoothly joining together charts in $\mathcal{A}$ via the true transition maps (so $\mathcal{N}$ is truly an atlas for $\mathcal{M}$). Define an equivalence relation $\sim$ on $\mathcal{A}$ (the symbol is the same for connections between net indices, but by the argument it should always be clear which one is meant):

$$(x, i) \sim (y, j) \iff i \sim j \text{ and } \Phi_i(\Phi_j^{-1}(y)) = x \tag{4.8}$$

This definition means that there is a point $z \in \mathcal{M}$ such that $\Phi_i(z) = x$ and $\Phi_j(z) = y$ provided $(x, i) \sim (y, j)$. This is a natural way of defining an equivalence between points in neighboring charts. Let $\mathcal{N} = \mathcal{A}/\sim$ be the quotient space. Define $\Phi : \mathcal{M} \to \mathcal{N}$ by

$$\Phi(z) = \{(x, i) \,:\, \Phi_i(z) = x\} \tag{4.9}$$

We can see that Φ maps to equivalence classes in $\mathcal{N}$ by directly comparing to the definition of $\sim$ in (4.8). For any point $z \in \mathcal{M}$, by the construction of the net, there is a net point y_k such that $|z - y_k| \leqslant \delta$. Let $\Psi_k(x, k) = \{(y, j) \,:\, (x, k) \sim (y, j)\}$ be the mapping that takes points in chart k and maps them to their equivalence class in $\mathcal{N}$. Then there exists a neighborhood around z on which $\Phi(\cdot)$ agrees with $\Psi_k(\Phi_k(\cdot), k)$. For each k, Φ_k is smooth and invertible (proved in lemma 6) on any neighborhood contained in a ball of radius 2δ. Also for each k, Ψ_k is smooth and invertible on any neighborhood contained in a ball of radius 2δ. Since z is arbitrary, Φ agrees with a smooth invertible mapping on a neighborhood around any point $z \in \mathcal{M}$, which

implies that Φ is smooth and invertible everywhere and thus a diffeomorphism. This implies $\mathcal{N}$ is in fact a smooth manifold because $\mathcal{M}$ is.

We define several processes on $\mathcal{N}$. We start with process $X_t := \Phi(Y_t)$; X_t is the solution of an SDE, most easily written in local coordinates: if $X_0 = \Psi_i(x_0, i) \in \mathcal{N}$, let τ be the time when X_t first leaves C_i (in the sense that X_τ no longer contains an element with second entry i). We can track the position in $\mathbb{R}^d$ on the chart C_i. With some abuse of notation, in the following equation we will let $\{X_t\}_{t \leqslant \tau}$ represent the coordinates in chart i, even though X_t should refer to an equivalence class in $\mathcal{N}$. Let $\Phi_{i,k}$ denote the k^{th} coordinate of Φ_i. Then X_t solves for $t \leqslant \tau$, the Itô SDE

$$dX_t = b_i(X_t)dt + \sigma_i(X_t)dB_t \tag{4.10}$$

$$(b_i)_k(x) := \nabla\Phi_{i,k}(\Phi_i^{-1}(x)) \cdot b(\Phi_i^{-1}(x)) + \frac{1}{2}\sum_{j,l} \frac{\partial\Phi_{i,k}}{\partial x_j \partial x_l}(\Phi_i^{-1}(x))(\sigma\sigma^T)_{j,l}(\Phi_i^{-1}(x)) \tag{4.11}$$

$$(\sigma_i)_{k,j}(x) := \sum_l \frac{\partial\Phi_{i,k}}{\partial x_l}(\Phi_i^{-1}(x))\sigma_{l,j}(\Phi_i^{-1}(x)) \tag{4.12}$$

Let $\mathcal{L}$ be the generator (see Øksendal (2003)) for X_t, and $\mathcal{L}_i$ its restriction to chart C_i. $\mathcal{L}$ is uniformly elliptic on $\mathcal{N}$, since Y_t is uniformly elliptic on $\mathcal{M}$ and Φ is a diffeomorphism.

We will be comparing a number of simulators to bridge the gap between the simulation scheme and the true simulator. For this reason we will write the Atlas $\widehat{X}_t$ starting at $z = (x, i)$ as

$$\bar{X}_t^z = \Phi_{i'}\left(\Phi_i^{-1}(x)\right) + \bar{b}_{i'}t + \bar{\sigma}_{i'}B_t \tag{4.13}$$

$$\widetilde{X}_t^z = S_{i,i'}(x) + \bar{b}_{i'}t + \bar{\sigma}_{i'}B_t \tag{4.14}$$

$$\widehat{X}_t^z = W\big(\widetilde{X}_t^z\big) \tag{4.15}$$

where i' is defined as in (4.3). The processes $\bar{X}_t, \widetilde{X}_t$ are natural stepping stones from X_t to $\widehat{X}_t$: $\bar{X}_t$ is the process on $\mathcal{C}_{i'}$ which differs from X_t only in that it uses the

learned drift and diffusion coefficients. $\widetilde{X}_t$ differs from $\bar{X}_t$ only in that it uses the learned transition map $S_{i,i'}$ rather than the true transition map $\Phi_{i'} \circ \Phi_i^{-1}$. Given any initial condition $z = (x, i)$, the three processes $\bar{X}_t^z, \widetilde{X}_t^z, \widehat{X}_t^z$ are solutions of SDE's with generators $\bar{\mathcal{L}}_z, \widetilde{\mathcal{L}}_z, \widehat{\mathcal{L}}_z$, respectively, on chart i'. These generators clearly depend on the chart i', but as we will see in Lemma 6, they will also depend on x, and we keep track of this by putting z as a subscript on the generators. With similar notation, we let $\mathcal{L}_z = \mathcal{L}_{i'}$, the generator of the true process X_t on chart i'. We will prove that these generators are close to one another for all $z \in \mathcal{A}$, and then show that this is enough to imply long time bounds on the stationary distributions.

Let $Z_k = \left(X_{\Delta t}^{Z_{k-1}}, i_k\right)$ denote the Markov chain on $\mathcal{A}$ given by running k steps of the Atlas starting at $Z_0 \in \mathcal{A}$. The Markov chain Z_k is the process which we will show to be ergodic (see Lemma 3). Note that a continuous time version of Z_k would not be time homogeneous since the process is only allowed to transition between charts at fixed time intervals.

4.3 Proofs

Before we begin the proof of Theorem 2, we state some Lemmata which we will prove later, in order to keep the details until the end, while first showing the main ideas of the proof.

Lemma 3. *The process Z_k is ergodic with stationary distribution μ.*

Lemma 4. *For any smooth test function $f : \mathcal{A} \to \mathbb{R}$ and initial condition,*

$$\frac{1}{n} \sum_{k=1}^{n} \int_0^{\Delta t} f\left(Z_k, \widehat{X}_t^{Z_k}\right) dt \to \mathbb{E}_\mu \left[\mathbb{E} \left[\int_0^{\Delta t} f\left(z, \widehat{X}_t^z\right) dt \right] \right] \tag{4.16}$$

a.s. as $n \to \infty$. Here $\mathbb{E}_\mu$ means taking the expectation over the initial condition $z \sim \mu$, and $\mathbb{E}$ is taking the expectation over the transition probabilities of $\widehat{X}_t^z$.

Lemma 5. *There exists a constant C such that for any test function f, and initial condition $z \in \mathcal{A}$, with probability at least $1 - 4\exp(-\tau^2)$*

$$\mathbb{E}\left[\frac{1}{\Delta t}\int_0^{\Delta t}\left(\mathcal{L}_z - \bar{\mathcal{L}}_z\right) f\left(\widehat{X}_t^z\right) dt\right] \leqslant C\delta\tau\sqrt{\ln(1/\delta)}||f||_{C^2} \tag{4.17}$$

Lemma 6. *There exists a constant C such that for any test function f, and initial condition $z \in \mathcal{A}$,*

$$\mathbb{E}\left[\frac{1}{\Delta t}\int_0^{\Delta t}\left(\bar{\mathcal{L}}_z - \widetilde{\mathcal{L}}_z\right) f\left(\widehat{X}_t^z\right) dt\right] \leqslant C\delta\ln(1/\delta)||f||_{C^2} \tag{4.18}$$

Lemma 7. *There exists a constant C such that for any test function f, and initial condition $z \in \mathcal{A}$,*

$$\mathbb{E}\left[\frac{1}{\Delta t}\int_0^{\Delta t}\left(\widehat{\mathcal{L}}_z - \widetilde{\mathcal{L}}_z\right) f\left(\widehat{X}_t^z\right) dt\right] \leqslant C\delta||f||_{C^2} \tag{4.19}$$

Proof of Theorem 2. This proof follows the ideas and techniques from Mattingly et al. (2010) for proving long time convergence of numerical schemes. We present here a short version of the arguments contained in that paper. By assumption, the operator $\mathcal{L}$ is uniformly elliptic on $\mathcal{N}$ (for more details on how to define these operators on manifolds see Stroock (2008), Hsu (2002)). Let $\phi : \mathcal{N} \to \mathbb{R}$ be a smooth test function on $\mathcal{N}$ and define the average $\bar{\phi}$ by

$$\bar{\phi} = \int_{\mathcal{M}} \phi(\Phi(y))dq(y) = \int_{\mathcal{N}} \phi(x)d(\Phi_* q)(x)\,, \tag{4.20}$$

where $\Phi_* q$ is the push forward of q through Φ. Defined by:

$$\Phi_* q(A) = \int_{\mathcal{M}} \mathbb{1}_A(\Phi(x))dq(x) \tag{4.21}$$

$\Phi_* q$ is stationary for X_t since q is stationary for Y_t. Therefore $(\phi - \bar{\phi}) \perp \text{Null}(\mathcal{L}^*)$, and by the Fredhölm alternative there exists a unique solution ψ to the Poisson equation $\mathcal{L}\psi = \phi - \bar{\phi}$. Uniform ellipticity implies, via standard estimates Krylov (1996), $||\psi||_{C^2} \leqslant C_{\mathcal{M},\lambda}||\phi||_\infty$.

For ease of notation, let C_k be the chart associated with the index i_k, $\widehat{\mathcal{L}}_k = \widehat{\mathcal{L}}_{Z_{k-1}}$, $\widehat{X}_t^{(k)} = \widehat{X}_t^{Z_{k-1}}$, and $\psi_k = \psi\big|_{C_k}$. Also let $\left\{B_t^{(k)}\right\}_{k=1}^{\infty}$ denote independent Brownian motions. The function ψ_k is smooth on C_k, so by Itô's formula:

$$\psi_k\big(\widehat{X}_{\Delta t}^{(k)}\big) - \psi_k\big(\widehat{X}_0^{(k)}\big) = \int_0^{\Delta t} \widehat{\mathcal{L}}_k \psi_k\big(\widehat{X}_t^{(k)}\big) dt + \int_0^{\Delta t} \nabla\psi_k\big(\widehat{X}_t^{(k)}\big)\widehat{\sigma}_{i_k} dB_t^{(k)} \tag{4.22}$$

By Itô's isometry, letting $||A(\cdot)||_{F,\infty} := ||\,||A(x)||_F\,||_{L^\infty(\mathcal{M})}$,

$$\mathbb{E}\left[\left(\int_0^{\Delta t} \nabla\psi_k\big(\widehat{X}_t^{(k)}\big)\widehat{\sigma}_{i_k} dB_t^{(k)}\right)^2\right] \leqslant \Delta t ||\psi||_{C^1}^2 ||\widehat{\sigma}||_{F,\infty}^2 \tag{4.23}$$

Define the martingale M_n by

$$M_n = \frac{1}{n\Delta t}\sum_{k=1}^{n}\int_0^{\Delta t} \nabla\psi_k\big(\widehat{X}_t^{(k)}\big)\widehat{\sigma}_{i_k} dB_t^{(k)}$$

When calculating the variance of M_n, cross terms vanish by independence. Then from equation (4.23) we obtain the bound

$$\mathbb{E}[M_n^2] \leqslant \frac{1}{n\Delta t}||\psi||_{C^1}^2 ||\widehat{\sigma}||_{F,\infty}^2$$

which implies $M_n \to 0$ a.s. as $n \to \infty$ by the martingale convergence theorem. Summing equation (4.22) and dividing by $n\Delta t$,

$$\frac{1}{n\Delta t}(\psi_k(Z_n) - \psi_k(Z_0)) = M_n + \frac{1}{n\Delta t}\sum_{k=1}^{n}\int_0^{\Delta t} \widehat{\mathcal{L}}_k \psi_k\big(\widehat{X}_t^{(k)}\big) dt$$

ψ is bounded so $\psi_k(Z_n)/n\Delta t \to 0$. Taking $n \to \infty$ on both sides and using Lemma 4,

$$0 = \mathbb{E}_\mu\left[\mathbb{E}\left[\frac{1}{\Delta t}\int_0^{\Delta t} \widehat{\mathcal{L}}_z \psi_z\big(\widehat{X}_t^z\big) dt\right]\right]$$

Where if $z = (x, i)$, $\psi_z = \psi_{i'}$ with i' defined as in equation (4.3). Using Lemma 5, 6 and 7, we have with probability at least $1 - 4\exp(-\tau^2)$,

$$\mathbb{E}_\mu\left[\mathbb{E}\left[\frac{1}{\Delta t}\int_0^{\Delta t} \mathcal{L}_z \psi_z\big(\widehat{X}_t^z\big) dt\right]\right] \leqslant c\delta\tau\ln(1/\delta)||\phi||_\infty$$

Define the mapping $I : \mathcal{A} \to \mathcal{N}$ to be the mapping which takes a point in $\mathcal{A}$ and maps it to its equivalence class in $\mathcal{N}$. Since ψ solves the Poisson equation $\mathcal{L}\psi = \phi - \bar{\phi}$ and $\mathcal{L}_z\psi_z$ agrees with $\mathcal{L}\psi$ on the appropriate chart,

$$\left| \int_{\mathcal{N}} \phi \, d(\Phi_* q) - \int_{\mathcal{N}} \phi \, d(I_* \widehat{q}) \right| \leqslant c\delta\tau \ln(1/\delta) ||\phi||_\infty$$

Since equation (4.24) holds for all ϕ,

$$||\Phi_* q - I_* \widehat{q}||_{\mathbb{L}^1(\mathcal{N})} \leqslant c\delta\tau \ln(1/\delta) \tag{4.24}$$

Pushing the measures through Φ^{-1} yields the result since $\Phi^{-1} \circ I = G$. □

Proof of Lemma 3. Now suppose we use the update rule starting at $Z_0 = (x_0, i_0)$

$$i_{k+1} = \text{argmin}_j \left| x_k + \eta u - c_{i_k,j} \right| \tag{4.25}$$

$$x_{k+1} = W\left(S_{i_k,i_{k+1}}(x_k) + \bar{b}_{i_{k+1}} \Delta t + \bar{\sigma}_{i_{k+1}} B_{\Delta t} \right) \tag{4.26}$$

$$Z^\eta_{k+1} = (x_{k+1}, i_{k+1}) \tag{4.27}$$

with u a uniform random variable, and $\eta > 0$. Note that $\eta = 0$ in the algorithm detailed in section 3 and in equation (4.1). If $\eta = 0$, the process Z^0_n is not Feller continuous ($\mathbb{E}[Z^0_1]$ does not depend continuously on the initial conditions), a common assumption implying that Z_n has a stationary distribution. We start with $\eta > 0$ and later will take $\eta \to 0$ and show that the η-dependent stationary measures converge to a new stationary measure. For now, we drop the superscript η to simplify notation, but we will come back to it.

First we show that the process Z_n is Feller continuous. Let f be a bounded function on $\mathcal{A}$ and $(x, i) \in \mathcal{A}$. Let $p_j(x, i)$ denote the probability of transitioning to chart j from i starting at x. Note that p_j is continuous and bounded provided $\eta > 0$. If $n = 1$,

$$\mathbb{E}_{(x,i)}[f(Z_1)] = \sum_{j \sim i} p_j(x, i) \mathbb{E}_{(x,i)}[f(Z_1) | i_1 = j] \tag{4.28}$$

Since $\widehat{X}_t^{Z_0}$ conditioned on i_1 is an Itô process, it is Feller continuous (see Øksendal (2003)). Thus $\mathbb{E}_{(x,i)}[f(\widehat{X}_{\Delta t}^{Z_0})|i_1 = j]$ is continuous and bounded and then by equation (4.28), $\mathbb{E}_{(x,i)}[f(Z_1)]$ is continuous and bounded. By the induction step, assume $u(x,i) = \mathbb{E}_{(x,i)}[f(Z_n)]$ is continuous and bounded. Then

$$\mathbb{E}_{(x,i)}[f(Z_{n+1})] = \mathbb{E}_{(x,i)}\big[\mathbb{E}_{(y,j)}[f(Z_n)]\big|Z_1 = y, i_1 = j\big] = \mathbb{E}_{(x,i)}[u(Z_1)]\,, \tag{4.29}$$

which is continuous and bounded. Thus by induction on n, Z_n is Feller continuous for all $\eta > 0$.

Next we show the transition density of Z_n is tight for all n. Fix $\varepsilon > 0$. Let $z = Z_{n-1}$. Then $\widetilde{X}_{\Delta t}^z$ is Gaussian with mean $\widetilde{b}_z \Delta t$ and variance $\widetilde{\sigma}_z \widetilde{\sigma}_z^T \Delta t$ (see equation (4.52) for their definitions). $\sup_{z\in\mathcal{A}} b_z$ and $\sup_{z\in\mathcal{A}} \widetilde{\sigma}_z$ are bounded. Thus there exists an R such that $\mathbb{P}[\widetilde{X}_{\Delta t}^z \in B_R(0)] > 1 - \varepsilon$ for all $z \in \mathcal{A}$, and thus $\mathbb{P}[Z_n \in W(B_R(0))] > 1 - \varepsilon$. $W(B_R(0))$ is compact which implies the transition density of Z_n is tight. The transition density is tight and Feller continuous, so by the Krylov-Bogolyubov Theorem Zabczyk (1996) there exists an invariant measure.

Now observe that for any $(x,i),(y,j) \in \mathcal{A}$, if A_y is a neighborhood of y in chart C_j then $\mathbb{P}[Z_n \in A_y] > 0$ for large enough n. The $\delta-$net is connected since $\mathcal{M}$ is connected. Thus there exists a finite length path $\{i_k\}$ with $i_k \sim i_{k-1}$, $i_0 = i$, $i_n = j$. The probability of such a path occurring is strictly positive since the probability of jumping from i_{k-1} to i_k is strictly positive for all k. The density of Z_n is strictly positive on the chart i_n intersected with $\mathcal{A}$, thus $\mathbb{P}[Z_n \in A_y] > 0$. Because the density of Z_n is positive on any open set in $\mathcal{A}$ for large enough n, the stationary distribution is unique, and thus Z_n is ergodic.

Now let μ_η denote the stationary measure for Z_n^η for each $\eta > 0$. The family of measures μ_η is tight, and so there exists a subsequence $\{\mu_{\eta_k}\}_{k=1}^\infty$ which converges in probability to some measure μ (see Billingsley (1999)). It is left to show that μ is stationary for the process Z_n. Let $\varepsilon > 0$ and let f be a bounded function on $\mathcal{A}$. Let

$\mu_k = \mu_{\eta_k}$. Then $|f\mu_k - f\mu| \to 0$. Let P_k denote the transition density for the process with stationary distribution μ_k. Similarly let P denote the transition kernel of the process Z_n with $\eta = 0$.

$$|fP\mu - f\mu| \leqslant |fP\mu_k - fP_k\mu_k| + |fP\mu - fP\mu_k| + |f\mu_k - f\mu| \tag{4.30}$$

The last two terms go to zero as $k \to \infty$ because μ_k converges to μ in probability and fP is bounded. It is left to show that $(P - P_k)\mu_k \to 0$. Let E denote the boundary set defined by

$$E = \{(x, i) : \exists j \text{ with } |x| = |x - c_{i,j}|\}$$

The chart centers $c_{i,j}$ are a finite set and so E has μ measure zero. Let E_k denote the set E thickened by η_k:

$$E_k = \big\{(x, i) : \exists j \text{ with } \big||x| - |x - c_{i,j}|\big| < \eta_k\big\}$$

Fix $z = (x, i) \in \mathcal{A}$ and notice that the probability density starting at z, $P_k\delta_z$, for any k is of the form

$$P_k\delta_z = \sum_{j\sim i} p_j^k(z)\nu_j$$

with $p_j^k(z)$ being the probability of transitioning to chart j from i with $\eta = \eta_k$, and ν_j independent of k. For any j, ν_j is absolutely continuous with respect to lebesgue measure on $\mathbb{R}^d$, and $\nu_j(E_k) \to 0$. Then it follows that

$$\mu_k(E_k) \leqslant \sup_{z\in\mathcal{A}} \mathbb{1}_{E_k} P_k\delta_z \to 0.$$

Since P and P_k agree on the set E_k^c, $|fP_k\mu_k - fP\mu| \to 0$. Thus, $fP\mu = f\mu$ for all test functions f, and therefore $P\mu = \mu$. □

Proof of Lemma 4. First let

$$I_n = \int_0^{\Delta t} f\big(Z_n, \widehat{X}_t^{(n)}\big)dt \tag{4.31}$$

Then define a new Markov chain $Q_n = (Z_n, Z_{n+1}, I_n)$. Define a family of measures ν on $\mathbb{R}$ by

$$\nu(z_1, z_2, A) = \mathbb{P}[I_n \in A | Z_n = z_1, Z_{n+1} = z_2] \tag{4.32}$$

Let γ be a measure on $\mathcal{A} \times \mathcal{A} \times \mathbb{R}$ so that

$$\gamma(A) = \int_{\mathcal{A}\times\mathcal{A}\times\mathbb{R}} \mathbb{1}_A(r)\nu(z_1, z_2, dr)P(z_1, dz_2)\mu(dz_1) \tag{4.33}$$

Where P is the transition density for Z. Because Z_n is ergodic, $P^n(\delta_{(x,i)}, \cdot) \to \mu(\cdot)$ weakly. Then by the dominated convergence theorem as $n \to \infty$,

$$\int_{\mathcal{A}\times\mathcal{A}\times\mathbb{R}} \mathbb{1}_A(r)\nu(z_1, z_2, dr)P(z_1, dz_2)P^n(\delta_{(x,i)}, dz_1) \to \gamma(A) \tag{4.34}$$

The last statement shows that the density of Q_n converges weakly to γ, and so Q is ergodic. Pick $\phi(Q_n) = I_n$. Then by Birkhoff's ergodic theorem (see Yuri (1998)),

$$\frac{1}{n}\sum_{k=1}^{n} \phi(Q_k) \to \int \phi d\gamma = \mathbb{E}_\mu \left[\mathbb{E}\left[\int_0^{\Delta t} f\left(z, \widehat{X}_t^z\right) dt \right] \right] \tag{4.35}$$

□

Proof of Lemma 5. Choose some $z = (x, i) \in \mathcal{A}$ and $f \in C^2$. Then the generators $\mathcal{L}_z, \bar{\mathcal{L}}_z$ are given by

$$\mathcal{L}_z f(y) = \sum_j \left(b_i(y)\right)_j \frac{\partial f}{\partial y_j}(y) + \frac{1}{2}\sum_j \sum_k \left(\sigma_i(y)\sigma_i^T(y)\right)_{j,k} \frac{\partial^2 f}{\partial y_j \partial y_k}(y)$$

$$\bar{\mathcal{L}}_z f(y) = \sum_j \left(\bar{b}_i\right)_j \frac{\partial f}{\partial y_j}(y) + \frac{1}{2}\sum_j \sum_k \left(\bar{\sigma}_i\bar{\sigma}_i^T\right)_{j,k} \frac{\partial^2 f}{\partial y_j \partial y_k}(y)$$

It suffices to show that $b_i(y)$ is close to $\bar{b}_i$ and $\sigma_i(y)\sigma_i^T(y)$ is close to $\bar{\sigma}_i\bar{\sigma}_i^T$ for each y within 2δ of 0 and all i. Let $x_k = x'_{i,k}$ be the projections of our random samples onto $\mathcal{N}$. These are samples of X_t starting at $c_{i,i} = 0$; as $p \to \infty$,

$$t\bar{b}_i \to \mathbb{E}\left[X_t\right] \qquad , \qquad t\bar{\sigma}_i\bar{\sigma}_i^T \to \mathrm{Cov}\left(X_t\right)$$

a.s. by the strong law of large numbers. Next in order to use finite sample bounds, we show that the random variables x_k are sub-gaussian with sub-gaussian norm $t_0\kappa(|\sigma|_{F,\infty} + t_0|b|_\infty)$ for some universal constant κ. To do this, we first show Y_{t_0} is sub-gaussian.

Rewrite the process Y_s by the definition of the Itô integral (see Øksendal (2003)). Here we use a uniform partition of $(0, s)$ with n subintervals so $\Delta s = s/n$, $s_j = j\Delta s$, $Y_j = Y_{s_j}$ and z_j are independent standard random normal vectors in $\mathbb{R}^d$. Then $Y_s - Y_0$ can be written

$$Y_s - y_0 = \lim_{n\to\infty} \frac{1}{n} \sum_{j=0}^{n-1} b(Y_j)\Delta s + \sigma(Y_j)\sqrt{\Delta s} z_j \tag{4.36}$$

Note that we can always think of equations (4.6),(4.36) as being in $\mathbb{R}^D$ by the Whitney Embedding Theorem. If one is concerned about how to make sense of equation (4.36) on a manifold in $\mathbb{R}^D$, see Hsu (2002). Using proposition 5.10 in Vershynin (2012) on the right hand side of equation (4.36), we can see that there is a universal constant c so that for each n,

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{j=0}^{n-1} b(Y_j)\Delta s + \sigma(Y_j)\sqrt{\Delta s} z_j\right| > \alpha\right] \leqslant \exp\left(\frac{-c\alpha^2}{s(|\sigma|_{F,\infty} + s|b|_\infty)}\right) \tag{4.37}$$

Taking $n \to \infty$ we conclude that the sub-gaussian norm of Y_{t_0} is bounded by $\sqrt{t_0\kappa(|\sigma|_{F,\infty} + t_0|b|_\infty)}$ for a universal constant κ. Then X_{t_0} is also sub-gaussian with the same sub-gaussian norm since Φ_i is a projection. Then $|t_0\bar{b}_i - t_0\mathbb{E}[\bar{b}_i]|$ can be written as a sum of mean zero sub-gaussians and by Vershynin (2012) there exists a c_1 such that,

$$\mathbb{P}[\,|\bar{b}_i - \mathbb{E}[\bar{b}_i]| < \varepsilon] \geqslant 1 - e \cdot \exp\left(-c_1\varepsilon^2 p t_0\right) \tag{4.38}$$

Again by Vershynin (2012) bounds on finite sample covariance estimation yields for

some c_2,

$$\mathbb{P}\Big[\,\big|\big|\bar{\sigma}_i\bar{\sigma}_i^T - \mathbb{E}\left[\bar{\sigma}_i\bar{\sigma}_i^T\right]\big|\big|_2 < \varepsilon\big|\big|\mathbb{E}\left[\bar{\sigma}_i\bar{\sigma}_i^T\right]\big|\big|_2\Big] \geqslant 1 - 2e^{-c_2 d\alpha^2} \tag{4.39}$$

provided $p > d\alpha^2/\varepsilon^2$. Notice that t_0 appears in the bound in equation (4.38), but not in equation (4.39). This is due to the fact that for $t_0 \ll 1$, the mean is much smaller than the standard deviation (and thus harder to estimate). Estimating the covariance to within accuracy ε takes $O(d/\varepsilon^2)$ samples, but estimating the drift to within accuracy ε takes $O(1/t_0\varepsilon^2)$ samples. Assuming $t_0 = \delta^2 \ll 1/d$, the mean will be more difficult to estimate. For simplicity we will assume that the covariance has the same bound as the drift.

Next we must ensure that the probabilistic bound holds for all indices $i \in \Gamma$. Since the volume of $\mathcal{M}$ is fixed, $|\Gamma| = c_3(1/\delta)^d$ for some c_3. Next set the accuracy to $\varepsilon = \delta\ln(1/\delta)$, and the confidence $\tau^2 = c_1\varepsilon^2 p t_0 - (1 + \ln(c_3) + d\ln(1/d))$. When we take a union bound over $i \in \Gamma$ we have with probability at least $1 - 2e^{-\tau^2}$,

$$|\bar{b}_i - \mathbb{E}[\bar{b}_i]| < \delta\ln(1/\delta) \text{ and } \big|\big|\bar{\sigma}_i\bar{\sigma}_i^T - \mathbb{E}\left[\bar{\sigma}_i\bar{\sigma}_i^T\right]\big|\big|_2 < \delta\ln(1/\delta) \tag{4.40}$$

$$\text{if } p > \frac{c_1}{t_0\delta^2}\left(\frac{\tau^2 + 1 + \ln(c_3)}{\ln(1/\delta)^2} + \frac{d}{\ln(1/\delta)}\right) \tag{4.41}$$

We can think of equation (4.41) as telling us $p > c_4/\delta^4$ since $t_0 = \delta^2$ and $\tau, d, \ln(1/\delta)$ all behave like $O(1)$ constants.

Φ_i is smooth, so b_i, σ_i are Lipschitz and bounded since b, σ are Lipschitz and bounded, by some constant M. By the Cauchy-Schwarz inequality and Itô's isometry,

$$\mathbb{E}[|X_{t_0}|^2] \leqslant M^2 t_0 + O({t_0}^{3/2}) \qquad , \qquad \mathbb{E}\left[\int_0^{t_0} |X_s|^2 ds\right] \leqslant \frac{1}{2}M^2 {t_0}^2 + O({t_0}^{5/2}) \tag{4.42}$$

Let $A = \int_0^{t_0} b_i(X_s) - b_i(0)ds$ and $B = \int_0^{t_0} \sigma_i(X_s)dB_s$. Then by Jensen's inequality,

$$|\mathbb{E}[X_{t_0}] - t_0 b(0)|^2 \leqslant \mathbb{E}[|A|^2] \leqslant \mathbb{E}\left[t_0\int_0^{t_0} |b_i(X_s) - b_i(0)|^2 ds\right] \leqslant \frac{1}{2}C^2M^2{t_0}^3 + O({t_0}^{7/2}) \tag{4.43}$$

Dividing by t_0 and taking a square root,

$$|\mathbb{E}[\bar{b}_i] - b_i(0)| \leqslant \sqrt{\frac{t_0}{2}} MC + O({t_0}^{3/4}) \tag{4.44}$$

By Itô's isometry we have

$$\mathbb{E}[|B|^2] \leqslant M^2 t_0 \tag{4.45}$$

Combining equations (4.45) and (4.43),

$$|\mathrm{Cov}(A+B) - \mathrm{Cov}(B)|_2 \leqslant \mathbb{E}[|A|^2] + 2\mathbb{E}[|A|^2]^{1/2}\mathbb{E}[|B|^2]^{1/2} \leqslant \sqrt{2}CM^2{t_0}^2 + O({t_0}^{9/4}) \tag{4.46}$$

Using Itô isometry and the Lipschitz condition on σ_i,

$$|\mathrm{Cov}(B) - t_0\sigma_i(0)\sigma_i^T(0)|_2 = \left|\mathbb{E}\left[\left(\int_0^{t_0} \sigma_i(X_s) - \sigma_i(0)dB_s\right)\left(\int_0^{t_0} \sigma_i(X_s)dB_s\right)^T \right.\right. \tag{4.47}$$

$$\left.\left. + \left(\int_0^{t_0} \sigma_i(0)dB_s\right)\left(\int_0^{t_0} \sigma_i(X_s) - \sigma_i(0)dB_s\right)^T \right]\right| \tag{4.48}$$

$$\leqslant \sqrt{2}KM{t_0}^{3/2} + O({t_0}^{7/4}) \tag{4.49}$$

Combine equations (4.44) with the concentration inequality (4.40) along with $t_0 = \delta^2$ implies for some c_5 with probability at least $1 - 2e^{-\tau^2}$,

$$|\bar{b}_i - b_i(0)| \leqslant c_5\delta\ln(1/\delta) \tag{4.50}$$

Combine equations (4.46), (4.49) with the concentration inequality (4.40) to find for some c_6 with probability $1 - 2e^{-\tau^2}$,

$$\left|\left|\bar{\sigma}_i\bar{\sigma}_i^T - \sigma_i(0)\sigma_i^T(0)\right|\right|_2 \leqslant c_6\delta\ln(1/\delta) \tag{4.51}$$

Lipschitz conditions on b_i and σ_i yield the result.

□

Proof of Lemma 6. Fix a starting location $z = (x, i) \in \mathcal{A}$. We can write an SDE for $\widetilde{X}_t^z$ starting at $\Phi_{i'}\left(\Phi_i^{-1}(x)\right)$ in the next chart i' by

$$\begin{aligned} d\widetilde{X}_t^z &= \widetilde{b}_z dt + \widetilde{\sigma}_z dB_t \\ \widetilde{b}_z &= \frac{1}{\Delta t}\big(S_{i,i'}(x) - \Phi_{i'}\left(\Phi_i^{-1}(x)\right)\big) + \overline{b}_i \\ \widetilde{\sigma}_z &= \overline{\sigma}_{i'} \end{aligned} \tag{4.52}$$

Writing this equation in this form spreads the transition error out over the course of one timestep of length Δt. Thus, proving $\widetilde{\mathcal{L}}_z$ is close to $\bar{\mathcal{L}}_z$ reduces to showing that the transition error is sufficiently small after dividing by Δt (so that it can be combined in the drift term). Allowing the transition error to affect the drift forces us to have the drift $\widetilde{b}$ (and thus $\widetilde{\mathcal{L}}$) depend on the starting location z.

By the Whitney embedding theorem, $\mathcal{M}$ can be smoothly embedded into $\mathbb{R}^D$ for $D \geqslant 2d$. In $\mathbb{R}^D$, the LMDS mapping Φ_i reduces to principal component analysis, which is simply a projection onto the top d eigenvectors of the covariance matrix of the landmarks. Thus we can think of Φ_i as a matrix acting on vectors. To be consistent with the algorithm, vectors will be written as row vectors and the matrix Φ_i will act on the left.

Fix $k \in \Gamma$. Let $\Pi_k \in \mathbb{R}^{D\times d}$ denote the projection from $\mathcal{M}$ onto T_{y_k}, the tangent plane of $\mathcal{M}$ at y_k. Then Π_k is invertible on a ball of radius 2δ on T_{y_k} by assumption. Also since $\mathcal{M}$ is smooth, Taylor's theorem tells us that for some c_1 and all $x \in \mathcal{M}$ near y_k,

$$|x\Pi_k - x| \leqslant c_1 |x - y_k|^2 \tag{4.53}$$

Let $L_k = \{l_{k,i}\}$ denote the collection of landmarks associated with the neighbors of y_k, and μ_k denote their mean. The matrix Φ_k minimizes the squared error on the landmarks given by:

$$\sum_i |l_{k,i}\Phi_k - l_{k,i}|^2 \tag{4.54}$$

Inserting Π_k into equation (4.54) yields a bound of $c^2\delta^4$. The landmarks are well spread through the space by construction and the ellipticity condition. Thus, L will have covariance at least δ along each direction in the tangent plane. Then $\widetilde{L} = (L - \mu)\Pi = \{\widetilde{l}_i\}$ must have smallest singular value at least δ, and thus any vector v in the tangent plane can be written $v = a\widetilde{L}$ with $a = v\widetilde{L}^\dagger$. The bound on the singular values imply $|a| \leqslant \delta^{-1}|v|$. Then using Cauchy-Schwarz,

$$|v\Phi_k - v| \leqslant |v|c_1\delta \tag{4.55}$$

Since Φ_k, Π_k are projections, this implies that $||\Phi_k - \Pi_k||_2 \leqslant c_1\delta$. Let $j \in \Gamma$ such that $j \sim k$. By a Taylor expansion, $||\Pi_k - \Pi_j|| \leqslant c_2\delta$ for some c_2. Thus there exists a constant c_3 such that

$$||\Phi_k - \Phi_j||_2 \leqslant c_3\delta \tag{4.56}$$

The properties (4.55) (4.56) allow us to treat Φ_k like Π_k, the projection onto the tangent plane. Also since $||\Phi_k - \Pi_k||_2 \leqslant c_1\delta$ and $\delta \ll 1$, Φ_k will be invertible whenever Π_k is since Π_k has singular values equal to 1.

Next let $A = A_{k,j} = \{a_{k,i}\} \cup \{a_{j,i}\}$ be the collection of landmarks common to L_k and L_j. Let $\mu = \mu_{k,j}$ be the mean of A. Now we can write $T_{i,j}$ as

$$T_{i,j} = [(A - \mu)\Phi_i]^\dagger(A - \mu)\Phi_j \tag{4.57}$$

By definition of the pseudoinverse, $T_{i,j}$ minimizes

$$||(A - \mu)\Phi_i T - (A - \mu)\Phi_j||_2 \tag{4.58}$$

over all choices of $T \in \mathbb{R}^{d\times d}$. Choose T to be the restriction of Φ_j onto chart C_i. Then $T \in \mathbb{R}^d \times \mathbb{R}^d$ and

$$||(A - \mu)\Phi_i T - (A - \mu)\Phi_j||_2 = ||((A - \mu)\Phi_i - (A - \mu))(\Phi_j - \Phi_i)|| \leqslant c_1c_2\delta^3 \tag{4.59}$$

Since T is a possible choice for $T_{i,j}$,

$$||(A-\mu)\Phi_i T_{i,j} - (A-\mu)\Phi_j||_2 \leqslant c_1 c_3 \delta^3 \tag{4.60}$$

The matrix of landmarks $(A-\mu)$ spans the chart C_i, so there is a constant c_4 such that for any x in the chart C_i,

$$|S_{i,j}(x) - \Phi_j(\Phi_i^{-1}(x))| \leqslant c_4 \delta^3 \tag{4.61}$$

Using $\Delta t = \delta/\ln(1/\delta)$ and equation (4.61), the result follows. □

Proof of Lemma 7. Fix a starting location $z = (x, i) \in \mathcal{A}$. Then the process $\widetilde{X}_t^z$ is the solution of an SDE on chart i' with smooth coefficients. Thus, $\widehat{X}_t^z = W\big(\widetilde{X}_t^z\big)$ is also the solution of an SDE on chart i' with smooth coefficients.

$$d\widehat{X}_t = \widehat{b}(z, \widehat{X}_t^z)dt + \widehat{\sigma}(z, \widehat{X}_t^z)dB_t \tag{4.62}$$

$$\tag{4.63}$$

Using Itô's formula on $W(\widetilde{X})$,

$$\widehat{b}_j(z, \widehat{X}_t^z) = \sum_k \frac{\partial W_j}{\partial x_k}(\widetilde{X}_t^z)\widetilde{b}_k(z) + \frac{1}{2}\sum_k\sum_l \frac{\partial^2 W_j}{\partial x_k \partial x_l}(\widetilde{X}_t^z)(\widetilde{\sigma}\widetilde{\sigma}^T)_{k,l}(z) \tag{4.64}$$

$$\widehat{\sigma}_{j,l}(z, \widehat{X}_t^z) = \sum_k \frac{\partial W_j}{\partial x_k}\widetilde{\sigma}_{k,l}(z) \tag{4.65}$$

with $\widetilde{b}(z), \widetilde{\sigma}(z)$ defined as in Lemma 6. Note that since W is invertible, we could replace $\widetilde{X}_t^z$ with $W^{-1}(\widehat{X}_t^z)$ so that $\widehat{b}, \widehat{\sigma}$ can be thought of as a function of z and $\widehat{X}_t^z$. Direct computation shows that for some c_1, c_2,

$$\sum_k \left(\frac{\partial W_j}{\partial x_k}(\widetilde{X}_t)\right)^2 \leqslant c_1 \qquad , \qquad \sum_k\sum_l \left(\frac{\partial^2 W_j}{\partial x_k \partial x_l}(\widetilde{X}_t)\right)^2 \leqslant \frac{c_2}{\delta^2}$$

Let E_t denote the set $\left\{t : |\widetilde{X}_t^z| > \dfrac{3\delta}{2}\right\}$. By definition of W, $\widehat{b}$ and $\widetilde{b}$ agree on E_t^c. $|\widetilde{X}_0^z| \leqslant \delta + O(\delta^2)$ so the Brownian motion must push the process at least $O(\delta)$ in time t. In other words there are constants c_3, c_4 such that,

$$\mathbb{E}_z[\mathbb{1}_{E_t}] \leqslant \mathbb{P}[|B_t| > c_3\delta] \leqslant \exp(-c_4\delta^2/4t) \tag{4.66}$$

Next we can bound the effect of the boundary function W on the drift and diffusion terms:

$$\mathbb{E}\left[\int_0^{\Delta t}\left|\widehat{b}-\widetilde{b}\right|^2\left(z,\widehat{X}_t^z\right)dt\right]=\mathbb{E}\left[\int_0^{\Delta t}\mathbb{1}_{E_t}\left|\widehat{b}-\widetilde{b}\right|^2\left(z,\widehat{X}_t^z\right)dt\right]\leqslant\frac{c_5\Delta t}{\delta^2}\exp\left(-c_4\frac{\delta^2}{\Delta t}\right)$$

for some new constant c_5. By equations (4.65), (4.66) and the fact that $\widehat{\sigma}$ agrees with $\widetilde{\sigma}$ on E_t^c,

$$\mathbb{E}\left[\int_0^{\Delta t}\left|\left|\widehat{\sigma}\widehat{\sigma}^T-\widetilde{\sigma}\widetilde{\sigma}^T\right|\right|_F^2\left(z,\widehat{X}_t^z\right)dt\right]\leqslant\Delta t c_6\exp\left(-c_4\frac{\delta^2}{\Delta t}\right)\tag{4.67}$$

The result follows for $\Delta t=\delta/\ln(1/\delta)$. □

5

Examples

5.1 Simulator Comparison

In order to see how well the Atlas is doing we will need to have a criterion for comparing simulators. Since we are interested in the behavior of the system over multiple timescales, we will simulate 10,000 paths from each simulator and record the positions at times $\{t_k := 2^k\}$. The smallest time scale will be at the size of one step of the original simulator and the largest time scale will be at some time T (example dependent) at which point systems have reached equilibrium.

In order to understand motivation for how to compare simulators, we start with a 1-d example. For a fixed $t_j \leqslant T$, we can bin samples into equal spaced bins, and compare them. Next we would like to compare the probabilities of landing in each bin as in figure 5.1 by overlaying them. We can next vary k (and thus t_k) to obtain overlaid bar graphs for multiple time scales as in figure 5.2. We see that the real quantity of interest is the difference between these two histograms, and we will sum the absolute values of their difference to approximate the L^1 distance between the measures.

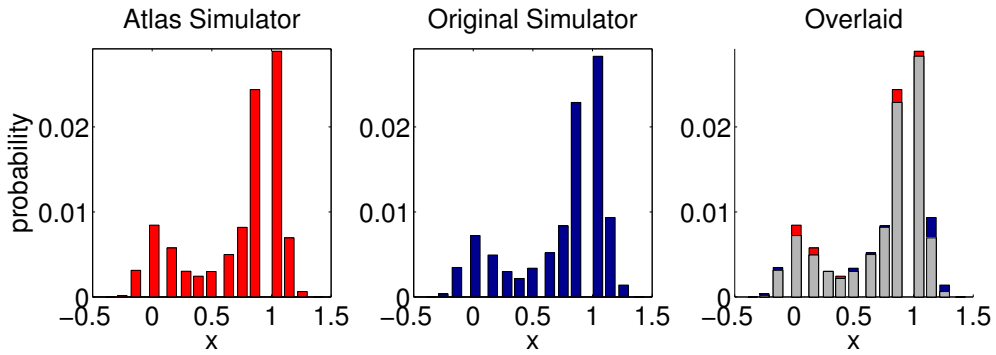

FIGURE 5.1: Comparing distributions obtained from two simulators at time $T = 0.2$ (orginal and Atlas) in example 5.2.2.

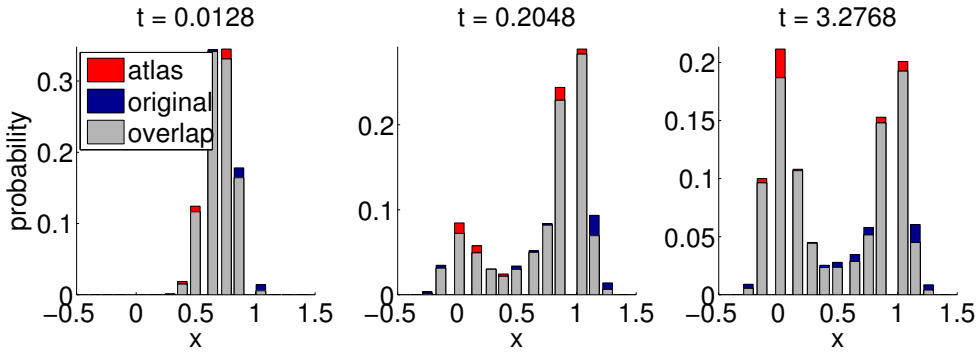

FIGURE 5.2: Comparing overlaid distributions obtained from two simulators at multiscale times 2^k (orginal and Atlas) in example 5.2.2.

Our next goal is to generalize this to high dimensional spaces. Here the bins we will use can be given naturally by the Atlas we construct. Instead of using a "hard" binning procedure by assigning each point to the closest bin, we will assign smooth weights to the nearest neighbors. This smooth binning procedure will help to wash out the small scale errors we make, so that we can measure the large scale errors.

The first step is to explain the smooth map which takes a distribution ν on $\{x_i\}_{i=1}^n$ to a distribution μ on a set $\{y_j\}$. We think of $\{y_j\}$ as a coarser binning of the

distribution ν on $\{x_i\}$. First assign weights $w_{i,j}$ to each (x_i, y_j) pair given by:

$$w_{ij} = \begin{cases} \exp\left(\frac{-|x_i - y_j|^2}{\delta^2}\right) & |y_i - x_j| < 2\delta \\ \\ 0 & \text{otherwise} \end{cases}$$

Then we normalize the weights so that they sum to 1 when summed over j.

$$\mu_j = \sum_i \nu_i \frac{w_{i,j}}{\sum_j w_{i,j}} \tag{5.1}$$

Fix a time slice t_k, then assign equal weights $\nu_i = 1/n$ to the set of samples $\{x_i\}_{i=1}^n$ given by a the original simulator, and map them to a distribution μ on the net Γ using (5.1),(5.1) and the distance function in the ambient space. Next we will assign equal weights to the samples $\{\widehat{x}_i\}_{i=1}^n$ and map them to weights $\widehat{\mu}$ on the net Γ using the euclidean chart distances.

Once we have μ, $\widehat{\mu}$, we could compare them directly. However, we know that the Atlas makes errors on this spatial scale, and so we would like to smooth these distributions out to a coarser net with $\delta_c \geqslant \delta$. This will also allow us to compare simulators with varying δ while keeping the number of bins fixed. For each example, we will fix a coarse grained δ_c equal to the largest δ used for that example, and obtain a net $\{z_l\}$. Then we can push μ, $\widehat{\mu}$ to distributions p, $\widehat{p}$ on $\{z_l\}$ again using (5.1),(5.1) and the distance function in the ambient space. This gives us two probability distributions, one for each simulator, at time t_k on the coarse net.

Given a single initial condition, we will calculate the L^1 distance between p and $\widehat{p}$ for each time slice t_k. Then we will repeat this procedure for 10 fixed initial conditions (randomly chosen) to compare the transition densities over a wide range of time scales and initial conditions. In examples where only one Atlas is used, we plot one thin colored line for each initial condition, then a thick line representing the mean $\pm$ one standard deviation (see figures 5.7, 5.11, 5.14, 5.19). In examples where

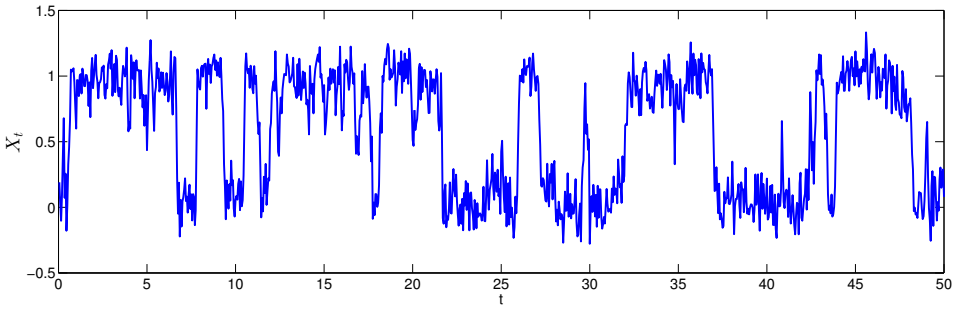

FIGURE 5.3: Sample trajectory of X_t for the two well example 5.2.1.

we compare multiple Atlas's, we just plot the thick line representing the mean $\pm$ one standard deviation (see figures 5.5, 5.10, 5.13, 5.18).

5.2 One dimensional Example

5.2.1 Smooth Potential

The first example presented is a simple one dimensional two-well example. We will use the potential

$$U_1(x) = 16x^2(x-1)^2$$

and use a simulator which approximates

$$dX_t = -\nabla U_1(X_t)dt + dB_t$$

using an Euler-Maruyama scheme which takes timesteps of size 0.005. A sample path of this system is shown in figure 5.3. The initial point set we use to generate the δ-net is linearly spaced points with spacing 0.01. It is important to note that the distribution of the initial point set does not play an important role in the resulting Atlas. The Atlas algorithm performs equally well on any initial point set that has no holes of size order δ. We subsample this initial point set to obtain a δ-net with δ parameter 0.1 using the brute force method described in section 3.1.

Once we run the Atlas algorithm in this case, it is simple to map estimated drift vectors from the chart coordinates back to the original space. In general for an

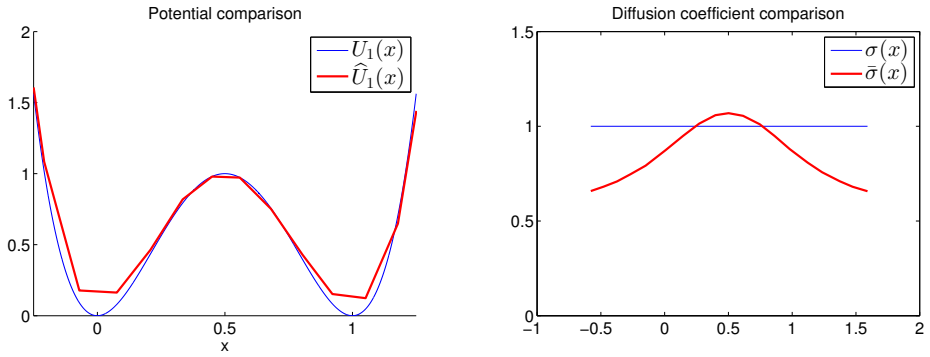

FIGURE 5.4: Left: original potential U (shown in blue) and effective potential of the Atlas $\widehat{U}$ (shown in red). Right: comparing original diffusion coefficient (blue) with that of the Atlas(red) with $\delta = 0.1$ in example 5.2.1.

arbitrary metric space this is a hard problem, but in 1-d we need only multiply by ± 1 to undo MDS. In 1-d, the estimated drift vectors can easily be integrated to obtain an effective potential $\widehat{U}$ for the system. We can also bring back the diffusion coefficients and see how they compare to the truth. Inverting MDS and comparing the coefficients we obtain with the true coefficients of the underlying system is a procedure we will only be able to do for this 1-d system, but it gives interesting insight into the working of the homogenizing nature of the Atlas. See Figure 5.4 showing the resulting comparisons between drift and diffusion coefficients.

Next we generate four nets (and four Atlas simulators) with δ values 0.05, 0.10, 0.15 and 0.20 by subsampling from a fine mesh. In each example we have used $p = 10,000$ simulations per net point, and $t_0 = \delta^2$. The number of landmarks is irrelevant because as long as $m \geqslant 1$, there will be enough landmarks to exactly recover the local space. When simulating, we set the simulation time step $\Delta t = \delta^2/5$. Then for each of 10 randomly chosen staring locations, we run $10,000$ long paths up to time $T = 50$. Using the simulator comparison method from section 5.1, we obtain figure 5.5. As we expect from theorem 2, the long time error is decreasing as δ decreases. Figure 5.5 shows that the transition kernels are close for all time scales,

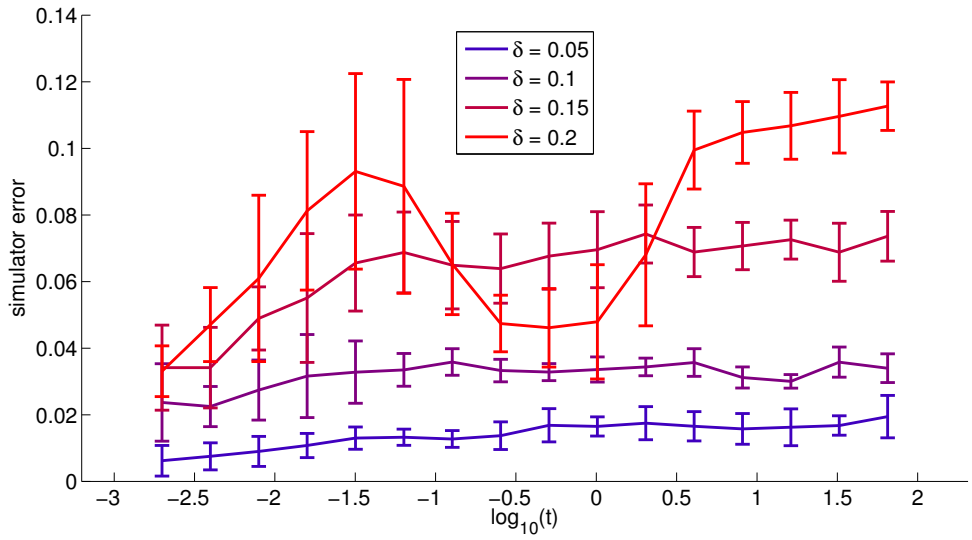

FIGURE 5.5: Simulator comparison for example 5.2.1. Each line represents the average simulator error for a single net of the specified δ value.

which is a stronger experimental result than given by theorem 2. Theorem 2 only tells us that the stationary distributions are $O(\delta \log(1/\delta))$ far from each other.

5.2.2 Rough Potential

In order to make a more interesting example, we add high frequency ridges to the potential well to emulate microscale interactions. This example is a case where it is of interest to approximate a homogenized system which behaves like the original system above a certain temporal/spatial scale. Define

$$V_1(x) = U_1(x) + \frac{1}{6}\cos(100\pi x) \tag{5.2}$$

where $U_1(x)$ is defined in example 5.2.1. For our initial point set, we could again use evenly spaced grid points as in example 5.2.1. Since one might wonder if this is a "fair" input we run each grid point through the simulator for a small time $t = 0.01$ to obtain our initial point set. As long as these points have no holes of size order δ, the Atlas will return a robust result with high probability.

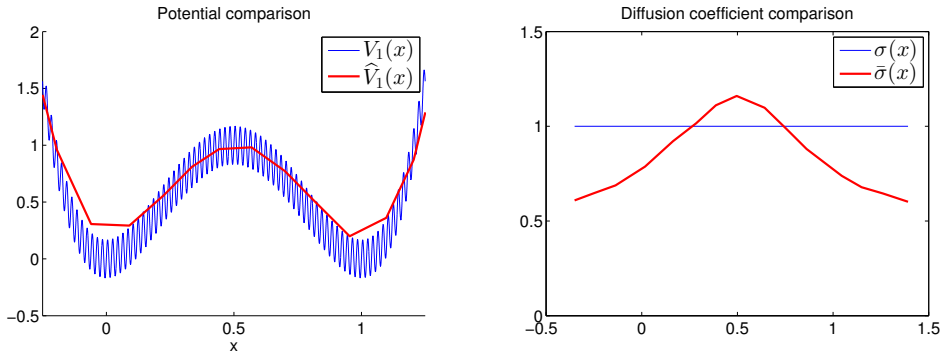

FIGURE 5.6: Left: original potential U (shown in blue) and effective potential of the Atlas $\widehat{U}$ (shown in red). Right: comparing original diffusion coefficient (blue) with that of the Atlas(red) with $\delta = 0.1$ in example 5.2.2.

Even though the new potential well is infinitely differentiable, the Lipschitz constant of the drift in this example is 625. In order to accurately simulate Brownian motion in this potential well, we decrease the time step to 0.00005. These microscale interactions are determining our timestep, and thus becoming a bottleneck for running long time simulations.

If we were to apply theorem 2 directly to this example, it will guarantee a relatively useless error bound on the stationary distribution (since the error bound depends on the Lipschitz constant). Instead, the way we think of theorem 2 applying to this problem is that there is a time scale t_0 at which the system with potential well V_1 behaves like a homogenized version with smooth potential and small Lipschitz constant. Multiscale systems of this form have been studied (see Pavliotis and Stuart (2007) and references therein), and it is known that such systems behave like an SDE with smooth parameters at a large scale. If we only observe samples at time t_0, then we can pretend our samples come from the homogenized system rather than the microscale simulator.

In this example, we learn the Atlas using the parameters $\delta = 0.1$, $t_0 = 2\delta^2 = 0.02$, $p = 10,000$, and $\Delta t = t_0/5$. Again we can map the drift and diffusion back to

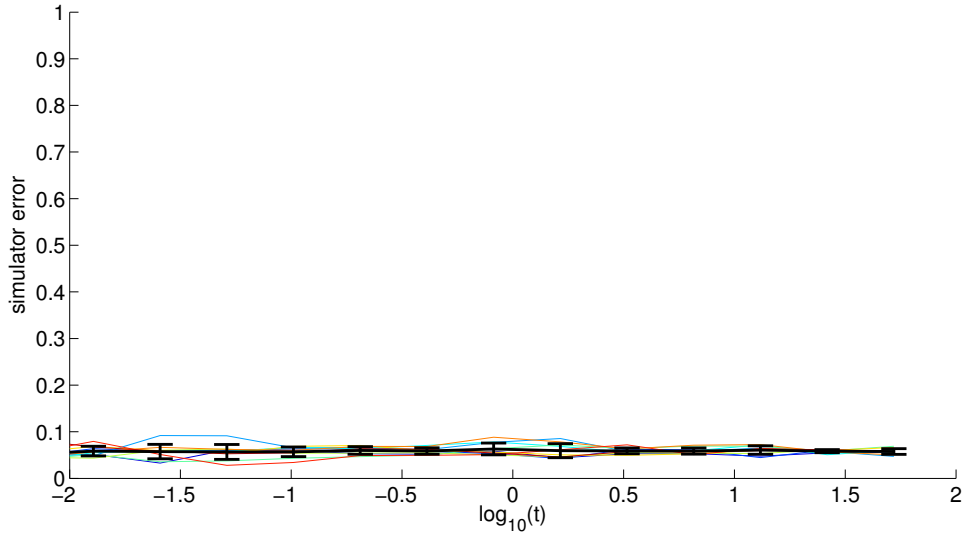

FIGURE 5.7: Comparing true simulator with the Atlas with $\delta = 0.1$ on example 5.2.2.

the original space and compare with the true simulator. Figure 5.6 shows that the resulting drift is a homogenized version of the original system. The time scale the local simulator uses is 100 times larger than that of the original system. This will result in long simulations being about 100 times faster than using the original simulator.

Next we have run 10,000 long paths from the Atlas with $\delta = 0.1$ shown above in figure 5.6. Figure 5.7 shows that again the distribution of paths is similar over multiple timescales, indicating that transition rates are preserved between states.

In this example we only show results for $\delta = 0.1$ because that is the spatial scale where it makes sense to homogenize. For smaller values of δ, the Atlas becomes less stable as the estimated drift becomes less smooth. For larger values of δ, the macroscale features of the system begin to wash out, and the two wells merge into one.

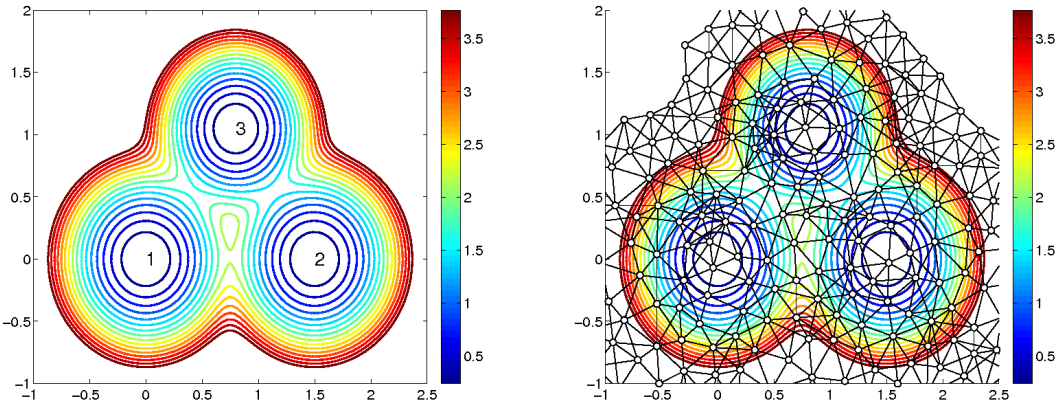

FIGURE 5.8: Left: Potential for three well example. Right: $\delta = 0.2$ net overlaid. Circles represent net points, black lines represent connections between net points.

5.3 Two Dimensional Example

5.3.1 Smooth Potential

In this example, consider the 2-d potential well $U_2(x)$ shown below.

$$U_2(x) = -\ln\left(\exp\left(\frac{-||x-p_1||^2}{c_1}\right) + \exp\left(\frac{-||x-p_2||^2}{c_2}\right) + \exp\left(\frac{-||x-p_3||^2}{c_3}\right)\right)$$

$$p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad p_2 = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}, \quad p_3 = \begin{bmatrix} 0.8 \\ 1.05 \end{bmatrix}, \quad c = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{6}\right]$$

The stationary distribution is a mixture of Gaussians given by $\exp(-U_2/2)$. There are three clearly defined minima near p_1, p_2, p_3. The parameters of the problem were chosen in such a way that the transition regions between wells lie on different level sets of the potential (see figure 5.8).

We will see a simulator which approximates the process X_t given by

$$dX_t = -\nabla U_2(X_t)dt + dB_t. \tag{5.3}$$

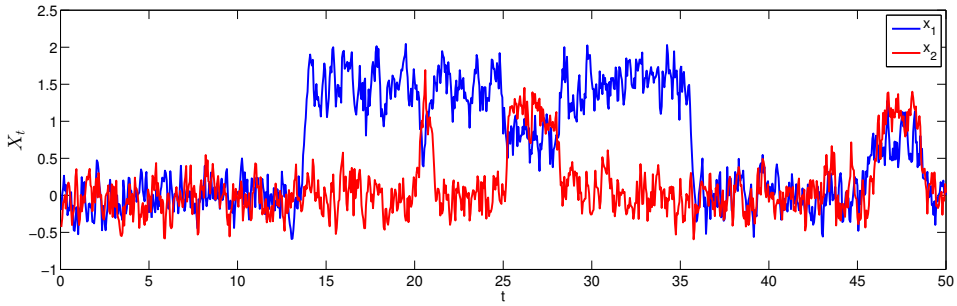

FIGURE 5.9: Sample trajectory for three well example

Figure 5.9 shows a sample trajectory of the process X_t using a simple Euler-Maruyama scheme with timestep 0.005. This is the simulator given to the Atlas algorithm. The initial point set we use is a grid spaced by 0.01, discarding points with $U_2(x) \geqslant 10$. Figure 5.8 shows an example net for $\delta = 0.2$.

When generating the Atlas in this example, we use $p = 10,000$, $t_0 = \delta^2$, $\Delta t = t_0/5$. Again the number of landmarks does not matter since LMDS will return the exact result (up to machine precision) every time. Next for each of 10 randomly chosen starting locations we run 10,000 paths from each simulator. Then we compare them using a common coarse grained net with $\delta_c = 0.2$ as in section 5.1. The output is shown in figure 5.10. Again we notice that the errors are small for all times, including the range of timescales where transitions occur. This means we must be accurately capturing transition rates from each of the 10 randomly chosen starting locations.

5.3.2 Rough Potential

In the next example we take $U_2(x)$ and add a fast oscillating component to simulate small scale interactions as in example 5.2.2. The new potential well is

$$V_2(x) = U_2(x) + \frac{1}{6}\cos(100\pi x_1) + \frac{1}{6}\cos(100\pi x_2). \tag{5.4}$$

And again see a simulator which approximates the process X_t.

$$dX_t = -\nabla V_2(X_t)dt + dB_t \tag{5.5}$$

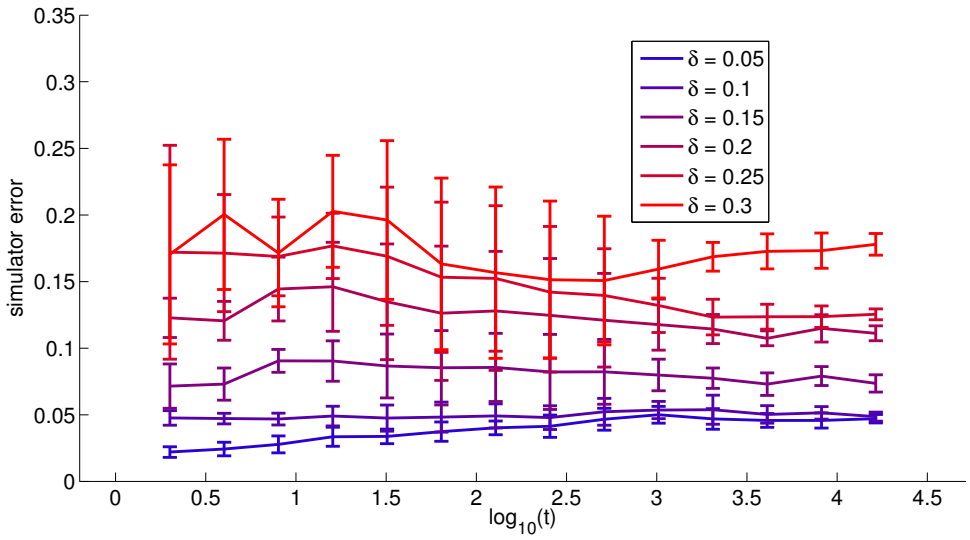

FIGURE 5.10: Comparison of Atlas's with original simulator in the smooth three well potential from example 5.3.1.

As a result of the high frequency oscillations, the the timesteps will be of size 0.00005. This example will show that our algorithm is robust to fast oscillations of the potential even in a more complicated system. In this example we will again avoid using evenly spaced points as input, and run the grid points through the simulator for a short time $t = 0.01$. These are samples we could obtain from running the original simulator for a long time, or using some kind of fast exploration technique. Again, the distribution of this point set is irrelevant as long as there are no holes of size δ.

For this system we will use $\delta = 0.2$ which will return δ nets with ≈ 230 net points. We will again use use $p = 10,000$, $t_0 = \delta^2$, $\Delta t = t_0/5$ for consistency, even though p could be chosen smaller (since δ is larger). Again, the timestep of the Atlas is $\Delta t = 0.004$ which is over 100 times larger than the timesteps of the original simulator, and thus the Atlas runs about 100 times faster. For the simulator comparison with this example see figure 5.11.

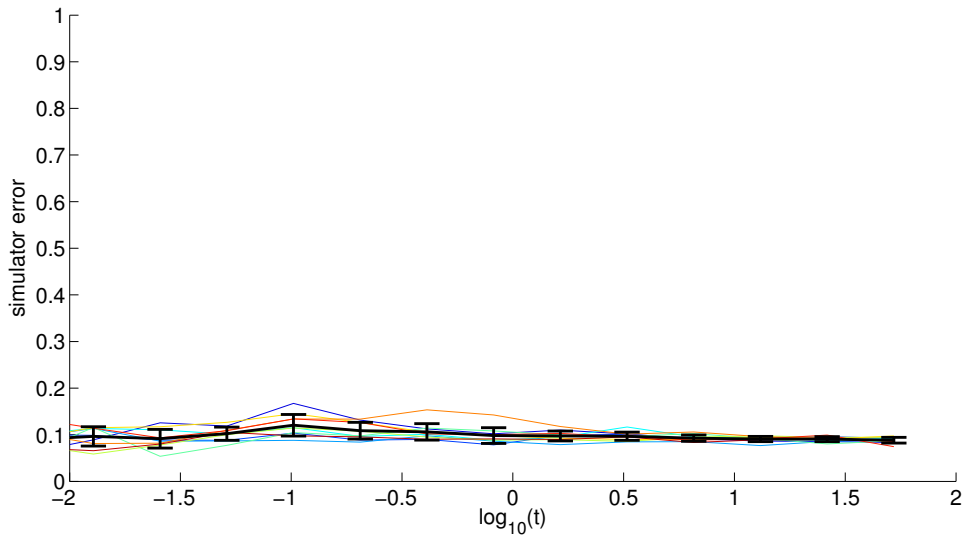

FIGURE 5.11: Comparison of original simulator with the Atlas ($\delta = 0.2$) in the rough three well potential from example 5.3.2.

5.4 Random Walk on Images

Next we will embed the two dimensional three well examples from sections 5.3.1 and 5.3.2 into $D = 12,500$ dimensions. The high dimensional embedding is given by the following algorithm given a two dimensional point x:

1. Generate a mesh $\{z_j\}$ on $[-1.5, 3.5] \times [-1.5, 2.5]$ with evenly spaced grid points and spacing 0.04.

2. The output vector v at position j is 1 if $|z_j - x| < 1/2$ and 0 otherwise.

See figure 5.12 for an example image generated by this algorithm run on the point $(0, 0)$. The natural distance to use in this space is the hamming distance, which counts the number of different entries. It induces a norm, which we call $||v||_1$ since this is the same as the 1-norm of the vector on $\mathbb{R}^D$. Given a binary vector v, we can write the "inverse" $\widetilde{x}$

$$\widetilde{x} = ||v||_1^{-1} \sum_j v_j z_j \tag{5.6}$$

This just averages the positions of the pixels $\{z_j\}$, which should roughly return the center of the circle in the image. Any two dimensional simulator now can be mapped to a simulator on $\mathbb{R}^D$ in the following way:

1. Given input $v \in \mathbb{R}^D$ and a time t_0, calculate the two dimensional point $\widetilde{x}$ from the approximate inverse mapping.

2. Run the 2-d simulator for time t_0 with initial condition $x_0 = \widetilde{x}$.

3. Take the output of the simulator, X_{t_0} and map it to $\mathbb{R}^D$ with the high dimensional embedding.

Next, we rescale the distance function by the constant $(0.04)^2/2$ so that the new norm is locally equivalent to the original distance. In so doing, we can continue using values of δ that made sense to us in the original space. This high dimensional mapping is nontrivial, and all the possible vectors v we could see span the entire $12,500$ dimensional space. The space can be locally approximated by a 2-d plane for a ball of radius $r < 1/2$, and so we expect the Atlas to find the appropriate local spaces to estimate the dynamics.

5.4.1 Smooth Potential

First we will apply the high dimensional mapping to the simulator with smooth potential well U_2 from example 5.3.1. Next we start with a set of points in $\mathbb{R}^D$ which cover the known state space (the same covering set from before only mapped to $\mathbb{R}^D$). The Atlas algorithm is given the rescaled hamming distance function for computing distances between vectors, and it is given the simulator which takes points in $\mathbb{R}^D$ and

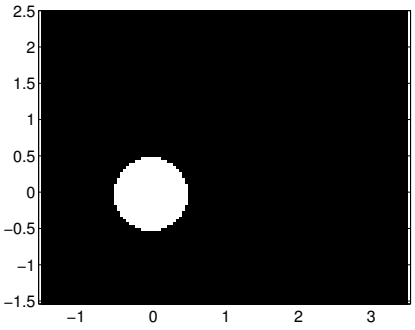

FIGURE 5.12: Circle image corresponding to the point [0,0].

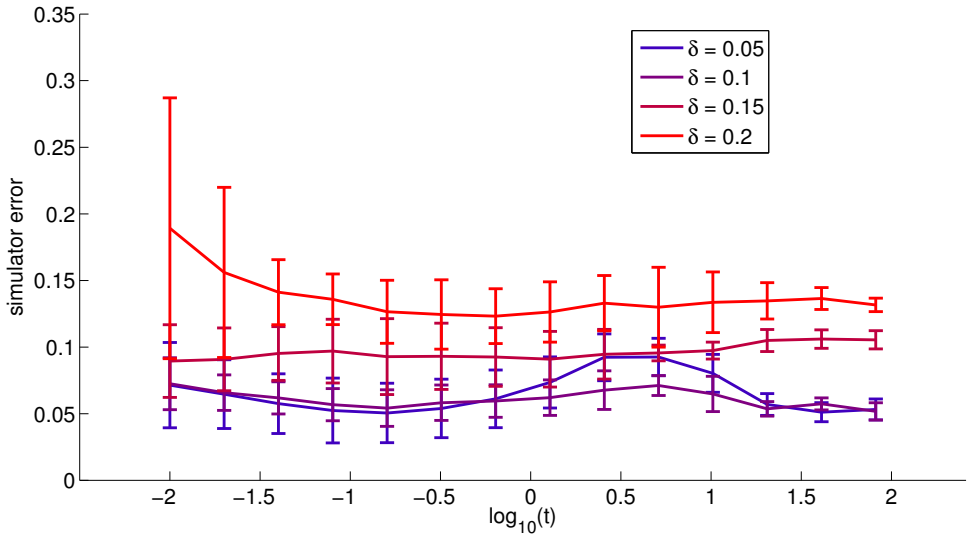

FIGURE 5.13: Comparison of Atlas's with original simulator for example 5.4.1.

a time t_0 and returns points in $\mathbb{R}^D$. Because distances are now $12,500$ times more expensive to compute, for this example we set $p = 1000$ and $m = 20$ landmarks per point. Again keep $t_0 = \delta^2$ and $\Delta t_0 = t/5$.

After constructing multiple Atlas's for varying values of δ, we find that the distributions are well approximating the original given simulator. See figure 5.13 for details. The small number of samples, along with the width of the pixels limits the accuracy for small values of δ. In fact we can see that $\delta = 0.05$ returns a simulator

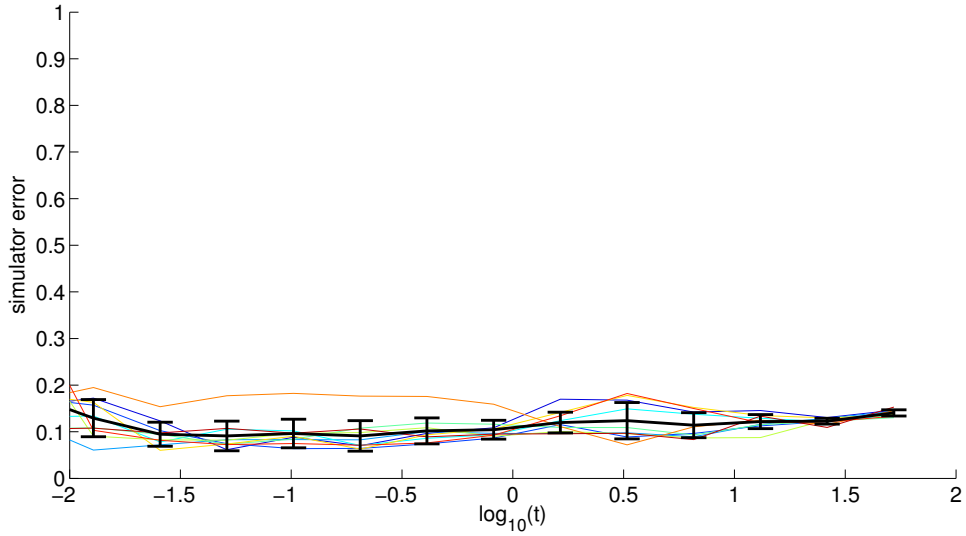

FIGURE 5.14: Comparison of the Atlas $\delta = 0.2$ with original simulator for example 5.4.2.

which is worse than $\delta = 0.1$.

5.4.2 Rough Potential

In the next example of this paper, we will apply the high dimensional transformation to the rough potential well V_2 from example 5.3.2. Again, we give the algorithm the same set of initial points from example 5.3.2 mapped to $\mathbb{R}^D$ along with the simulator using V_2 embedded in high dimensions. In this example we use $\delta = 0.2$, $p = 2000$, $m = 40$, $t_0 = \delta^2$ and $\Delta t = t_0/5$. Again the simulation timescale of the local simulator is 100 times larger than that of the original simulator. The Atlas has a running time which depends only on the local dimensionality of the system, and so the ambient dimension only enters in the construction phase. After simulating $10,000$ paths for each of 10 different initial conditions, we can test the simulator error (see figure 5.14). Because running the original simulator is very expensive for this system, we used the same original simulator samples (mapped to $\mathbb{R}^D$) for comparison as in figure 5.11.

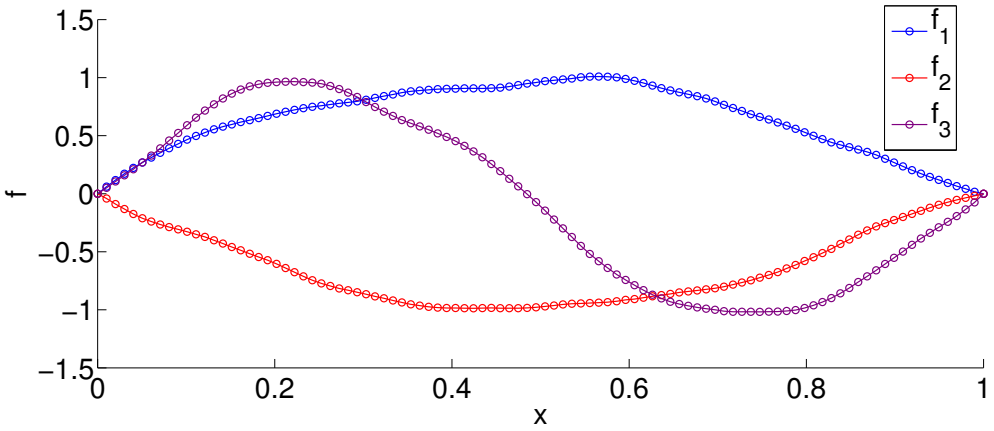

FIGURE 5.15: Three typical outputs of the simulator from section 5.5.

5.5 Random walk on Functions

In this last example, we are given a dynamical system in the form of a random walk on functions on $[0, 1]$ with endpoints fixed at zero. These functions are represented by values on a grid of 100 evenly spaced points (including the ends). Typical functions seen as output from the simulator are shown in figure 5.15. The distance we will use is euclidean distance in $\mathbb{R}^{100}$, rescaled by $1/100$ to approximate the L^2 distance on functions. A single step of the simulator is done by adding a Brownian path fixed at the endpoints, then smoothing the result and renormalizing. The pseudocode is shown in figure 5.16.

This behavior of this system in characterized by large dwelling times near the smoothest functions (f_1 and f_2 from figure 5.15) with rare transitions ($10^3 - 10^4$ steps) across functions like that of f_3 in figure 5.15. The three constraints $f(0) = 0, f(1) = 0, ||f|| = ||f_0||$ force the functions to live on S^{97}, a 97 dimensional sphere with radius $||f_0||$. Although we expect these functions to lie near a low dimensional submanifold $\mathcal{M} \subset S^{97}$ because of the smoothing step, a single step of the simulator could take us anywhere on S^{97}; this means the outputs of our simulator are never exactly on $\mathcal{M}$. This is an important aspect of this example, as real world data

Function Simulator

function $f = \mathcal{S}(f)$

% *simulate Brownian bridge*
$W = \text{cumsum}(\text{randn}(1{,}100))$
$W = W - W(1)$
$W = W - x * W(100)$

% *Add bridge to f, smooth and renormalize*
$f = f + (1/100) * W$
$f = \text{smooth}(f)$
$f = f * (f_{\text{norm}}/\text{norm}(f))$

FIGURE 5.16: Pseudocode for a single step of the simulator used in example 5.5. $f_{\text{norm}} = ||\sin(\pi x)|| = 0.0704$. The function smooth is MATLAB's default smoothing algorithm.

typically will have small noise in the ambient space.

One can think of this simulator as a discretization of the SDE on S^{97}

$$dX_t = F(X_t)dt + \sigma(X_t)dW_t \tag{5.7}$$

For an appropriate choice of F, σ. One can also think of this as a discretization scheme for a stochastic partial differential equation of the form

$$\frac{\partial}{\partial t} f_t = \frac{\partial^2}{\partial x^2} f_t + b(f_t) + \sum_{j=1}^{\infty} g_j(f_t) dW_t^j \tag{5.8}$$

for an appropriate choice of drift b and orthogonal functions $\{g_j\}$. One can think of (5.8) as an infinite dimensional analogue to (5.7) with each coordinate $X_t^j = \langle f_t, g_j \rangle$ being driven by a one dimensional brownian motion.

In order to generate the Atlas, first we must generate an initial sampling of the space. In order to do this, we start with $50,000$ renormalized gaussian vectors, the uniform distribution on S^{97}. Next we want to "heal" these samples by running them through the simulator. One can see with some observation that 250 steps is large enough that the noise is killed; samples with 250 steps of healing are similar to those

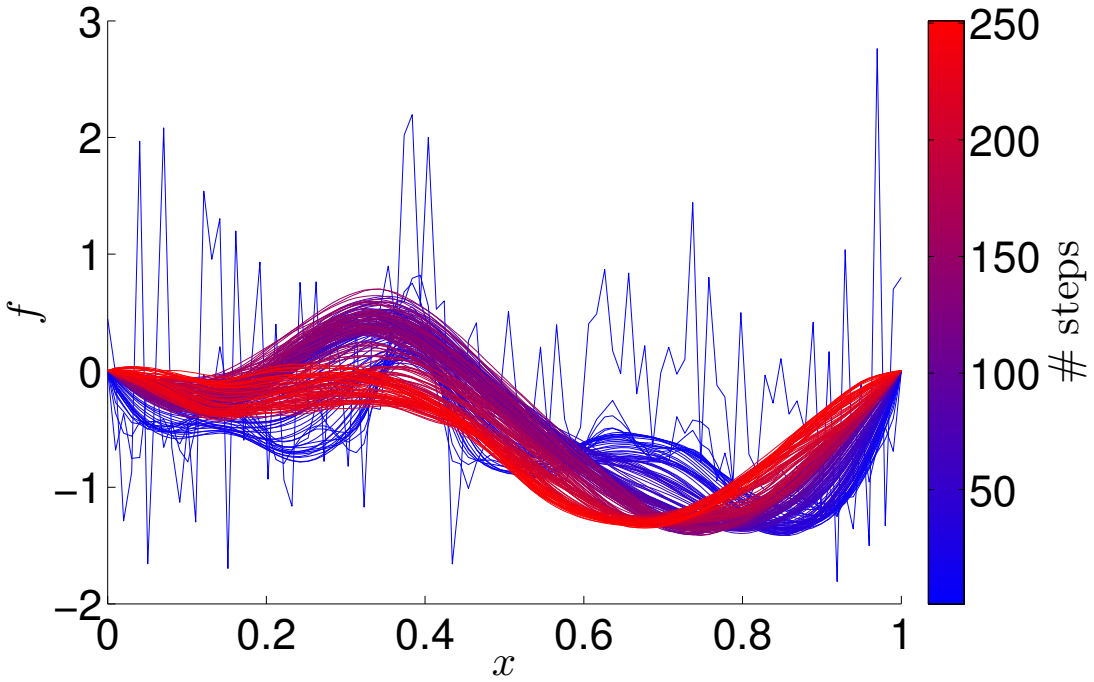

FIGURE 5.17: Example "healing" starting from random normal initial conditions. Color indicates the number of steps taken through the simulator.

with 500 steps of healing. See figure 5.17 to see an example simulation starting at random normal initial conditions, run for 250 steps.

Next we wish to select parameters δ, t_0. We expect that the system may be homogenized at a time scale of $t_0 = 250$ steps for the following reasons: t_0 is an order of magnitude below the scale of major events of the system, t_0 is an order of magnitude above the scale of the noise. The parameter δ is closely tied to the choice of t_0. We measure the average distance moved by paths of length t_0 starting from our healed samples to be $0.3 = \delta$. Next we choose the minor parameters p, m, d. In these experiments, we use $p = 5000$ and $m = 40$. As discussed in section 3.2, we can choose d based upon the singular values obtained through LMDS. Choosing a cutoff of $(\delta/4)^2$ for the eigenvalues yeilds $d = 3$ over 99% of the time. Using $d = 3$ and comparing with the original simulator in the usual way yeilds figure 5.18.

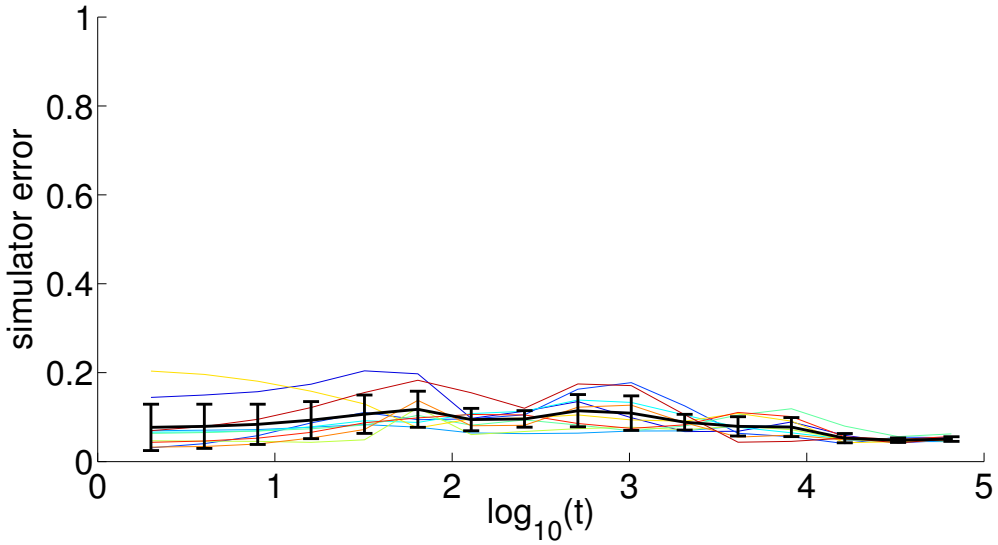

FIGURE 5.18: Comparison of the Atlas with original simulator for example 5.5.

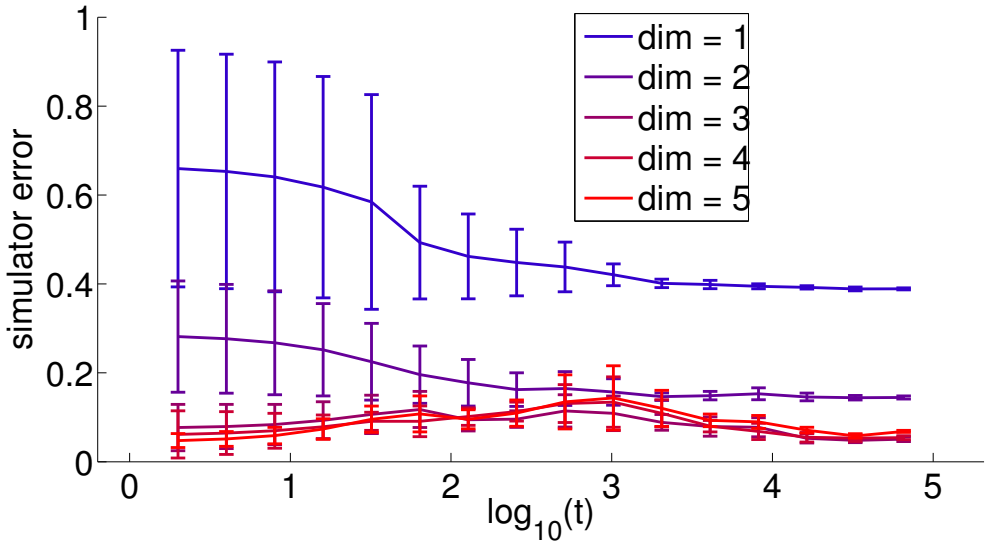

FIGURE 5.19: Comparison for example 5.5 varying d, the dimension of the Atlas.

In general, it is better to overestimate d than underestimate; underestimating d may lose important degrees of freedom causing failure, while overestimating d will only affect the computational cost mildly. In fact the algorithm is robust to the choice of d, provided d is large enough to capture the important degrees of freedom. See figure 5.19 to see results for varying values of the choice of d.

The Atlas constructed for this example again captures the important aspects of the original simulator. The Atlas is again faster in this example due to two factors: decreased dimensionality and increased timestep. The dimensionality of the Atlas is 3 as compared to the original 100, and the timestep of the simulator is equivalent to 50 of the original steps (250/5 since $\Delta t = t_0/5$).

6

Extensions and Discussion

There are many open problems related to this work, some of which we mention here.

Theorem 2 reveals that the local learning algorithm works well on compact SDEs with Lipschitz drift and diffusion. We consider only bounded domains in the proof to make thing simpler, although the same framework can be applied to the unbounded case with tight transition density. In this case, one has to worry about parts of the space which are unexplored, but seldomly reached. Indeed we see that some of our examples have unbounded state spaces, and the algorithm performs as desired.

The framework we introduced may be generalized to richer families of local simulators, enabling the approximation of larger classes of stochastic systems. Proving large time accuracy may be difficult for such systems, so it is an open problem how much one is allowed to change these local simulators. Many molecular dynamics (MD) systems remember the velocity of atoms and so do not follow an SDE of the form (4.6) which is memoryless. A subject of ongoing research is to use more complex models locally to be able to capture dynamics of typical MD systems.

Another subject of future work is efficient computation of the function G, which is the inverse MDS mapping. In some cases, such as when ρ is the root mean square

distance (RMSD), it is possible to create an inverse mapping which has error of order δ^2 instead of order δ.

Using the Atlas as a basis for generating samples from the stationary distribution is useful for quickly computing diffusion maps for these systems. A subject of interest is to understand how the errors made by the Atlas propagate through diffusion maps. How similar do diffusion maps look generated by samples from the Atlas as compared to diffusion maps generated directly from the original simulator?

In some problems, choosing δ and t_0 is difficult. Another subject of ongoing research is a robust way of choosing these parameters based on short simulations. For simplicity in this paper we have assumed that δ and t_0 is constant for each $k \in \Gamma$, but it is possible to have these parameters depend on the location y_k (and perhaps statistics of short sample paths).

Last but not least, this construction as described here still requires a large number of steps to sample rare events and reach stationarity, i.e. it does not address the problem of accelerating the sampling of rare events or overcoming energy barriers. In many important applications, e.g. molecular dynamics, such barriers force the simulations to be extremely long (e.g. $10^{12} - 10^{14}$ time-steps is common). The point of this work is to produce a simulator that is much faster (in real world time) than the original fine scale simulator. It is important to note that any of the many techniques developed over the years to attempt to overcome this problem may be used in conjunction with our construction, i.e. it can be run on our Atlas, instead of the original expensive fine scale simulator. This yields a double gain in simulation speed, combining the gains of a faster simulator with those of an importance sampler that efficiently samples rare events.

Bibliography

A. J. Majda, I. T. and Vanden-Eijnden, E. (2001), "A Mathematical Framework for Stochastic Climate Models," *Comm. Pure App. Math.*, 8, 891–974.

Allard, W. K., Chen, G., and Maggioni, M. (2012), "Multi-scale geometric methods for data sets II: Geometric Multi-Resolution Analysis," *Applied and Computational Harmonic Analysis*, 32, 435–462, (submitted:5/2011).

Antoulas, A. C., Sorensen, D. C., and Gugercin, S. (2001), "A survey of model reduction methods for large-scale systems," *Contemporary Mathematics*, 280, 193–219.

Bensoussan, A., Lions, J., and Papanicolaou, G. (2011), *Asymptotic analysis for periodic structures*, vol. 374, American Mathematical Soc.

Beygelzimer, A., Kakade, S., and Langford, J. (2006), "Cover trees for nearest neighbor," in *Proceedings of the 23rd international conference on Machine learning*, pp. 97–104, ACM.

Billingsley, P. (1999), *Convergence of Probability Measures*, Wiley-Interscience Publication.

Bowman, G., Beauchamp, K., Boxer, G., and Pande, V. (2009), "Progress and challenges in the automated construction of Markov state models for full protein systems," *The Journal of chemical physics*, 131, 124101.

Brand, M. (2002), "Charting a manifold," in *Advances in neural information processing systems*, pp. 961–968.

Coifman, R. and Lafon, S. (2004), "Diffusion maps," *submitted to Applied and Computational Harmonic Analysis*.

Coifman, R., Kevrekidis, I., Lafon, S., Maggioni, M., and Nadler, B. (2008), "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *Multiscale Model Sim*, 7, 842–864.

Crosskey, M. M., Nixon, A. T., Schick, L. M., and Kovačič, G. (2011), "Invisibility cloaking via non-smooth transformation optics and ray tracing," *Physics Letters A*, 375, 1903–1911.

Faradjian, A. and Elber, R. (2004), "Computing time scales from reaction coordinates by milestoning," *The Journal of Chemical Physics*, 120.

G. Chen, A. Little, M. M. (2013), "Multi-Resolution Geometric Analysis for Data in High Dimensions," in *Excursions in Harmonic Analysis, Volume 1*, eds. T. D. Andrews, R. Balan, J. J. Benedetto, W. Czaja, and K. A. Okoudjou, Applied and Numerical Harmonic Analysis, pp. 259–285, Birkhuser Boston.

G. Pavliotis, A. S. (2008), *Multiscale Methods*, Springer.

Gear, C., Kaper, T., Kevrekidis, I., and Zagaris, A. (2005), "Projecting to a Slow Manifold: Singularly Perturbed Systems and Legacy Codes," *SIAM Journal on Applied Dynamical Systems*, 4, 711–732.

Gilbert, A. (1998), "A comparison of multiresolution and classical one-dimensional homogenization schemes," *Applied and Computational Harmonic Analysis*, 5, 1–35.

Hornung, U. (1997), *Homogenization and Porous Media*, Springer.

Hsu, E. (2002), *Stochastic Analysis on Manifolds*, American Mathematical Society.

Huisinga, W., Schütte, C., and Stuart, A. (2003), "Extracting Macroscopic Stochastic Dynamics," *Comm. Pure Appl. Math.*, 56, 234 – 269.

Kevrekidis, I., Gear, C., Hyman, J., and Kevrekidid, P. (2003a), "Equation-Free, Coarse-Grained Multiscale Computation: Enabling Mocroscopic Simulators to Perform System-Level Analysis," *Communications in Mathematical Sciences*, 1, 715—762.

Kevrekidis, I., Gear, W., Hyman, J., Kevrekidid, P., Runborg, O., Theodoropoulos, C., et al. (2003b), "Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis," *Communications in Mathematical Sciences*, 1, 715–762.

Krylov, N. (1996), *Lectures on Elliptic and Parabolic Equations in Hölder Spaces*, American Mathematical Society.

Little, A., Jung, Y.-M., and Maggioni, M. (2009), "Multiscale Estimation of Intrinsic Dimensionality of Data Sets," in *Proc. A.A.A.I.*

Little, A., Maggioni, M., and Rosasco, L. (2012), "Multiscale Geometric Methods for Data Sets I: Multiscale SVD, Noise and Curvature," Tech. rep., MIT-CSAIL-TR-2012-029/CBCL-310, MIT, Cambridge, MA.

Maggioni, M., Minsker, S., and Strawn, N. (2013), "Geometric Multi-Resolution Dictionary and Manifold Learning: Non-asymptotic Bounds, Noise, and Rough Sets," *in preparation*.

Mattingly, J., Stuart, A., and Tretyakov, M. (2010), "Convergence of Numerical Time-Averaging and Stationary Measures via Poisson Equations," *SIAM Journal on Numerical Analysis*, 48, 552–577.

Melbourne, I. and Stuart, A. (2011), "A note on diffusion limits of chaotic skew-product flows," *Nonlinearity*, 24, 1361.

Moore, B. C. (1981), "Principal Component Analysis in Linear System: Controllability, Observability and Model Reduction," *IEEE Transactions on Automatic Control*, pp. 17–32.

Øksendal, B. (2003), *Stochastic Differential Equations*, Springer.

Pande, V., Beauchamp, K., and Bowman, G. (2010), "Everything you wanted to know about Markov State Models but were afraid to ask," *Methods*, 52, 99–105.

Pardoux, E. (1999), "Homogenization of linear and semilinear second order parabolic PDEs with periodic coefficients: a probabilistic approach," *Journal of Functional Analysis*, 167, 498–520.

Pavliotis, G. and Stuart, A. (2007), "Parameter estimation for multiscale diffusions," *Journal of Statistical Physics*, 127, 741–781.

Penrose, R. (1956), "On best approximate solutions of linear matrix equations," in *Proceedings of the Cambridge Philosophical Society*, vol. 52, pp. 17–19, Cambridge Univ Press.

Rohrdanz, M. A., Zheng, W., Maggioni, M., and Clementi, C. (2011), "Determination of reaction coordinates via locally scaled diffusion map," *J. Chem. Phys.*, p. 124116.

Saul, L. and Roweis, S. (2003), "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, 4, 119–155.

Shardlow, T. and Stuart, A. M. (2000), "A Perturbation Theory for Ergodic Properties of Markov Chains," *SIAM J. Num. Anal*, 37.

Silva, V. D. and Tenenbaum, J. (2004), "Sparse multidimensional scaling using landmark points," Tech. rep., Technical report, Stanford University.

S.Larson, Snow, C., Shirts, M., et al. (2002), "Folding@ Home and Genome@ Home: Using distributed computing to tackle previously intractable problems in computational biology," .

Stroock, D. (2008), *Partial Differential Equations for Probabilists*, Cambridge University Press.

Vanden-Eijnden, E. (2003), "Fast Communications: Numerical techniques for multiscale dynamical systems with stochastic effects," *Communications in Mathematical Sciences*, 1, 385–391.

Vershynin, R. (2012), "Introduction to the non-asymptotic analysis of random matrices," *Compressed Sensing, Theory and Applications*, Chapter 5.

Weinan, E., Engquist, B., Li, X., Ren, W., and Vanden-Eijnden, E. (2007), "Heterogeneous multiscale methods: a review," *Commun Comput Phys*, 2, 367–450.

Yuri, M. P. . M. (1998), *Dynamical Systems and Ergodic Theory*, London Mathematical Society.

Z. Zhang, H. Z. (2002), "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *CoRR*, cs.LG/0212008.

Zabczyk, G. D. P. . J. (1996), *Ergodicity for Infinite Dimensional Systems*, Cambridge University Press.

Zheng, W., Rohrdanz, M., and Clementi, C. (2013), "Rapid Exploration of Configuration Space with Diffusion-Map-Directed Molecular Dynamics," *The Journal of Physical Chemistry B*.

Biography

Miles Crosskey was born on September 24th, 1986 in Hartford, CT. He graduated from New Milford High School in 2004. After that, he went on to study Mathematics at Rensselaer Polytechnic Institute. There he began research working on the mathematics behind invisibility cloaks with some undergraduate comrades, under the supervision of Gregor Kovačič. The work was then published in Physics Letters A Crosskey et al. (2011). Then he went on to graduate work at Duke University where this work will earn him a Ph.D. in Mathematics in March 2014.