# Manifold Learning to Enable Equation-Free Modeling of Chemical Engineering Systems

Proposed by David Sroczynski

under the supervision of

Professor Yannis Kevrekidis

11/04/2015

# Contents

# 1 Introduction

Virtually every field of science and engineering uses modeling to expand and elucidate the information available from experiments. These models range from simple one variable ODEs for a CSTR to computational fluid dynamics code. Experimental data is inherently messy; a well-developed model can bring clarity to complicated systems. Models have the additional advantage that they can be initialized as often as desired in whatever state is desired, while experiments can be challenging or expensive to initiate. The obvious caveats are that a model must be accurate enough to capture the "important" information while being compact enough to be solved in a reasonable timeframe.

The field of dimensionality reduction explores the question of what information is "important". The concept is very old; for example, we learned early on that in the analysis of a chemical reaction, the important variables often include temperature and concentrations. The reaction may involve collisions between individual atoms, but we only need to know certain things about the average behavior to predict the results that we are interested in. However, as we encounter new systems (which are often more complex), we desire more systematic ways to extract these important variables. The most ubiquitous method is principal components analysis (PCA), which can be traced back to as early as 1901 [13], but was developed across many fields under many names over much of the $20^{th}$ century [8]. PCA can be thought of as an extension of the line-of-best-fit method to higher dimensions: if the data is viewed as a cloud of points in some high dimensional space, PCA determines which directions characterize the highest variability in the data. Projecting the data onto only these directions results in a reduced representation of the data that still captures most of the variability. One drawback to PCA is that it only captures linear relationships (consider the case of a line-of-best-fit vs. a nonlinear curve fit). The field of nonlinear dimensionality reduction has grown rapidly in recent years with the introduction of methods like kernel PCA [15], local linear embedding [14], Isomap [18], and diffusion maps [3] [4] [2].

Dimensionality reduction methods have the potential to greatly improve models in terms of clarity and speed by reducing the model to the lower-dimensional representations that are easier to visualize and faster to compute. We are interested in using diffusion maps to improve model performance in molecular simulation and other relevant chemical engineering systems. Diffusion maps has recently shown promise in data analysis for models in transport [17], chemical kinetics [1], and molecular dynamics [5] [12] [11]. We intend to expand on this work in the context of the "equation-free" (EF) framework, which uses short simulations of detailed models as the basis for projection in a coarse model (whose variables can be determined by diffusion maps). This idea of data analysis for the purpose of computation includes the idea that diffusion maps can aid in the development of a model or the characterization of dynamical systems by biasing where to next take data to gain "important" information. This in particular has applications to molecular simulation, where the characterization of rare events can be extremely expensive; methods to speed up these computations often require a good coarse variable to bias the simulation.

This document will first explain the EF framework and then discuss diffusion maps and some of the challenges in its implementation. We will describe current work being done on a data-centric application in developmental dynamics, and then we will conclude with systems of interest in future work.

# 2 Equation-Free Modeling

EF modeling refers to the framework of using exisiting detailed microscopic simulators in a black-box manner to enable solutions to macroscopic tasks that are intractable for the microscopic simulator in its original formulation [16] [9] [10]. Consider such a simulator (e.g., MD or CFD code) which performs detailed time-stepping on the microscopic level given certain initial conditions and parameters. EF modeling requires the existence of an appropriate coarse (lower dimensional) description, where appropriate in this context means that it should capture the "important" information. We want the detailed simulation to track along some lower dimensional manifold (which can be represented by our coarse description) embedded in the detailed, high dimensional space. Simulations that are initialized away from this manifold should very quickly move to the manifold in the spirit of chemical kinetics systems that follow the quasi-steady-state approximation.

EF modeling also requires that we be able to move between coarse (low dimensional) and fine (high dimensional) descriptions through some reasonable tractable operators. A restriction operator converts a fine description to a coarse description; for example, we might restrict the speeds of each molecule in MD simulation to the coarse variable temperature. Lifting operators, which do the reverse, often have the added complication that there are many fine descriptions that correspond to the same coarse description, so some care needs to be taken in systematically choosing a fine description. In many cases, however, even if the fine simulation is initialized poorly, it will quickly move back to the appropriate manifold. This is typically referred to as healing.

Given the detailed simulator, the appropriate coarse description, and the two operators, the most straightforward application of the EF framework is coarse projective integration (CPI), where we step the coarse variables forward in time based on properly initialized fine simulations. For each integration step, the procedure is as follows:

1. Use the lifting operator to determine initial conditions for the fine simulation based on the current values of the coarse variables.

2. Run the fine simulation for a short healing period to bring it back to the appropriate manifold.

3. Continue to run the fine simulation until sufficient information is captured about about the progression of the coarse variables.

4. Project the coarse variables forward in time using forward Euler integration or any other integration scheme.

If the simulation data actually lie on or near a manifold characterized by the coarse variables, coarse projective integration offers vast speed-ups compared to straight integration of the fine simulation. This framework of using fine simulations to determine the coarse behavior also has applications in stability/bifurcation analysis [19] [7] and the exploration of potential surfaces [6]. We are interested in applying this approach to systems where the coarse variables are not known *a priori* but must be determined through dimensionality reduction.

# 3 Diffusion Maps

## 3.1 Overview

Diffusion maps is a method of analyzing the geometry of data and discovering lower-dimensional manifolds that the data approximates. The algorithm is designed to approximate the continuous Laplace-Beltrami operator (which has been shown to provide good parametrizations of nonlinear manifolds) to discrete data. Suppose you have $m$ data points in $n$-dimensional space represented by $y_1$,...,$y_m$. The algorithm first constructs a weight matrix such that

$$W_{ij} = exp(-\frac{||y_i - y_j||^2}{\sigma^2})$$

where $|| \bullet ||$ is an appropriate norm or distance metric between to data points and $\sigma$ is a characteristic distance such that points are considered close. Diffusion maps treats distances smaller than $\sigma$ as important but treats distances much longer than $\sigma$ as meaningless. The matrix is then made row-stochastic by dividing each row by its sum so that the rows sum to 1. This gives $W$ the interpretation of a Markov matrix such that the elements represent transition probabilities from one data point to another. Variations to the algorithm exist which can, among other things, account for variations in sampling density.

The eigendecomposition of W yields eigenvalues $\lambda_0$,...,$\lambda_{m-1}$ and eigenvectors $\phi_0$,...,$\phi_{m-1}$. Due to row-stochasticity, the first eigenvector $\phi_0$ is a trivial constant vector with $\lambda_0 = 1$. The other eigenvectors provide a new coordinate system such that the $k^{th}$ component of $y_i$ is given by the $i^{th}$ component of $\phi_k$, scaled by $\lambda_k$. In this new coordinate system, distance between two points is referred to as the diffusion distance. This diffusion distance represents moving from one point to another by diffusion, where you can only move to nearby points based on the probabilities in the weight matrix. Since diffusion distance is based on moving from point to point, it approximates the distance along the manifold that the points lie on. If there is a large spectral gap, meaning that some eigenvalues are significantly larger than others, then the diffusion distance can be accurately approximated using only coordinates

with the largest eigenvalues. The number of these large eigenvalues gives information about the true dimensionality of the manifold.

## 3.2 Choosing a distance metric and kernel scale

Since the diffusion maps algorithm is based entirely on the distances between points, choosing an approriate distance metric is critical. Assuming that each data point is represented by a vector such that each element represents one variable that describes the state (e.g., concentration of one chemical species, signal intensity at some point in space, etc.), then the obvious solution is simply the Euclidian distance between the two points described by the two data vectors. Unfortunately, in many instances, straightforward application of the Euclidean distance is not an informative measure of similarity. Here is a listing of some alternate approaches, and a short explanation of when they might be useful:

1. **Data preprocessing.** This is a general term for cleaning up data so that the Euclidean distance becomes more informative. This can include actions like standardization of variables which span disparate scales, blurring of spatially organized data to remove uninformative small-scale structure, and contrast boosting or other image enhancement.

2. **Feature extraction.** In certain cases, *a priori* knowledge about the system can justify the selection of informative features. These functions of the original data can include ratios of quantities, times to achieve certain benchmarks, or quantities of distributions. This is essentially an initial step of dimensionality reduction before allowing the formal algorithm to finish the job. Unfortunately, this *a priori* knowledge is often unavailable for new systems.

3. **Mahalanobis distance.** This metric is primarily useful in stochastic systems where movement along the slow manifold is obscured by independent white noise. Based on the covariance of the data, the Mahalanobis distance ignores directions which do not indicate meaningful dissimilarity. Given column vector observations $y_i$ and $y_j$ from a distribution with covariance $C$, the Mahalanobis distance is given by

$$||y_i - y_j||_M^2 = (y_i - y_j)^T C^{-1} (y_i - y_j)$$

   The covariance matrix typically must be estimated from the data, and often the estimation is not full-rank so a pseudoinverse is required.

4. **Earth mover's distance**. Also known as the Wasserstein metric, this is a measure of distance between two probability ditributions, or more practically, between two normalized histograms. The earth mover's distance quantifies how much work is required to change one histogram into another. The choice to use the histogram is a form of feature extraction that is useful when the individual quantities are less informative than their distribution, as is often the case in applications such as molecular simulation.

5. **Graph metrics** A graph is a representation of data that defines nodes (which can be people, websites, locations, etc.) and edges (connections) between nodes. These edges can be either weighted or unweighted, and either directed or undirected. Methods for graph similarity include edit distance, maximal common subgraph, and graph kernels. These methods are often computationally intractable, but methods exist for their approximation.

Once the distance metric is chosen, the kernal scale $\sigma$ must also be chosen. This often requires some trial and error, but a good initial guess is to use some fraction (e.g., $\frac{1}{7}2$) of the median pairwise distance between data points. An alternate approach is to use the maximum distance to some number of nearest neighbors, averaged over each data point.

## 3.3 Synchronization of data

Another issue that can occur in the application of diffusion maps is when data points have some degree of freedom among some group that obscures the dynamics. One example is dynamics under periodic boundary conditions where we often want to factor out the translation. Other examples include 2D images that can be arbitrarily rotated, or molecular simulations of a complex molecule that can be arbitrarily shifted and rotated in 3D space. A common way to solve this problem is to align each images based on some template, but when the dynamics are complicated, it is not always obvious how to choose a template without *a priori* knowledge, and pairwise alignment between a data point and a template can be noisy. The eigenvector alignment method solves this problem by considering each data point as the template for each of the other points and finding globally optimal alignments for each image.

The algorithm requires that the transformation between members of the group be represented by some operator $g_{ij}$ such that the operator satisfies the triplet consistency relation:

$$g_{ik} = g_{ij}g_{jk}$$

In the case of 1D periodic boundary conditions, we can consider the dynamics to be taking place on a ring, and then the operator $g_{ij}$ becomes $exp(-\theta_{ij})$, where $\theta_{ij}$ is the shift that takes data point $j$ to data point $i$. For general rotations, $g_{ij}$ becomes the rotation matrix $R_{ij}$, which can be 2 by 2 for rotations in the plane or 3 by 3 for rotations in 3d space. Once this operator is defined, we constrict the matrix $G$ such that each the $ij^{th}$ element (or $ij^{th}$ block) of $G$ is the operator $g_{ij}$. The top eigenvector (or eigenvectors in the block matrix case) gives the operators $g_i$ which transformed the optimal representation into data point $i$. This representation has degrees of freedom within the group (in the case of rotation within the plane, the solutions could be arbitrarily rotated within the plane), but it is optimized based on all pairwise comparisons.

An extension called vector diffusion maps exists which simultaneously aligns the images while performing diffusion maps. VDM is useful in cases where the dynamics are dramatic,

and pairwise alignments between dynamically different data points become meaningless. VDM ignores these alignments and simultaneously provides global alignment information and dimensionality reduction.

### 3.4 Lifting and Restriction Operators

Various choices exist for lifting and restriction operators, which are essentially interpolation/curve fitting schemes. Several choices are summarized here; more detailed explanations and analysis can be found in [1]. The problem statement is as follows: given a set of detailed descriptions $y_1$,...,$y_m$, along with a set of corresponding coarse descriptions $\phi_1$,...,$\phi_m$ from diffusion maps, how do we assign a new coarse description $\phi$ to a new detailed state $y$ and vice versa.

1. **Nyström Extension.** This is one of the most common methods for finding diffusion maps coordinates for a new detailed state. The new coordinate is essentially calculated as a weighted sum of all of the previous coordinates with weights based on the diffusion kernel. The equation for the $\alpha^{th}$ component of the new coarse description is given by

$$\phi_\alpha = \frac{1}{\lambda_\alpha} * ($$

### 3.5 Extending outside of the support

## 4 Application to Data Fusion: *Drosophila* embryos

### 4.1 Data description and motivating problem

One application of diffusion maps is in the study of developmental dynamics. This application is not within the EF framework and is more focused on the data itself rather than using the data for calculations, but it does showcase many of the relevant tools in data analysis that form the backbone of EF modeling. The specific case is the development of *Drosophila* embryos, where researchers are interested in tracking the embryo structure (as represented by the locations of cell nuclei) as well as the distribution of various relevent proteins. There are two relevant types of data sets:

1. A live movie set consists of images every 30 seconds of the same embryo over some range of its development time. These images have nuclei labeled by Histone-RFP, and also have associated time stamps.

2. Fixed snapshot data sets consist of one image from each of a set of embryo. These embryo have three channels; they are stained different colors for the nuclei as well as two proteins, Twist and dpERK. These images do not have time stamps, and they may also be arbitrarily rotated in the plane.

The motivating problem is to construct an average developmental trajectory that contains information about all three channels as well as approximate timings. Prior work in the group has shown that the vector diffusion maps algorithm can very accurately reconstruct the ordering of the live movies even when they are scrambled and arbitrarily rotated. It has also been shown that VDM can also correctly rotate a data set of fixed snapshots, and the ordering is reasonably well consistent with manual ordering by an expert. The next step is to fuse these data sets together to produce a trajectory with color and time.

## 4.2 Challenges in data preprocessing

Each image consists of 100 by 100 pixels. The live movies have just one channel value at each pixel, while the fixed snapshots have three. Each image can thus be viewed as a vector of length $100,000$ (or $300,000$ for fixed snapshots), and a simple first choice of metric is the Euclidean distance between two such vectors. Unfortunately, some significant work needs to be done to make that Euclidean distance informative. The imaging process can produce various image artifacts that are not relevent to the developmental dynamics, especially for the fixed snapshots which are all different embryo. Images need to be centered in the field of view and resized to occupy a consistent portion of the frame. Contrast limited adaptive histogram equalization (CLAHE) is used to normalize the images. We recently found that boosting the contrast significantly improves the performance, probably because whether or not a pixel is occupied is more important than the specific brightness level. Finally, we blur the images using a Gaussian filter to remove small scale structure that distorts the comparison. Variability between embryo means that features will not be in the exact same place; blurring allows such features to still be properly compared.

## 4.3 Synchronization of rotations and movie times

While the embryos in the live movies stay at the same orientation throughout development, the embryos in the fixed snapshots can be in any arbitrary orientation within the plane. While manual registration is possible, it is extremely tedious. The eigenvector alignment method is capable of registering the fixed snapshots with respect to each other in just seconds of computational time. As previously discussed, the final orientation is arbitrary, so there is a final step of rotating the snapshots to align with the live movies in order to do any comparison. The eigenvector alignment method also provides a framework for aligning the different live movies and the snapshots with respect to each other, but with only 7 live movies and 1 data set of snapshots, the gains over manual registration are minimal.

Another synchronization of interest is that of movie times. When diffusion maps is run on multiple live movies simultaneously, we find that within each live movie there is a good monotonic relationship between time and the first diffusion maps coordinate. However, these different functional relationships are offset from each other. We believe that while some of this error is due to noise and embryo variability, issues with experimental design could be

causing some of the discrepancy. For example, lack of precision in the fertilization time could mean that the times for one embryo are shifted in relation to another. Differences in ambient temperature or other environmental factors could cause one embryo to develop at a faster rate. We are interesting in finding factors by which to scale and shift the movie times so that they optimally align with each other. If we consider operations of the form

$$t_j = a_{ij} * t_i + b_{ij}$$

then the eigenvector alignment method can be used by considering the matrix operator

$$g_{ij} = \begin{bmatrix} 1 & 0 \\ a_{ij} & b_{ij} \end{bmatrix}$$

The pairwise $a_{ij}$ and $b_{ij}$ can by quickly found by minimizing the least squares error between the two relationships for time as a function of the first diffusion maps coordinate. We then form the block matrix G, and the top two eigenvectors can be used to form a block eigenvector which contains the scales and shifts from the globally optimal solution. Again, there are degrees of freedom in the scale and in the shift, so we apply a final scale and shift such that the mean shift that we apply to the data is 0 and the mean log scale we apply is also 0. This is in an effort to modify the data as little as possible, and also to make the algorithm deterministic.

### 4.4 Data fusion to construct a representative trajectory

## 5 Systems of Interest for Future Work

## 6 Conclusions

## References


[1] E. Chiavazzo, C. Gear, C. Dsilva, N. Rabin, and I. Kevrekidis. Reduced Models in Chemical Kinetics via Nonlinear Data-Mining. *Processes*, 2(1):112–140, Jan. 2014.

[2] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[3] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–31, May 2005.

[4] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: multiscale methods. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7432–7, May 2005.

[5] A. L. Ferguson, A. Z. Panagiotopoulos, P. G. Debenedetti, and I. G. Kevrekidis. Systematic determination of order parameters for chain dynamics using diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 107(31):13597–602, 2010.

[6] T. A. Frewen, G. Hummer, and I. G. Kevrekidis. Exploration of effective potential landscapes using coarse reverse integration. *Journal of Chemical Physics*, 131(13), 2009.

[7] C. W. Gear, I. G. Kevrekidis, and C. Theodoropoulos. 'Coarse' integration/bifurcation analysis via microscopic simulators: micro-Galerkin methods. *Computers & Chemical Engineering*, 26(7–8):941–963, 2002.

[8] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

[9] I. Kevrekidis and C. Gear. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

[10] I. G. Kevrekidis, C. W. Gear, and G. Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7):1346–1355, July 2004.

[11] S. B. Kim, C. J. Dsilva, I. G. Kevrekidis, and P. G. Debenedetti. Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. *The Journal of Chemical Physics*, 142(8):085101, 2015.

[12] L. V. Nedialkova, M. A. Amat, I. G. Kevrekidis, and G. Hummer. Diffusion maps, clustering and fuzzy Markov modeling in peptide folding transitions. *The Journal of chemical physics*, 141(11):114102, Sept. 2014.

[13] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(1):559–572, 1901.

[14] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[15] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[16] C. I. Siettos, C. C. Pantelides, and I. G. Kevrekidis. Enabling Dynamic Process Simulators to Perform Alternative Tasks: A Time-Stepper-Based Toolkit for Computer-Aided Analysis. *Industrial & Engineering Chemistry Research*, 42(26):6795–6801, Dec. 2003.

[17] B. E. Sonday, M. Haataja, and I. G. Kevrekidis. Coarse-graining the dynamics of a driven interface in the presence of mobile impurities: Effective description via diffusion maps. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 80(3):1–11, 2009.

[18] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–2323, 2000.

[19] C. Theodoropoulos, Y.-H. Qian, and I. G. Kevrekidis. "Coarse" stability and bifurcation analysis using time-steppers: A reaction-diffusion example. *Proceedings of the National Academy of Sciences*, 97(18):9840–9843, 2000.