EQUATION-FREE ANALYSIS FOR
AGENT-BASED COMPUTATION

PING LIU

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
CHEMICAL AND BIOLOGICAL ENGINEERING
ADVISER: PROFESSOR IOANNIS G.KEVREKIDIS

JANUARY 2013
Abstract

Due to the advances in computing technology, agent-based modeling (ABM) has become a powerful tool in addressing a wide range of problems. However, there is a challenge which modelers often encounter: the effective nonlinear and stochastic nature of individual dynamics and the inherent complexity of microscopic descriptions; closures that allow us to write macroscopic evolution equations for the coarse-grained dynamics are usually not available.

As an attempt to overcome this difficulty and to enable system-level analysis of agent-based simulations, the Equation-Free (EF) approach is explored in this Thesis in studying two different agent-based models. The first agent-based model describes the dynamic behavior of many interacting investors in a financial market in the presence of mimesis. Three aspects of the EF framework are successfully applied to this model: (1) in the coarse bifurcation analysis, using appropriately initialized short runs of the microscopic agent-based simulations, bifurcation diagrams of the identified coarse variables are constructed, and the stability of its multiple solution branches are determined; (2) in the coarse rare event analysis, an effective Fokker-Planck (FP) equation is constructed on a coarse-grained one dimensional reaction coordinate. The mean escape time of the associated rare event computed using this effective FP shows good agreement with the results from direct agent-based simulations, but requires only 3.2% of the computational time; (3) utilizing the smoothness of coarse variables, a patch-dynamics scheme is successfully designed which allows expensive agent-based simulations to be performed in small “patches” (2%) of the full spatio-temporal domain, while giving comparable system-level solutions.

The second agent-based model describes the dynamic behavior of a group of swarming animals. Using a recently developed data-mining technique - Diffusion Maps (DMAP) - interesting coarse level features about the swarming dynamics were successfully captured. The first two dominant DMAP coarse variables character-
ize the “up-down” and “left-right” directions of collective group motion respectively. Based on these two DMAP coarse variables, a reduced stochastic differential equation (SDE) model is successfully constructed using the EF framework. Using the reduced SDE model, the associated coarse rare events are efficiently studied, circumventing expensive long-term agent-based simulations.
Acknowledgements

I am heartily thankful to my advisor, Prof. Ioannis G. Kevrekidis, not only for patiently guiding me through the journey of research, but also for the warm help he provided beyond that. I would also like to thank Prof. Stanislav Y. Shvartsman, Prof. C. William Gear, Prof. Constantinos I. Siettos, Prof. Giovanni Samaey, and Prof. Iain D. Couzin for their critical guidance and advice on the multiple projects I describe in this Thesis.

I am grateful to the previous and current Kevrekidis group members: Michail E. Kavousanakis, Thomas A. Frewen, Dror Givon, Lilia V. Nedialkova, Yu Zou, Benjamin Sonday, Karthikeyan Rajendran, Carlos E. Colosqui, Andreas C. Tsoumanis, Arta Jamshidi, Miguel A. Amat, Carmline Dsilva, Long Chen, and Matthew Williams, for their invaluable help and support provided in the last five years. I would like to give special thanks to Dr. Michail E. Kavousanakis for his kind guidance about hands-on skills for the Equation-Free analysis in general, to Dr. Thomas A. Frewen for offering me helpful tutorials about Diffusion Maps, to Dr. Dror Givon for teaching me stochastic calculus basics, and to Lilia V. Nedialkova for her help about eigencomputations for large scale data sets, and much more, beyond research. There are many more things for which I am indebted to you all, but to list all of them would make this acknowledgement long. Here I would like to express my deepest gratitude to all of you again, for your help, support and friendship.

Many thanks to Robin Block, Karen Oliver, and Heather Yacone for their kind administrative help, to Eric J. Paul for his patient IT support, and to Bill Wichser for his voluntary system support about the high performance computer cluster.

This Thesis is dedicated to my grandfather, my parents, my wife and my daughter. I thank my grandfather and my parents for their love, care and encouragement provided to me along the way I grew up. I thank my wife Li for her love and support.
without which this Thesis would not be possible. I thank my daughter Aiwei for bringing me joy and courage while I was working on my thesis. Thank you.
To my grandfather, parents, wife and daughter.
Contents

Abstract ........................................................................................ iii
Acknowledgements ...................................................................... v
List of Tables ........................................................................... xi
List of Figures ........................................................................... xii

1 Introduction ........................................................................... 1
  1.1 the Equation-Free (EF) Approach ........................................ 2
    1.1.1 Coarse Time Stepper .................................................. 2
    1.1.2 Coarse Projective Integration ....................................... 5
    1.1.3 Coarse Bifurcation Analysis ........................................ 7
  1.2 A Nonlinear dimensionality reduction tool - Diffusion Maps (DMAP) 11
    1.2.1 The construction of DMAP ........................................ 13
    1.2.2 Manifold interpolation and the Nyström formula .......... 15
  1.3 Contributions and Outline ............................................... 16

2 Equation-Free analysis for a financial market model ................. 18
  2.1 Introduction .................................................................... 18
  2.2 The agent-based financial market model ............................ 20
  2.3 Coarse bifurcation and stability analysis ............................ 22
  2.4 Bifurcation analysis for the continuum approximation .......... 24
    2.4.1 Population dynamics and the Fokker-Planck approximation 26
5 Substrate-dependent ERK phosphorylation can lead to oscillations

5.1 Introduction ................................................................. 105
5.2 Methods ................................................................. 108
  5.2.1 Mathematical model ................................................... 108
  5.2.2 Nondimensionalization ................................................ 110
  5.2.3 Quasi-steady-state approximation and the reduced model ........ 111
  5.2.4 The auxiliary model ................................................... 112
5.3 Results and Discussion .................................................... 113
  5.3.1 Separation of time scales in the full model ..................... 116
  5.3.2 Connection between bistability and oscillations ............... 122
5.4 Conclusions .............................................................. 124

Bibliography ............................................................... 126
List of Tables

3.1 Numerical results of the order of consistency analysis .......................... 81
3.2 Order of consistency estimates for the two limits ................................. 81
4.1 Comparison of the statistics between ABM and SDE models .................. 101
5.1 A representative parameter set for oscillations ................................. 114
List of Figures

1.1 Framework of coarse time stepper ........................................... 3
1.2 Schematic of the Coarse Projective Integration Scheme .................. 6
1.3 Schematic of Newton-GMRES method ....................................... 9
1.4 Dmap embedding of a closed spring curve .................................. 12

2.1 Coarse bifurcation diagram .................................................... 25
2.2 Two-parameter bifurcation diagram ........................................... 27
2.3 Three dimensional one-parameter bifurcation diagram ..................... 29
2.4 Bifurcation results for the asymmetric case ................................ 30
2.5 Comparison of analytical and discretized PDE solutions ................... 31
2.6 Illustration of the point markers ............................................. 36
2.7 Probability density distributions of the coarse variable ................. 40
2.8 Autocorrelation function C(t) of the noise term .......................... 41
2.9 Profiles of coarse characteristic functions ................................ 43
2.10 Histogram of escape times ................................................... 44
2.11 First two dominant DMAP eigenvectors .................................... 47
2.12 Top 50 dominant eigenvalues ................................................. 47
2.13 Comparison between DMAP and intuition-based coordinates ........... 48

3.1 Illustration of a finite volume method ....................................... 59
3.2 Illustration of the gap-tooth scheme ......................................... 62
3.3 Illustration of the lifting procedure ........................................ 64
3.4 Illustration of the simulation step (bulk domain) ......................... 67
3.5 Illustration of the simulation step (boundary) ............................. 68
3.6 Comparison of the solutions of different schemes ....................... 85
3.7 Numerical results of the order of consistency analysis .................. 86
3.8 Snapshots along a patch dynamics agent-based trajectory ............ 87

4.1 Sample snapshots of the ABM animal swarming model ................. 92
4.2 Illustration of the intermediate coarse variables .......................... 93
4.3 Diffusion maps embedding and the microscopic snapshots ............ 96
4.4 Snapshots for representative migration states ............................. 97
4.5 Plots of log-densities for the three cases studied ....................... 98
4.6 Comparison of solutions between ABM and the reduced SDE ......... 101
4.7 Illustration of the mean exit time ........................................... 102
4.8 Comparison of the mean exit time distributions ......................... 102

5.1 Model biochemical network and one-parameter bifurcation diagrams . 107
5.2 A sample limit cycle in the full model ..................................... 117
5.3 Representative one-parameter bifurcation diagrams ..................... 120
5.4 Partition of input/load parameter space .................................. 121
5.5 Limit cycle in the reduced system ......................................... 123
Chapter 1

Introduction

Over the last years, agent-based modeling has become a powerful simulation modeling technique with a number of applications, varying from ecology [9, 39, 10] to economics and financial systems [88, 92, 105], and from traffic and supply chain networks [40, 76, 101] to biological [12, 100, 109] and social [38, 41, 71] systems.

In agent-based models, agents are treated as unique and discrete entities. They interact with each other and make their own decisions through prescribed microscopic rules. Modelers expect to study the macroscopic, population level dynamical behavior (we call it emergent behavior) through the evolution of the detailed system itself. This bottom-top kind of approach is usually more realistic than the traditional state-variable-based (or top-down) approach because more details can be specified at the microscopic level. However, this bottom-top approach in nature has a spatio-temporal gap between the available description level (individual, microscopic-level) and the desired description level (population, system-level). This gap caused a critical challenge in connecting mathematical approaches (e.g. Newton-Raphson method) to the current state-of-art agent-based modeling, since traditional computational tools usually require a coarse-grained model in closed form (i.e. macroscopic equations at the population level). To overcome this obstacle, over the last years Kevrekidis et al.
have developed a systematic computational methodology aimed at extracting and understanding coarse-grained, emergent dynamical phenomena by bridging modern agent-based modeling with macroscopic, systems-level, continuum numerical analysis. This methodology is termed the “Equation-Free approach” as it bypasses the derivation of macroscopic evolution equations when these equations conceptually exist but are not available in closed form. This is achieved by using appropriately initialized short bursts of the agent-based simulator to estimate coarse-grained quantities required for continuum numerics (e.g. coarse residuals, coarse Jacobians etc.). This Thesis focuses on applying the EF approach towards efficiently performing macroscopic-tasks for agent-based modeling problems. The EF approach is introduced in the next section.

1.1 the Equation-Free (EF) Approach

A persistent feature of many complex systems is the emergence of macroscopic, coherent behavior from the interactions of microscopic agents such as molecules, cells or individuals in a population. The implication is that macroscopic rules (a description of the system at a coarse-grained, high level) can somehow be deduced from microscopic ones (a description at a much finer level) even when the macroscopic equation is not available. The objective of the equation free approach is to find such macroscopic rules directly from microscopic descriptions when macroscopic equations are not available (i.e. the traditional computational tools based on macroscopic equations cannot be applied).

1.1.1 Coarse Time Stepper

The backbone of Equation-Free approach is the design of computational experiments. In traditional numerical analysis, the main code ‘pings’ a subroutine containing the
model (from the macroscopic equation), and uses the returned information (time derivatives, etc.) to perform computer-assisted analysis. In the equation free approach, the same main code ‘pings’ a subroutine that runs an ensemble of appropriately initialized computational experiments from which the same quantities are estimated. This ensemble of appropriately initialized computational experiments together with associated lifting and restriction procedures forms the core part of the equation free approach - the coarse time stepper [46], which allows the performance of numerical tasks at the macroscopic level using microscopic/stochastic simulations. As shown in Fig. 1.1 the coarse time stepper provides a bridge between the microscopic agent based simulator on the one hand and traditional numerical analysis, such as control design and optimization techniques, parametric continuation, coarse stability analysis (the numerical superstructure which usually work on coarse descriptions) on
the other. The computational methodology of coarse time stepper consists of the following elements:

(a) Choose the appropriate statistics for describing the long-term behavior of the system. For some systems, the statistics can be obvious. For example, in a gas simulation at the particle level, the statistics would probably be the density and momentum (zeroth and first moment of the particle distribution over velocities). Let us denote the identified statistics as $x$. Then a restriction operator $M$ can be determined accordingly, from the microscopic description $X$ to the macroscopic description: $x = MX$.

(b) Choose an appropriate lifting operator $\mu$, from the macroscopic description $x$, to a set of consistent microscopic descriptions $X$. Here consistent means $M\mu = I$, i.e. lifting from the macroscopic to the microscopic and then restricting down again should have no effect (except for the round off errors).

(c) Start with a macroscopic condition (e.g., concentration profile) $x(t_0)$, lift it to an ensemble of consistent microscopic realizations $X(t_0) = \mu x(t_0)$.

(d) Evolve these realizations using the agent-based simulator for an appropriately chosen short macroscopic time horizon, $\Delta T$, to obtain the value(s) $X(t_0+\Delta T) = \Phi_{\Delta T}(X(t_0)) = \Phi_{\Delta T}(\mu x(t_0))$. The choice of $\Delta T$ is estimated by the separation of time scales between the individual and the coarse grained dynamics of the system under study [47].

(e) Obtain the restrictions $x(t_0 + \Delta T) = MX(t_0 + \Delta T)$, and average over them.

The above procedure for the coarse time stepper can be considered as a black box mapping, from an initial macroscopic condition to the macroscopic property after $\Delta T$. If this map is accurate enough, we can wrap around the coarse time stepper numerical superstructures such as the projective integration technique [34].
matrix-free Newton-GMRES [45] method and Arnoldi iterative eigen-solver [84] to find coarse steady states, and to perform coarse bifurcation analysis, controller design [1] [93] and optimization tasks [8].

1.1.2 Coarse Projective Integration

Coarse projective integration enables “short time” coarse time stepper to perform “long time” tasks. It exploits the smoothness in time of the unavailable macroscopic equation in order to project (jump) to the future. For this reason, the coarse projective integration method can significantly reduce the simulation time for systems where this technique is applicable.

The separation of time scales is the underlying assumption for coarse projective integration to work on complex multiscale systems, that is, there are many fast components which can quickly be slaved to a few slow components. This bears strong analogy to stiff differential equations whose eigenvalues are separated into two clusters, one containing “stiff” or fast components and one containing the slow components. The projective integration technique has been demonstrated to be effective in integrating stiff differential equations [34]. For multiscale agent-based models when macroscopic equations are not available, with the help of the coarse time stepper, we expect that coarse projective integration can accelerate the computation of the long time behavior and will also be able to find the stable coarse steady states of the system (when they exist) if the assumption of separation of time scales is valid.

Fig. 1.2 illustrates the coarse projective integration scheme. Suppose the appropriate statistics (the coarse variables), the lifting and restriction operators have all been determined as mentioned in the previous coarse time stepper section; then coarse projective integration can be conducted by repeating the following steps (step (d) below is illustrated by using the explicit forward Euler method):

(a) Lift the coarse initial condition $x_1$ to the microscopic description $X_1$. 
Figure 1.2: Schematic of the Coarse Projective Integration Scheme

(b) Run the microscopic simulator for \( k \) time steps, generating the values \( X_2, X_3, \ldots, X_k, X_{k+1} \).

(c) Obtain the restrictions \( x_n = MX_n \), for \( n = k, k+1 \).

(d) Perform the extrapolation over \( N \) steps using \( x_k, x_{k+1} \) to compute \( x_{k+1+N} \) as

\[
x_{k+1+N} = (N + 1)x_{k+1} - Nx_k
\]

The coarse variable \( x_{k+1+N} \) value obtained in step (d) is used in (a) as the new initial condition. Steps (a)-(d) are repeated to continue the coarse projective integration process.

The first \( k - 1 \) steps in (b) are the healing steps in which the errors, due to the lifting of the coarse initial condition in (a), get healed. We will call them inner integration steps, by analogy to similar steps in projectively integrating stiff differential equations [34]. The number of simulation steps needed for the healing depends on our restriction and lifting procedure. The healing time can be estimated through numerical tests before the actual projective integration is conducted. Steps (c) and
(d) contribute the outer integrator, which projects the simulation forward through, in this case, the forward Euler scheme. Higher order polynomial extrapolation as well as implicit methods can also be applied [35].

Analogous to the coarse projective integration scheme, we can use smoothness in space to do what is called the “gaptooth” scheme [87] - do simulations only in small boxes in space, and interpolate coarsely between them. We can also combine these two schemes to do simulations in little patches of the spatial-temporal domain. The combined scheme is therefore called the “patch-dynamics” scheme [47]; once successfully applied, it can yield greater computational savings. The details of the patch-dynamics scheme are illustrated through an agent-based model in Chapter 3.

1.1.3 Coarse Bifurcation Analysis

In this section, we will discuss another useful building block of the EF approach - coarse bifurcation analysis. This technique enables computations to locate fixed points and periodic orbits, determining the stability of solutions and continuing solutions in parameter space [35] for problems whose coarse equations are not available in closed form.

Fixed point algorithms like Newton-Raphson, provide an efficient way of finding steady-states compared to direct integration (given a good initial guess). Such algorithms can locate both stable and unstable solutions (the latter are impossible to find with direct simulation). However, these algorithms are usually applicable to macroscopic equations. When the model comes in the form of microscopic and stochastic simulators at a much finer scale, it seems that these numerical tools become useless as the “right hand side” for the time derivative is not available, lacking the closed formula for the equation. Fortunately, by coupling the previously discussed coarse time stepper and matrix-free iterative algorithms, such as GMRES [45], the derivation of the coarse equations can be bypassed. The Newton-Raphson algorithm then instead
relies on the evaluation of the matrix-vector inner products, which are approximated by the coarse time stepper integrations.

The coarse time stepper can be written in a general form as:

\[ x_{n+1} = \Phi(x_n). \]  

(1.1)

where \( x \) denotes the \( n \) dimensional coarse variable, and \( \Phi \) denotes the coarse time stepper. From Eq. (1.1), we can get the coarse residual function:

\[ F(x_n) = x_n - \Phi(x_n). \]  

(1.2)

The equation for Newton-Raphson iteration is written as,

\[ \left[ \frac{dF}{dx} \right]_{x_n} (x_{n+1} - x_n) = -F(x_n). \]  

(1.3)

Recent developments in large scale computational linear algebra (the so-called matrix free solvers) can help us solve Eq. (1.3) in an efficient way. The integrated computational scheme is shown in Figure 1.3. There are three main components in this scheme, the coarse time stepper for \( F(x) \), the Newton-Raphson iterative loop for the computation of the coarse fixed point, and the inner iterative GMRES solver for solving the linear system Eq. (1.3) at each Newton step. The first two components have been discussed before. We will now look at how the GMRES solver can be incorporated to solve Eq. (1.3).

The GMRES method is a type of Krylov space method which solves a linear system that has the general form \( Ax = b \), based on the construction of a series of Krylov space vectors:

\[ K_k = span(r_0, Ar_0, ..., A^{k-1}r_0), \]  

(1.4)
\begin{equation}
D_h F(x; v_k) = \left| \left| \left. \left( \frac{x F}{h v} \right) \right| \right|_k \right|_k
\end{equation}

\begin{enumerate}
\item Solve upper-triagonal matrix $H \Delta x = g$ for $\Delta x$
\item $x = x + \Delta x$
\item Test orthogonality of Krylov vectors and do renormalization if necessary
\item Update $\eta$
\item Find elements of Hessenberg Structure, $h_{jk} = (v_{k+1})^* v_j, j = 1, 2, \ldots, k$
\item Do Given’s rotation, Construct rotated Hessenberg matrix $H$ and right-hand vector $g$
\end{enumerate}

Figure 1.3: Schematic of Newton-GMRES method for the computation of coarse steady states
where \( r_0 \) is the initial residual which equals \( b - Ax_0 \). During each GMRES iteration, an additional Krylov vector \( v_{k+1} \) is calculated from the product of the system matrix \( A \) and the previously computed Krylov vector \( v_k \) (followed by Gram-Schmidt orthogonalization to make \( v_{k+1} \) perpendicular to the previous Krylov vectors). During the same process of constructing a Krylov subspace, the Hessenberg matrix \( H \) (coefficients of the Krylov vectors) as well as the right hand side vector \( g \) are found. Eventually the iterative residual \( \rho \) and solution \( x_k \) are found in an efficient manner.

The typical number of iterations \( k \) required to find a satisfactory solution is usually much less than the dimension \( N \) of our linear system. The main advantage of the Newton-GMRES scheme is the beauty of “matrix free” calculation. In Eq. (1.3), we note that the matrix of this linear system \( A \) is actually the Jacobian matrix. Hence, in order to find the product \( Av_k \) for the computation of \( v_{k+1} \), we can instead find the directional derivative \( D_h F(x : v_k) \) without constructing the Jacobian matrix \( A \) explicitly. This can lead to significant savings of the required computational work because each directional derivative approximation requires only one coarse time stepper simulation (so \( k \) iterations requires \( k \) simulations) while the Jacobian matrix \( A \) requires \( N \) simulations (for large systems \( k \) is usually much less than \( N \)). One last thing we need to note is that by adaptively adjusting the forcing factor \( \eta \) for the inner GMRES loop according to the convergence rate of the outer Newton-Raphson process, we can effectively avoid over-calculation in the inner loop.

The above discussed Newton-GMRES scheme enables the efficient computation of the fixed point at each parameter value. Coupling the Newton-GMRES scheme with parameter continuation techniques like the pseudo arc-length continuation \cite{13}, we are able to systematic compute the fixed point solutions of the system over a wide parameter space.

A crucial step in the EF framework is to identify the appropriate coarse variables. Although coarse variables can be identified based on physical intuition about
the problem, such problem-dependent intuition is not always available and usually requires extensive observations and testing of the simulations. Aiming at overcoming this limitation, one goal of this Thesis is to explore the feasibility of a recently developed data-mining tool - *Diffusion Maps (DMAP)* \[15, 16\] - to systematically detect the coarse variables. This data-mining based approach has two main advantages. First, it is a generic approach as it only requires the dataset generated by the simulations - prior knowledge or intuition about the macroscopic level dynamics is not required. Second, this approach can tell us not only what the underlying coarse variables are, but also how many of them are required to sufficiently describe the coarse level dynamics. In the next section the details of DMAP will be discussed.

### 1.2 A Nonlinear dimensionality reduction tool - Diffusion Maps (DMAP)

The well-known and widely used dimensionality reduction technique, known as principal component analysis (PCA, \[44\]) - works well when the data live on or close to a linear manifold; however, it has the severe limitation that it cannot capture nonlinear dependencies among the data. Over the past decade, nonlinear dimensionality reduction techniques such as Local Linear Embedding (LLE, \[82\]), Isomap \[99\], and Laplacian Eigenmaps \[6\] have been developed to uncover nonlinear manifolds from data and have drawn great interest. Diffusion Maps (DMAP) \[15, 16\] is a relatively more recent nonlinear dimensionality reduction technique. Based on a constructed diffusion process along a given high-dimensional data set, this technique can identify the underlying low-dimensional, nonlinear manifold. Expressing the data points in terms of coordinates on this lower dimensional manifold, therefore, results in dimensionality reduction. For example, in Fig. \[14\] the given data points of the “spring” curve originally live in a three dimensional space; however, the underlying manifold is
The DMAP approach is based on the construction of a Markov transition probability matrix corresponding to a random walk on a graph whose vertices are the data points. The transition probabilities are used to measure the local similarities between pairs of data points. The first few eigenvectors of this Markov matrix are used as coordinates, called the DMAP coordinates, which are also the coarse variables we want to identify.

DMAP is a successful generalization of PCA to nonlinear data sets whose geometric structure is not known in advance. Different from PCA, the DMAP coordinates are nonlinear projections of the data. Another distinct feature is that, DMAP uses local similarities of neighboring data points to infer the global geometric structure of the data - the nonlinear embedding. This makes sense because for data living on a nonlinear manifold, a large Euclidean distance is not a meaningful measure of similarities between the data points (i.e., they can be moderately different, or they
can also be significantly different), while a small Euclidean distance almost always indicates the closeness of the data points.

1.2.1 The construction of DMAP

The DMAP is constructed as follows. Given a set of $N$ data points $x_1, x_2, ..., x_N \in \mathbb{R}^d$, we first build a weight matrix $W$ by using a positive semi-definite kernel function $k$,

$$W_{ij} = k(d(x_i - x_j)),$$  \hspace{1cm} (1.5)

where $d$ is a relevant distance metric (e.g. it can be the Euclidean distance). A popular choice for $k$ is the Gaussian kernel $k(d(x_i, x_j)) = \exp(-d^2(x_i, x_j)/\epsilon)$, although other choices are also possible. The parameter $\epsilon$ is a characteristic length scale which corresponds to the bandwidth of the kernel. Note that the matrix elements $W_{ij}$ almost vanish for pairwise distances larger than $\epsilon$, so in practice $W$ is usually a sparse matrix.

The matrix $W$ can also be viewed as an adjacency matrix for a graph with $n$ nodes, with $W_{ij}$ being the weight of the edge between nodes $i$ and $j$. The weight matrix $W$ is then normalized to be row stochastic, by using a diagonal matrix $D$ whose elements are the row sums of $W$,

$$D_{ii} = \sum_{j=1}^{N} W_{ij},$$ \hspace{1cm} (1.6)

$$A = D^{-1}W.$$ \hspace{1cm} (1.7)

The matrix $A$ is a Markov transition matrix which describes the transitions of a Markov chain involving the nodes of the graph defined by $W$. The probability of a random walker starting at point $i$ to arrive at point $k$ at time $t$ is given by $A_{ik}^t$. Then,
to compare points \(i\) and \(k\) at time \(t\), we should compare the rows of \(A^t\),

\[
D_t^2(i, j) = \frac{\sum_{k=1}^{n} (p^t_{i \to k} - p^t_{j \to k})^2}{D_{kk}} = \frac{\sum_{k=1}^{n} (A^t_{ik} - A^t_{jk})^2}{D_{kk}},
\]

(1.8)

where \(D_{kk}\) is the degree of the vertex \(k\). This is to make sure each vertex will have a similar contribution to the difference measure. We refer to \(D_t(i, j)\) as the diffusion distance between \(i\) and \(j\) at time \(t\). It can be related to the eigenvectors and eigenvalues of the Markov transition matrix \(A\) as follows,

\[
D_t^2(i, j) = \sum_{k=1}^{n} \lambda_{2t}^k (\psi_k(i) - \psi_k(j))^2 \tag{1.9}
\]

where \(\psi_1, \psi_2, ..., \psi_N\) and \(\lambda_1, \lambda_2, ..., \lambda_N\) are the eigenvectors and eigenvalues of \(A\) respectively. Because \(A\) is similar to the symmetric matrix \(S = D^{-1/2}WD^{-1/2}\), its eigenvalues \(|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_N|\) are real. Since \(A\) is row stochastic, it can be shown that \(\lambda_1 = 1\) and the corresponding eigenvector \(\psi_1 = (1, 1)^T\). For many practical problems a spectral gap can be observed at some \(\lambda_M\), e.g., \(|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_M| \gg |\lambda_{M+1}| \geq ... \geq |\lambda_N|\). In such a case the diffusion distance can be approximated as

\[
D_t^2(i, j) = \sum_{k=2}^{M} \lambda_{2t}^k (\psi_k(i) - \psi_k(j))^2 \tag{1.10}
\]

The truncated DMAP embedding at time \(t\) is then defined as

\[
x_i \mapsto (\lambda_2^t \psi_2(i), \lambda_3^t \psi_3(i), ..., \lambda_M^t \psi_M(i)). \tag{1.11}
\]

Usually the following \(t = 0\) embedding is used,

\[
x_i \mapsto (\psi_2(i), \psi_3(i), ..., \psi_M(i)). \tag{1.12}
\]
In this Thesis we call the embedding defined in Eq. (1.12) the DMAP embedding. Basically we are using the first \( M - 1 \) DMAP eigenvectors as our coarse variables, and it is common that \( M - 1 \ll d \), where \( d \) is the dimension of the space of the given data before the DMAP technique is applied. The DMAP embedding, therefore, gives a good low-dimensional representation of the data set. It is also interesting to note that if the data come from a Markovian stochastic process, the eigenvectors and eigenvalues are approximations to the eigenfunctions and eigenvalues of the corresponding backward Fokker-Planck operator [68].

1.2.2 Manifold interpolation and the Nyström formula

DMAP is a useful tool for extracting key features of a data set resulting from a dynamical process. However, to successfully incorporate this tool into the existing coarse-graining framework to accelerate simulations, we still need to be able to find the corresponding reduced DMAP coordinates for new data points. This can be done by manifold interpolation through what is known as the Nyström extension [7]. Before we introduce the Nyström formula, we first look at the identities of the DMAP eigenvectors and eigenvalues,

\[
\lambda_j \psi_j(i) = \sum_{k=1}^{N} A_{ik} \psi_j(k) = \frac{1}{D_{ii}} \sum_{k=1}^{N} W_{ik} \psi_j(k), \quad (1.13)
\]

Given a new data point \( x_{\text{New}} \in R^d \), an analogous form of Eq. (1.13) can be written as,

\[
\psi_j(\text{new}) = \frac{1}{\lambda_j D_{\text{new}}} \sum_{k=1}^{N} k(x_{\text{new}}, x_k) \psi_j(k), \quad (1.14)
\]

where,

\[
D_{\text{new}} = \sum_{j=1}^{N} k(x_{\text{new}}, x_j). \quad (1.15)
\]
Equation Eq. (1.14) is the Nyström formula which we use to find the DMAP coordinates for a new data point.

1.3 Contributions and Outline

This Thesis focuses on applying EF approach in efficiently performing macroscopic-tasks for the agent-based modeling problems. The approach demonstrated in this Thesis is generally applicable to wide classes of agent-based problems.

Two illustrative agent-based models have been studied in this Thesis. The first one is an agent-based financial market model which describes the dynamic behavior of many interacting investors in the presence of mimesis. Through the investigation of this model, Chapter 2 and 3 presents three aspects of the EF approach, namely, coarse bifurcation analysis (Chapter 2), coarse rare event analysis (Chapter 2) and the patch-dynamics scheme (Chapter 3). Using appropriately initialized short runs of the microscopic agent-based simulations, coarse bifurcation analysis enables efficient studies of the macroscopic level dynamical behavior of the system. In a parameter regime where an interesting “explosion” happens at random time intervals, by identifying a relevant coarse reaction coordinate and by constructing an effective Fokker-Planck (EF) equation, based on it statistics of the rare event (“explosion” in this case) are efficiently computed, and expensive long time agent-based simulations are bypassed.

Based on the smoothness of the coarse variables, a patch-dynamics scheme in Chapter 3 is also designed. Using this scheme the macroscopic tasks are performed by running agent-based simulations in only a small fraction of spacial-temporal domain (i.e. the patches), which greatly saves the computational cost. Some of the content of Chapter 2 and 3 has been presented in the conference talk given in the 2010 AIChE Annual Meeting at Salt Lake City [59].
The second agent-based model studied in this thesis is an animal swarming model. In Chapter 4, aiming to extend the scope of the current EF approach, the DMAP technique introduced in the previous section is applied to the simulation data of this model to systematically identify the coarse variables of the system. The two dominant DMAP coarse variables characterize how the group of animals switches its direction of collective motion, with the two coordinates corresponding to the left-right and up-down directions of motion respectively. Based on the identified DMAP coarse variables and using the EF approach, a reduced stochastic differential equation (SDE) model is successfully constructed. Using the reduced SDE model, the associated coarse rare event is efficiently studied, and the expensive agent-based simulations are avoided. With more effort, the DMAP based approach has the potential to become a common method in reducing the high computational cost for agent-based simulations.

In Chapter 5 as an interesting side project, a reaction network model for a biologically important gene signaling pathway - the extracellular signal-regulated Kinase (ERK) pathway - is studied. This chapter is based on work published in collaboration with Prof. Stanislav Y. Shvartsman and Prof. Ioannis G. Kevrekidis in [60]. Some of the content has also been presented in the 2011 AIChE Annual Meeting at Minneapolis [61]. The ERK protein can use the same domains to interact with phosphatases, which dephosphorylate and deactivate ERK, and with substrates, which connect ERK to its downstream effects. As a consequence, substrates can compete with phosphatases and control the level of ERK phosphorylation. Using a combination of numerical solution of the model and its steady-state bifurcation analysis, a parameter regime which supports large amplitude limit cycles is successfully identified. In Chapter 5 we will discuss the details of the modeling work, the associated interesting separation of time scales for the identified oscillatory solutions, as well as the mechanism of the identified oscillations.
Chapter 2

Equation-Free analysis for an agent-based financial market model.

2.1 Introduction

Our first illustrative example is a financially motivated agent-based model based on the work of Omurtag and Sirovich [74]. It simulates the behavior of a large population of investors who buy or sell on the basis of information from the external environment and the action of other agents.

This chapter focuses on applying the EF approach to study the macroscopic level dynamical properties of the system. Section 2.3 presents the coarse bifurcation analysis for this agent-based model. First we identify a set of coarse variables and then we construct the coarse time stepper. Based on the coarse time stepper, we construct the coarse bifurcation diagram of the agent density distribution with respect to a bifurcation parameter quantifying the strength of the interaction between agents. Coarse bifurcation diagrams are constructed in two different representative parameter
regimes (i.e. the so called “symmetric” and “asymmetric” parameter regimes). We observe a pitchfork bifurcation in the “symmetric” parameter regime and a saddle-node type bifurcation in the “asymmetric” parameter regime.

To validate the results of the coarse bifurcation analysis, and to gain a deeper understanding of the underlying dynamics, in section 2.4 we perform a detailed bifurcation analysis for a continuum approximation (PDE) of the discrete agent-based model. Based on the analytical solution of the continuum approximation, a two-parameter bifurcation diagram is constructed. Then, for a selected parameter regime, we construct representative one-parameter bifurcation diagrams. By exploring a wide parameter space, we observe several interesting phenomena: additional turning points at the unstable branch (in addition to the turning point we present in the previous work), a special triangle-like probability density profile and the so-called termination points (beyond which no steady state solution exits). It is also shown that the bifurcation diagram of the agent-based model (constructed using the EF approach) matches qualitatively with the one for the continuum approximation.

The remainder of this chapter demonstrates how the EF approach can be applied to efficiently study the stochastic behavior of agent-based models through a coarse rare event analysis. In the one-parameter coarse bifurcation diagram (saddle-node type) in the region far from the turning point, the problem appears effectively deterministic and the coarse level dynamics may be assumed to be governed by partial differential equations (PDE). However, in the region near the critical point, due to the stochasticity and also due to the closeness between the stable and unstable stationary states, the problem is better suited to modeling by stochastic differential equations (SDE). In addition, due to the stochastic nature of the microscopic agent-based simulations, the stable stationary states near the saddle-node turning point are actually metastable ones: the state of the population escapes from the region of attraction of the “stable” stationary state after random (long) time intervals. This escaping event
is the rare event in this problem. A question then arises - how often does this rare event take place? The answer comes from a coarse rare-event analysis which we will present in section 2.5 of this chapter.

To apply the EF approach to a general complex system problem, we need not know the macroscopic equations for the coarse-level attributes in closed form, but we at least need to know the right statistics (we call them coarse variables) of the complex system for which we can then write a deterministic or Markov-type reduced model. Although in many cases we may know such coarse variables from experience or physical intuition as in the example in Section 2.5, very often such intuition is not available. It is for such situations that recently developed data-mining techniques, which are capable of passing low-dimensional nonlinear manifolds through simulation data and simultaneously providing information on the dimensionality of such manifolds, can be most helpful. In section 2.5.5 of this chapter we demonstrate how such data-mining techniques (specifically, the DMAP technique) can help us detect the actual coarse-grained variables in terms of which the macroscopic equations may be written. We conduct the rare event analysis with the coarse variables detected by the DMAP technique. It is shown that the DMAP-based approach can give us comparable results to those from intuition-based coarse variables.

2.2 The agent-based financial market model

The agent-based model we study in this chapter and in chapter 3 is a financial market model initially described by Omurtag and Sirovich [74]. This model simulates the actions of buying and selling by a large population of interacting individuals in the presence of mimesis. In this model, each agent’s propensity to buy or sell is indicated by its preference state. The preference state $X_{i=1,2...N}(t)$ of each individual $i = 1, 2...N$ evolves with time in a one-dimensional artificial world $X_i(t) \in (-1, 1)$ according to two
coupled processes. The first one is the exponential decay of $X_i(t)$ towards the neutral state 0 at the constant decay rate $\gamma$. This decay mimics the assumption that each agent tends to gradually forget its current preference state and tends to be neutral in its preference of buying or selling. The second one is a stochastic process denoted by $I_i(t)$ which represents the effect of the incoming information to each preference state $X_i(t)$. $I_i(t)$ tells us three things: (1) when the incoming information arrives; (2) what kind of information (i.e. good or bad) it is; (3) how this information affects the state $X_i(t)$. The arrival of the incoming information is assumed to follow a Poisson distribution with average arrival frequency ($\nu^+ + \nu^-$), where $\nu^+$ denotes the average arrival frequency for good news and $\nu^-$ denotes the one for bad news. The average arrival frequencies are defined as:

$$\nu^\pm = \nu^\pm_{ex} + gR^\pm,$$

where the parameter $\nu^\pm_{ex}$ (or $\nu^-_{ex}$) represents the contribution of the external information (e.g. mass media news or opinions of stock market consultants) each individual draws from its environment. $R^+$ and $R^-$ are the “buy” and “sell” rates respectively and are defined as the number of buys or sells that happened in the market per unit time normalized by the total number of agents in the market (we assume there is a very large external pool, so that $R^+$ and $R^-$ do not need to be equal). The parameter $g$ is a feedback constant which tells us to what extent the average arrival frequency is determined by the buy (or sell) rate. Since the buy (or sell) rate is a collective property of the population (i.e. all the agents in the market), the second term of Eq. (2.1), $gR^\pm$, tells us how each individual agent’s preference state is affected by the collective behavior of the entire population. In this model, the “buy” and “sell” rates as well as the information arrival frequencies $\nu^+$ and $\nu^-$ follow discrete time evolu-
tion. They have fixed values within discrete time intervals called the time reporting horizon, and are only updated after each time reporting horizon.

The information that arrives has probability \( \nu^+/\left(\nu^+ + \nu^-\right) \) to be good news and probability \( \nu^-/\left(\nu^+ + \nu^-\right) \) to be bad news. When good news arrives, the value of \( X_i(t) \) is increased instantaneously by a fixed amount \( \epsilon^+ \) \( (\epsilon^+ > 0) \); similarly, when bad news arrives, \( X_i(t) \) is decreased instantaneously by a fixed amount \( -\epsilon^- \) \( (\epsilon^- < 0) \). If after a positive jump the value of \( X_i(t) \) exceeds the positive boundary (i.e., \( X_i(t) > 1 \)), then a “buy” is considered to occur and the number of buys for that time interval is increased by one. Similarly after \( X_i(t) \) crosses the negative boundary (i.e., \( X_i(t) < -1 \)), the number of “sells” is increased by one. In either case \( X_i(t) \) is set back to the neutral state (i.e., \( X_i(t) = 0 \)) after the number of buys or sells is updated. In this way, each individual agent’s decision can also affect the population’s collective behavior.

This discrete jump process and the previously mentioned exponential decay process together form the evolutionary rule for each agent’s preference state \( X_i \), which can be summarized as the following equation:

\[
dX_i = -\gamma X_i(t) dt + dI_i(t).
\] (2.2)

### 2.3 Coarse bifurcation and stability analysis

For the agent-based model introduced in the previous section, the microscopic variables are the preference states \( X_i(t), i = 1, 2, ..., N \) of the \( N \) agents in the market (in this case, \( N = 50000 \)) and the information arrival frequencies are \( \nu^+ \) and \( \nu^- \). That is, we have 50002 microscopic variables in total. At the macroscopic level we approximate the distribution of \( X_i(t) \) by approximating its cumulative distribution function (CDF) through a piecewise linear polynomial of 40 segments. The coarse-grained variables are the 39 percentile points \( X_j, j = 1, 2, ..., 39 \) and \( \nu^+ \) and \( \nu^- \) - we
have 41 coarse variables in total. Here the actual number of percentile points we use 
does not matter much, as we rely on the microscopic simulator to heal the discrep-
ancies caused by the approximation. After defining the microscopic and macroscopic 
variables, we proceed to the construction of the coarse time stepper which is the core 
of the EF approach. It has been established [46, 48, 47] that the coarse time stepper 
provides a bridge between microscopic (e.g. agent-based) modeling on the one hand 
and and macroscopic systems-level continuum numerical analysis on the other. The purpose of constructing this coarse time stepper is to bypass the derivation 
of the underlying system-level evolutionary equations which usually is an impossi-
ble or overwhelmingly difficult task. The coarse time stepper consists of two main 
procedures, lifting and restriction. In the lifting procedure, the microscopic variables 
(preference states $X_i(t_0)$ of the agents) are generated consistently with a set of given 
coarse variables (the percentile points $x_i(t_0)$ of the cumulative distribution function) 
and are used to initiate the microscopic simulation through the agent-based simulator 
for a relatively small time horizon $\Delta t$. At the end of this simulation the restriction 
procedure is performed on the microscopic variables $X_i(t_0 + \Delta t)$ to obtain the macro-
scopic coarse variable $x_i(t_0 + \Delta t)$. We have thus constructed the coarse time stepper 
$x_i(t_0 + \Delta t) = \Phi_{\Delta t}(x_i(t_0))$.

Next, the coarse-grained steady states are computed by wrapping the Newton-
GMRES scheme as a shell around the coarse time stepper. Continuation around the 
turning point is accomplished by coupling the fixed point algorithm with the pseudo-
arc-length continuation method. The critical eigenvalues corresponding to each coarse 
fixed point are computed by implementing the Arnoldi iterative eigensolver around the 
coarse timestepper. The dominant eigenvalues are used to determine the stabilities 
of each fixed point. This coarse bifurcation analysis is performed for two different 
parameter cases, the symmetric case ($\epsilon^+ = -\epsilon^- = 0.072$) and the asymmetric case 
($\epsilon^+ = 0.075, \epsilon^- = -0.072$). The corresponding coarse bifurcation diagrams are shown
The stationary states on the coarse bifurcation diagram were obtained as fixed points of the agent-based time stepper averaged over 1000 realizations. The time horizon used in this case is $\Delta t = 0.25 \times 4 = 1$. In the bifurcation diagrams the solid line corresponds to the branch with stable fixed points while the dotted one corresponds to the branch with unstable fixed points. The standard error for each fixed point is estimated based on 20 separate runs. The rightmost inset in 2.1(b) shows good agreement between fixed-point solutions obtained using the EF methodology and the ones generated from long time direct agent-based simulations. We need to note that the unstable steady state branches computed here can not, in general, be obtained from direct agent-based simulations.

The dynamics of the system reveals interesting phenomenological behavior at the chosen parameter values. For the asymmetric case, a saddle-node type bifurcation is observed. Starting the simulations with all agents at the neutral position ($X_i(t = 0) = 0$), the system approaches stable stationary states for low values of the influence factor $g$, while for larger values of $g$ the buying rate appears to eventually blow up. More interestingly, for the symmetric case, a pitchfork type bifurcation is observed. Starting the simulations with all agents at the neutral position ($X_i(t = 0) = 0$), the system approaches stable stationary states for low values of the influence factor $g$ while for larger values of $g$ either the buy or sell rate will eventually blow up - which one depends on the noise along the evolution process.

2.4 Bifurcation analysis for the continuum approximation

In this section we study a continuum approximation (PDE) of the agent-based model. An extended bifurcation and stability analysis is performed based on this continuum approximation.
Figure 2.1: Coarse bifurcation diagrams constructed using the EF approach. (a) the symmetric case ($\epsilon^+ = -\epsilon^- = 0.072$) and (b) the asymmetric case ($\epsilon^+ = 0.075, \epsilon^- = -0.072$). The solid line corresponds to the branch with stable fixed points while the dotted one corresponds the branch with unstable fixed points. The standard error for each fixed point is estimated based on 20 separate runs. The stability of each branch is determined based on the algebraically largest eigenvalues calculated by applying a matrix-free Arnoldi algorithm around the coarse timestepper. The rightmost inset compares the coarse fixed points obtained using the EF methodology with the ones from long term microscopic simulations.
approximation in order to acquire a deeper understanding of the underlying dynamics as well as to validate the results of our previous coarse bifurcation analysis.

2.4.1 Population dynamics and the Fokker-Planck approximation

It is possible to achieve a concise description of the dynamics of a large assembly of agents by keeping track of only the number of agents in each preference state rather than the preference state of every agent in the population. This is accomplished by deriving an equation for $\rho(x,t)$, the probability density of agents in preference state $x$ at time $t$. We can think of $\rho(x,t)$ as the density of agents averaged over a large number of replicas of the population. From the continuity equation, Omurtag and Sirovich [74] derived the following population equation which describes the evolution of the probability density $\rho(x,t)$.

$$
\frac{\partial \rho}{\partial t} = -\gamma \frac{\partial (x \rho)}{\partial x} + \sum_{k=+,-} \nu^k (\rho(x - \epsilon^k, t) - \rho(x, t)) + (R^+ + R^-) \delta(x). \tag{2.3}
$$

For small $\epsilon^k$ by expanding $\rho(x - \epsilon^k, t)$ about $\rho(x, t)$ in terms of $\epsilon^k$ and truncating terms higher than second order in $\epsilon^k$, one obtains the following Fokker-Planck type approximation:

$$
\frac{\partial \rho}{\partial t} = \gamma \frac{\partial ((x - p) \rho)}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 \rho}{\partial x^2} + (R^+ + R^-) \delta(x), \tag{2.4}
$$

where $p = (\nu^+ \epsilon^+ + \nu^- \epsilon^-)/\gamma$ and $\sigma^2(t) = \nu^+(\epsilon^+)^2 + \nu^-(\epsilon^-)^2$. The fact that agents leaving at $x = \pm 1$ are brought back to the origin dictates that $\rho(x = \pm 1, t) = 0$. In order to conserve probability, the buy and sell rates are defined as $R^\pm = \mp \frac{1}{2} \sigma^2 \frac{\partial \rho}{\partial x} \mid_{x=\pm 1}$. By setting the time derivative to zero, we obtain the analytical
Figure 2.2: Two-parameter bifurcation diagram generated by MATCONT based on analytical solutions. The two active parameters are the positive jump size $\epsilon^+$ and the feedback factor $g$. The negative jump size $\epsilon^-$ is fixed at the value of -0.072.

solution for the stationary states of Equation (2.4) in the following form

$$\rho_L(x) = f_1(x; \epsilon^+, \epsilon^-, \nu e^x, \nu e^- x, \gamma, g, R^+, R^-)$$  \hspace{1cm} (2.5)$$
$$\rho_R(x) = f_2(x; \epsilon^+, \epsilon^-, \nu e^x, \nu e^- x, \gamma, g, R^+, R^-)$$  \hspace{1cm} (2.6)$$
$$R^+ = h_1(\epsilon^+, \epsilon^-, \nu e^x, \nu e^- x, \gamma, g, R^+, R^-)$$  \hspace{1cm} (2.7)$$
$$R^- = h_2(\epsilon^+, \epsilon^-, \nu e^x, \nu e^- x, \gamma, g, R^+, R^-).$$  \hspace{1cm} (2.8)$$

The detailed expressions for $f_1, f_2, h_1, h_2$ are given in the Appendix at the end of this chapter. By inserting the above set of nonlinear equations into sophisticated continuation software such as AUTO/MATCONT [21, 22, 55] we can perform the bifurcation analysis for the Fokker-Planck approximation (Eq.(2.4)) of the continuous model. Fig. 2.2 shows the two-parameter bifurcation diagram generated by MAT-
CONT based on the analytical expressions (2.5) - (2.8). The two free parameters in Fig.2.2 are the positive jump size $\epsilon^+$ and the feedback factor $g$. We fix the negative jump size $\epsilon^-$ at $-0.072$. A cusp point appears when the positive jump size equals the negative one.

Next, we select two representative regimes in the two-parameter bifurcation diagram to perform one-parameter bifurcation analysis. In addition to $\epsilon^-$, we also fix $\epsilon^+$ leaving the feedback constant $g$ the only free parameter for this one-parameter bifurcation analysis. We study two representative cases – the symmetric and the asymmetric case respectively. In the symmetric case, $\epsilon^+$ and $\epsilon^-$ have equal magnitudes (in particular, we set $\epsilon^+ = -\epsilon^- = 0.072$); while in the asymmetric case their magnitudes are not equal ($\epsilon^+ = 0.075$, $\epsilon^- = -0.072$). Fig. 2.3(a) and Fig. 2.3(b) show the three dimensional one-parameter bifurcation diagram for the symmetric and asymmetric case respectively. The diagrams have three dimensions because we are plotting one free parameter versus two observable variables. The first observable variable is the mean preference state $\bar{x}$ (plotted in y-axis), the second is the quadratic mean of the buy and sell rates $[(R^+)^2 + (R^-)^2]/2$ (plotted in z-axis). Fig. 2.4(a) justifies the stabilities of various branches with the plots of the dominant eigenvalues of the Jacobian matrix associated with the discretized PDEs (with discretization in a sufficient number of bins). Fig. 2.4(b) shows the probability density profiles of the preference states at representative parameter values. Fig. 2.5 shows that the discretized PDE solution converges to the analytical solution. Fig. 2.5(a) and Fig. 2.5(b) indicate that in the region of relatively small $g$ values, there exist a pitchfork bifurcation for the symmetric case and a saddle-node type bifurcation for the asymmetric case. These observations are consistent with the results of the previous coarse bifurcation analysis for the agent-based model. The branch point for the Pitchfork bifurcation occurs when $g$ is around 84; while the turning point for the saddle-node bifurcation occurs when $g$ is around 42. For the symmetric (asymmetric) case, by
One free parameter: $g$ (x-axis)
Two observables:
1. Mean preference state (y-axis)
2. $[(R^+)^2 + (R^-)^2]/2^{1/2}$ (z-axis)

Figure 2.3: Three dimensional one-parameter bifurcation diagrams constructed based on the analytical solutions for (a) the asymmetric case ($\epsilon^+ = 0.075$, $\epsilon^- = -0.072$) and (b) the symmetric case ($\epsilon^+ = -\epsilon^- = 0.072$). The active parameter is the feedback factor $g$. The two observable quantities in the $y$ and $z$ coordinates are the mean preference state $\bar{x}$ and the quadratic mean of the buy and sell rate $[(R^+)^2 + (R^-)^2]/2^{1/2}$ respectively.
Figure 2.4: Bifurcation results for the asymmetric case ($\epsilon^+ = 0.075, \epsilon^- = -0.072$): (a) stability analysis for various fixed point branches. For each branch the top eigenvalues of the discretized PDE system at various fixed points are plotted. (b) Probability density profiles of the preference states at different representative parameter values.
Figure 2.5: Comparison between the analytical solution and discretized PDE solutions for (a) symmetric case, and (b) asymmetric case. Discretized PDE solution with discretization of different number of bins (N=501 in red dash dot and N=101 in green dash) is shown to converge to the analytical solution (in blue solid line). Detailed probability density profiles are also compared at some specific locations of different bifurcation branches to show the convergence.
extending the middle (lower) unstable branch to much greater \( g \) values, we observe an additional turning point which occurs when \( g \) is around 276 (248). The turning point can be seen more clearly in the side-view (i.e. in the \( x - z \) plane). As Fig. 2.4(a) shows, after passing the turning point the number of eigenvalues greater than zero increased from one to two, meaning the solution becomes even more unstable. The bifurcation diagrams also indicate that for values of the parameter \( g \) greater than the value at the turning point there exists no steady state solution, not even an unstable one. More interestingly (see Fig. 2.4), as the unstable fixed point branch goes beyond the additional turning point, the probability density profile of the agents’ preference states changes from a shape characteristic of the normal distribution to a triangular shape. This is because in the region close to and beyond the additional turning point the arrival frequencies \( \nu^+ \) and \( \nu^- \) both have a very large magnitude which leads to a large \( \sigma^2 \) (see Eq. (2.4)) value and a small \( p \) value at the same time. These two facts together result in the triangular shape. At these locations, it appears the buy/sell rates grow without bound.

By extending the unstable branches toward the left-hand side of the parameter space we observed some “termination points” – the points at which the unstable branches terminate. We found that the termination point for both symmetric and asymmetric cases is located between \( g = 13 \) and \( g = 14 \). For parameter values less than the critical termination value, there exists no unstable steady state solution.

Overall, by comparing Fig. 2.5 and Fig. 2.1 we can see that the coarse bifurcation diagrams computed through the coarse timestepper are qualitatively comparable but quantitatively different from the ones computed by PDE approximation. The discrepancies are most likely caused by the bias introduced in the continuum approximations.
2.5 Coarse rare event analysis

For the asymmetric case of the agent-based problem, in the region close to the coarse saddle-node bifurcation point the stable steady state is quite close to its corresponding unstable steady state for each parameter value \( g \). Due to this fact and also due to the stochasticity of the process itself, a system starting in the region of attraction of the stable steady state will eventually escape from that region and become unstable after a long enough time. An interesting question naturally arises - how long (on average) does it take for this escape event to happen at a particular value of the parameter \( g \)? Although we may run the agent-based code repeatedly to get the statistics for the escape times and then calculate the average value, that would be too costly. In this section we will apply the EF approach to perform an “economical” coarse-grained rare event analysis to estimate the escape time. We will show that the equation-free approach not only saves a significant portion of time (compared with direct simulations) but also gives a result that is reasonably close to the actual one from costly direct simulations.

This section starts with a brief discussion about the relevant theoretical background. Then, an intuition-based 1D coarse reaction coordinate is proposed, followed by a validation of Fokker-Planck equation assumptions along this reaction coordinate. Finally, the rare event analysis results are presented.

2.5.1 Theoretical background

For a general one-dimensional stochastic process \( \psi(t) \), the evolution of the probability density \( P(\psi, t) \) of \( \psi \) obeys the following integral equation [80]:

\[
P(\psi, t + \tau) = \int W(\psi, t + \tau | \psi', t) P(\psi', t) d\psi',
\]

(2.9)
where $W(\psi, t + \tau | \psi', t)$ is the transition probability from point $\psi'$ at time $t$ to point $\psi$ at time $(t + \tau)$. The differential form of this equation, known as the Kramers-Moyal expansion, is as follows:

$$\frac{\partial P(\psi, t)}{\partial t} = \sum_{n=1}^{\infty} (-\frac{\partial}{\partial \psi})^n D^{(n)}(\psi, t) P(\psi, t), \tag{2.10}$$

where

$$D^{(n)}(\psi, t) = \frac{1}{n!} \lim_{\tau \to 0} \frac{1}{\tau} \langle [\xi(t + \tau) - \xi(t)]^n \rangle |_{\xi(t) = \psi} \tag{2.11}$$

are the differential moments of the transition probability $W$. The angular brackets here denote ensemble averaging and $\xi$ denotes a realization of the stochastic process. At the starting point $t$ it has a $\delta$-function distribution, $\xi(t) = \psi$.

If the process is Gaussian and Markovian, then all the higher order terms other than the first two in Eq. (2.11) will vanish [37]. The Kramers-Moyal expansion Eq. (2.10) is then reduced to the following useful differential equation called the forward Fokker-Planck equation:

$$\frac{\partial P(\psi, t)}{\partial t} = [-\frac{\partial}{\partial \psi} v(\psi, t) + \frac{\partial^2}{\partial \psi^2} D(\psi, t)] P(\psi, t). \tag{2.12}$$

Furthermore, if the stochastic process is temporally homogeneous (i.e. if it is invariant with respect to the shift in time) which is true for processes without external time-dependent forcing, then the forward Fokker-Planck equation can be written as follows:

$$\frac{\partial P(\psi, t)}{\partial t} = [-\frac{\partial}{\partial \psi} v(\psi) + \frac{\partial^2}{\partial \psi^2} D(\psi)] P(\psi, t) \tag{2.13}$$

where $v(\psi) \equiv D^{(1)}(\psi)$ is called the drift coefficient, and $D(\psi) \equiv D^{(2)}(\psi)$ is called the diffusion coefficient. The drift and diffusion coefficients are the two characterizing functions which completely determine the Markov process $\psi(t)$ in the stochastic sense [37]. One way these two functions determine the process $\psi(t)$ is through the forward
Fokker-Planck equation (2.13). The forward Fokker-Planck equation is useful in the sense that, with appropriate initial and boundary conditions, it uniquely determines the Markov state density function \( P(\psi, t|\psi_0, t_0) \) of the process. However, it does not describe the evolution of \( \psi(t) \) itself which we may also want to study. Fortunately, the drift and diffusion coefficients also provide us a way to study the evolution of \( \psi(t) \) itself through the following stochastic differential equation, called the Langevin equation:

\[
\psi(t + dt) = \psi(t) + D^{1/2}(\psi(t))dW(dt) + A(\psi(t))dt,
\]

where, \( A(\psi(t)) \) and \( D(\psi(t)) \) are the drift and diffusion coefficients, \( dW(dt) \) is a normal random variable with mean 0 and variance \( dt \). So far we can see that as long as we obtain the drift and diffusion coefficients, we can essentially reconstruct both the forward Fokker-Planck equation and the associated Langevin equation. After the reconstruction of these two very useful analytical tools, several global characteristics of the system will be available, such as the potential function \( G(\psi) \) and the mean escape time \( \tau \) (the mean escape time is the average time for the system to escape from a potential well). The potential function \( G(\psi) \) can be obtained from the equilibrium probability distribution which is the solution of the steady-state Fokker-Planck equation,

\[
0 = -\frac{d}{d\psi}[A(\psi)P_s(\psi)] + \frac{d^2}{dx^2}[D(\psi)P_s(\psi)].
\]

Substituting the ansatz \( P_{eq}(\psi) \propto \exp[-G(\psi)] \) into Eq. (2.15), we obtain the equation for the potential function,

\[
G(\psi) = -\int \frac{v(\psi')}{D(\psi')}d\psi' + lnD(\psi) + const.
\]
Another function which is similar but different from the potential function is defined as follows:

$$G_0(\psi) = - \int \frac{v(\psi')}{D(\psi')} d\psi' + \text{const.}$$  \hspace{1cm} (2.17)$$

We call $G_0(\psi)$ the pseudo-potential function to differentiate it from the actual potential function $G(\psi)$. $G_0(\psi)$ will be identical to $G(\psi)$ when the diffusion coefficient $D(\psi)$ is not constant. Although it is not the actual potential function, $G_0(\psi)$ finds its use in the mean escape time calculations. In fact, the mean escape time from a potential well can be written as [37]:

$$\tau(\psi_0 \rightarrow \psi_e) = \int_{\psi_0}^{\psi_e} dy e^{G_0(y)} \int_{\infty}^{y} dz \frac{e^{-G_0(z)}}{D(z)},$$  \hspace{1cm} (2.18)$$

where, as illustrated in Fig. 2.6, $\psi_0$ is a starting point inside the well and $\psi_e$ is a point beyond the potential well where the probability density is nearly zero. When the y-integrand and z-integrand contribute significantly only for small regions around $\psi_e$.
and \( \psi_0 \) respectively, the analytical approximation of Eq. (2.18) is available as follows:\cite{37}:

\[
\tau(\psi_0 \rightarrow \psi_b) \approx 2\pi \left| \frac{D(\psi_b)}{D(\psi_0)v'(\psi_b)(v'(\psi_0) - D''(\psi_0))} \right|^{\frac{1}{2}} e^{(G_0(\psi_b) - G_0(\psi_0))},
\]

(2.19)

where \( \psi_b \) is the peak of the barrier adjacent to the potential well (see Fig. 2.6). For a constant diffusion, Eq. (2.19) further simplifies to the well-known Kramers’ escape time:

\[
\tau(\psi_0 \rightarrow \psi_b) \approx 2\pi e^{[G_0(\psi_b) - G_0(\psi_0)]} \sqrt{G''_0(\psi_0)G''_0(\psi_b)}
\]

(2.20)

### 2.5.2 Coarse-grained one-dimensional reaction coordinate

In this section we will construct an intuition-based one-dimensional reaction coordinate in the parameter space where the coarse rare event analysis will be performed. After the construction of this reaction coordinate, an effective Fokker-Planck equation will be constructed and the mean escape time will be calculated in the following sections.

In our rare event analysis there are three different levels of description, namely, the microscopic level, the intermediate coarse level and the coarse level. The microscopic level description refers to the level the agent-based simulator operates. At this level the microscopic variable \( X(t) \) is defined as the input/output variable of the agent-based simulator. This microscopic variable is of 25002 dimensions, 25000 of which represent the preference states of the 25000 agents we specified for the system, and the remaining two represent the positive and negative information arrival frequencies (\( \nu^+ \) and \( \nu^- \) respectively). The intermediate coarse level refers to the level the coarse timestepper is working on. It bridges the microscopic level and the coarse level which we will introduce shortly. The intermediate coarse variable \( x(t) \) is defined
as the input/output variable of the coarse timestepper. It has 41 dimensions, 39 of which represent the 39 percentile points of the probability density distribution of the agents’ preference states, the remaining two are the positive and negative information arrival frequencies. Ultimately, we define the coarse level description as the system/population level description which measures the collective behavior of the entire agent population. We refer to the variable associated with this level as the coarse variable and denote it $\psi(t)$. This coarse variable is a one-dimensional variable varying along a one-dimensional reaction coordinate with each coordinate value representing a particular state of the population at time $t$. The construction of this one-dimensional reaction coordinate is described in the following paragraph.

We study the rare event in the parameter space just before the saddle-node turning point. In this parameter region, the unstable eigenvector of the Jacobian matrix of the coarse timestepper at the saddle is close to the dominant stable eigenvector (i.e. the eigenvector corresponding to the eigenvalue with largest real part) of the Jacobian matrix of the coarse timestepper at the node. These two eigenvectors get closer and closer to each other as the parameter $g$ approaches the turning point value. Eventually as $g$ reaches the turning point these two eigenvectors coincide and form the center manifold of the turning point. Therefore, in the parameter space slightly before the saddle-node turning point, the collective behavior of the agent population can be approximately described as moving along the vector pointing from the saddle $x_{saddle}$ to the node $x_{node}$. This assumption implies that all other variables quickly approach this slow, attracting, one-dimensional manifold; the statistics of the simulation quickly become functions of one coarse variable. Based on this, the ultimate one-dimensional reaction coordinate $\psi$ can be defined as follows:

$$\psi = (x_{node} - x)^T \frac{(x_{saddle} - x_{node})}{\|(x_{saddle} - x_{node})\|}$$  (2.21)
2.5.3 Validity of the Fokker-Planck equation assumptions

Based on the coarse reaction coordinate proposed in the last section, we will briefly validate two important assumptions, which must be satisfied before we can employ an effective Fokker-Planck equation: that the stochastic process is Gaussian and Markovian.

The process $\psi(t)$ is demonstrated qualitatively to be Gaussian in the following way: (1) by initializing many (in this case, 20000) separate microscopic configurations corresponding to the same coarse coordinate value $\psi_0$, we construct a pseudo-delta function at some specific coarse coordinate value $\psi_0$; (2) we then propagate this pseudo-delta function by running the microscopic simulator for each of those 20000 microscopic initial configurations for some time $t$, and during this process we record the microscopic configurations at different transient times $t_i$ with $i = 1, 2, 3...n$; (3) at each $t_i$ by restricting the microscopic configurations to their corresponding coarse coordinate values $\psi_j(t_i)$, $j = 1, 2, 3, ...20000$, we construct the evolutionary probability density distributions of the coarse variable $\psi(t)$ at different times $t_i$. As Fig. 2.7 shows, the propagated distributions of the coarse variable are well fitted by Gaussian distributions, which validates the assumption that the stochastic process $\psi_i(t)$ is Gaussian. To show that the process is Markovian is equivalent to proving that the noise term (the second term) of the Langevin Eq. (2.14) has zero-correlation time (i.e. the autocorrelation function has value zero for any finite correlation time). For a process $X(t)$ with mean $\mu$ and variance $\sigma^2$, its autocorrelation function is computed as follows,

$$C(t) = \frac{E[(X_\tau - \mu)(X_{\tau+t} - \mu)]}{\sigma^2}, \quad (2.22)$$

where “$E$” denotes the expected value operator. Manipulation of the Langevin equation shows that the correlation time of the noise is proportional to the correlation
time of $d\hat{\psi}(t)/dt$, where
\[ \hat{\psi}(t) = \psi(t) - \langle \psi(t) \rangle \]  
(2.23)
is the fluctuation of $\psi(t)$. Therefore, to show that the noise term has zero correlation time we just need to show that $d\hat{\psi}(t)/dt$ has zero correlation time. The term $d\hat{\psi}(t)/dt$ is estimated using the forward differences,
\[
\left. \frac{d\hat{\psi}(t)}{dt} \right|_{t=t_i} \approx \frac{\hat{\psi}(t_{i+1}) - \hat{\psi}(t_i)}{\Delta t},
\]  
(2.24)
where $\Delta t$ is the step size of the coarse time stepper. A normalized autocorrelation function of $d\hat{\psi}(t)/dt$ (and therefore of the noise) for the initial coarse variable value
Figure 2.8: Autocorrelation function $C(t)$ of $d\dot{\psi}(t)/dt$ (and therefore of the noise term in the Langevin equation) computed using Eq. (2.22); this function is normalized so that $C(0) = 1$. The shown function is computed for the initial coarse variable value at $\psi_0 = -0.25$ and is typical for the coarse coordinate value between $\psi_{node}$ and $\psi_{saddle}$ which is the region we are interested in for the rare event analysis.

$\psi_0 = -0.25$ is computed using Eq. (2.22) and shown in Fig. 2.8. This indicates that the coarse stochastic process $\psi(t)$ is Markovian.

2.5.4 Mean escape time results – based on the empirical coordinate

In this section we first show how the drift and diffusion profiles along the coarse reaction coordinate (see Eq. (2.21)) can be constructed through the short time simulations of the agent-based code. Then based on the constructed drift and diffusion functions we calculate the mean escape time.

We discretize the coarse reaction coordinate using 30 bins of equal size. The grid points range from $\psi = -3.3$ to $\psi = -1.2$ in order to cover the coordinate space between the saddle and node (and slightly beyond them). Since the drift and diffusion
coefficients are actually the first two moments of Eq. (2.11), at each grid point $\psi_i$, they can be approximated by the following equations:

$$\nu(\psi_i) \approx \frac{1}{\Delta t} \langle \xi(t_0 + \Delta t) - \xi(t_0) \rangle|_{\xi(t_0)=\psi_i}, \quad (2.25)$$

$$D(\psi_i) \approx \frac{1}{2\Delta t} \langle [\xi(t_0 + \Delta t) - \xi(t_0)]^2 \rangle|_{\xi(t_0)=\psi_i}, \quad (2.26)$$

where the angular brackets denote ensemble averaging, and $\xi$ denotes a realization of the stochastic process with a $\delta$-function distribution at the starting point $t_0$, $\xi(t_0) = \psi_i$. Eq. (2.25) and Eq. (2.26) indicate that the drift and diffusion coefficients can be estimated through multiple parallel short time simulations.

In practice, the initial delta function is approximated in the following way: for each coarse grid point $\psi_i$, first we find its corresponding intermediate coarse variable $x_i$ along the vector pointing from the saddle $x_{saddle}$ to the node $x_{node}$ using Eq. (2.21); then we initialize 4000 copies of microscopic configurations $X_j, j = 1, 2, \ldots 4000$ consistent with the intermediate coarse variable $x_i$ by randomly allocating agents’ preference states between each two adjacent percentile points stored in $x_i$.

After the initialization, we run the agent-based simulator for each copy of the 4000 microscopic configurations for a short time $\Delta t$ which is equal to the time horizon of the coarse time stepper; we then restrict the evolved microscopic variables back to the corresponding intermediate coarse variables $x_j(t_0 + \Delta t)$, and finally further restrict each intermediate coarse variable to the corresponding coarse coordinate values $\xi_j(t_0 + \Delta t)$. Now we are able to apply Eq. (2.25) and Eq. (2.26) to estimate the drift and diffusion coefficients at grid point $\psi_i$. The above procedures are repeated for all the grid points to construct the drift and diffusion profiles along the coarse reaction coordinate. After that the profiles for the effective potential function $G(\psi)$ and the pseudo-potential function $G_0(\psi)$ can be approximated by numerical integration using Eq. (2.16) and Eq. (2.17). The results are shown in Fig. 2.9. Based on these
constructed characteristic functions, we can apply the equations mentioned at the end of section 2.5 to estimate the mean escape time. Because Eq. (2.18) is computationally more expensive to apply and requires data beyond the potential barrier which are difficult to properly estimate, we decided to apply Eq. (2.5.4) for our coarse grained calculation. We specify the starting state $\psi_0$ for the rare event as the bottom state at the effective potential function (as Fig. 2.9(c) shows); the mean escape time obtained using Eq. (2.5.4) is $\tau = 788$, which compares reasonably well with the mean escape time $\tau = 599$ observed from the full-scale agent-based simulations (see Fig. 2.10). Fig. 2.9(b) shows that the variation of the diffusion coefficient is relatively small between $\psi_b$ and $\psi_0$. If we treat the diffusion coefficient as constant with $D \approx (D(\psi_0) + D(\psi_b))/2$, and then apply Kramers’ equation Eq. (2.20), we obtain the
mean escape time $\tau = 725$, which is in even better agreement with the equilibrium result. In terms of computational cost, the full-scale simulation requires $5.28 \times 10^5$ steps of agent-based simulation, while the coarse grained approach only takes $5.28 \times 10^5$ steps, as little as 3.2% of the effort required for the full-scale simulation.

### 2.5.5 Data mining tool - DMAP - based approach

In the previous section, we constructed the coarse reaction coordinate based on physical intuition about the problem. However, such intuition is not always available. To overcome this limitation, in this section we will explore the recently developed data mining tool - DMAP [15] [16] - to systematically identify the coarse variables. This data-mining based approach has two main advantages. First, it is a generic approach as it only requires the dataset generated by the simulations - prior knowledge or intuition about the macroscopic level dynamics is not required. Second, this approach...
can suggest to not only what the underlying coarse variables are, but also how many
of them are required to sufficiently describe the coarse level dynamics.

The details of the DMAP construction were discussed previously in section 1.2. Here we recall the main steps of the technique. The DMAP technique is based on
the construction of a Markov matrix describing a random walk over a data set, where
the probability of transitioning from one data point to another is based on a pairwise
similarity metric. The similarity kernel is defined as follows:

\[
W(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\varepsilon}\right).
\] (2.27)

Next, we define the diagonal matrix \(D\) by,

\[
D_{ii} = \sum_{j=1}^{N} W_{ij},
\] (2.28)

and then we compute the first few right eigenvectors and eigenvalues of the stochastic
matrix:

\[
A = D^{-1}W.
\] (2.29)

We are interested in the set of truncated real eigenvalues, \(\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_M \geq 0\)
(usually there is a spectral gap between the \(M^{th}\) and the \((M + 1)^{th}\) eigenvalue),
with corresponding eigenvectors \(\{\psi_j\}_{j=1}^{M}\). Since \(A\) is row-stochastic, \(\lambda_1 = 1\) and
\(\psi_1 = [1...1]^T\). The \((M - 1)\)-dimensional representation of a particular \(d\)-dimensional
data vector, \(x_i\), is given by the DMAP as follows,

\[
x_i \mapsto (\psi_2(i), \psi_3(i), ..., \psi_M(i)).
\] (2.30)

Since usually \(M \ll d\), the first several dominant eigenvectors of this Markov ma-
trix reveal the underlying low-dimensional manifold of the original data set. The
Euclidean distance in the diffusion map space corresponds to the diffusion distance (proximity) in the original space.

In the DMAP-assisted study of the agent-based problem, the first step is to generate data for the construction of the Markov matrix. We perform multiple agent-based simulations starting from a microscopic initial condition corresponding to the coarse saddle solution. We then set two “stop” points: one at the microscopic configuration corresponding to \( x_{saddle} + \delta_1(x_{saddle} - x_{node}) \); and the other at the microscopic configuration corresponding to \( x_{node} - \delta_2(x_{saddle} - x_{node}) \) (both \( \delta_1 \) and \( \delta_2 \) are small positive numbers). Beyond either of these two points we stop and re-execute the simulations. In total this sampling process takes 893 steps of the microscopic agent-based simulations in this case, which does not cause much additional computational burden.

Once the microscopic configurations \( (X_i) \) are collected, we apply the previously defined lifting procedure to compute the intermediate coarse variables (the percentile points, \( x_i \)). Using these intermediate coarse variables as the raw data, and applying the procedures described above and in section 1.2, we construct the Markov matrix and compute the dominant DMAP eigenvectors. The plot for the first and second dominant DMAP eigenvectors is shown in Fig.2.11. It suggests that the top two dominant eigenvectors actually describe the same one-dimensional space. Fig.2.12 shows the plot of the top 50 dominant eigenvalues. We observe a spectral gap between the first dominant eigenvalue \( (\lambda_2) \) and the remaining eigenvalues \( (\lambda_1 \text{ is the trivial eigenvalue with value 1}) \). That single dominant eigenvalue confirms the existence of an underlying one-dimensional slow manifold for the dynamics of our coarse grained system. In fact, the dominant slow time scale in this case corresponds to the time scale of the rare event, while the other fast time scales correspond to the relaxation time scales in this case. Fig.2.13 compares several possible coarse reaction coordinates. It shows a good correlation between the DMAP generated reaction coordinate and the
Figure 2.11: First two dominant eigenvectors of DMAP Markov matrix.

Figure 2.12: Top 50 dominant eigenvalues.
intuition-based reaction coordinate. It also shows that the correlation will be poor if we compare the intuition-based coordinate with some “not so good” coarse property like the positive information arrival frequency $\nu^+$. We applied this DMAP generated coordinate to the same data set (the short time simulation data) we used in the previous section to construct the characteristic function profiles $-\nu(\psi_{dmap})$, $D(\psi_{dmap})$, $G(\psi_{dmap})$, and $G_0(\psi_{dmap})$. We then calculate the mean escape times based on these profiles. Using Eq. (2.19), we obtain $\tau = 862$; using Kramers’ Eq. (2.20) and assuming constant diffusion, we obtain $\tau = 596$, both of which are reasonably close to the results given by the intuition-based coordinate as well as the full-scale agent-based simulations.
2.6 Summary

In this chapter, we applied the EF approach and performed coarse bifurcation analysis as well as coarse rare event analysis for an agent-based financial market model based on the work of Omurtag and Sirovich. Coarse bifurcation diagrams are successfully constructed in two interesting parameter regimes (i.e. the so called “symmetric” and “asymmetric” parameter regimes). A saddle-node type bifurcation is observed in the “asymmetric” parameter regime, and a pitchfork bifurcation is observed in the “symmetric” parameter regime. The unstable steady states branches identified using this approach cannot be found through the conventional method (e.g. direct agent-based simulations). To validate the results of the coarse bifurcation analysis, and to gain a deeper understanding of the underlying dynamics, we performed an extended bifurcation analysis for the continuum approximation of the discrete agent-based model. Two-parameter as well as 3D one-parameter bifurcation diagrams were constructed for the continuum approximation. Several interesting phenomena were observed, such as the additional turning point at the unstable branch, a special triangle probability density profile, and so-called termination points.

In the coarse rare event analysis, an intuition-based coarse reaction coordinate was computed in a parameter regime where a stochastic rare event arises. The underlying effective FP equation was constructed from ensembles of short bursts of microscopic simulations with judiciously chosen initial conditions. The resulting mean escape time for the rare event compares reasonably well to the actual mean escape time from costly long-time agent-based simulations. A data-mining tool (DMAP) based approach was also explored and found to produce results consistent with those from the intuition-based approach. Still, in this preliminary study, the initial configurations required to perform the short bursts of simulations were generated with the assistance of physical intuition about the system. In the future, additional effort will be required to construct a systematic lifting procedure which can reliably generate microscopic
configurations corresponding to a given DMAP coordinate value without the necessity of prior physical intuition of the system.

2.7 Appendix: Steady State Solution of the Continuum Approximation

We use subscripts ‘L’ and ‘R’ to denote the properties at the two sides of the spacial domain - \([-1, 0]\) and \([0, 1]\) - respectively.

The boundary conditions of the continuum model Eq.(2.4) are:

\[
\begin{align*}
\rho_L(t, -1) & = 0 \\
\rho_R(t, 1) & = 0 \\
\rho_L(t, 0) & = \rho_R(t, 0) \\
R^- & = \frac{1}{2}\sigma^2 \frac{\partial \rho}{\partial x}(t, -1) \\
R^+ & = -\frac{1}{2}\sigma^2 \frac{\partial \rho}{\partial x}(t, 1)
\end{align*}
\]

The integral condition reads,

\[
\int_{-1}^{1} \rho(x)dx = 1
\]

At steady state, the continuum model is written as follows,

\[
0 = \gamma \frac{\partial((x - p)\rho)}{\partial x} + \frac{1}{2}\sigma^2 \frac{\partial^2 \rho}{\partial x^2}
\]

Integrate Eq. (2.37) in space,

\[
(x - p)\rho + \frac{1}{2} \left( \frac{\sigma^2}{\gamma} \right) \frac{\partial \rho}{\partial x} = c
\]
Let
\[ \beta = \frac{\sigma^2}{\gamma} \]  
(2.39)

Eq. (2.38) becomes,
\[ (x - p)\rho + \frac{1}{2} \beta \frac{\partial \rho}{\partial x} = c \]  
(2.40)

The solution of the homogenous part of Eq. (2.40) (i.e. with \( c = 0 \)) is,
\[ u(x) = \alpha e^{-(x-p)^2/\beta} \]  
(2.41)

By looking for a polynomial expression \( w(x) \), Eq. (2.40) has the following general solution:
\[ \rho(x) = u(x) + c \cdot w(x) \]  
(2.42)

where, \( w(x) \) is given by
\[ w(x) = (x - p) \left( 1 - \frac{z}{3} + \frac{z^2}{3 \times 5} - \frac{z^3}{3 \times 5 \times 7} + \ldots \right) \]  
(2.43)

with \( z = 2(x - p)^2/\beta \). Substituting Eq. (2.42) into Eq. (2.31) and (2.32),
\[ c_L = -\frac{\alpha_L}{w(-1)} e^{-(1-p)^2/\beta} \]  
(2.44)
\[ c_R = -\frac{\alpha_R}{w(1)} e^{-(1-p)^2/\beta} \]  
(2.45)

The probability density profiles at the two sides of the domain can be expressed as,
\[ \rho_L(x) = \alpha_L \left( e^{-(x-p)^2/\beta} + c_L w(x) \right) \]  
(2.46)
\[ \rho_R(x) = \alpha_R \left( e^{-(x-p)^2/\beta} + c_R w(x) \right) \]  
(2.47)
Substituting Eq.(2.46) and Eq.(2.47) into Eq.(2.33), we obtain,

\[ \alpha_L = F \cdot \alpha_R \]  

(2.48)

with,

\[ F = \frac{e^{-\frac{p^2}{\beta}} - c_R w(0)}{e^{-\frac{c_L^2}{\beta}} - c_L w(0)} \]  

(2.49)

Substituting Eq.(2.48) into Eq.(2.46) and Eq.(2.47), and then apply the integral condition Eq.(2.36), we have:

\[ \alpha_L = A + B + F(D + E) \]  

(2.50)

\[ \alpha_R = \frac{A + B}{F} + D + E \]  

(2.51)

By defining a function \( G(x) = \frac{1}{2} \sum_{i=1}^{\infty} \frac{x^i}{i!(2i-1)!} \), \( A,B,D,E \) are expressed in terms of \( p \) and \( \beta \) as follows,

\[ A = -\sqrt{\frac{3\pi}{4}} \text{erf}(p \cdot \sqrt{\frac{1}{\beta}}) \]  

+ \[ \sqrt{\frac{3\pi}{4}} \text{erf}((p + 1) \cdot \sqrt{\frac{1}{\beta}}) \]  

(2.52)

\[ B = \frac{c_R \beta}{2} \left( G\left( \frac{2(1-p)^2}{\beta} \right) - G\left( \frac{2p^2}{\beta} \right) \right) \]  

(2.53)

\[ D = -\sqrt{\frac{3\pi}{4}} \cdot \text{erf}(p - 1) \cdot \sqrt{\frac{1}{\beta}} \]  

+ \[ \sqrt{\frac{3\pi}{4}} \cdot \text{erf}(p \cdot \sqrt{\frac{1}{\beta}}) \]  

(2.54)

\[ E = \frac{c_L \beta}{2} \left( G\left( \frac{2p^2}{\beta} \right) - G\left( \frac{2(-1-p)^2}{\beta} \right) \right) \]  

(2.55)

So far, we have expressed \( c_L, c_R, \alpha_L, \alpha_R \) in terms of \( p \) and \( \beta \). Recall,

\[ \nu^+ = \nu_{cs}^+ gR^+ \]  

(2.56)

52
\[ 
\nu^- = \nu_{ex}^- g R^- \quad (2.57) \\
\sigma^2 = \nu^+(\epsilon^+)^2 + \nu^-(\epsilon^-)^2 \quad (2.58) \\
p = \frac{\nu^+ \epsilon^+ + \nu^- \epsilon^-}{\gamma} \quad (2.59) 
\]

From Eq. (2.56-2.59) and Eq. (2.39), we can see that \( p \) and \( \beta \) are functions of the model parameters \( \epsilon^+, \epsilon^-, \nu_{ex}^+, \nu_{ex}^-, \gamma, g \), and variables \( R^+, R^- \). Therefore, \( c_L, c_R, \alpha_L, \alpha_R \) are functions of these model parameters and variables as well. According to Eq. (2.46-2.47), the density profiles can be written in the following form,

\[
\rho_L(x) = f_1(x; \epsilon^+, \epsilon^-, \nu_{ex}^+, \nu_{ex}^-, \gamma, g, R^+, R^-) \quad (2.60)
\]

\[
\rho_R(x) = f_2(x; \epsilon^+, \epsilon^-, \nu_{ex}^+, \nu_{ex}^-, \gamma, g, R^+, R^-) \quad (2.61)
\]

Finally, by substituting Eq. (2.60-2.61) into boundary conditions Eq. (2.34-2.35), we obtain a set of nonlinear equations as follows:

\[
R^+ = h_1(\epsilon^+, \epsilon^-, \nu_{ex}^+, \nu_{ex}^-, \gamma, g, R^+, R^-) \quad (2.62)
\]

\[
R^- = h_2(\epsilon^+, \epsilon^-, \nu_{ex}^+, \nu_{ex}^-, \gamma, g, R^+, R^-) \quad (2.63)
\]

Eq. (2.62-2.63) can then be plugged into AUTO/MATCONT for the fixed point computation and bifurcation analysis. For each fixed point pair \( (R^+, R^-) \), the associated probability density profile can be obtained using Eq. (2.46-2.47).
Chapter 3

A patch dynamics scheme for agent-based computations

3.1 Introduction

Running agent-based simulations (especially for cases with a large number of agents) in the entire spatial and temporal domain could be prohibitively expensive. In this chapter we explore the application of another important building block of the EF approach - the patch dynamics scheme [86] - in efficiently extracting macroscopic information from agent-based simulations. We implement this scheme to the same agent-based financial market model studied in the previous chapter.

The patch dynamics scheme is designed to perform numerical simulations of a theoretically existing (but unavailable) macroscopic equation on macroscopic time and length scales; it uses appropriately initialized simulations of the available microscopic agent-based model in a number of small patches, which cover only a fraction of the spatio-temporal domain. The solution for the whole region is then approximated by interpolation/extrapolation of the macroscopic variables computed in these patches. The smoothness of the macroscopic variables allows us to use interpolation. The
advantage of this scheme is that the combined savings in both the space and time domain can yield significant overall savings in terms of the computational cost.

The patch dynamics scheme is a combination of two sub schemes - the gap-tooth scheme (interpolate macroscopic properties in space) [87] and the coarse projective integration scheme (extrapolate macroscopic properties in time) [79]. In the gap-tooth scheme [87], microscopic simulations are only performed in a number of small intervals in the spatial domain; in between these small intervals are what we call gaps. A coarse time-$\delta t$ map for this scheme is constructed as follows. We first choose a number of macroscopic grid points. Then, we choose a small interval around each grid point; initialize the fine scale, apply the microscopic solver (e.g. the agent-based simulator in this case) within each interval consistently with the macroscopic initial condition profiles; and provide each box with appropriate boundary conditions. Subsequently, we use the microscopic model in each interval to simulate until time $\delta t$, and obtain macroscopic information (e.g. by computing the average density in each box) at time $\delta t$. This amounts to a coarse time-$\delta t$ map; the procedure is then repeated. The resulting scheme has already been used with particle-based simulations of the viscous Burgers equation [36].

Analogously, exploiting the smoothness of coarse variables in the temporal domain, one can use the gap-tooth scheme in conjunction with any method-of-lines time integration scheme, such as a projective integration scheme [79]. We then perform a number of gap-tooth steps of size $\delta t$ to obtain an estimate of the time derivative of the unavailable macroscopic equation. Based on this estimate, a projective integration scheme is subsequently used to perform a time step of size $\Delta t \gg \delta t$. This combination has been termed patch dynamics [47].

In previous work, gap-tooth schemes [87] and patch dynamics schemes [86] have been applied to study a diffusion homogenization problem (a model problem). For the kind of agent-based problems we study in this chapter, a conservation law needs
to be satisfied. For this reason, instead of using the finite difference scheme, a finite volume based patch dynamics scheme has been designed. We first apply this scheme to an approximate continuum model (a PDE) of the agent-based model to illustrate the scheme as well as to facilitate our error analysis. In this case, our microscopic model is the continuum model (PDE), and our microscopic simulator is a finite volume scheme. Before each coarse time step, we construct the microscopic initial conditions inside many small intervals (which we call a simulation unit) in the spatial domain. Each simulation unit consists of one tooth and two buffers which are used to “protect” the boundaries of the tooth. In general, a given microscopic code only allows us to run with a set of predefined boundary conditions. It is highly non-trivial to impose macroscopically inspired boundary conditions on such microscopic codes, see, e.g. [58] for a control-based strategy. First proposed in [87], buffer regions are used to protect the tooth inside each simulation unit from the boundary artifacts. After running the microscopic simulations for time $\delta t$, we update the macroscopic properties inside the teeth and gaps through the restriction procedure. We then project the macroscopic properties forward in time for a bigger step $\Delta t$. Based on the projected macroscopic properties we construct microscopic initial conditions and the same cycle repeats. We studied the performance of the patch dynamics scheme on the continuum model, and computed order of accuracy estimates both analytically and numerically. Subsequently the patch dynamics scheme has been applied to the agent-based model after a small modification. The agent-based patch dynamics scheme requires the expensive microscopic simulations in only 20% of the spatial domain and 10% of the temporal domain.

We note that the so-called heterogeneous multiscale method [26, 27] shares a similar spirit in addressing the same problem of simulating only the macroscopic behavior of a multi-scale model. In what is termed the heterogeneous multi-scale method, a macro-scale solver is combined with an estimator for quantities that are unknown
because the macroscopic equation is not available. This estimator subsequently uses appropriately constrained runs of the microscopic model [26].

This chapter is organized as follows: in section 3.2.1 we recall the agent-based financial market model and its associated continuum model. In section 3.3 we describe the finite volume scheme for the continuum model. In section 3.4 we describe the patch dynamics scheme. In section 3.5 we first present the patch dynamics results based on the continuum model. Then we analyze the order of consistency of this scheme (based on the continuum model) both analytically and numerically. Finally we present the agent-based patch dynamics results. We conclude in section 3.6.

3.2 The models

3.2.1 The agent-based model

The agent-based model based on which we implemented the patch-dynamics scheme is the same financial market model studied in the previous chapter. This model simulates the actions of buying and selling by a large population of interacting individuals in the presence of mimesis. Recall briefly that each agent has an internal state whose value affects the individual’s decision-making in buying a selling a stock. The state varies in some artificial one-dimensional space, whose dynamics is subjected to: (1) an exponential decay process (reflecting the “forgetting effect” of individuals); (2) external market news; (3) and the effect of the behavior of the other agents (through the aggregate property - the buying and selling rates) - of the population. Mathematically, each internal state $X_i$ is governed by a stochastic different equation (SDE) in the following form:

$$dX_i = -\gamma X_i(t) dt + dI_i(t),$$

(3.1)
where the first and second term on the right hand side of the equation denote the
deterministic and stochastic effects respectively. The detailed agent-based rules can
be found in Section 3.5.3 of Chapter 2.

3.2.2 The continuum model

It is possible to reach a concise description of the dynamics of a large assembly of
agents by keeping track of only the number of agents in each preference state rather
than the preference states of every agents in the population. This is accomplished by
deriving an equation for \( \rho(x, t) \), the probability density of agents in preference state
\( x \) at time \( t \). We can imagine \( \rho(x, t) \) as the density of agents averaged over a large
number of replicas of the population. From the continuity equation, Omurtag and
Sirovich [74] derived the following population equation which describes the evolution
of the probability density \( \rho(x, t) \),

\[
\frac{\partial \rho}{\partial t} = -\gamma \frac{\partial (x \rho)}{\partial x} + \sum_{k=\pm} \nu^k (\rho(x - \epsilon^k, t) - \rho(x, t)) + (R^+ + R^-) \delta(x). \tag{3.2}
\]

For small \( \epsilon^k \) by expanding \( \rho(x - \epsilon^k, t) \) about \( \rho(x, t) \) in terms of \( \epsilon^k \) and truncating
terms higher than second order in \( \epsilon^k \), one obtains the following Fokker-Planck type
approximation:

\[
\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x} (\mu \rho) + \frac{1}{2} \sigma^2 \frac{\partial^2 \rho}{\partial x^2} + (R^+ + R^-) \delta(x), \tag{3.3}
\]

where,

\[
\mu(x, t) = \gamma x - (\nu^+ \epsilon^+ + \nu^- \epsilon^-), \tag{3.4}
\]

\[
\sigma^2(t) = \nu^+ (\epsilon^+)^2 + \nu^- (\epsilon^-)^2. \tag{3.5}
\]

The fact that agents leaving at \( x = \pm 1 \) are restored at the origin dictates that
\( \rho(x = \pm 1, t) = 0 \). The buying and selling rates are defined as
\( R^\pm = \mp \frac{1}{2} \sigma^2 \partial \rho / \partial x \mid_{x=\pm 1} \),

which are the outgoing fluxes at the boundaries. The Fokker-Planck approximation
Figure 3.1: Illustration of a finite volume method for updating the cell average $U_{i-1/2}^{n}$ by fluxes at the cell edges. Shown in x-t space.

Eq. (3.3) is the one we call “the continuum model” in this study. Before we apply the patch-dynamics scheme to the agent-based model we apply it to this continuum model (to validate this scheme as well as to facilitate the order of consistency analysis discussed in the subsequent sections.).

### 3.3 Finite Volume Scheme

Our patch dynamics scheme was inspired by a finite volume scheme. This finite volume scheme is introduced in this section through the continuum model introduced in the previous section. We divide the one dimensional spatial domain into equal-sized grid cells and keep track of our approximations to the average densities in these cells. In each time step we update the average cell densities using approximations of the fluxes through the edges of the cells. As shown in Fig. 3.1 we denote the $i^{th}$ grid cell by

$$C_i = (x_{i-1}, x_i).$$

(3.6)

$U_{i-1/2}^{n}$ denotes the approximated average cell density in the $i^{th}$ cell at time $t_n$:

$$U_{i-1/2}^{n} \approx \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} \rho(x, t_n)dx \equiv \frac{1}{\Delta x} \int_{C_i} \rho(x, t_n)dx ,$$

(3.7)

where $\Delta x = x_i - x_{i-1}$ denotes the length of the cell.
Except for the central grid cell which will be treated separately, Eq. \((3.3)\) can be rewritten as:

\[
\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x} \left( \mu \rho + \frac{1}{2} \sigma^2 \frac{\partial \rho}{\partial x} \right). \tag{3.8}
\]

Define the flux term as,

\[
f(x, t) = - (\mu(x, t) \rho(x, t) + \frac{1}{2} \sigma^2(t) \frac{\partial \rho(x, t)}{\partial x}), \tag{3.9}
\]

the integral form (in space) of Eq. \((3.8)\) gives

\[
\frac{d}{dt} \int_{C_i} \rho(x, t) dx = f(x_{i-1}, t) - f(x_i, t). \tag{3.10}
\]

Integrating Eq. \((3.10)\) in time from \(t_n\) to \(t_{n+1}\) and dividing by \(\Delta x\) gives

\[
\frac{1}{\Delta x} \int_{C_i} \rho(x, t_{n+1}) dx = \frac{1}{\Delta x} \int_{C_i} \rho(x, t_n) dx - \frac{1}{\Delta x} \left[ \int_{t_n}^{t_{n+1}} f(x_i, t) dt - \int_{t_n}^{t_{n+1}} f(x_{i-1}, t) dt \right]. \tag{3.11}
\]

Based on Eq. \((3.7)\), Eq. \((3.11)\) can be rewritten as

\[
U_{i-1/2}^{n+1} = U_{i-1/2}^n + \frac{\Delta t}{\Delta x} (F_{i-1}^n - F_i^n), \tag{3.12}
\]

where, \(U_{i-1/2}^n\) is the average density defined in eq.\((3.7)\), \(F_{i-1}^n\) and \(F_i^n\) denote the average fluxes across cell edges:

\[
F_{i-1}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(x_{i-1}, t) dt \tag{3.13}
\]

\[
F_i^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(x_i, t) dt. \tag{3.14}
\]

Based on Eq. \((3.9)\), the average fluxes are approximated by

\[
F_{i-1}^n \approx F(U_{i-3/2}^n, U_{i-1/2}^n). \tag{3.15}
\]
\begin{align}
\approx & -\left(\mu_{i-1} \frac{U_{i-3/2}^n + U_{i-1/2}^n}{2} + \frac{1}{2} \sigma^2 \frac{U_{i-1/2}^n - U_{i-3/2}^n}{\Delta x}\right) \quad (3.15) \\
F_i^n & \approx F\left(U_{i-1/2}^n, U_{i+1/2}^n\right) \\
& \approx -\left(\mu_i \frac{U_{i-1/2}^n + U_{i+1/2}^n}{2} + \frac{1}{2} \sigma^2 \frac{U_{i+1/2}^n - U_{i-1/2}^n}{\Delta x}\right). \quad (3.16)
\end{align}

For a detailed discussion about the construction of diffusion fluxes for finite volume scheme one can refer to [33]. The fluxes at the two boundaries are expressed as (recall that \(x_0 = -1\) and \(x_N = 1\)):

\begin{align}
F_0^n &= -R^- \approx -\sigma^2 \frac{U_{1/2}^n}{\Delta x} \quad (3.17) \\
F_N^n &= R^+ \approx \sigma^2 \frac{U_{N-1/2}^n}{\Delta x}. \quad (3.18)
\end{align}

We use an odd number of grid cells \(N\) for this scheme. Since the central grid cell centered at \(x_{N/2}\) contains the source point at \(x = 0\), two additional terms need to be added to Eq. \(3.12\) for this cell:

\begin{align}
U_{N/2}^{n+1} &= U_{N/2}^n + \frac{\Delta t}{\Delta x} \left(F_{(N-1)/2}^n - F_{(N+1)/2}^n + (R^+ + R^-)\right). \quad (3.19)
\end{align}

The above scheme is a conservative scheme. The patch dynamics scheme introduced in the next section is inspired by this scheme, and is such that when applied to agent-based problem the patch dynamics scheme is conservative. Before applying the patch dynamics scheme to the agent-based problem, we first apply it to a more tractable case - the continuum model (PDE) - described in section 3.2.2, to study its performance as well as to analyze its order of accuracy. The above mentioned finite volume scheme is used as the microscopic solver in this case to mimic the agent-based simulator. The only reason for implementing the microscopic simulation on the continuum equation is to study the errors involved - something we cannot do when dealing with an agent-based simulator.


3.4 Patch Dynamics Scheme

As mentioned previously, the patch dynamics scheme is actually a combination of two subsystems - the gap tooth scheme and the projective integration scheme. The gap-tooth scheme is illustrated in Fig. 3.2. We have divided the one-dimensional space into two kinds of unequal-sized grid cells: the narrower cells which we call teeth and the wider cells which we call gaps. We intend to solve the microscopic model only within the teeth (plus some buffer regions to handle the teeth boundary conditions) but to compute the average densities for both the teeth and the gaps and use these as the macroscopic properties so as to get a conservative scheme. As in the finite volume scheme, we want to update the average cell densities based on the fluxes at the edges of the cells:

\[
U_i^{n+\delta} = U_i^n + \frac{\delta t}{L_T} (F_{i,L}^n - F_{i,R}^n), \tag{3.20}
\]

\[
U_{i+1/2}^{n+\delta} = U_{i+1/2}^n + \frac{\delta t}{L_G} (F_{i,R}^n - F_{i+1,L}^n), \tag{3.21}
\]

where \(L_T\) and \(L_G\) denote the tooth and gap size respectively. However, as opposed to the finite volume scheme, the fluxes \(F_{i,L}^n\) and \(F_{i,R}^n\) are not computed based on a known macroscopic equation (which is usually not available for agent-based models). Instead, our goal is to compute these fluxes based on the “microscopic simulations” inside a small fraction of the spatial domain - the simulation unit (“\(\alpha \Delta x\)” shown in Fig. 3.3). These separated simulation units are the only places we run microscopic simula-

---

**Figure 3.2: Illustration of the gap-tooth scheme.**
tions. The time integration of the microscopic model in each simulation unit provides information on the evolution of the global problem at the spatial location of the simulation unit, as if we were running the microscopic simulations in the entire spatial domain. Therefore, it is crucial to choose some appropriate boundary conditions such that the solutions inside the teeth evolve as if the simulations were performed for the entire domain. For some cases it is possible to impose macroscopically-inspired constraints on the microscopic model as boundary conditions \[87\]. However, this is not always the case. To overcome this difficulty, as first proposed in \[86\], we use a larger box - the simulation unit - to run the microscopic simulations, but still only use the flux at the teeth boundaries to update macroscopic variables. Each simulation unit consists of one tooth and two surrounding buffers (the two simulation units on the boundaries of the spatial domain are treated slightly differently). The purpose of the additional computational domains, the buffers, is to protect the teeth from boundary artifacts. This can be accomplished over short enough time intervals, provided the buffers are large enough.

As Fig. 3.3 shows, we divide the one-dimensional spatial domain into an odd number, \(N\), grid cells \(C_i, i = 1, 2, \ldots, N\). The size of each cell is \(\Delta x = 2/N\). We then put teeth (narrow bins) at the edges of these cells and put gaps (wide bins) between the teeth. There are \(N + 1\) teeth and \(N\) gaps. The average densities for the teeth at time \(t_n\) are denoted as \(U_i^n, i = 0, 2, \ldots, N\), and the densities for the gaps are denoted as \(U_i^{n-1/2}, i = 1, 2, \ldots, N\). We want to update these macroscopic properties at time \(t_{n+1}\) based on the microscopic simulations inside the simulation units during the time step from \(t_n\) to \(t_n + \delta t\). The patch dynamics algorithm to proceed from \(U_i^n\) to \(U_i^{n+1}(i = 0, \frac{1}{2}, 1, \ldots, N)\), is given below:

1. **Lifting.** At time \(t_n\) create initial conditions \(u^i(x, t_n), i = 0, \ldots, N\) inside each simulation unit for the microscopic simulator, consistent with the spatial profile
Figure 3.3: Illustration of the lifting procedure. (a) lifting procedure in the bulk of the domain. A local quadratic polynomial is constructed such that the averaged polynomial values inside the tooth (region 3), and inside the gaps (region 1 and 5) equal the average densities in these regions. (b) lifting procedure in the left boundary. (c) lifting procedure in the right boundary. One point value and two average values are used in constructing the local polynomials for the simulation units at the two boundaries.

of the macroscopic properties - the average densities inside the teeth and gaps $U_i^n (i = 0, 1/2, 1, ... N)$.

2. Simulation. Based on the microscopic initial condition constructed in step 1 compute $F_{i,L}$ and $F_{i,R}$ by running the microscopic simulator inside the simulation units from time $t_n$ to $t_n + \delta t$.

3. Restriction. Obtain the spatially averaged densities inside the teeth and
gaps $U_i^{n+\delta}(i = 0, 1/2, 1, ...N)$ based on Eq.(3.20)-(3.21).

4. **Projective step.** Estimate the macroscopic time derivative at time $t_n$, e.g., as

$$F(U_i^n; \delta t) = \frac{U_i^{n+\delta} - U_i^n}{\delta t},$$

and use with a time integration method of choice, e.g., forward Euler,

$$U_i^{n+1} = U_i^n + \Delta t F(U_i^n; \delta t).$$

### 3.4.1 Lifting Procedure

Except for the two end teeth that get special treatment, we create the microscopic initial condition inside the $i^{th}$ simulation unit by constructing a local quadratic polynomial approximation, $u^i(x, t)$, to the density that matches the average density in a tooth and its two neighboring gaps as follows. Let $H$ be the width of the tooth (so $\Delta x - H$ is the width of the gap) and require:

$$\frac{1}{H} \int_{-H/2}^{H/2} u^i(x, t_n) dx = U_i^n,$$

$$\frac{1}{\Delta x - H} \int_{-\Delta x + H/2}^{-\Delta x - H/2} u^i(x, t_n) dx = U_{i-1/2}^n,$$

$$\frac{1}{\Delta x - H} \int_{H/2}^{\Delta x - H/2} u^i(x, t_n) dx = U_{i+1/2}^n.$$

As shown in Fig. 3.3(a), the microscopic initial condition $u^i(x, t_n), (i = 1, 2, ..., N - 1)$ is constructed such that the averaged function values inside the tooth (region 3), and inside the gaps (region 1 and 5) equal the average densities in these regions. At the two boundaries, as Fig. 3.3(b) and (c) show, there is only one neighboring gap for each of the two boundary teeth. Therefore, the microscopic initial condi-
tions are created in a slightly different way. At the left boundary, a local quadratic polynomial \( u^0(x, t_n) \) is constructed such that:

\[
\begin{align*}
  u^0(-1, t_n) &= 0 \\
  \frac{1}{H} \int_{-H/2}^{H/2} u^0(x, t_n) dx &= U_0^n \\
  \frac{1}{\Delta x - 3H/2} \int_{H/2}^{\Delta x - H} u^0(x, t_n) dx &= U_{1/2}^n
\end{align*}
\]

Similarly, at the right boundary a local quadratic polynomial \( u^N(x, t_n) \) is constructed such that,

\[
\begin{align*}
  u^N(1, t_n) &= 0 \\
  \frac{1}{H} \int_{-H/2}^{H/2} u^N(x, t_n) dx &= U_N^n \\
  \frac{1}{\Delta x - 3H/2} \int_{-H/2}^{\Delta x - H} u^N(x, t_n) dx &= U_{N-1/2}^n
\end{align*}
\]

Note that it is the left edge of the leftmost tooth (and similarly the right edge of the rightmost tooth) which are put at the two boundaries \( x = \pm 1 \), instead of their respective centers. For this reason, the leftmost and rightmost gaps are of a slightly different size \( (\Delta x - \frac{3}{2}H) \) compared to the rest of the gaps (which are of size \( \Delta x - H \)).

3.4.2 Simulation step

As mentioned previously, for our initial experiment and its analysis a fine-grid finite volume scheme is used as the microscopic simulator. We emphasize again that the only purpose for doing the microscopic simulation on the continuum equation is to facilitate our investigations about the errors involved - something we cannot do when dealing with an agent-based simulator.

We divide each simulation unit into \( N_t + 2N_b \) fine bins, where \( N_t \) denotes the number of fine bins inside the tooth while \( N_b \) denotes the number of fine bins inside
the buffers. All these fine bins are of equal size $h$. Based on the previously constructed local quadratic polynomial $u^i(x, t_n)$ we initialize the averaged densities in the fine bins as follows:

$$\frac{1}{h} \int_{x_j-h/2}^{x_j+h/2} u^i(x, t_n) dx = u^i_j,$$  \hspace{1cm} (3.33)

where $x_j$ denotes the center of each fine bin and $u^i_j$ denotes the average densities inside the $j^{th}$ fine bin of the $i^{th}$ simulation unit (as the red cross marks in Fig. 3.4 and 3.5 denote). Fig. 3.4 illustrates the simulation step in the bulk of the spatial domain. Starting from the previously constructed local quadratic polynomial, we first initialized the average densities inside the fine bins using Eq. (3.33). To evolve $u^i_j(t_0)$ for one microscopic step $dt$, as Eq. (3.12) and (3.15-3.16) show, it requires $u^i_{j-1}(t_0)$, $u^i_j(t_0)$ and $u^i_{j+1}(t_0)$. Because of that, we are not able to evolve the solutions for the two bins at the boundaries (the two bins denoted in grey in the upper-left part of Fig. 3.4). Therefore, after the first microscopic time step $dt$, these two bins are discarded. For the same reason, after another microscopic time step $dt$, two more fine bins need to be discarded, and so on until all the bins in the buffer regions are discarded. There
Figure 3.5: Illustration of the simulation step for the simulation unit at the left boundary.

are $N_b$ fine bins in each buffer, so we can run the microscopic simulator for $N_b$ steps. During each step we save the fluxes computed at the edges of each tooth, $F_{i,L}^k$ and $F_{i,R}^k$, which denote the fluxes at the left and right edges of the $i^{th}$ tooth at the $k^{th}$ step ($k = 1, 2, \ldots N_b$).

The simulations in the first and last simulation units are performed slightly differently. As Fig. 3.5 shows, buffer is only used on one side (the side to the interior of the spatial domain) of the tooth. On the other side, the side outside the spatial domain, we use a ghost bin (colored pink in Fig. 3.5) such that the density of this ghost bin is of the same magnitude but opposite sign to the the density of the bin next to it at the boundary of the domain. We use this artificial density to update the density of the bin next to it as well as the flux at the edge of the tooth. After each microscopic time step we set the density inside the ghost bin to be of the same magnitude but opposite sign to the bin next to it again for the evolution of the following step. In this way the boundary condition $\rho(x = \pm 1) = 0$ is satisfied. The schematic we illustrated in Fig. 3.5 is for the simulation unit at the left boundary, the one for the right boundary is done in a similar way.
3.4.3 Restriction procedure

At the end of the $N_b$th step, we update the total fluxes at the edges of each tooth for the coarse time interval $\delta t = N_b \cdot dt$ as follows:

$$F_{i,L} = \frac{1}{N_b} \sum_{k=1}^{N_b} F_{k,L}$$

(3.34)

$$F_{i,R} = \frac{1}{N_b} \sum_{k=1}^{N_b} F_{k,R}.$$  (3.35)

Then based on Eqn. (3.20 - 3.21), our coarse variables - the average densities inside the teeth and gaps $U_{i}^{n+\delta}(i = 0, 1/2, 1, ...N)$ - are updated.

3.4.4 Projective step

Eq. (3.22) is used to estimate the macroscopic time derivative and Eq. (3.23) is used to project the macroscopic solutions forward to obtain $U_{i}^{n+1}$.

3.5 Results and discussion

3.5.1 Comparison of patch-dynamics scheme with fine and coarse finite volume schemes for the continuum equation

We compare the performance of patch-dynamics scheme with two other schemes: one is the fine grid finite volume scheme i.e., our microscopic simulator applied to the entire region; the other is a coarse grid finite volume scheme which mimics the usually unavailable macroscopic solver. All the three schemes are first applied to the continuum approximation of the agent-based model. Fig. 3.6 shows several snapshots of the solutions of these three schemes along the path towards their steady states. Starting from the same Gaussian-like initial conditions (Fig. 3.6a) we evolve the
solutions of the three systems for a long time (Fig. 3.6b shows one snapshot of the transient solutions) until they have reached their steady states (Fig. 3.6d). As we can see from the figure, the solution of the patch-dynamics scheme agrees well with the other two schemes along the whole process. The steady state solutions of the three schemes also agree well with the analytical steady state solution of the problem (Fig. 3.6d). In Fig. 3.6c and e we plot the differences between the fine finite-volume scheme and the other two methods, patch dynamics and coarse finite volume scheme. The solution of the fine finite-volume scheme is an approximation to the true solution, so these differences can be viewed as a first approximation to the errors in the methods. Also, since the patch dynamics uses the fine finite-volume scheme as its microscopic solver, the differences between it and patch dynamics are an indication of the error introduced by the patch dynamics scheme. As we can see from (Fig. 3.6c and e), at the transient time the two schemes give comparable errors, while, as we further evolve the system to steady state, the patch dynamics scheme gives smaller errors. This is probably due to the fact that our patch dynamics scheme utilizes more microscopic level information.

We now look at the computational savings of the patch dynamics scheme. For the patch-dynamics scheme, we divide the spatial domain into \( N = 41 \) big cells which gives 42 teeth and 41 gaps. We run “microscopic simulations” in only \( \alpha = 10\% \) of the spatial domain (since the width of each simulation unit is equal to only 10% of each big cell). Inside each simulation unit we put 10 equal-sized fine bins inside each tooth and in each buffer to run the microscopic simulations (which is mimicked by the fine grid finite volume scheme). We use \( dt = 4 \times h^2 \) as the time step for the microscopic simulator. After running the microscopic simulator for \( n_b = 10 \) microscopic steps (\( dt \)), or equivalently one coarse time step \( \delta t = n_b \times dt \), we project macroscopic variables 90 steps forward in time. This means the microscopic simulations are performed in only 1.11% of the temporal domain. For the fine finite volume scheme, we use
\( N_{\text{fine}} = 1271 \) fine bins, and use the same time step size \( dt \) as the one used in the microscopic simulator for the patch dynamics scheme. For the coarse finite volume scheme we use the same number of big cells \( (N = 41) \) as the one used for the patch-dynamics scheme. The time step size of this scheme is set to be equal to the effective time step size \( \Delta t \) of the patch dynamics scheme (recall that \( \Delta t = (M + 1)\delta t \), with \( M = 90 \)). Overall, in this illustrative example the patch dynamics scheme runs the microscopic simulations in only 10\% of the spatial domain and 1.11\% of the temporal domain.

### 3.5.2 Consistency Analysis

In this section, we perform consistency analysis for the patch-dynamics scheme using the continuum equation of the agent-based model. We present analytical derivations of the order of consistency first, followed by numerical results.

**Analytical Results**

We analyze the errors due to the quadratic interpolation scheme used in lifting procedure. We assume the spatial domain is infinitely large (no issues about boundary conditions), and perform the analysis in the \( i^{th} \) tooth and \( i-1^{th} \) gap. Substituting Eq. (3.20) and (3.22) into Eq. (3.23), we obtain the equation for updating the tooth density at \( t_{n+1} \) as follows,

\[
U_{i}^{n+1} = U_{i}^{n} + \frac{\Delta t}{H}(F_{i,L}^{n} - F_{i,R}^{n}).
\]  

Similarly, for gap density \( U_{i}^{n+1} \),

\[
U_{i-1/2}^{n+1} = U_{i-1/2}^{n} + \frac{\Delta t}{\Delta x - H}(F_{i-1,R}^{n} - F_{i,L}^{n}).
\]
Let us define the truncation error for the tooth to be \( T_i^n \) and the one for the gap to be \( T_{i-1/2}^n \). Our goal in this section is to express the truncation errors in terms of orders of \( \Delta x \). Let us further assume the exact solution \( v(x, t) \) for the continuum model is available, Eq. (3.36) can then be rewritten as,

\[
\frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} v(x, t_{n+1}) dx + \Delta t T_i^n = \frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} v(x, t_n) dx + \frac{\Delta t}{H} (F_{i,L}^n - F_{i,R}^n). \tag{3.38}
\]

Taylor expanding \( v(x, t_{n+1}) \) in time, Eq. (3.38) becomes,

\[
\frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} v(x, t_n) + \Delta t v_t(x, t_n) + O(\Delta t^2) dx + \Delta t T_i^n = \frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} v(x, t_n) dx + \frac{\Delta t}{H} (F_{i,L}^n - F_{i,R}^n).
\] (3.39)

Canceling the first term of each side and then divide both sides by \( \Delta t \) to get,

\[
\frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} v_t(x, t_n) dx + T_i^n + O(\Delta t) = \frac{1}{H} (F_{i,L}^n - F_{i,R}^n). \tag{3.40}
\]

Substitute the flux form of the continuum equation \( v_t(x, t_n) = -F(v(x, t_n))_x \) into Eq. (3.40) to get,

\[
\frac{1}{H} \int_{x_i-H/2}^{x_i+H/2} -F(v(x, t_n))_x dx + T_i^n + O(\Delta t) = \frac{1}{H} (F_{i,L}^n - F_{i,R}^n). \tag{3.41}
\]

After integration,

\[
\frac{1}{H} (F(v(x_i - H/2, t_n)) - F(v(x_i + H/2, t_n))) + T_i^n + O(\Delta t) = \frac{1}{H} (F_{i,L}^n - F_{i,R}^n). \tag{3.42}
\]

Rearranging Eq. (3.42), we obtain the truncation error for the teeth as,

\[
T_i^n = \frac{1}{H} (F_{i,L}^n - F_{i,R}^n) - \frac{1}{H} \left( F(v(x_i - H/2, t_n)) - F(v(x_i + H/2, t_n)) \right) + O(\Delta t) . \tag{3.43}
\]
where, $F_{n,i,L}$ and $F_{n,i,R}$ are the fluxes at the left and right edges of the $i^{th}$ tooth computed from the lifting procedure through the second order polynomial interpolation; $F(v(x_i - H/2, t_n))$ and $F(v(x_i + H/2, t_n))$ are the corresponding fluxes computed from the exact solution. Following the same approach, the truncation error for each gap reads

$$
T_{i-1/2}^{n} = \frac{1}{\Delta x - H} \left( F_{n,i-1,R}^{n} - F_{n,i,L}^{n} \right) - \frac{1}{\Delta x - H} \left( F(v(x_i - \Delta x + H/2, t_n)) - F(v(x_i - H/2, t_n)) \right) + O(\Delta t),
$$

(3.44)

where, $F_{i-1,L}^{n}$ and $F_{i,L}^{n}$ are the fluxes at the left and right edges of the $(i - 1/2)^{th}$ gap computed from the lifting procedure while $F(v(x_i - \Delta x + H/2, t_n))$ and $F(v(x_i - H/2, t_n))$ are the ones computed from the exact solution.

In the following analysis we will express the above fluxes in terms of the spatial Taylor expansions of the exact solution around $v(x_i, t_n)$. After that the truncation errors can be expressed in terms of various orders of $\Delta x$ and spatial derivatives of the exact solution at $(x_i, t_n)$, from which we can tell the orders of consistency for the patch dynamics scheme. For simplicity, we will drop the superscript “$n$” since all the terms in the following analysis are at time $t_n$.

To find the fluxes computed from the lifting procedure, let us first look at the local quadratic polynomial in the $i^{th}$ simulation unit:

$$
u_i(x) = a_i \cdot x^2 + b_i \cdot x + c_i
$$

(3.45)

$x$ here has its origin at the center of the tooth. We require this polynomial to have the correct averages over the tooth as well as the two adjacent gaps. This leads to the constraints shown in Eq. 3.24, 3.26. The coefficients $a_i$, $b_i$, $c_i$ are solved in
terms of the tooth and gaps densities $U_i$, $U_{i-\frac{1}{2}}$, $U_{i+\frac{1}{2}}$ as follows:

\[
a_i = \frac{3(U_{i+\frac{1}{2}} - 2U_i + U_{i+\frac{1}{2}})}{\Delta x(2\Delta x - H)} \tag{3.46}
\]

\[
b_i = \frac{U_{i+\frac{1}{2}} - U_{i-\frac{1}{2}}}{\Delta x} \tag{3.47}
\]

\[
c_i = U_i - \frac{1}{12}H^2 a_i. \tag{3.48}
\]

Defining $\beta = \nu^+\epsilon^+ + \nu^-\epsilon^-$, so $\mu(x) = \gamma x - \beta$, based on Eq. (3.9) we can write the fluxes at the left and right edges of the $i$th tooth as (the prime sign denotes spatial derivative):

\[
F_{i,L} = -\mu(x_i - \frac{H}{2})u_i(x_i - \frac{H}{2}) - \frac{1}{2}\sigma^2 u'_i(x_i - \frac{H}{2})
= -\left(\gamma \cdot (x_i - \frac{H}{2}) - \beta\right)\left(U_i - \frac{b_i H}{2} + \frac{a_i H^2}{6}\right) - \frac{\sigma^2}{2}\left(-a_i H + b_i\right) \tag{3.49}
\]

\[
F_{i,R} = -\mu(x_i + \frac{H}{2})u_i(x_i + \frac{H}{2}) - \frac{1}{2}\sigma^2 u'_i(x_i + \frac{H}{2})
= -\left(\gamma \cdot (x_i + \frac{H}{2}) - \beta\right)\left(U_i + \frac{b_i H}{2} + \frac{a_i H^2}{6}\right) - \frac{\sigma^2}{2}\left(a_i H + b_i\right). \tag{3.50}
\]

Similarly, the flux at the right edge of the $i-1$th tooth can be expressed as,

\[
F_{i-1,R} = -\left(\gamma \cdot (x_i - \Delta x + \frac{H}{2}) - \beta\right)\left(U_{i-1} + \frac{b_{i-1} H}{2} + \frac{a_{i-1} H^2}{6}\right) - \frac{\sigma^2}{2}\left(a_{i-1} H + b_{i-1}\right). \tag{3.51}
\]

The first terms on the right hand side of truncation error equations Eqs. (3.43) - (3.44) read

\[
\frac{1}{H} (F_{i,L} - F_{i,R}) = \frac{1}{2\Delta x(2\Delta x - H)} \left( (4\beta \Delta x - 2\beta H + \gamma H^2 + 6\sigma^2 - 4\Delta x\gamma x_i + 2\gamma H x_i)U_{i-1/2} + 2(2\Delta x^2\gamma - \Delta x\gamma H - \gamma H^2 - 6\sigma^2)U_i + ( -4\beta \Delta x + 2\beta H + \gamma H^2 + 6\sigma^2 + 4\Delta x\gamma x_i - 2\gamma H x_i)U_{i+1/2} \right), \tag{3.52}
\]
\[
\frac{1}{\Delta x - H} \left( F_{i-1,R} - F_{i,L} \right) = -\frac{1}{2\Delta x(\Delta x - H)(2\Delta x - H)} \left( -2(\beta + \Delta x \gamma - \frac{1}{2}\gamma(H + 2x_i)) \left( H(-\Delta x + H)U_{i-3/2} + (2\Delta x^2 - \Delta x H - H^2)U_{i-1} + \Delta x HU_{i-1/2} \right) + \sigma^2 \left( -2(\Delta x - H) \right) \right),
\]

Next, we will replace the average densities in Eqs. (3.52) - (3.53) in terms of Taylor expansions of the exact solution as below,

\[
U_i = \frac{1}{H} \int_{x_i - H/2}^{x_i + H/2} v(x, t_n) dx = \frac{1}{H} \int_{x_i - H/2}^{x_i + H/2} v(x_i, t_n) + \frac{1}{2} v''(x_i, t_n)(x - x_i)^2 dx + O(H^4)
\]

\[
U_{i-1/2} = \frac{1}{\Delta x - H} \int_{x_{i-1} - \Delta x + H/2}^{x_{i-1} + H/2} v(x, t_n) dx = \frac{1}{\Delta x - H} \int_{x_{i-1} - \Delta x + H/2}^{x_{i-1} + H/2} v(x_i, t_n) + \frac{1}{2} v''(x_i, t_n)(x - x_i)^2 dx + O(\Delta x^4)
\]

\[
U_{i+1/2} = \frac{1}{\Delta x - H} \int_{x_{i+1} - \Delta x + H/2}^{x_{i+1} + H/2} v(x, t_n) dx = \frac{1}{\Delta x - H} \int_{x_{i+1} - \Delta x + H/2}^{x_{i+1} + H/2} v(x_i, t_n) + \frac{1}{2} v''(x_i, t_n)(x - x_i)^2 dx + O(\Delta x^4)
\]
\[
\begin{align*}
U_i & = v(x_i, t_n) + \frac{1}{2} \Delta x v'(x_i, t_n) + \frac{1}{6} \Delta x^2 v''(x_i, t_n) - \frac{1}{12} \Delta x H v''(x_i, t_n) \\
& + \frac{1}{24} H^2 v''(x_i, t_n) + \frac{1}{24} \Delta x^3 v^{(3)}(x_i, t_n) - \frac{1}{24} \Delta x^2 H v^{(3)}(x_i, t_n) \\
& + \frac{1}{48} \Delta x H^2 v^{(3)}(x_i, t_n) + O(\Delta x^4),
\end{align*}
\]

\[
\begin{align*}
U_{i-1} & = \frac{1}{H} \int_{x_i - \Delta x + H/2}^{x_i} v(x, t_n) \, dx \\
& = \frac{1}{H} \int_{x_i - \Delta x - H/2}^{x_i - \Delta x + H/2} v(x_i, t_n) + v'(x_i, t_n)(x - x_i) + \frac{1}{2} v''(x_i, t_n)(x - x_i)^2 \\
& + \frac{1}{6} v^{(3)}(x_i, t_n)(x - x_i)^3 \, dx + O(\Delta H^4) \\
& = v(x_i, t_n) - \Delta x v'(x_i, t_n) + \frac{1}{2} \Delta x^2 v''(x_i, t_n) + \frac{1}{24} H^2 v''(x_i, t_n) \\
& - \frac{1}{6} \Delta x^3 v^{(3)}(x_i, t_n) - \frac{1}{24} \Delta x H^2 v^{(3)}(x_i, t_n) + O(\Delta H^4),
\end{align*}
\]

\[
\begin{align*}
U_{i-\frac{3}{2}} & = \frac{1}{\Delta x - H} \int_{x_i - 2\Delta x + H/2}^{x_i - \Delta x + H/2} v(x, t_n) \, dx \\
& = \frac{1}{\Delta x - H} \int_{x_i - 2\Delta x - H/2}^{x_i - \Delta x - H/2} v(x_i, t_n) + v'(x_i, t_n)(x - x_i) + \frac{1}{2} v''(x_i, t_n)(x - x_i)^2 \\
& + \frac{1}{6} v^{(3)}(x_i, t_n)(x - x_i)^3 \, dx + O(\Delta x^4) \\
& = v(x_i, t_n) - \frac{3}{2} \Delta x v'(x_i, t_n) + \frac{7}{6} \Delta x^2 v''(x_i, t_n) - \frac{1}{12} \Delta x H v''(x_i, t_n) \\
& + \frac{1}{24} H^2 v''(x_i, t_n) - \frac{5}{8} \Delta x^3 v^{(3)}(x_i, t_n) + \frac{1}{8} \Delta x^2 H v^{(3)}(x_i, t_n) \\
& - \frac{1}{16} \Delta x H^2 v^{(3)}(x_i, t_n) + O(\Delta x^4).
\end{align*}
\]

Substituting Eq. (3.54)-(3.58) into Eq. (3.52)-(3.58),

\[
\begin{align*}
\frac{1}{H} (F_{i,R} - F_{i,L}) & = \frac{1}{24} \left( 12\sigma v''(x_i, t_n) - \beta \left( 24v'(x_i, t_n) + \left( 2\Delta x^2 - 2\Delta x H + H^2 \right) v^{(3)}(x_i, t_n) \right) \\
& + \gamma \left( 24v(x_i, t_n) + 3H^2 v''(x_i, t_n) + x_i(24v'(x_i, t_n)) \right) \right) + O(\Delta x^2),
\end{align*}
\]

\[
\begin{align*}
\frac{1}{\Delta x - H} (F_{i-1,R} - F_{i,L}) & = \frac{1}{24} \left( 6\sigma^2 (2v''(x_i, t_n) - \Delta x v^{(3)}(x_i, t_n)) - \beta \left( 24v'(x_i, t_n) + 4\Delta x^2 v^{(3)}(x_i, t_n) \right) \\
& + H^2 v^{(3)}(x_i, t_n) - 4\Delta x (3v''(x_i, t_n) + H v^{(3)}(x_i, t_n)) \right) + \gamma (24v(x_i, t_n))
\end{align*}
\]
\[+24x_i v'(x_i, t_n) + 3H^2 v''(x_i, t_n) - 4\Delta x^3 v^{(3)}(x_i, t_n) + H^2 x_i v^{(3)}(x_i, t_n)\]
\[+\Delta x^2 (12v''(x_i, t_n) + (5H + 4x_i)v^{(3)}(x_i, t_n)) - 2\Delta x(12v'(x_i, t_n)\]
\[+(H + 2x_i)(3v''(x_i, t_n) + Hv^{(3)}(x_i, t_n))) + O(\Delta x^2).\]  \hspace{1cm} (3.60)

Based on Eq. (3.59) the second terms of the right hand side of Eq. (3.38 - 3.44) are expressed as follows,
\[
\frac{1}{H} \left( F(v(x_i - H/2, t_n)) - F(v(x_i + H/2, t_n)) \right)
\]
\[= \frac{1}{H} \left( (-\mu(x_i - H/2)v(x_i - H/2, t_n) - \frac{\sigma^2}{2}v'(x_i - H/2, t_n))\]
\[-( -\mu(x_i - H/2)v(x_i + H/2, t_n) - \frac{\sigma^2}{2}v'(x_i + H/2, t_n)) \right)\]
\[= \frac{1}{H} \left( (-\gamma \cdot (x_i - H/2) - \beta)v(x_i - H/2, t_n) - \frac{\sigma^2}{2}v'(x_i - H/2, t_n))\]
\[-( -\gamma \cdot (x_i + H/2) - \beta)v(x_i + H/2, t_n) - \frac{\sigma^2}{2}v'(x_i + H/2, t_n)) \right),\]  \hspace{1cm} (3.61)
\[
\frac{1}{\Delta x - H} \left( F(v(x_i - \Delta x + H/2, t_n)) - F(v(x_i - H/2, t_n)) \right)
\]
\[= \frac{1}{\Delta x - H} \left( (-\mu(x_i - \Delta x + H/2)v(x_i - \Delta x + H/2, t_n)\]
\[-\frac{\sigma^2}{2}v'(x_i - \Delta x + H/2, t_n)) - ( -\mu(x_i - H/2)v(x_i - H/2, t_n)\]
\[-\frac{\sigma^2}{2}v'(x_i - H/2, t_n)) \right)\]
\[= \frac{1}{\Delta x - H} \left( (-\gamma \cdot (x_i - \Delta x + H/2) - \beta)v(x_i - \Delta x + H/2, t_n)\]
\[-\frac{\sigma^2}{2}v'(x_i - \Delta x + H/2, t_n)) - ( -\gamma \cdot (x_i - H/2) - \beta)v(x_i - H/2, t_n)\]
\[-\frac{\sigma^2}{2}v'(x_i - H/2, t_n)) \right).\]  \hspace{1cm} (3.62)

Taylor expanding the exact solution about \(x_i\), Eq. (3.61 - 3.62) become,
\[
\frac{1}{H} \left( F(v(x_i - H/2, t_n)) - F(v(x_i + H/2, t_n)) \right)
\]
\[= \frac{1}{24} \left( 12\sigma^2 v''(x_i, t_n) - \beta \left( 24v'(x_i, t_n) + H^2 v^{(3)}(x_i, t_n) \right) \right) + \gamma(24v(x_i, t_n)\]
\[+ 24x_i v'(x_i, t_n) + 3H^2 v''(x_i, t_n) + H^2 x_i v^{(3)}(x_i, t_n)) + O(H^2),\]  \hspace{1cm} (3.63)

77
\[
\frac{1}{\Delta x - H} \left( F(v(x_i - \Delta x + H/2, t_n)) - F(v(x_i - H/2, t_n)) \right) \\
= \frac{1}{24} \left( 6\sigma^2 \left( 2v''(x_i, t_n) - \Delta x v^{(3)}(x_i, t_n) \right) - \beta \left( 24v'(x_i, t_n) + 4\Delta x^2 v^{(3)}(x_i, t_n) \\
+ H^2 v^{(3)}(x_i, t_n) - 2\Delta x (6v''(x_i, t_n) + H v^{(3)}(x_i, t_n)) \right) + \gamma \left( 24v(x_i, t_n) \\
+ 24x_i v''(x_i, t_n) + 3H^2 v''(x_i, t_n) - 4\Delta x^3 v^{(3)}(x_i, t_n) + H^2 v^{(3)}(x_i, t_n) \\
- 2\Delta x (12v'(x_i, t_n) + 3H v''(x_i, t_n) + 6x_i v''(x_i, t_n) + H^2 v^{(3)}(x_i, t_n) \\
+ Hx_i v^{(3)}(x_i, t_n)) + 4\Delta x^2 (3v''(x_i, t_n) + (H + x_i) v^{(3)}(x_i, t_n)) \right) \right) \\
\right) \\
+ O(\Delta x^2) .
\] (3.64)

Substituting Eqs. (3.63) - (3.64) and Eqs. (3.59) - (3.60) into Eqs. (3.53) - (3.54), the equations for the truncation errors now read,

\[
T_i^n = \frac{1}{H} \left( F^n_{i,L} - F^n_{i,R} \right) - \frac{1}{H} \left( F(v(x_i - H/2, t_n)) \right) + O(\Delta t) \\
= -\frac{1}{12} \Delta x (\Delta x - H)(\beta - \gamma x_i) v^{(3)}(x_i, t_n) + O(\Delta x^2) + O(\Delta t) .
\] (3.65)

\[
T_{i-1/2}^n = \frac{1}{\Delta x - H} \left( F^n_{i-1,R} - F^n_{i,L} \right) - \frac{1}{\Delta x - H} \left( F(v(x_i - \Delta x + H/2, t_n)) \right) \\
- F(v(x_i - H/2, t_n)) \right) + O(\Delta t) \\
= \frac{1}{24} \Delta x H (2\beta + \gamma (\Delta x - 2x_i)) v^{(3)}(x_i, t_n) + O(\Delta x^2) + O(\Delta t) .
\] (3.66)

There are two interesting limits for the above truncation errors. The first limit corresponds to infinitely small tooth size \((H \to 0)\):

**Limit 1**

\[
\lim_{H \to 0} T_i^n = -\frac{1}{12} \Delta x^2 (\beta - \gamma x_i) v^{(3)}(x_i, t_n) + O(\Delta x^2) + O(\Delta t) ,
\] (3.67)

\[
\lim_{H \to 0} T_{i-1/2}^n = O(\Delta x^2) + O(\Delta t) .
\] (3.68)
In the second limit, the ratio between the size of each simulation unit (3H, since in the current setting the size of the teeth equals the size of the buffers) and the size of each big cell (Δx) is fixed at α (e.g. when α = 0.1 we are running microscopic simulations in only 10% of the spatial domain). At this limit, Limit 2

\[
\lim_{H \to a\Delta x/3} T_i^n = \frac{1}{36}(\alpha - 3)\Delta x^2(\beta - \gamma x_i)v^{(3)}(x_i, t_n) + O(\Delta x^2) + O(\Delta t) \quad (3.69)
\]

\[
\lim_{H \to a\Delta x/3} T_{i-1/2}^n = \frac{1}{72}\alpha\Delta x^2(2\beta + \gamma(\Delta x - 2x_i))v^{(3)}(x_i, t_n) + O(\Delta x^2) + O(\Delta t) . \quad (3.70)
\]

In practice, we set the effective time stepper size (including the projection) to be \(O(\Delta t) \sim O(\Delta x^2)\). From Eq. (3.67)-(3.71), we can see that the truncation errors for both of the two limits are of order \(O(\Delta x^2)\). Therefore, the consistency for the patch dynamics scheme is second order.

**Numerical Results**

To numerically validate that the order of consistency for the patch dynamics scheme is order two, we compare solutions of the patch-dynamics scheme (using a fine-grid finite volume scheme as the microscopic simulator) with the solutions of a reference scheme which is also a fine-grid finite volume scheme. In the reference scheme, the spatial domain is divided into 963 fine bins with bin width being \(dx = 2/963\). The microscopic time step is set to be \(dt = 2dx^2\). As Fig. 3.7a shows, starting from the same Gaussian-like initial condition, we evolve both the patch-dynamics scheme and the reference scheme for some time (\(t = 0.086\), or equivalently 10000 time steps in the reference scheme), and then compute the differences between the solutions of these two schemes in order to construct the subsequent log-log plot for the order of consistency analysis.
For the patch-dynamics scheme, two limits analyzed in the previous section have been numerically simulated. The first limit corresponds to cases when the tooth size is very small. In practice, the tooth size in this limit is set to be equal to the fine bin size in the reference scheme ($dx = 2/963$). The size of the buffers is set to be the same as the one of the teeth. By fixing the size of the teeth and buffers, we studied cases with different sizes of big cells and therefore different sizes of gaps. Recall that, as Fig. 3.3 shows, each big cell is of size $\Delta x$, which is equal to the sum of the sizes of a pair of tooth and gap. In the second limit, instead of fixing the teeth size to be of a very small value, we keep the ratio between the sizes of the simulation unit and the big cell fixed while varying both of them simultaneously. The second limit is closely related to practical cases, because in practice we need to use teeth with some finite size in order to capture meaningful macroscopic properties.

For each limit, we compute the differences between the average densities inside the teeth and the gaps of the patch dynamics scheme and the average densities inside the corresponding regions of the reference scheme. We define the errors for the teeth and gaps as the L2 norm of these differences:

\[
\|U_t - U_{ref}\| = \sqrt{\frac{1}{N+1} \sum_{i=0}^{N} (U^t_i - U^i_{ref})^2} \tag{3.71}
\]

\[
\|U_g - U_{ref}\| = \sqrt{\frac{1}{N} \sum_{j=1/2}^{N-1/2} (U^g_j - U^j_{ref})^2} \tag{3.72}
\]

where, $U^t_i$ and $U^g_j$ denote the average densities in the teeth and gaps for the patch-dynamics scheme, while $U^t_{ref}$ and $U^j_{ref}$ denote the average densities inside the corresponding regions of the reference scheme. The detailed results of the different cases in these two limits are shown in Table 3.1.

To find the order of consistency estimates, we generated the $\log(\text{error})$ v.s. $\log(\Delta x)$ plot and performed linear fitting. The fitted plots for the two limits are shown in
Table 3.1: Numerical results of the order of consistency analysis for the two limits. 
N is the number of big cells in the patch-dynamics scheme. ∆x denotes the size for the big cells. The error terms are defined in Eq. (3.71-3.72).

<table>
<thead>
<tr>
<th>N</th>
<th>∆x</th>
<th>Limit 1</th>
<th>Limit 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>1.82E-1</td>
<td>6.76E-03</td>
<td>9.12E-03</td>
</tr>
<tr>
<td>15</td>
<td>1.33E-1</td>
<td>4.07E-03</td>
<td>5.01E-03</td>
</tr>
<tr>
<td>21</td>
<td>9.52E-2</td>
<td>2.29E-03</td>
<td>2.63E-03</td>
</tr>
<tr>
<td>31</td>
<td>6.45E-2</td>
<td>1.15E-03</td>
<td>1.20E-03</td>
</tr>
<tr>
<td>41</td>
<td>4.88E-2</td>
<td>6.61E-04</td>
<td>6.92E-04</td>
</tr>
<tr>
<td>61</td>
<td>3.28E-2</td>
<td>2.88E-04</td>
<td>3.02E-04</td>
</tr>
</tbody>
</table>

Table 3.2: Order of consistency estimates for the two limits. 95% confidence intervals for the order estimates are shown in the brackets.

<table>
<thead>
<tr>
<th></th>
<th>Limit 1</th>
<th>Limit 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_t$</td>
<td>1.839 (1.715, 1.964)</td>
<td>1.818 (1.732, 1.903)</td>
</tr>
<tr>
<td>$U_g$</td>
<td>1.991 (1.939, 2.043)</td>
<td>1.974 (1.948, 2.001)</td>
</tr>
</tbody>
</table>

Fig. 3.7 (b) and (c). The slope of the fitted lines together with their respective 95% confidence interval estimates are shown in Table 3.2. It shows that the numerical order of consistency estimates are all close to 2, which is consistent with the analysis performed in the previous section. The slight deviation is most likely due to the discontinuity at the origin and the boundary conditions.

3.5.3 Patch dynamics scheme for the agent-based model

Finally, we apply the patch-dynamics scheme to the agent-based model (discussed in section 3.2.1) for which the patch dynamics scheme is designed. We divide the spatial domain into $N = 21$ big cells, and set the ratio of the size of the simulation unit to the big cell ($\alpha$) to be equal to 0.2. This means we are running the microscopic simulations in 20% of the spatial domain. Our macroscopic properties are the average
agent densities inside the teeth and gaps. Inside each simulation unit the tooth and
the two buffers are of equal size, and each of them has been divided into 10 fine bins.
Based on the lifting procedure discussed in section 3.4.1 microscopic agent densities
are assigned to these fine bins. We then convert these density values into numbers of
agents based on the total amount of agents available in the system, in this case the
total number of agents we use is \( N_{agents} = 3 \times 10^6 \). Once the number of agents for each
fine bin is assigned, we start the agent-based simulations. As illustrated in section
3.4.2 the coarse time step size \( \delta t \) - the time we can run the agent-based simulator
before we stop for reconstruction - is closely related to the size of the buffers. This is
because the buffers are gradually “contaminated” due to the boundary artifacts, so
some time before the artifacts get propagated to the tooth we need to stop. In this
case, the coarse time step size \( \delta t \) is chosen to be \( 2 \times 10^{-3} \). Within this time interval,
the chance for any agent to travel a distance of the size of a buffer or more is very
low. In this way, agents outside each simulation unit will not affect the solution of the
tooth (although they do affect the solution of the buffers), and the tooth is therefore
protected. Along the simulation process, the fluxes of agents across the edges of the
teeth are kept track of. In section 3.4.2 the fluxes are updated based on equations of
the artificial microscopic simulator (a PDE). In the agent-based case we keep track
of the fluxes by counting the number of agents going across those edges.

Due to the stochastic nature of the agent-based modeling, we also need to reduce
the noise for the macroscopic variables. The agent-based simulations are therefore
repeated inside each simulation unit for \( N_{realizations} = 20 \) times before the restriction
procedure is applied. At the end of each coarse time step, we perform the restriction
procedure to compute the averaged fluxes at the edges of the teeth. Based on these
fluxes we update the number of agents inside the teeth as well as the neighboring gaps,
and compute the corresponding densities based on the numbers of agents. Following
the same procedure discussed in section 3.4.4 based on the macroscopic solutions
(the densities) at the start and the end of each coarse time step, we estimate local
time derivatives of the macroscopic variables and project them forward in time. What
allows us to do this is the smoothness of these coarse variables in time domain. In
this case, we run the agent-based simulations for one coarse time step, and jump nine
steps ahead. In other words, we are running the agent-based simulations in 10% of
the temporal domain.

Fig. 3.8 shows four snapshots along a patch dynamics agent-based trajectory on its
way to its steady state. The solutions for the agent-based patch-dynamics simulations
are plotted in blue (teeth) and red (gaps) respectively. To compare the performance,
the solutions of a fine-grid finite volume scheme for the continuum model are also
plotted (in blue curve) as a reference. The analytical steady state for the continuum
model is shown in Fig. 3.8(d). Overall, the solutions of the agent-based patch-
dynamics simulations match well (although not perfectly) with the solutions of the
continuum model. The deviations are most likely due to the difference between the
continuum and the agent-based models. In Fig. 3.8(d) we can see that the steady
state of the patch-dynamics agent based simulations match better with the steady
state of the full scale agent-based simulations compared to the analytical steady state
of the continuum model.

Regarding the computational savings, the patch-dynamics agent-based simulations
require the expensive agent-based computations to be performed in only 10% of the
temporal domain and 20% of the spatial domain compared to the full scale agent-
based simulations.

3.6 Conclusions

We describe a patch dynamics scheme for agent-based problems. This scheme ap-
proximates an unavailable effective equation over macroscopic time and length scales,
when only a microscopic agent-based simulator is given. It only uses appropriately initialized simulations of the agent-based model over small subsets (patches) of the spatial-temporal domain, therefore significantly reduces the computational cost. Because this scheme mimics a finite volume scheme for the underlying effective equation, it is conservative by construction. Since it is often not possible to impose macroscopically inspired boundary conditions on a microscopic agent-based simulation, we have used buffer regions around the patches, which temporarily shield the internal region of the patches from boundary artifacts. We have performed the order of accuracy analysis based on the continuum version (a PDE) of the agent-based model, and demonstrated the effectiveness of this scheme for agent-based computations through the application to a financial market agent-based model. In this study the agent-based model was mainly used for illustration purposes to demonstrate the effectiveness of this approach. Factors which affect the performance of the patch dynamics scheme are not studied in detail. It will be interesting for future studies to investigate the effects of the noise (in the teeth and gaps) to the performance of this scheme.
Figure 3.6: Comparison of the solutions of different schemes. (a) The common Gaussian-like initial condition for the three schemes. (b) Solutions of the three schemes at time $4 \times 10^3 \Delta t$. (c) Comparison of the errors of the patch-dynamics scheme and the coarse finite volume scheme at time $4 \times 10^3 \Delta t$. (d) Steady state solutions. (e) Comparison of the errors of the patch-dynamics scheme and the coarse finite volume scheme at steady state. In (a), (b) and (c), The blue curve denotes the solution of the fine finite volume scheme; the red curve denotes the solution of the coarse finite volume scheme; the blue dots denote the solutions of the teeth for the patch dynamics scheme. In (c) and (e), the blue dots denotes the errors (in teeth) of the patch dynamics scheme; the red cross marks denotes the errors of the coarse finite volume scheme.
Figure 3.7: (a) The blue curve corresponds to the Gaussian-like initial condition for both the patch-dynamics scheme and the reference scheme. The red curve corresponds to the solution of the reference scheme at $t = 0.086$. This corresponds to running the reference scheme for 10000 time steps. The patch-dynamics scheme was running for the same amount of time. The differences of the average densities inside the teeth and gaps between these two schemes were taken at this time point to generate the log-log plots. The model parameters used in these simulations are the same as the ones listed in Fig. 3.6. (b) Log(Error) v.s. log ($\Delta x$) plot for Limit 1. (b) Log(Error) v.s. log ($\Delta x$) plot for Limit 2. In (b) and (c), the dots denote the log(Error) v.s. log($\Delta x$) data points for the teeth, while the cross marks denote the corresponding data points for the gaps. The blue and red curves denote the linear fit for the data sets of the teeth and gaps respectively. The R-square values for the fits are all very close to one (0.998 and above). The slopes of the fitted lines (order of consistency estimates) together with their respective 95% confidence intervals are presented in Table 3.2.
Figure 3.8: Snapshots along a patch dynamics agent-based trajectory on its way to the steady state. The average agent densities inside the teeth are denoted by red dots, while the ones inside the gaps are denoted by blue dots. To compare the performance, solutions of a fine-grid ($N = 629$) finite volume scheme for the continuum model are plotted in (a-c) in blue curve. In (d) the analytical steady state of the continuum model is plotted in blue curve while the approximated steady state of the full agent-based simulations is plotted in red curve. Parameter values for patch dynamics scheme are as follows: $N = 21$, $\alpha = 0.2$, $g = 1$, $\epsilon^+ = 2.9 \times 10^{-3}$, $\epsilon^- = -2.89 \times 10^{-3}$, $\gamma = 1 \times 10^{-3}$, $N_{\text{agents}} = 3 \times 10^6$, $N_{\text{realizations}} = 20$, $\delta t = 2 \times 10^{-3}$, $M = 9$. 
Chapter 4

Diffusion maps assisted coarse-graining approach for an animal swarming model

4.1 Introduction

A persistent feature of agent-based problems is the emergence of macroscopic, collective behavior from the interactions of microscopic agents (e.g. cells, individual animals in a population) between themselves and with their environment. This implies that the description of macroscopic behavior can somehow be deduced from the microscopic rules. However, as discussed in previous chapters, identifying the “right” coarse level descriptors is a nontrivial task, which usually requires extensive observations and physical intuitions about the system.

Aiming to overcome this limitation and to develop a systematic approach in identifying the coarse descriptors which can rely less on intuition, in this chapter, we use a data mining approach to help identify the coarse variables required. In particular, based on an agent-based animal swarming model developed by Couzin et al. [18],
we use diffusion maps (DMAP) to capture key features of the collective motion of the animals to understand how a group of socially interacting animals switches its direction of motion. Based on the identified DMAP coarse variables, we apply the EF approach and construct a reduced stochastic differential equation (SDE) model which allows us to efficiently study the stochastic direction switching behavior of a group of animals.

This chapter is organized as follows. In Section 4.2, the agent-based animal swarming model is introduced. In Section 4.3, the implementation of the DMAP approach and the choice of a similarity measure between nearby system snapshots are discussed. In Section 4.4, the extraction of a two dimensional effective description of the collective animal motion is presented. In Section 4.5, the construction of the reduced SDE model using EF approach and the DMAP coarse variables is discussed, followed by the computation of the mean escape time of the rare event - in this case the time (on average) it takes for the group to switch from upward motion to downward motion and vice versa. We conclude and discuss future work in Section 4.7.

4.2 Model Description

The agent-based animal swarming model we study in this chapter was developed by Couzin et al. [18]. We use this model to study how collective decisions are made when two sub-groups of leaders (informed individuals) have different opinions about the locations of the food resources. We consider a group of \( N \) individuals. Each individual with position vector \( \mathbf{c}_i(t) \), direction vector \( \mathbf{d}_i(t) \), and speed \( s \) (assuming all the individuals have the same constant speed) attempt to maintain a minimum distance \( r_p \) between itself and other individuals. This avoidance is the highest priority in each individual’s rule of motion. It is reflected in the model by defining a repulsion zone, \( \Omega_p \), which is a local neighborhood of radius \( r_p \). The desired direction of travel
\( \mathbf{d}_i \) is updated as follows,

\[
\mathbf{d}_i(t + \Delta t) = -\sum_{j \in \Omega_p, j \neq i} \frac{\mathbf{c}_j(t) - \mathbf{c}_i(t)}{|\mathbf{c}_j(t) - \mathbf{c}_i(t)|}.
\] (4.1)

If no other individuals are detected within the repulsion zone, then the individual is attracted towards, and align their direction of travel with, other individuals in a larger neighborhood called the *attraction zone*, \( \Omega_a \). The region has radius \( r_a \). The resulting desired direction of travel is given by,

\[
\mathbf{d}_i(t + \Delta t) = \sum_{j \in \Omega_a, j \neq i} \frac{\mathbf{c}_j(t) - \mathbf{c}_i(t)}{|\mathbf{c}_j(t) - \mathbf{c}_i(t)|} + \sum_{j \in \Omega_a} \frac{\mathbf{d}_j(t)}{|\mathbf{d}_j(t)|}.
\] (4.2)

Here \( \mathbf{d}_i(t + \Delta t) \) is converted to the corresponding unit vector \( \hat{\mathbf{d}}_i(t + \Delta t) = \frac{\mathbf{d}_i(t + \Delta t)}{|\mathbf{d}_i(t + \Delta t)|} \). In the model there are three different subgroups of agents; we denote them as Group A, Group B and Group U respectively. Group A and Group B are the informed agents with different preferred directions (simulated as unit vectors \( \mathbf{g}_A \) and \( \mathbf{g}_B \) respectively) representing, for example, the directions to two different known food resources. All the other individuals (Group U) are naive and have no preference to move in any particular direction. Informed individuals balance their preferred directions and the social interactions with others with weighting factor \( \omega \),

\[
\mathbf{d}'_i(t + \Delta t) = \frac{\hat{\mathbf{d}}_i(t + \Delta t) + \omega \mathbf{g}_i}{|\hat{\mathbf{d}}_i(t + \Delta t) + \omega \mathbf{g}_i|}.
\] (4.3)

In this study we let Group A agents have preferred directions pointing upward (i.e. \( \mathbf{g}_A = (0, 1)^T \) in the 2D physical space), and let Group B agents have preferred directions pointing downward (\( \mathbf{g}_A = (0, -1)^T \)). When the weighting factor \( \omega \) is relatively small, the informed individuals have weak desire to move toward the preferred directions, so does the entire group. As \( \omega \) increases, the level of influence by the preferred direction increases while influence due to the social interactions decreases. At in-
intermediate values of $\omega$, the group randomly switches between upward motion and downward motion. As $\omega$ further increases, the group breaks into two parts each consists of one informed group plus a couple of uninformed individuals. In this study we focus on the case when $\omega$ is at some intermediate value, and study how the group switches its direction of motion.

To account for an individual’s error in perception and motion, the direction of motion $d'_i(t + \Delta t)$ is rotated by a Gaussian random angle with a standard deviation $\sigma$, resulting in $d''_i(t + \Delta t)$. There is also a constraint on the rate of turning (the turning angle per unit time $\eta$). If the angle between $d''_i(t + \Delta t)$ and the individual’s previous direction of motion $d''_i(t)$ is less than $\eta \Delta t$, then the desired direction of motion at time $t + \Delta t$ is set to $d''_i(t + \Delta t)$. Otherwise, $d''_i(t + \Delta t)$ is obtained by turning $d'_i(t + \Delta t)$ by an angle $\eta \Delta t$ towards $d''_i(t + \Delta t)$. Finally, the position of each individual is updated as follows,

$$c_i(t + \Delta t) = c_i(t) + d''_i(t + \Delta t)s\Delta t$$

where, $s$ denotes the speed of each individual (we set it to be a constant). A sample snapshot of the simulation results is shown in Fig. 4.1.

### 4.3 Similarity measure between snapshots for the DMAP construction

The details of the DMAP construction have been discussed before in Section 1.2. A crucial step in the DMAP construction is the selection of the appropriate similarity measure between data snapshots. The Euclidean distance is a good choice for many data sets; however, often a more complex distance is required. For example, if the data has one or more underlying symmetries that need to be factored out before measuring the distance, Euclidean distance is not a good choice. For the agent-based
animal swarming model we study in this chapter, there are three distinct subgroups of agents. Each subgroup is invariant under a permutation of \( N \) indistinguishable agents. For this reason, the Euclidean distance between the data snapshots is not an appropriate choice for this problem.

Optimization-based distance measures like the Earth Mover’s distance [83] can be applied to compute the similarity measure between data snapshots with indistinguishable agents. With this technique, one considers all possible permutations of the particles in each pair of data points and computes all of the corresponding pairwise distances. The minimum of these distances is then used as the similarity measure between the two data snapshots. The disadvantage of this approach is the high computational cost, since the enumeration of all possible \( n! \) permutations becomes impractical even when \( n \) is a modest double digit number.
Figure 4.2: Illustration of the intermediate coarse variables for the similarity measure. There are three different subgroups, each with two properties, so in total there are 6 property-groups. For each property-group we use the first several statistical moments (mean, variance and covariance in x- and y- coordinates) as a coarse representation, which gives us a $5 \times 6 = 30$ dimensional intermediate coarse variable. Each element of this vector is weighted properly so that every element will have similar magnitude. The choice for the weighting factors are made based on observations of the simulation results, and the detailed values of these factors are shown in the top-right corner of this figure.
In this study, we use the first few statistical moments to compute the similarity measure between each pair of data snapshots. Before we present the details of the similarity measure computation, several terms need to be defined first. The microscopic variables of the agent-based animal swarming model are the positions and directions of the individuals in the 2D physical space. So for \( N \) individuals the microscopic variables have dimension \( 4N \). We define the intermediate coarse variables as \( x_i \), which will be used for the computation of the similarity measure \( k_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\epsilon}\right) \) between each pair of data points (a data point here actually refers to a data snapshot which contains the positions and directions for the \( N \) agents in the system). The reason we call \( x_i \) the “intermediate coarse variables” is because later on we will call the first several DMAP eigenvectors as our ultimate coarse variables.

The construction of the intermediate coarse variables \( x_i \) has been illustrated in the chart shown in Fig. 4.2. There are three different subgroups, each with two properties. So in total there are 6 property-groups. For each property-group we decide to use the first several statistical moments (mean, variance and covariance in x- and y-coordinates) as the coarse representation, which gives us a \( 5 \times 6 = 30 \) dimensional intermediate coarse variable. Before we use it for the similarity computation, each element of this vector is weighted properly so that every element has similar magnitude. The choice for the weighting factors is made based on observations of the simulation results, and the detailed values of these factors are shown in the top-right corner of Fig. 4.2.

4.4 Resulting two dimensional discription

Based on the similarity measure chosen in the previous section, we apply DMAP to the simulation data of three different cases: (1) symmetric case, with equal number of individuals in each of the two informed groups \( (N_A = 7, N_B = 7, N_U = 36) \); (2)
asymmetric case, with slightly more individuals in one of the two informed groups
\((N_A = 8, N_B = 7, N_U = 35)\); (3) asymmetric case but with lower proportion of
informed individuals \((N_A = 6, N_B = 5, N_U = 39)\).

To generate the raw data set for the DMAP construction, we run the agent-based
simulations and save the data snapshots at a frequency of one snapshot per every five
time steps. For each case, we generate 60000 data snapshots, with each data snapshot
consists of positions and directions for all the individuals. However, to utilize the
symmetry of the problem, not all the data snapshots are generated directly from
the agent-based simulations. For case (1), because the problem is symmetric about
both the x- and the y- axis, we generated 15000 data snapshots from the agent-based
simulations and then performed a reflection about x-axis, y-axis and also about both
x- and y- axis. However, caution must be taken for this case when we perform the
reflection about the x- axis: in addition to reflecting the positions and the directions
of each agents, the identities of the agents of the two informed subgroups also need
to be swapped in order to create the correct image. For case (2) and (3), because
the problem is only symmetric about the y-axis, we generated 30000 data snapshots
from the agent-based simulations, and then did the reflection about the y-axis.

Once the raw data set is generated, we compute the intermediate coarse variables
defined in the last section - the rescaled first few statistical moments of positions
and directions for each subgroup. Following the procedures discussed in Section
1.2, the DMAP embeddings are obtained. Fig. 4.3 shows the DMAP embedding
and the associated representative microscopic snapshots for the symmetric case (case
(1)). In the embedded two dimensional DMAP space, the first meaningful diffusion
maps eigenvector \((\psi_2)\) corresponds to the up-down direction of motion in the original
physical space, and the second meaningful DMAP eigenvector \((\psi_3)\) corresponds to
the left-right direction of motion in the original physical space. For example, for the
DMAP embedding located on the middle left of the 2D map \((\psi_2 \sim -5, \psi_3 \sim 0)\), the
Figure 4.3: Diffusion maps embedding and the associated representative microscopic snapshots for the symmetric case (case (1)). In the embedded two dimensional diffusion maps space, the first meaningful diffusion maps eigenvector \( (\psi_2) \) corresponds to the up-down direction of motion in the original physical space, and the second meaningful DMAP eigenvector \( (\psi_3) \) corresponds to the left-right direction of motion in the original physical space. For each pair of displayed snapshots, the left one presents the direction of motions for the agents; and the right one shows the relative positions of the agents.
direction of collective motion is vertically upward; for the DMAP embedding located on the bottom of the 2D map ($\psi_2 \sim 0, \psi_3 \sim -0.008$), the direction of collective motion is to the left; at the center of the map ($\psi_2 \sim 0, \psi_3 \sim 0$), the group “jiggles” and there is no consensus among the agents about where to move. There are also interesting microscopic features for the different migration states classified with the DMAP embedding. As Fig. 4.4 (a) and (b) show, when the group is in the state of upward motion, the group is elongated. The informed agents (red circles) who prefer the upward direction of motion tend to cluster at the moving front of the group to

Figure 4.4: Sample snapshots of different migration states (case 1). (a-b): upward motion. (c-d): rightward motion. (e-f): “jiggling” state. The right column presents the directions of the agents; the left column presents the positions of the agents.
lead the migration. The other subgroup of informed agents (black triangles) who prefer the downward direction of motion tend to spread and stay at the tale of the group. The two subgroups of informed agents are separated by the uninformed ones (blue asterisks). When the group is in the state of rightward motion, as Fig. 4.4 (c) and (d) show, the two subgroups of informed agents tend to spread at the two sides of the bulk of the uninformed agents, with some of the agents from one informed subgroup occupying the moving front of the entire group to lead the migration. In the “jiggling state”, the agents from the three subgroups are mixed with each other, the whole group “jiggles” and there is no consensus among the agents about which direction to move.

The DMAP embedding gives us the same interpretation for the other two cases. Based on the DMAP embedding computed for all the three cases, we plot the log-arithmetic of the densities in the DMAP space. As Fig. 4.5 shows, in the symmetric case (case 1), the group spends most of its time moving either upward or downward with equal probability; in the asymmetric case (case 2), the group spends more time in traveling along the preferred direction of one of the informed subgroups which contains more individuals; as the proportion of uninformed the individuals increases (case 3), the group spends less time along the preferred direction of the dominant informed subgroup.

Figure 4.5: Plots of log-densities for the three cases studied.
4.5 Coarse Graining: SDE reduced model approximation

In the symmetric case, as Fig. 4.5(a) shows, the group spends most of its time in either the state of upward motion or downward motion. Due to the intrinsic stochastic nature of the motion, detailed simulation shows that the group switches its direction of motion (from upward to downward and vice versa) at random time intervals. An interesting question naturally arises: how much time (on average) does it take for this switch to happen? In this section, based on the two dimensional DMAP embedding found in the last section, a reduced stochastic differential equation (SDE) model is constructed using the EF approach to address this question in a computationally efficient way.

The reduced two dimensional SDE model is in the following form:

\[
\begin{align*}
    d\psi_2 &= \mu_1 dt + \sigma_1 dW_1 \\
    d\psi_3 &= \mu_2 dt + \sigma_{21} dW_1 + \sigma_{22} dW_2
\end{align*}
\] (4.5)

where \( W_1 \) and \( W_2 \) denote independent Wiener processes. \( \mu_1, \mu_2 \) are the drift coefficients, and \( \sigma_1, \sigma_{21}, \sigma_{22} \) are the diffusion coefficients. The following equation can be used to compute these coefficients [56]:

\[
\begin{align*}
    \mu_i(\psi_2, \psi_3) &= \lim_{\Delta t \to 0} \frac{\langle \Psi_i(t + \Delta t) - \Psi_i(t) \rangle}{\Delta t} \bigg|_{\Psi_2(t) = \psi_2, \Psi_3(t) = \psi_3} \\
    D_{ij}(\psi_2, \psi_3) &= \lim_{\Delta t \to 0} \frac{\langle (\Psi_i(t + \Delta t) - \Psi_i(t)) (\Psi_j(t + \Delta t) - \Psi_j(t)) \rangle}{\Delta t} \bigg|_{\Psi_2(t) = \psi_2, \Psi_3(t) = \psi_3} ;
\end{align*}
\] (4.6)

where the angled brackets denote expectation and \( D_{ij} \) are the diffusion coefficients of the corresponding Fokker-Planck equation of the SDE (4.5). The drift coefficients \( \mu_i \) govern the deterministic part of the macroscopic dynamics, while the diffusion coefficients \( D_{ij} \) represent the stochastic aspect of the SDE. We estimate the drift and
diffusion coefficients from a finite-length simulation with finite $\Delta t$,

$$
\begin{align*}
\mu_i(\psi_2, \psi_3) & \approx \frac{\langle \Psi_i(t + \Delta t) - \Psi_i(t) \rangle}{\Delta t} \bigg|_{\Psi_2(t) = \psi_2, \Psi_3(t) = \psi_3} \\
D_{ij}(\psi_2, \psi_3) & \approx \frac{\langle (\Psi_i(t + \Delta t) - \Psi_i(t))(\Psi_j(t + \Delta t) - \Psi_j(t)) \rangle}{\Delta t} \bigg|_{\Psi_2(t) = \psi_2, \Psi_3(t) = \psi_3} \\
& - \Delta t \mu_i(\psi_2, \psi_3) \mu_j(\psi_2, \psi_3)
\end{align*}
$$

(4.7)

where the last term in Eq. (4.7) helps to correct for the finite size of $\Delta t$. The diffusion coefficients in Eq. (4.5) are calculated via Cholesky decomposition as follows

$$
\sigma_1 = \sqrt{D_{11}}, \sigma_{21} = \frac{D_{12}}{\sigma_1}, \sigma_{22} = \sqrt{D_{22} - \sigma_{21}^2}.
$$

(4.8)

To numerically estimate the drift and diffusion coefficients using the above equations, the two-dimensional DMAP space is discretized into 10000 (100 by 100) small “boxes”. For each small “box”, we locate several instances arising in long agent-based simulations, and then for each of these instances, record DMAP coordinates $\Delta t$ later. Averaging results as shown above provides numerical estimates of the drift and diffusion coefficients. After the drift and diffusion coefficients are computed, the reduced SDE model is obtained. Fig. 4.6 compares time histories of the agent-based simulations (after converting the solutions to DMAP variables using the Nyström formula discussed in Section 1.2.2) with the ones of the reduced SDE model. The solutions of the two models are qualitatively similar to each other.

### 4.6 Mean exit time computation

Once the reduced SDE model is available, we are able to efficiently compute various macroscopic level properties such as the mean exit time of rare events. In this problem, an interesting event to investigate is the exit of the system from “vertical” upward motion or downward motion to some “non-vertical” state of motion. Quantitatively,
Figure 4.6: A comparison between the solutions of the agent-based model (a,c) and the ones from the reduced SDE model (b,d).

Table 4.1: Comparison of the statistics (the first three moments) of the mean exit time distribution between the original ABM and the reduced SDE models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDE</td>
<td>759.4</td>
<td>677.9</td>
<td>1.874</td>
</tr>
<tr>
<td>ABM</td>
<td>511.2</td>
<td>441.2</td>
<td>1.771</td>
</tr>
</tbody>
</table>

as Fig. 4.7 illustrates, we define the mean exit time here as the average time it takes for the system to travel from the red dashed line to the green dashed line. Fig. 4.8 shows the distributions of the exit times for the reduced SDE and the original agent-based model (ABM). They qualitatively resemble each other. The statistics of the exit times of the two models are computed and summarized in Table 4.1 and the results are reasonably close. Moreover, simulation of the SDE for $1 \times 10^6$ only takes about half a second, while the original agent-based model requires over 37 minutes for the same number of time steps (a difference factor of more than 70).
Figure 4.7: The mean exit time here is defined as the mean time it takes for the system to travel from the region marked by the red dashed line to the region marked by the green dashed line. (a) Log-probability density plot for the symmetric case; (b) sample time series of the agent-based simulations in the DMAP space.

Figure 4.8: Comparison of the mean exit time distributions from (a) the effective reduced SDE and (b) the original ABM.
4.7 Conclusion and future work

In the work presented in this chapter, we explored the use of nonlinear dimensionality reduction techniques, in particular, diffusion maps (DMAP), to systematically extract an effective macroscopic description for a complex agent-based animal swarming model. The advantage of this approach is that it does not require extensive physical intuition about the problem, which is not available in general.

The first few statistical moments of the properties of each subgroup were used as the “intermediate coarse variables” to compute the similarity measure between data snapshots. Using this similarity measure, the constructed low dimensional (2D) DMAP embedding characterized the up-down and left-right directions of motion for the group of swarming agents. Based on the identified DMAP coarse variables, we showed how to use the EF approach to construct an effective reduced model, here in the form of a two dimensional stochastic different equation (SDE). The sampled time series of the reduced SDE solution as well as the statistics of the exit time distribution closely resemble those obtained from the original agent-based simulations (but with significantly higher computational cost).

As a preliminary study, it is promising that the DMAP variables identified in this work successfully captured macroscopic level features about the collective direction of motion for the group of simulated agents. However, in this particular case, these coarse features can also be “deduced” by human thought (i.e. we can roughly tell the collective direction of motion by looking at a given data snapshot). Therefore, to make this approach more useful, it is important for future work to study those more “difficult” cases, the ones whose coarse features are not obvious to a human being. An interesting task is to apply this approach to study how a coherent group of migrating animals break into separate groups, and the coarse level dynamics associated with the fragmentation process. This can be achieved by running multiple agent-based
simulations and save the data snapshots just before fragmentation occurs. The DMAP embedding is then constructed based on the saved data set.

In the current study, the lifting step is done by running long time agent-based simulations and then assign instances to the corresponding small boxes of the DMAP space. The drawback of this approach is some boxes may have insufficient data points for the ensemble average computations. Therefore, it is important for future research to design more advanced lifting procedure to efficiently generate the microscopic configurations, especially for those small boxes whose instances arise less frequently. With more effort, the DMAP based approach has the potential to become a common method in extracting effective coarse level features as well as in reducing the high computational cost for the agent-based simulations.
Chapter 5

Substrate-dependent control of ERK phosphorylation can lead to oscillations

This chapter is based on work published in collaboration with Prof. Stanislav Y. Shvartsman and Prof. Ioannis G. Kevrekidis in [60].

5.1 Introduction

The ERK/MAPK pathway is organized around a cascade of three kinases: Raf, the topmost kinase in the cascade, phosphorylates and activates MEK, which goes on to phosphorylate and activate ERK [14]. In contrast to MEK, which phosphorylates only ERK, ERK has multiple substrates, which enables it to regulate multiple cellular processes, such as expression of genes and remodeling of cytoskeleton [91]. ERK signaling has been associated with countless developmental events across species [32, 17, 62, 81, 64]. ERK regulates tissue homeostasis and repair [50, 96] in adulthood, and is also related to longevity, memory, and metabolism [73, 4, 66].
Depending on cellular context, the ERK pathway can function as a rheostat, a switch, or an oscillator. Here, we focus on the mechanisms leading to oscillations of ERK activity, which have been observed in two different experimental systems. Previous studies show that the emergence of oscillatory ERK activity is most commonly associated with negative feedback control. In this study, we propose that ERK oscillations can also arise through a different mechanism that relies on substrate-dependent control of ERK phosphorylation. In this mechanism, ERK substrates compete with ERK phosphatases, due to the fact that substrates and phosphatase can bind to the same docking site on the ERK molecule. Recently, substrate-dependent control of ERK phosphorylation has been documented both in vitro and in vivo. Based on the presented computational analysis, we suggest that this effect can lead to oscillations.

Our reasoning was based on a model network where ERK is regulated by opposing activities of ERK kinase and phosphatase and where, in addition, active ERK phosphorylates a substrate, thereby inducing its degradation; a scenario that has been identified in several experimental systems. In this mechanism, substrate is synthesized at a constant rate and degraded via two pathways: constitutive (slow) and ERK-dependent (fast). In such a network, high levels of ERK activity, realized at high levels of ERK phosphorylation, would degrade most of the substrate, making active ERK more available to phosphatases. This would reduce the total level of active, double phosphorylated ERK, reducing the ERK-dependent substrate degradation and therefore enabling the substrate levels to build up. As a consequence of substrate-dependent control of ERK phosphorylation, the increased substrate will compete with phosphatases and reduce the dephosphorylation rate of the double phosphorylated ERK. This would in turn reduce the level of substrate expression, and so on, repeating the cycle of oscillations.
Figure 5.1: (A) Model biochemical network: an enzyme (M) is regulated by kinase (\(E_1\)) and phosphatase (\(E_2\)). Enzymatic activity requires dual phosphorylation. The mechanisms of enzyme phosphorylation and dephosphorylation are distributive. Active enzyme (\(M_2\)) phosphorylates a substrate (\(S\)), thereby increasing the rate of its degradation. The substrate is synthesized at a constant rate and, in the absence of enzyme activity, is degraded via a constitutive degradation pathway. The full model consists of two subsystems: the enzyme regulation module (inside the green dashed box) and the substrate degradation module (inside the red dashed box). (B) One parameter bifurcation diagram for the isolated dual phosphorylation module. (Black solid curves) Stable steady-state branches. (Red dashed curve) Unstable steady-state branch. As the input level is changed this module can be bistable. (C) One-parameter bifurcation diagram for the isolated substrate module (the part of the network in the red box). The nondimensional concentration of the double-phosphorylated enzyme (\(m_2\)) is used one constitutive, and one dependent on \(M_2\). \(C_5\) denotes the complex of \(M_2\) and \(S\).
We use a mathematical model to demonstrate that this mechanism can generate robust periodic oscillations of ERK phosphorylation. Our model neglects several important aspects of ERK regulation in cells, including compartmentalization and feedback control [107]. At the same time, this model emphasizes the functional significance of competing docking interactions, which have been largely ignored in mathematical models of cell signaling. In the context of ERK signaling, computational studies and chemical reaction network theory approaches revealed that competing docking interactions can lead to bistability [57, 19, 65, 75]. In the case of bistability, different phosphostates of ERK compete for the enzymes regulating ERK. We demonstrate that competing docking interactions can also lead to oscillations. In this case, competition is among the binding partners of ERK. In our model, the emergence of oscillations depends on bistability, similar to the canonical van der Pol oscillator [94]. Thus, competing docking interactions provide a rich source of dynamic regimes in biochemical networks.

5.2 Methods

5.2.1 Mathematical model

The model is based on the following network of reactions (Fig. 5.1A):

M0, M1, and M2 denote the unphosphorylated, monophosphorylated, and double phosphorylated forms of the enzyme (ERK); E1, and E2 denote the kinase and phosphatase which regulate this enzyme. C1-4 denote the complexes comprising this enzyme and its binding partners. S is the substrate, synthesized at a constant rate $V$ (as shown in reaction R5) and degraded by two pathways, one constitutive (R6), and one dependent on M2 (R7). C5 denotes the complex of M2 and S. We assume that phosphorylated substrate, $S^*$, does not interact with the enzyme and the rest of the species in the mechanism. As shown in Fig. 5.1A, reactions (R1)-(R4) comprise the
"enzyme regulation module". Reactions (R5)-(R7) constitute the "substrate degradation module". Based on mass action kinetics, we obtain the following system of equations:

\[
\begin{align*}
\frac{d[M_0]}{dt} &= -k_1^+ [M_0][E_1] + k_1^- [C_1] + k_4^+ [C_4], \\
\frac{d[M_1]}{dt} &= -k_2^+ [M_1][E_1] + k_2^- [C_2] + k_1^+ [C_1] - k_4^- [M_1][E_2] + k_4^- [C_4] + k_3^- [C_3], \\
\frac{d[M_2]}{dt} &= -k_3^+ [M_2][E_2] + k_3^- [C_3] + k_2^+ [C_2] - k_5^+ [S][M_2] + (k_5^+ + k_5^-) [C_5], \\
\frac{d[S]}{dt} &= V - k_6 [S] - k_5^+ [S][M_2] + k_5^- [C_5], \\
\frac{d[C_5]}{dt} &= k_5^+ [S][M_2] - (k_5^+ + k_5^-) [C_5], \\
\frac{d[C_1]}{dt} &= k_1^+ [M_0][E_1] - (k_1^+ + k_1^-) [C_1],
\end{align*}
\]
$$\frac{d[C_2]}{dt} = k_2^+ [M_1][E_1] - (k_2^- + k_2^e)[C_2]$$
$$\frac{d[C_3]}{dt} = k_3^+ [M_2][E_2] - (k_3^- + k_3^e)[C_3]$$
$$\frac{d[C_4]}{dt} = k_4^+ [M_1][E_2] - (k_4^- + k_4^e)[C_4]$$
$$\frac{d[E_1]}{dt} = -k_1^+ [M_0][E_1] + k_1^- [C_1] + k_1^e [C_1] - k_2^+ [M_1][E_1] + k_2^- [C_2] + k_2^e [C_2]$$
$$\frac{d[E_2]}{dt} = -k_3^+ [M_2][E_2] + k_3^- [C_3] + k_3^e [C_3] - k_4^+ [M_1][E_2] + k_4^- [C_4] + k_4^e [C_4].$$

The total amounts of the enzyme (M), kinase (E1) and phosphatase (E2) are conserved:

$$[E_1] + [C_1] + [C_2] = E_1^{tot},$$
$$[E_2] + [C_3] + [C_4] = E_2^{tot},$$
$$[M_0] + [M_1] + [M_2] + [C_1] + [C_2] + [C_3] + [C_4] + [C_5] = M^{tot}.$$

### 5.2.2 Nondimensionalization

The system is rendered dimensionless by the following set of transformations:

$$\tau = t \frac{k_3^+ E_2^{tot}}{m_0}$$
$$m_0 = \frac{[M_0]}{M^{tot}}, m_2 = \frac{M_2}{M^{tot}},$$
$$c_1 = \frac{[C_1]}{E_1^{tot}}, c_2 = \frac{[C_2]}{E_1^{tot}}, c_3 = \frac{[C_3]}{E_2^{tot}},$$
$$c_4 = \frac{[C_4]}{E_2^{tot}}, c_5 = \frac{[C_5]}{M^{tot}}, s = \frac{[S]}{V/k_6}.$$

Thus, time is rescaled by the time scale of binding to the phosphatase. Variables related to the free enzyme, either phosphorylated or unphosphorylated, are rescaled by the total amount of the enzyme. Complexes in the phosphorylation/dephosphorylation reactions are rescaled by total concentrations of kinase and phosphatase. Free sub-
strate is rescaled by the steady concentration in the absence of the enzyme. Finally, the enzyme-substrate complex is rescaled by the total amount of the enzyme. Using the conservation equations, we obtain the following system (all parameters are defined in Table 1): 

\[
\begin{align*}
\frac{dm_0}{d\tau} &= -p_1p_3m_0(1 - c_1 - c_2) + p_{14}p_7c_1 + p_{10}c_4, \\
\frac{dm_2}{d\tau} &= -m_2(1 - c_3 - c_4) + p_5c_3 + p_6p_1c_2 - p_{15}p_{16}p_{14}m_2s + p_{15}p_{17}c_5, \\
\frac{dc_1}{d\tau} &= p_3m_0(1 - c_1 - c_2) - p_4c_1, \\
\frac{dc_2}{d\tau} &= p_8(1 - m_0 - m_2 - p_1p_2c_1 - p_1p_2c_2 - p_2c_3 - p_2c_4 - c_5)(1 - c_1 - c_2) - p_9c_2, \\
\frac{dc_3}{d\tau} &= m_2(1 - c_3 - c_4) - (p_{13} + p_5)c_3, \\
\frac{dc_4}{d\tau} &= p_{11}(1 - m_0 - m_2 - p_1p_2c_1 - p_1p_2c_2 - p_2c_3 - p_2c_4 - c_5)(1 - c_1 - c_2) - p_{12}c_4, \\
\frac{ds}{d\tau} &= p_{15}(p_{16}p_{14}m_2s - p_{17}c_5), \\
\end{align*}
\]

System (5.1) corresponds to the full model, as we will refer to it in our subsequent discussions. For this model, we performed extensive parametric analysis and identified an oscillatory regime.

### 5.2.3 Quasi-steady-state approximation and the reduced model

In the parameter regime where oscillatory behavior was discovered, it is useful to consider a related simplified model, which we will refer to as the reduced model. Specifically, when \( p_2'0 \) (that is, when the total amount of phosphatase compared to the total amount of enzyme is very small) the Quasi-Steady-State Assumption can be applied to the fast variables (C1-4) in the full system, and the full system then
reduces to the following reduced model:

\[
\begin{align*}
\frac{dm_0}{d\tau} &= \frac{(p_7 - p_4)p_1m_0}{m_0 + \frac{p_4}{p_5} + \frac{p_6p_8(p_13 + p_5)}{p_3p_9}(1 - m_0 - m_2 - c_5)} \\
&\quad + \frac{p_{12}}{p_{11}(p_13 + p_5)} m_2 + \frac{p_{12}}{p_{11}} + (1 - m_0 - m_2 - c_5),
\end{align*}
\]

\[
\begin{align*}
\frac{dm_2}{d\tau} &= -\frac{p_{12}m_2}{m_2 + (p_{13} + p_5) + \frac{p_{12}}{p_{11}}(p_{13} + p_5)(1 - m_0 - m_2 - c_5)} \\
&\quad + \frac{p_{12}p_1m_0 + \frac{p_6p_8m_0}{p_3p_9} + (1 - m_0 - m_2 - c_5)}{1 - m_0 - m_2 - c_5} - p_{15}p_{16}m_2s + p_{15}p_{17}c_5,
\end{align*}
\]

\[
\begin{align*}
\frac{dc_5}{d\tau} &= p_{15}(p_{16}p_{14}m_2s - p_{17}c_5),
\end{align*}
\]

\[
\begin{align*}
\frac{ds}{d\tau} &= p_{15}(1 - s - p_{16}m_2s + \frac{p_{18}}{p_{14}}c_5).
\end{align*}
\]

\[5.2\]

5.2.4 The auxiliary model

The oscillations present in the full model are well approximated by the oscillations in the reduced model above. In order to rationalize the origin of the identified oscillatory behavior, and to visualize it in three dimensions, we constructed a three-variable "auxiliary model". In this model, one of the variables in Eq. (5.1) and Eq. (5.2) - the substrate level s - is treated as a parameter. Later on, the one-parameter bifurcation diagram of this auxiliary model with respect to the parameter s will be used as a skeleton that helps understand the emergence oscillations:

\[
\begin{align*}
\frac{dm_0}{d\tau} &= \frac{(p_7 - p_4)p_1m_0}{m_0 + \frac{p_4}{p_5} + \frac{p_6p_8(p_13 + p_5)}{p_3p_9}(1 - m_0 - m_2 - c_5)} \\
&\quad + \frac{p_{12}}{p_{11}(p_13 + p_5)} m_2 + \frac{p_{12}}{p_{11}} + (1 - m_0 - m_2 - c_5),
\end{align*}
\]

\[
\begin{align*}
\frac{dm_2}{d\tau} &= -\frac{p_{12}m_2}{m_2 + (p_{13} + p_5) + \frac{p_{12}}{p_{11}}(p_{13} + p_5)(1 - m_0 - m_2 - c_5)} \\
&\quad + \frac{p_{12}p_1m_0 + \frac{p_6p_8m_0}{p_3p_9} + (1 - m_0 - m_2 - c_5)}{1 - m_0 - m_2 - c_5} - p_{15}p_{16}m_2s + p_{15}p_{17}c_5,
\end{align*}
\]

\[
\begin{align*}
\frac{dc_5}{d\tau} &= p_{15}(p_{16}p_{14}m_2s - p_{17}c_5),
\end{align*}
\]

\[5.3\]

112
Parametric analysis of steady states and limit cycles in the full and reduced models was done using continuation algorithms, implemented in AUTO \cite{23, 24, 25} and MATCONT \cite{21, 22, 55}.

5.3 Results and Discussion

Overview of reaction network The starting point for our analysis was provided by a mathematical model of a multisite phosphorylation cycle that controls ERK activity \cite{65, 75}. In this model, ERK activation by MEK follows a distributive mechanism, where MEK dissociates from monophosphorylated ERK after the first phosphorylation. ERK activation requires a second phosphorylation, which depends on an independent binding to MEK. The distributive nature of the mechanism of ERK phosphorylation by MEK is supported by kinetic experiments with purified components \cite{30, 11}. In another set of biochemical studies, it was demonstrated that ERK deactivation by phosphatases, such as MKP3, also follows a distributive mechanism \cite{110}. Mathematical analyses of a network comprising distributive phosphorylation and dephosphorylation mechanisms revealed that they can exhibit bistability, an effect where, for the same set of operating conditions, the network can be found in two different, stable steady states \cite{65, 104, 29}, (Fig. 5.1B). Which of the two steady states is realized depends on the starting composition of the mixture of ERK and its regulators. In our study, the enzyme regulation system (the part of the network encircled by the green box in Fig. 5.1A) provides an input to a second module, the substrate module (the part of the network inside the red box in Fig. 5.1A) that describes substrate synthesis and degradation. Several substrates are known to be degraded in response to their phosphorylation by ERK. Note that this type of effect of ERK on its substrate is certainly not the only one possible. In fact, the lifetime of
several substrates, such as Fos, is increased when they are phosphorylated by ERK \[69, 67\].

Table 5.1: Definitions and interpretation of dimensionless parameters and a representative parameter set for oscillations

<table>
<thead>
<tr>
<th>Dimensionless parameter</th>
<th>Definition</th>
<th>Sample value for oscillatory solution</th>
<th>Dimensionless parameter</th>
<th>Definition</th>
<th>Sample value for oscillatory solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>( \frac{E_1}{E_{ tot}} )</td>
<td>1.4</td>
<td>( p_{10} )</td>
<td>( \frac{k_1}{k_3 \cdot M_{tot}} )</td>
<td>4.0e - 3</td>
</tr>
<tr>
<td>( p_1 )</td>
<td>( \frac{E_2}{E_{ tot}} )</td>
<td>5.0e - 2</td>
<td>( p_{11} )</td>
<td>( \frac{k_1}{k_3} )</td>
<td>1.0e - 1</td>
</tr>
<tr>
<td>( p_3 )</td>
<td>( \frac{k_1}{k_5} )</td>
<td>4.0e - 3</td>
<td>( p_{12} )</td>
<td>( \frac{k_1 + k_5}{k_5 \cdot M_{tot}} )</td>
<td>8.0e - 3</td>
</tr>
<tr>
<td>( p_4 )</td>
<td>( \frac{k_1 + k_5}{k_5 \cdot M_{tot}} )</td>
<td>3.2e - 4</td>
<td>( p_{13} )</td>
<td>( \frac{k_1}{k_5 \cdot M_{tot}} )</td>
<td>4.0e - 2</td>
</tr>
<tr>
<td>( p_5 )</td>
<td>( \frac{k_1 \cdot M_{tot}}{k_5} )</td>
<td>4.0e - 2</td>
<td>( p_{14} )</td>
<td>( \frac{V/k_5}{M_{tot}} )</td>
<td>1.5e1</td>
</tr>
<tr>
<td>( p_6 )</td>
<td>( \frac{k_1}{k_5 \cdot M_{tot}} )</td>
<td>3.2e - 1</td>
<td>( p_{15} )</td>
<td>( \frac{k_1}{k_5 \cdot E_{ tot}} )</td>
<td>5.6e - 5</td>
</tr>
<tr>
<td>( p_7 )</td>
<td>( \frac{k_5}{k_5 \cdot M_{tot}} )</td>
<td>1.6e - 4</td>
<td>( p_{16} )</td>
<td>( \frac{k_1}{k_5 \cdot E_{ tot}} )</td>
<td>1.571e2</td>
</tr>
<tr>
<td>( p_8 )</td>
<td>( \frac{k_5}{k_3} )</td>
<td>8.0</td>
<td>( p_{17} )</td>
<td>( \frac{k_5 + k_5}{k_5} )</td>
<td>1.571e1</td>
</tr>
<tr>
<td>( p_9 )</td>
<td>( \frac{k_5 + k_5}{k_5 \cdot M_{tot}} )</td>
<td>6.4e - 1</td>
<td>( p_{18} )</td>
<td>( \frac{k_5}{k_5} )</td>
<td>1.429</td>
</tr>
</tbody>
</table>

Some of the most extensively characterized examples of ERK-dependent substrate degradation were provided by genetic and biochemical studies of gene regulation during Drosophila embryogenesis, where gene activation by ERK signaling depends on phosphorylation and degradation of transcriptional repressors \[78, 43\]. Recent experiments in the early Drosophila embryo revealed that several of these ERK substrates positively regulate ERK phosphorylation \[53, 51, 52\]. These motivate our analysis of a two-module network shown in Fig. 5.1A. The isolated substrate module (treating the level of the double phosphorylated enzyme as a parameter) is monostable (Fig. 5.1C). Below, we demonstrate that oscillations can occur when this substrate-dependent control of ERK phosphorylation is linked to the bistability of the enzyme regulation module: the network comprising distributive phosphorylation and dephosphorylation mechanisms. Identification of an oscillatory regime Our identification of
an oscillatory regime was guided by the van der Pol oscillator, which is based on a combination of a fast bistable system and a slow variable that switches the system between branches of two alternative stable steady states. This design has been identified in multiple biological oscillators, including cell cycle and calcium signaling networks [28, 102]. In our model, the role of the slow variable can be played by an ERK substrate. At high concentration (low level of its conversion by ERK), this substrate could stabilize the state with high level of ERK phosphorylation. High ERK activity associated with this state increases substrate conversion, eventually leading to a switch to a state with low ERK phosphorylation. To test whether this type of oscillations is indeed possible, we analyzed long-term dynamics in a model where an enzyme, which is controlled by two opposing distributive mechanisms, increases the degradation rate of a substrate, which is synthesized at a constant rate. A mass action kinetics description of this mechanism leads to a system of eight differential equations that describe the joint dynamics of chemical species in the model. After nondimensionalization, the model contains 18 dimensionless groups, each of which reflects ratios of characteristic concentrations and time scales in the underlying chemical mechanism (see Materials and Methods and Table 1). To determine whether this model can support oscillations, we need to determine whether there exist values of the model parameters for which the long-term dynamics of the model has stable periodic solutions. Identification of parameter sets that support limit cycles is greatly simplified by the conceptual analogy with the van der Pol oscillator, where a subset of the full model exhibits bistability. In our case, this subset of the full model corresponds to the enzyme regulation module (Fig. 5.1A). Upon disconnecting this module from the downstream substrate degradation module, this subsystem has a smaller number of parameters (p1-p13 in Table-1), which depend only on the concentrations of ERK and its regulators, and on the rates of binding and catalytic reactions within the ERK phosphorylation cycle (see Materials and Methods). Based on the results of
the previous studies, we could readily select parameter sets that support bistability in
this module. These parameters are characterized by saturation of kinase/phosphatase
regulating the enzyme and by asymmetry of the rate constants of catalytic reactions
comprising the dual phosphorylation/dephosphorylation cycle [65, 75]. At the next
step, we explored the space of the remaining five components of the parameter vector
(p14-p18), in order to identify values that support oscillations. Using a combination
of numerical solution of the model and its steady state bifurcation analysis, we iden-
tified a domain that supports large amplitude limit cycles. These periodic solutions
indeed correspond to the antiphase oscillations of the double phosphorylated enzyme
and substrate levels (Fig 5.2). The period of the identified periodic solution is deter-
mined by the time scale of constitutive substrate degradation, underscoring the fact
that oscillations emerge due to substrate dynamics.

5.3.1 Separation of time scales in the full model

The parameter regime for which we found oscillations displays a clear separation of
time scales between different model subsystems. In fact, according to Table 1, there
are two successive "gaps" between three characteristic time scales, namely, the time
scale corresponding to the change of the kinase/phosphatase in the enzyme regulation
subsystem, ; the time scale corresponding to the change of the enzyme, ; and the time
scale corresponding to the constitutive degradation of the substrate, in the substrate
degradation module.

When is small, there is a separation of time scales within the enzyme regulation
subsystem, whose model equations correspond to the first six equations of Eq. (1) in
the Materials and Methods section. This subsystem consists of two slow equations
(the two equations for unphosphorylated and double phosphorylated enzyme) and
four fast equations (the four equations for the complexes of the enzyme with its kinase
and phosphatase). In this regime, the unphosphorylated and double phosphorylated
Figure 5.2: (A) Sample limit cycle in the full model. The rescaled total concentration for the double-phosphorylated ERK ($m_2^{\text{tot}}$) is equal to $[M_2] + [C_3] + [C_5]/[M^{\text{tot}}]$. The rescaled substrate concentration is denoted by $s$. (B) Dynamics of $m_2^{\text{tot}}$ and $s$ along the periodic trajectory. (C) Dynamics of the rate of the change of $m_2^{\text{tot}}$ ($r_1$) and the rate of the change of $s$ ($r_2$) along the periodic trajectory. The labels a and b correspond to two different phases of the periodic trajectory.
enzyme evolve on a time scale that is slower than the respective complexes. In the limit p2’0, the pseudo-steady-state assumption can be applied to the complexes C1-4, so that the full system (Eq. (1)) reduces to the reduced system (Eq. (2)).

Furthermore, when is small, there is an additional gap between the previous two time scales and the timescale corresponding to the constitutive degradation of the downstream substrate (S). In this regime, the free substrate and its complex with the double phosphorylated enzyme evolve on a time scale that is slower than the upstream enzyme regulation subsystem.

While we cannot guarantee that the existence of oscillations requires such a clear separation of time scales, such a hierarchy allows us to rationalize model behavior in terms of dynamic regimes of isolated subsystems. A similar separation of time scales is characteristic of relaxation oscillations, which are organized around a fast bistable subsystem and a slow subsystem that mediates transitions between branches of two alternative steady states.

Parametric analysis of steady state and oscillatory solutions In analyzing the parametric dependence and stability of the identified periodic solutions, we focused on two dimensionless parameters. The first of them, p1, which is proportional to the concentration of the activating kinase (E1), can be viewed as an input to the network. The second parameter, p14, is the ratio of the maximal concentration of the substrate to the total concentration of the various forms of the enzyme. When p14 is small, the influence from the downstream substrate degradation module to the upstream enzyme regulation module is small. Therefore, dynamics of the full model can be understood in terms of the behavior of isolated modules. For instance, when the enzyme regulation module is bistable, there are three possible steady state values for the substrate level in the full system (Fig. 5.3A). The steady state with high substrate modification rate (low substrate level) corresponds to the state where most of the enzyme is in the active, double phosphorylated form. A second stable steady state, with low substrate
modification rate (high substrate level), corresponds to the state where most of the enzyme is in unphosphorylated.

The retroactive effect of the substrate on the enzyme regulation module can be explored using one-parameter continuation diagrams of steady states of the full model as a function of the parameter p14, which represents the loading level of the downstream substrate. Results of a representative one-parameter continuation are shown in Fig. 5.3B (and) C, where we started in a bistable regime of the enzyme regulation module. As expected, at small values of the loading parameter the system is still bistable. At a critical value of p14, bistability is destroyed as the unstable steady state branch (red dashed curve in the lower left of Fig. 5.3A) collides with its adjacent stable steady state branch and disappears in a saddle-node bifurcation, leaving the state with low substrate conversion as the unique steady state of the model. Upon subsequent increase of the loading parameter, this steady state undergoes a subcritical Hopf bifurcation, and is surrounded by a large amplitude limit cycle, where the system oscillates between high and low levels of enzyme phosphorylation.

Using continuation algorithms to follow the instabilities of steady states and oscillatory solutions, we constructed a two-parameter map for the long-term dynamics of the model. This map is partitioned into domains of oscillations, bistability, and unique steady states (Fig. 5.4). Stable oscillations are realized at intermediate loading level (quantified by p14) of the substrate. When the loading level is relatively low (compared to the magnitude corresponding to oscillations) the dynamics of the full model is close to the dynamics of the enzyme regulation subsystem, which functions as a bistable switch characterized by previous studies. At intermediate loading levels, substrate-dependent control of the enzyme phosphorylation leads to oscillations, where the enzyme phosphorylation is toggled between two, high and low, levels. In this diagram, beyond the critical substrate loading level, bistability is eliminated and
Figure 5.3: Representative one-parameter bifurcation diagrams. (A) One-parameter bifurcation diagram showing the steady-state substrate level as a function of the input to the network, $p_1$. The loading level is fixed at a low value ($p_{14} = 3$). (Solid black and red dashed curves) Stable and unstable steady states, respectively. (B) One-parameter bifurcation diagram showing the evolution of the steady states as a function of the loading level $p_{14}$, calculated at fixed input to the network ($p_1 = 1.6$). (Blue dots and open red circles) Stable and unstable periodic solutions, respectively. (C) The period of oscillations as a function of the loading level for the cross section shown in panel B; the period is rescaled by the timescale of constitutive substrate degradation.
Figure 5.4: (A) Partition of input/load parameter space into single-valued (light blue), bistable (gray), and oscillatory (red) regions (top left). (B) Two-dimensional phase portraits of the model corresponding to the single-valued, bistable, and oscillatory regions. They correspond to ($p_1 = 1.0$, $p_{14} = 15$), ($p_1 = 1.9$, $p_{14} = 6$), and ($p_1 = 1.4$, $p_{14} = 15$), respectively, with the remaining parameter values shown in Table 1. The two observables in the phase diagram are the rescaled substrate concentration $s$ and the nondimensional total concentration for the double-phosphorylated ERK ($m_{tot}^2$, which is equal to $[M_2] + [C_3] + [C_5]/[M_{tot}]$), respectively. Red solid dots and red asterisks denote the associated stable and unstable steady states, respectively.

The full model becomes monostable for all values of the model input (quantified by $p_1$).

The computed input/load diagram can be used to systematically explore how different dynamic regimes in the model depend on the values of original, dimensional problem parameters. As an example, let us discuss how the existence of oscillations depends on the time scale of substrate degradation. According to the definitions of
dimensionless parameters given in Table 1, large values of correspond to high loading levels, a regime where oscillations disappear. For small values, the steady state load is negligible and oscillations are again impossible. Thus, oscillations are realized for intermediate values of.

5.3.2 Connection between bistability and oscillations

Most of the inputs (p1) that lead to oscillations correspond to bistability in the isolated enzyme regulation module. In fact, the emergence of oscillations can be rationalized by analyzing bistability of an auxiliary problem (Eq.(3) in the Materials and Methods section), where the substrate concentration is not a variable, but a parameter. Since the substrate acts as a competitive inhibitor of the phosphatase, high levels of the substrate would stabilize the state with high levels of enzyme phosphorylation. Upon slowly varying the substrate level, we observed a counterclockwise hysteresis in the level of enzyme phosphorylation (black curve in Fig. 5.5A).

For a chosen set of parameters, at low substrate concentrations, only a state with low enzyme activation (and high rates of dephosphorylation reactions) is stable. On the other hand, at high substrate concentrations, the only steady state corresponds to high enzyme phosphorylation. This bistability provides a skeleton for the oscillations in the reduced model (Eq.(2) in the Materials and Methods section) when substrate concentration is allowed to vary as a new, coupled, dependent variable. Indeed, a limit cycle of the reduced model spends a significant fraction of its time close to the steady state hysteresis loop of the auxiliary problem (Fig. 5.5A,B). To summarize, the identified oscillations are found in a regime where the isolated enzyme regulation module is bistable and there is a clear separation of time scales between substrate degradation and enzyme regulation. This conclusion is based on parametric analysis of periodic solutions. In the future, it will be interesting to test whether this is a necessary condition for the emergence of oscillations. This can be accomplished
Figure 5.5: (A) Limit cycle in the reduced system and its connection to the bistability in the auxiliary model, at fixed substrate concentration. The hysteresis curve (black) is the input/output map of the auxiliary model, computed using the nondimensional substrate concentration $s$ as a bifurcation parameter. Substrate concentration becomes a dependent variable for the reduced model. (Red curve) Sample trajectory of the reduced model. The parameter values are the same as the ones shown in Table 1 with $p_1 = 1.45$. (B) Time traces of the four variables in the reduced system. For each variable, one oscillation period is shown.
using a combination of analytical and computational singular perturbation approaches [101, 54, 42, 20].

5.4 Conclusions

ERK can use the same binding domains to interact with its substrates and regulators. This provides a basis for a number of competitive effects [97, 5, 98, 108]. As a consequence, ERK substrates can compete with ERK phosphatases and control the level of ERK phosphorylation [5]. Experiments in cultured cells and in vivo demonstrate that such effects indeed exist and are of appreciable magnitude [5, 53, 78]. Here we use a mathematical model to suggest that substrate-dependent control of ERK phosphorylation can qualitatively change the dynamics of a biochemical network that controls ERK activation. On its own, this network can be either monostable or bistable [65, 75]. However, in a larger system, where active ERK accelerates the degradation rate of a substrate that competes with the ERK phosphatase, this network can give rise to self-sustained oscillations. During the time course of these oscillations, the system switches between states with low and high levels of substrate conversion. The proposed mechanism of oscillatory ERK signaling is different from the previously described mechanisms whereby active ERK reduces the positive input to the ERK cascade [70, 90]. In contrast to the mechanism where active ERK reduces the rate at which ERK is phosphorylated by its upstream kinase, in our mechanism active ERK downregulates a substrate, which protects active ERK from dephosphorylation by phosphatase. Importantly, these oscillations are realized at a constant activity of the ERK kinase. In addition to this biochemical difference, the two mechanisms differ from the systems-level perspective: In all previous models, oscillations emerge due to a time delay between ERK activation and negative feedback at the level of the ERK activation cascade [89, 90, 49]. In our case, oscillations result from the retroac-
tive interaction between ERK substrate and bistability of the ERK phosphorylation network. Retroactivity is a term used to describe an effect whereby the downstream target of an upstream module changes the internal state of this module [20, 103, 85]. In our case, an ERK substrate, which is commonly viewed as a downstream target of active ERK, changes the level of ERK phosphorylation, which is commonly believed to be controlled independently of ERK substrates. In our model, the ERK phosphorylation subsystem can be at most bistable, while the substrate synthesis/degradation subsystem has a unique steady state. Coupling of the two modules, realized by the reaction of ERK binding to its substrate, can generate a retroactive effect that leads to oscillations. From the nonlinear dynamics perspective, this effect is not surprising and fits the oscillatory mechanisms in other systems [72]. It is based on a bistable system coupled to a slow variable that switches the bistable system between the two stable branches of steady states.

Recent experiments in the early Drosophila embryo suggest that retroactive effects contribute to the spatial regulation of gene expression induced by ERK signaling [51]. Based on the computational studies in this paper, we propose that retroactive effects can also regulate the dynamics of ERK signaling. We hope that our work will motivate future studies of retroactive effects in pathways where substrates of a signaling enzyme can compete with its regulators.
Bibliography

control of distributed microscopic processes. *International Journal of Robust

G. Jimenez. A mapk docking site is critical for downregulation of capicua

making by early and delayed loops. *Nature Reviews Molecular Cell Biology*,


activated protein kinase (mapk) kinases, mar phosphatases and the elk-1 tran-
scription factor compete for mapk binding and are crucial for enzymic activity.

[6] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and

M. Ouimet. Learning eigenfunctions links spectral embedding and kernel p-

[8] A. Bindal, M. G. Ierapetritou, S. Balakrishnan, A. Armaou, A. G. Makeev,
and I. G. Kevrekidis. Equation-free, coarse-grained computational optimization

bier. Sworm: an agent-based model to simulate the effect of earthworms on soil

and S. J. Simpson. From disorder to order in marching locusts. *Science*,


