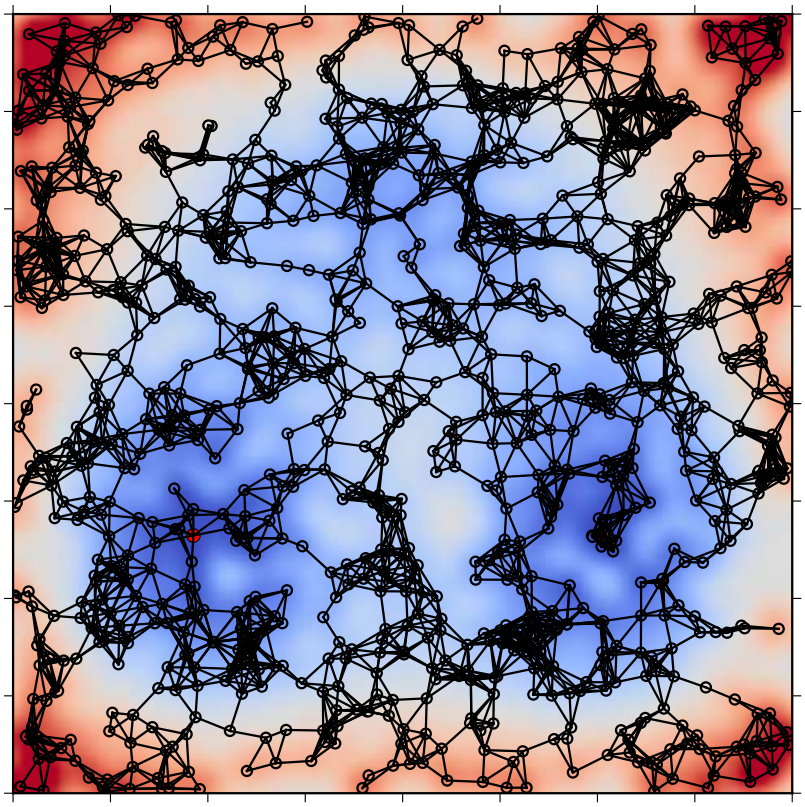
Nearly reducible finite Markov chains: theory and algorithms



Daniel Joshua Sharpe

Sidney Sussex College
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

May 2021

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Daniel Joshua Sharpe
May 2021

Abstract

Finite Markov chains are probabilistic network models that are commonly used as representations of dynamical processes in the physical sciences, biological sciences, economics, and elsewhere. Markov chains that appear in realistic modelling tasks are frequently observed to be nearly reducible, incorporating a mixture of fast and slow processes that leads to ill-conditioning of the underlying matrix of probabilities for transitions between states. Hence, the wealth of established theoretical results that makes Markov chains attractive and convenient models often cannot be used straightforwardly in practice, owing to numerical instability associated with the standard computational procedures to evaluate the expressions. This work is concerned with the development of theory, algorithms, and simulation methods for the efficient and numerically stable analysis of finite Markov chains, with a primary focus on exact approaches that are robust and therefore applicable to nearly reducible networks. New methodologies are presented to determine representative paths, identify the dominant transition mechanisms for a particular process of interest, and analyze the local states that have a strong influence on the characteristics of the global dynamics. The novel approaches yield new insights into the behaviour of Markovian networks, addressing and overcoming numerical challenges. The methodology is applied to example models that are relevant to current problems in chemical physics, including Markov chains representing a protein folding transition, and a configurational transition in an atomic cluster.

Relevant classical theory of finite Markov chains and a description of existing robust algorithms for their numerical analysis is given in Chapter 1. The remainder of this thesis considers the problem of investigating a transition from an initial set of states in a Markovian network to an absorbing (target) macrostate. A formal approach to determine a finite set of representative transition paths is proposed in Chapter 2, based on exact pathwise decomposition of the total productive flux. This analysis allows for the importance of competing dynamical processes to be rigorously quantified. A robust state reduction algorithm to compute the expectation of any path property for a transition between two endpoint states is also described in Chapter 2. Chapter 3 reports further numerically stable state

reduction algorithms to compute quantities that characterize the features of a transition at a statewise level of detail, allowing for identification of the local states that play a key role in modulating the slow dynamics. An expression is derived for the probability that a state is visited on a path that proceeds directly to the absorbing state without revisiting the initial state, which characterizes the dynamical relevance of an individual state to the overall transition process. In Chapter 4, an unsupervised strategy is proposed to utilize a highly efficient simulation algorithm for sampling paths on a Markov chain. The framework employs a scalable community detection algorithm to obtain an initial clustering of the network into metastable sets of states, which is subsequently refined by a variational optimization procedure. The optimized clustering is then used as the basis for simulating trajectory segments that necessarily escape from the metastable macrostates. The thesis is concluded with an overview of recent related advances that are beyond the scope of the current work (Chapter 5), and a discussion of potential applications where the novel methodology reported herein may be applied to perform insightful analyses that were previously intractable.

Acknowledgments

It has been a great pleasure to work with David Wales, and I cannot imagine that I would have learned so much under the supervision of another professor. I especially thank David for allowing me the creative freedom to pursue my own research interests, while providing prompt feedback, always sparing time for an invariably stimulating discussion, and guiding me to be a good researcher and communicator. Many of the mathematical topics that have now become my academic passions were introduced to me by David, and in this way I am certain that David will remain a positive influence on my work even when Cambridge is long distant in the memory.

I have also had the good fortune to collaborate with, and get to know, several talented academics from around the world, with whom I have worked on projects outside the scope of the present thesis: Dr Shiyan Xiao (USTC Hefei), Dr Leonardo Darre (Institut Pasteur de Montevideo), Prof. Jason R. Green (Univ. of Massachusetts Boston), Dr Thomas D. Swinburne (Aix-Marseille Univ.), Mark Clapp (Cambridge Part III), and Deepti Kannan (Cambridge MPhil, now Massachusetts Institute of Technology). The work that I have conducted with Deepti and Tom complements and extends the ideas presented in the current thesis, and I am grateful for the many hours of insightful discussion with them, which has helped to shape the direction of my own research.

I would be remiss not to give mention to the many people who have influenced me prior to my studies at Cambridge. In particular, I fondly remember my time at the University of Durham, and am indebted to my mentors there, Dr Mark Miller and Prof. David Tozer, and to many good friends who are too numerous to name exhaustively, but especially Ashley, Matt, Sophie, Callum, and Charley.

I thank my friends in the theory group; Alasdair, Sundeep, Luke, Fabio, Andreea, Nick, and Charlie, for making the office a fun place to work, and my friends in college; Ben, Lianne, Jake, Nakul, Josh, and others, whose company has made the PhD a more enjoyable experience.

I gratefully acknowledge financial support from the Cambridge Commonwealth, European,

and International Trust via the award of a Vice Chancellor's Scholarship, and the EPSRC for additional financial support.

Lastly, I am thankful for my wonderful partner, Kathleen, and for my parents and my sister, whose continual support has always been the reason for any successes I may have had, large or small.

List of Publications

The following publications are directly related to the work described in this thesis:

Chapter 1:

Daniel J. Sharpe and David J. Wales, *Nearly reducible finite Markov chains: theory and algorithms*, J. Chem. Phys. (accepted).

Chapter 2:

Daniel J. Sharpe and David J. Wales, *Graph transformation and shortest paths algorithms for finite Markov chains*, Phys. Rev. E (2021), **103**, 063306.

Chapter 3:

Daniel J. Sharpe and David J. Wales, *Numerical analysis of first passage processes in finite Markov chains exhibiting metastability*, Phys. Rev. E (2021), **104**, 015301.

Chapter 4:

Daniel J. Sharpe and David J. Wales, *Efficient and exact sampling of transition path ensembles on Markovian networks*, J. Chem. Phys. (2020), **153**, 024121.

I have also contributed to the following publications during my PhD, material from which does not appear in the current thesis:

Daniel J. Sharpe and David J. Wales, *Identifying mechanistically distinct pathways in kinetic transition networks*, J. Chem. Phys. (2019), **151**, 124101.

Shiyan Xiao, **Daniel J. Sharpe**, Debayan Chakraborty, and David J. Wales, *Energy landscapes and hybridization pathways for DNA hexamer duplexes*, J. Phys. Chem. Lett. (2019), **10**,

6771-6779.

Daniel J. Sharpe, Konstantin Roeder, and David J. Wales, *Energy landscapes of deoxyxylo- and xylo-nucleic acids*, J. Phys. Chem. B (2020), **124**, 4062-4068.

Thomas D. Swinburne, Deepti Kannan, **Daniel J. Sharpe**, and David J. Wales, *Rare events and first passage time statistics from the energy landscape*, J. Chem. Phys. (2020), **153**, 134115.

Deepti Kannan*, **Daniel J. Sharpe***, Thomas D. Swinburne, and David J. Wales, *Optimal dimensionality reduction of Markov chains using graph transformation*, J. Chem. Phys. (2020), **153**, 244108.

Contents

List of Acronyms

Acronym	Definition
BKL	Bortz-Kalos-Lebowitz algorithm
CTMC	continuous-time Markov chain
DPS	discrete path sampling
DTMC	discrete-time Markov chain
F	face-centered cubic state of LJ_{38} *or* folded state of TZ1
FPPE	first passage path ensemble
FPT	first passage time
FUND	FUND algorithm of Heyman and O'Leary
GMRES	generalized minimal residual
GT	graph transformation
GTH	Grassmann-Taksar-Heyman [algorithm]
HEM	heavy edge matching
IAD	iterative aggregation-disaggregation
I_h	incomplete Mackay icosahedron state of LJ_{38}
KTN	kinetic transition network
kMC	kinetic Monte Carlo
kPS	kinetic path sampling
LEA	local equilibrium approximation
LJ_{38}	cluster of 38 atoms bound by the Lennard-Jones potential
LU	lower triangular-upper triangular matrix decomposition
MCAMC	Monte Carlo with absorbing Markov chains
MCL	Markov clustering
MFPT	mean first passage time
MLR-MCL	multi-level regularized Markov clustering

Acronym	**Definition**
MSM	Markov State Model
REA	recursive enumeration algorithm
REFUND	recursive FUND algorithm
TPE	transition path ensemble
TPT	transition path theory
TSE	transition state ensemble
TZ1	tryptophan zipper peptide 1
U	unfolded state of TZ1
WE	weighted ensemble [sampling]

List of Mathematical Symbols

Bold uppercase Roman letters are used to denote matrices. Bold lowercase Roman or bold lowercase Greek letters are used to denote (column) vectors. The set of nodes to which a matrix or vector corresponds, and hence the dimensions of the array, is sometimes explicitly denoted by a subscript. For instance, $\mathbf{T}_{\mathcal{AQ}}$ is the $|\mathcal{A}|\times|\mathcal{Q}|$-dimensional matrix of probabilities for transitions from transient nodes, of the set $\mathcal{Q}$, to absorbing nodes, of the set $\mathcal{A}$.

Symbol	**Definition**
$\mathbf{1}_{\mathcal{S}}$	$\|\mathcal{S}\|$-dimensional unit column vector
$\partial\mathbb{A}$	subset of the absorbing set of nodes $\mathbb{A}$ directly connected to the basin $\mathbb{B}$ in kPS
$\partial\mathcal{B}$	subset of nodes at the boundary of the initial state $\mathcal{B}$
α	sampled node at the absorbing boundary $\partial\mathbb{A}$ in kPS
β	thermodynamic beta $1/k_{\mathrm{B}}T$
Γ	gamma function
γ_k	k-th eigenvalue of the transition rate matrix $\mathbf{K}$
δ_{ij}	Kroenecker delta
ϵ	initial node of the trapping basin $\mathbb{B}$ in kPS
ζ	reweighting parameter
$\boldsymbol{\zeta}$	vector of coupling factors
ζ_{K}	Kemeny constant (average mixing time)
η_j	total number of kMC transitions from the j-th node in kPS
θ_j	expected number of times the j-th node is visited on a $\mathcal{A}\leftarrow\mathcal{B}$ first passage path
$\widetilde{\theta}_j$	expected number of times the j-th node is visited on a $\mathcal{A}\leftarrow\mathcal{B}$ transition path
$\widetilde{\theta}_j^{\mathrm{SS}}$	expected number of times the j-th node is visited on a $\mathcal{A}\leftarrow\mathcal{B}$ steady state transition path

Symbol	Definition
λ_k	k-th eigenvalue of the discrete-time transition probability matrix $\mathbf{T}(\tau)$
μ_j	initial occupation probability distribution of reactive trajectories for the j-th node
μ_j^{SS}	initial probability distribution of steady state reactive trajectories for the j-th node
ξ	discrete path, i.e. ordered sequence of visited nodes
$\xi^{(m',m)}$	discrete path starting at node m and terminating at node m'
$\xi^k(j)$	k-th highest-probability first passage path from node j in the REA
π_j	stationary (equilibrium) occupation probability of the j-th node
τ	lag time (for a DTMC)
τ_j	mean waiting time for a transition from the j-th node (for a CTMC)
$\boldsymbol{\psi}^{(k)}$	k-th right eigenvector of the transition probability and rate matrices
$\boldsymbol{\phi}^{(k)}$	k-th left eigenvector of the transition probability and rate matrices
$\mathbb{A}$	set of absorbing nodes in kPS
$\mathcal{A}$	absorbing (target) state
$\mathbf{A}$	Markovian kernel *or* arbitrary square matrix
$\mathbf{A}^{\#}$	group inverse
$\mathbb{B}$	set of nodes that comprise the trapping basin in kPS
$\mathcal{B}$	initial state
B_{ij}	probability that a path initialized at the j-th node is absorbed at the i-th node
$\mathcal{C}$	set of N communities
$\mathbf{C}$	stochastic coupling (aggregation) matrix
$\mathcal{D}(j)$	set of nodes $\gamma \in \mathcal{D}(j)$ for which a direct $j \leftarrow \gamma$ connection exists
$\mathbb{E}$	subset of the basin nodes, of the set $\mathbb{B}$, to be eliminated in kPS
f_{ij}^{+}	net reactive stationary $\mathcal{A} \leftarrow \mathcal{B}$ flux for the $i \leftarrow j$ transition
$\mathbf{G}$	generalized inverse

Symbol	Definition
H_{ij}	probability that the i-th node is visited on a first passage path starting at node j
$\widetilde{H}_{ij}$	probability that the i-th node is visited on a transition path starting at node j
$H_{ij}^{(n)}$	number of $i \leftarrow j$ kMC transitions in kPS for transition matrix $\mathbf{T}^{(n)}$
$\mathbf{I}$	identity matrix
$\mathcal{I}$	set of intermediate nodes $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$
i, j	indices denoting nodes of a Markov chain
$\mathcal{J}_{\mathcal{AB}}$	reactive flux for the $\mathcal{A} \leftarrow \mathcal{B}$ transition at equilibrium steady state
$\mathcal{J}[\xi]$	reactive $\mathcal{A} \leftarrow \mathcal{B}$ flux associated with transition path ξ
K_{ij}	rate of the $i \leftarrow j$ transition
$k_{\mathcal{AB}}$	nonequilibrium rate constant for the $\mathcal{A} \leftarrow \mathcal{B}$ transition
$k_{\mathcal{AB}}^{\mathrm{SS}}$	steady-state rate constant for the $\mathcal{A} \leftarrow \mathcal{B}$ transition
k_{B}	Boltzmann constant
$\mathcal{M}(j)$	set of candidates for the next shortest path to the j-th node in the REA
M_{ij}	negative natural log of the transition probability for the $i \leftarrow j$ transition
m_j^{R}	probability that a trajectory is reactive for the j-th node
m_j	normalized probability that a trajectory is reactive for the j-th node
N	number of communities
$\mathbf{N}^*$	augmented transition probability matrix
N_{ij}	average number of times the i-th node is visited on a first passage path starting at node j
$\widetilde{N}_{ij}$	average number of times the i-th node is visited on a transition path starting at node j
$\mathcal{O}$	time complexity of an algorithm
$\mathcal{P}[\xi]$	path probability for trajectory ξ
P_{ij}	branching probability of the $i \leftarrow j$ transition
$\mathbf{p}(t)$	time-dependent occupation probability distribution vector

Symbol	Definition
$p_j(0)$	initial occupation probability for the j-th node
$\mathcal{Q}$	set of transient nodes
q_j^+	forward committor probability for the j-th node, $q_{b\in\mathcal{B}} = 0$
$q_j^{+\prime}$	forward committor probability for the j-th node, $q_{b\in\mathcal{B}} \neq 0$
$\mathcal{R}_{\mathcal{AB}}$	mean first passage reward for the $\mathcal{A} \leftarrow \mathcal{B}$ transition
$\mathcal{R}[\xi]$	expected reward for the path ξ
r	granularity parameter in Markov clustering
R_{ij}	reward associated with the $i \leftarrow j$ transition
r_j	average reward for a transition from the j-th node
r_j^+	probability that the j-th node is visited on a $\mathcal{A} \leftarrow \mathcal{B}$ transition path
$r_j^{+,\mathrm{SS}}$	probability that the j-th node is visited on a $\mathcal{A} \leftarrow \mathcal{B}$ steady state transition path
$\mathcal{S}$	set of nodes comprising a Markov chain, i.e. the state space
$\mathbb{T}$	subset of basin nodes that are noneliminated in kPS
$\boldsymbol{\mathcal{T}}$	matrix of MFPTs for all pairwise transitions between nodes
$\mathcal{T}_{\mathcal{AB}}$	MFPT for the $\mathcal{A} \leftarrow \mathcal{B}$ transition
T	temperature
$\mathbf{T}$	generic transition probability matrix ($\mathbf{T}(\tau)$,$\mathbf{T}_{\mathrm{lin}}(\tau)$, or $\mathbf{P}$)
$\mathbf{T}_{\mathcal{AQ}}$	block of the stochastic matrix $\mathbf{T}$, for transitions from nodes of the set $\mathcal{Q}$ to the set $\mathcal{A}$
$\mathbf{T}_{\mathcal{AQ}}^{\mathcal{N}}$	see $\mathbf{T}_{\mathcal{AQ}}$, explicitly indicating that nodes of the set $\mathcal{N}$ have been eliminated from $\mathbf{T}$ by renormalization
T_{ij}	probability of the $i \leftarrow j$ transition (discrete- or continuous-time)
T'_{ij}	renormalized probability for the $i \leftarrow j$ transition
$\widetilde{T}_{ij}$	reweighted *or* reactive probability for the $i \leftarrow j$ transition
$\mathbf{T}(\tau)$	discrete-time transition probability matrix parameterized at lag time τ
$\mathbf{T}_{\mathrm{lin}}(\tau)$	linearized transition probability matrix with uniform mean waiting time τ
$\mathbf{T}^{(n)}$	transition probability matrix after the elimination of n nodes by GT

Symbol	Definition
t	time
$t_{\mathbb{A}}$	time for a path escaping from the trapping basin $\mathbb{B}$ to the absorbing boundary $\partial\mathbb{A}$ in kPS
$\mathcal{V}_{\mathcal{AB}}$	variance of the FPT distribution for the $\mathcal{A} \leftarrow \mathcal{B}$ transition
$\mathcal{W}[\xi]$	path weight, i.e. product of transition probabilities, for trajectory ξ
$\mathcal{X}, \mathcal{Y}$	indices denoting communities of nodes
$\mathbf{Z}$	Kemeny and Snell's fundamental matrix

Chapter 1

Nearly reducible finite Markov chains: theory and algorithms

Finite Markov chains, memoryless random walks on complex networks, appear commonly as models for stochastic dynamics in condensed matter physics, biophysics, ecology, epidemiology, economics, and elsewhere. Here we review exact numerical methods for the analysis of arbitrary discrete- and continuous-time Markovian networks. We focus on numerically stable methods that are required to treat nearly reducible Markov chains, which exhibit a separation of characteristic timescales and are therefore ill-conditioned. In this metastable regime, dense linear algebra methods are afflicted by propagation of error in the finite precision arithmetic, and the standard kinetic Monte Carlo algorithm to simulate paths is unfeasibly inefficient. Furthermore, iterative eigendecomposition methods fail to converge without the use of nontrivial and system-specific preconditioning techniques. An alternative approach is provided by state reduction procedures, which do not require additional a priori knowledge of the Markov chain. Macroscopic dynamical quantities such as the mean first passage time (MFPT) for a transition to an absorbing state, higher moments of the FPT distribution, and the average mixing time, as well as microscopic dynamical quantities such as the stationary, committor, and absorption probabilities for nodes, can be computed robustly using state reduction algorithms. The related kinetic path sampling algorithm can be used to efficiently sample paths on a nearly reducible Markov chain. Thus, all information required to determine the kinetically relevant transition mechanisms, and to identify the states that have a dominant effect on the global dynamics, can be computed reliably even for computationally challenging models. Rare events are a ubiquitous feature of realistic dynamical systems, and so the methods described herein are valuable in many practical applications.

1.1 Introduction

Finite Markov chains[1–4] are commonly used to represent a variety of stochastic processes. They provide attractive coarse-grained representations of continuous-state models, such as the dynamics of many-particle systems,[5] since the high dimensionality of the coordinate space can be preserved.[6] That is, a Markov chain can be constructed as a discretized representation of a continuous state space with a one-to-one mapping between nodes of the network and contiguous regions of the full coordinate space, without projection onto representative or aggregated coordinates. Markov chains corresponding to a continuous-state system can be constructed by using explicit simulation data to estimate a network model by maximum-likelihood[7–11] or Gibbs sampling[12–15] approaches.[16–20] Alternatively, the energy landscape of a physical system can be mapped to a Markovian network using geometry optimization methods[21] to locate the stationary points.[22–29] The dynamics of the resulting Markov chain are described by a linear master equation,[30–32] a system of coupled first-order ordinary differential equations (ODEs). That is, the Markov chain corresponds to a complex network for which the edges are parameterized by rates K_{ij} for the transitions between nodes $i \leftarrow j$ (in the continuous-time case), or transition probabilities $T_{ij}(\tau)$ at a lag time τ (in the discrete-time case).[33] Continuous-time Markov chains are also frequently used to represent population dynamics processes, where the transitions correspond to discrete changes in the numbers of species. That is, each node of the network is associated with a vector specifying the population distribution. Since the populations of species are unbounded, the state space of the network is countably infinite,[34,35] but can be truncated to yield a finite Markov chain with negligible error.[36,37] Simple examples of such models include birth-death processes[38,39] and queuing networks.[40] More complex examples arise as representations of chemical[41,42] and biochemical[43–45] reaction cycles, gene regulatory networks,[46–49] epidemic spread,[50] and ecosystems.[51]

It is a ubiquitous feature of realistic models for complex dynamical processes that there exists a separation of characteristic timescales.[52–66] For instance, the extinction of a species in an ecosystem takes place over a long period of time, compared to short-timescale fluctuations in the size of the population arising from births and deaths.[51] In economic models, significant changes in the overall status of a market occur infrequently relative to the frequency of individual trades.[67] In molecular and condensed matter systems, the underlying energy landscape typically features a disparity in the heights of energy barriers separating regions of the state space.[68–72] In each of these applications, it is precisely the rare event that is the transition of interest.

In the present work, we review exact computational methods to analyze the dynamics of

arbitrary discrete- and continuous-time finite Markov chains. In particular, we are concerned with *nearly reducible* Markov chains,[73,74] which exhibit rare event dynamics and are consequently ill-conditioned.[75–80] The application of conventional dense linear algebra methods to nearly reducible Markovian networks is therefore usually prohibited by the severe propagation of error arising from the limits of numerical precision.[81] Similarly, the standard kinetic Monte Carlo[82,83] (kMC) algorithm for explicit simulation of the stochastic dynamics becomes unfeasibly inefficient, since the trajectories have a tendency to 'flicker' within the metastable sets of nodes.[84–88] We therefore focus on specialized methods that are capable of treating Markov chains featuring metastability. We consider a general Markov chain comprising the set of nodes $\mathcal{S}$. To formulate the problem of analyzing a particular transition, we consider initial and absorbing (target) sets of nodes, denoted $\mathcal{B}$ and $\mathcal{A}$, respectively. The nodes in $\mathcal{B}$ are a subset of the set of transient (nonabsorbing) nodes, denoted $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$.

Following an overview of the relevant theory of Markov chains and standard linear algebra methods for their exact analysis (Sec. 1.2), we provide a detailed review of procedures that have superior numerical stability, and are therefore recommended for application to Markovian networks exhibiting metastability. Specifically, we discuss state reduction methods that are inherently robust and do not involve preconditioning techniques. The mean first passage time (MFPT) for the $\mathcal{A} \leftarrow \mathcal{B}$ transition, which is the usual dynamical observable, can be computed using the graph transformation (GT) algorithm (Sec. 1.3).[89–94] The GT approach is closely related to uncoupling-coupling methods (Secs. 1.4.1 and 1.4.2) and the Grassmann-Taksar-Heyman (GTH) algorithm (Sec. 1.4.3) for computation of the stationary distribution. Further state reduction procedures exist to compute other macroscopic dynamical properties, such as the expected time to reach the equilibrium distribution, and higher moments of the first passage time distribution (Sec. 1.4.4). State reduction algorithms are also available to compute microscopic dynamical properties, including the committor probabilities for nodes (Sec. 1.5.1), defined as the probability that a trajectory initialized at that node hits state $\mathcal{A}$ before state $\mathcal{B}$, and other quantities that characterize the $\mathcal{A} \leftarrow \mathcal{B}$ transition path ensemble (TPE).[95–101] Finally, we discuss efficient methods to simulate paths on a nearly reducible Markov chain. The kinetic path sampling[84,85] (kPS) and Monte Carlo with absorbing Markov chains[102–105] (MCAMC) algorithms provide an efficient alternative to standard kMC simulations[82,83] for this purpose (Sec. 1.5). The former method extends the GT algorithm with an iterative reverse randomization procedure to sample the time associated with a path escaping from a metastable region of the network, while the latter approach uses local eigendecompositions. By employing the advanced methods described herein, it is in principle possible to extract any desired dynamical information from a nearly reducible Markov chain.

1.2 Background theory of Markov chains

1.2.1 Master equation dynamics

The dynamics of a continuous-time Markov chain (CTMC), parameterized by $i \leftarrow j$ internode transition rates K_{ij} and with state space $\mathcal{S}$, is governed by the linear master equation[30–32,106]

$$\frac{\mathrm{d}p_j(t)}{\mathrm{d}t} = \sum_{i \neq j} \Big(K_{ji} p_i(t) - K_{ij} p_j(t) \Big) \; \forall \, j \in \mathcal{S}, \tag{1.1}$$

where $p_j(t)$ is the time-dependent occupation probability of the j-th node. Eq. 1.1 can be written in matrix form as

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{K}\mathbf{p}(t), \tag{1.2}$$

where $\mathbf{p}(t) = (p_1(t), p_2(t), \ldots, p_{|\mathcal{S}|}(t))^\top$ is the time-dependent occupation probability vector for the nodes of the network, and $\mathbf{K}$ is the transition rate matrix. $\mathbf{K}$ has off-diagonal elements equal to the transition rates, and diagonal elements such that the columns of the matrix sum to zero, i.e. $K_{jj} = -\sum_{\gamma \neq j} K_{\gamma j}$. The right eigenvectors of $\mathbf{K}$ represent dynamical eigenmodes, where the magnitudes and signs of the elements reflect the extent and direction of probability flow for the corresponding nodes, respectively. The solution of Eq. 1.2 is a sum of contributions that decay exponentially with rates equal to the negatives of the eigenvalues $\{\gamma_k\}$ of $\mathbf{K}$.[8] This solution leads to a formal definition for metastability, namely that the Markov chain exhibits a spectral gap in its set of eigenvalues $\{\gamma_k\}$, and hence there exists a set of dynamical eigenmodes that represent comparatively slow relaxation processes.[64] By the Perron-Frobenius theorem,[107] $\mathbf{K}$ has a unique dominant zero eigenvalue $\gamma_1 = 0$ if the Markov chain is *irreducible*,[74] i.e. if every node of the network is reachable from all other nodes. The zero eigenvalue is associated with the stationary probability vector $\boldsymbol{\pi}$, and all other eigenvalues have a negative real component. The stationary distribution satisfies the global balance equation $\mathbf{K}\boldsymbol{\pi} = \mathbf{0}$, where $\mathbf{0}$ is the column vector with all elements equal to zero.[74] If the stationary distribution also satisfies the detailed balance condition, $K_{ij}\pi_j = K_{ji}\pi_i \; \forall \, i \neq j$, then the Markov chain is said to be *reversible*, and the eigenvalues of $\mathbf{K}$ are real.[2]

The transition rate (or *infinitesimal generator*) matrix $\mathbf{K}$ relates to a transition probability matrix $\mathbf{T}(\tau)$, which propagates the probability distribution vector $\mathbf{p}(t)$ at discrete time intervals τ according to the Chapman-Kolmogorov equation[2] $\mathbf{p}(t+\tau) = \mathbf{T}(\tau)\mathbf{p}(t)$, via[50]

$$\mathbf{T}(\tau) = \exp(\mathbf{K}\tau), \tag{1.3}$$

and

$$\mathbf{K} = \lim_{\tau \to 0} \frac{\mathbf{T}(\tau) - \mathbf{I}}{\tau}, \tag{1.4}$$

where $\mathbf{I}$ is the identity matrix. For an irreducible and aperiodic Markov chain, $\mathbf{T}$ has a single dominant eigenvalue $\lambda_1 = 1$ associated with the stationary distribution $\boldsymbol{\pi}$, which is the occupation probability distribution in the limit of infinite time. The absolute value of all other eigenvalues, λ_k for $k = 2, ...|\mathcal{S}|$, is less than unity.[1] The transition rate and probability matrices share the same set of right and left eigenvectors, $\{\boldsymbol{\psi}^{(k)}\}$ and $\{\boldsymbol{\phi}^{(k)}\}$, respectively. Their eigenvalues are related via $e^{\gamma_k \tau} = \lambda_k(\tau)$, *cf.* Eq. 1.3.[50] When the dynamics are reversible, the Markov chain has a complete set of orthonormal eigenvectors.[108] Specifically, the left and right eigenvectors satisfy the orthonormality conditions[8]

$$\begin{aligned} \sum_{j \in \mathcal{S}} \phi_j^{(k)} \phi_j^{(l)} \pi_j &= \sum_{j \in \mathcal{S}} \psi_j^{(k)} \psi_j^{(l)} / \pi_j \\ &= \sum_{j \in \mathcal{S}} \psi_j^{(k)} \phi_j^{(l)} = \delta_{kl}, \end{aligned} \tag{1.5}$$

where $\psi_j^{(k)}$ is the j-th element of the k-th right eigenvector, and δ_{kl} is the Kroenecker delta.

There are two possible choices of stochastic matrix for a CTMC.[109] The first is the branching probability matrix $\mathbf{P}$, with elements $P_{ij} = K_{ij} / \sum_{\gamma \neq j} K_{\gamma j}$.[92] $\mathbf{P}$ contains no self-loops, and the time associated with a transition from the j-th node is exponentially distributed with mean $\tau_j = 1 / \sum_{\gamma \neq j} K_{\gamma j}$,[110] referred to as the mean waiting time for the j-th node. The second valid stochastic matrix for a CTMC is the continuous-time linearized transition probability matrix[84]

$$\mathbf{T}_{\text{lin}}(\tau) = \mathbf{I} + \tau \mathbf{K}, \tag{1.6}$$

for which the mean waiting times are uniform for all nodes, equal to τ. $\mathbf{T}_{\text{lin}}$ has the same sparsity pattern as $\mathbf{P}$, except that the linearized matrix contains self-loop transitions, whereas $P_{jj} = 0 \; \forall \, j$. Provided that $\tau \leq \min\{-K_{jj}^{-1} : \forall \, j\}$, the linearized transition matrix is column-stochastic, and shares the same set of eigenvectors as $\mathbf{K}$, but the eigenvalues are shifted.[111] In the following exposition, we will use the notation $\mathbf{T}$ to denote any general stochastic matrix for a discrete- or continuous-time Markov chain, and draw attention to separate considerations for the different formulations where necessary.

1.2.2 Fundamental properties of irreducible Markov chains

Irreducible Markov chains have a stationary distribution $\boldsymbol{\pi}$ that satisfies the global balance equations $\mathbf{T}\boldsymbol{\pi} = \boldsymbol{\pi}$ and $\mathbf{K}\boldsymbol{\pi} = \mathbf{0}$. The Markovian kernel[112] $[\mathbf{I} - \mathbf{T}(\tau)]$ and the transition rate matrix $\mathbf{K}$, which we will collectively denote by $\mathbf{A}$, are singular, but there exist a class

of generalized inverses[113–117] $\mathbf{G}$ that satisfy $\mathbf{AGA} = \mathbf{A}$. Such matrices are *fundamental* in the sense that key global dynamical properties can be expressed straightforwardly in terms of $\mathbf{G}$ and the stationary distribution $\boldsymbol{\pi}$.[1,118] In the following discussion, we use $\mathbf{T}(\tau)$ to refer to a discrete-time stochastic matrix parameterized at lag time τ, or the linearized transition matrix of a CTMC with uniform mean waiting times τ.

Important examples of generalized inverses are Kemeny and Snell's fundamental matrix[1,109]

$$\mathbf{Z} = (\mathbf{I} - \mathbf{T}(\tau) + \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top})^{-1}, \tag{1.7}$$

where $\mathbf{1}_{\mathcal{S}}$ is the $|\mathcal{S}|$-dimensional column vector with elements equal to unity, and Meyer's group inverse,[119]

$$\mathbf{A}^{\#} = \mathbf{Z} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}, \tag{1.8}$$

with elements[120]

$$A_{ij}^{\#} = \sum_{n=0}^{\infty} \Big(T_{ij}^{n} - \pi_i \Big). \tag{1.9}$$

The fundamental matrix is a generalized inverse[113] that also satisfies $\mathbf{AZ} = \mathbf{ZA}$,[112,116,121,122] and the group inverse is the unique generalized inverse that additionally satisfies $\mathbf{A}^{\#}\mathbf{AA}^{\#} = \mathbf{A}^{\#}$.[112,115] The group inverse is also the unique solution to the Bellman-type equations[123]

$$\begin{aligned} \mathbf{A}^{\#} &= (\mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}) + \mathbf{T}(\tau)\mathbf{A}^{\#} \\ &= (\mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}) + \mathbf{A}^{\#}\mathbf{T}(\tau), \end{aligned} \tag{1.10}$$

with constraints $\mathbf{A}^{\#}\boldsymbol{\pi} = \mathbf{0}$ and $\sum_{\gamma} A_{\gamma j}^{\#} = 0 \; \forall \, j \in \mathcal{S}$. The diagonal elements of $\mathbf{A}^{\#}$ are strictly positive.

In practice, it is sometimes more convenient to compute and work with the group inverse rather than the fundamental matrix, since $\mathbf{A}^{\#}$ can be obtained without knowledge of the stationary distribution.[119] Furthermore, the elements of the group inverse have a probabilistic interpretation. Specifically, $A_{ij}^{\#}$ represents the expected deviation in the number of visits to the i-th node for the relaxation process to the stationary distribution initialized from the j-th node, compared to the average number of visits when starting at a node chosen randomly in proportion to the stationary distribution.[124] Formally, if $N_{ij}^{(n)}$ denotes the expected number of times that the i-th node is visited on a trajectory of n steps initialized from the j-th node, then[119]

$$\lim_{n\to\infty} \Big(N_{ij}^{(n)} - N_{ik}^{(n)} \Big) = A_{ij}^{\#} - A_{ik}^{\#}. \tag{1.11}$$

A key macroscopic quantity characterizing a particular transition is the mean first passage

time[119,125–136] (MFPT), defined as the expected first hitting time for trajectories to reach an absorbing (target) state given an initial condition.[2] The matrix of MFPTs for all pairwise internode transitions can be expressed directly in terms of the fundamental matrix or the group inverse,[119,129]

$$\boldsymbol{\mathcal{T}} = (\mathbf{I} - \mathbf{G} + \mathbf{E}\mathbf{G}_{\mathrm{d}})\mathbf{D}\tau, \tag{1.12}$$

where $\mathbf{E}$ is the $|\mathcal{S}| \times |\mathcal{S}|$-dimensional matrix with all elements equal to unity, $\mathbf{G}_{\mathrm{d}}$ is the matrix with diagonal elements of a generalized inverse $\mathbf{G}$ and off-diagonal elements equal to zero, and $\mathbf{D} = \mathrm{diag}(\boldsymbol{\pi})$. The matrix whose elements $\mathcal{V}_{ij}$ are the variances of the FPT distribution for all pairwise internode transitions is given by $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{T}}^{(2)} - \boldsymbol{\mathcal{T}} \circ \boldsymbol{\mathcal{T}}$, where $\circ$ denotes the element-wise product, and[115,117]

$$\boldsymbol{\mathcal{T}}^{(2)} = \Big(2[\boldsymbol{\mathcal{T}}\mathbf{G} - (\boldsymbol{\mathcal{T}}\mathbf{G})_{\mathrm{d}}\mathbf{E}] + \boldsymbol{\mathcal{T}}^{(2)}_{\mathrm{d}}\mathbf{D}\boldsymbol{\mathcal{T}}\Big)\tau^2, \tag{1.13}$$

with

$$\boldsymbol{\mathcal{T}}^{(2)}_{\mathrm{d}} = \mathbf{D}^{-1} + 2\mathbf{D}^{-1}[\mathbf{G}(\mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top})]_{\mathrm{d}}\mathbf{D}^{-1}. \tag{1.14}$$

More complicated expressions exist for matrices with elements corresponding to the higher moments of the FPT distributions for internode transitions.[116]

Another important property that characterizes the global dynamics of an irreducible Markov chain is the average mixing time,[1] which can be thought of as the expected time for an initial occupation probability distribution to relax to the stationary distribution. More formally, the average mixing time is the expected time for trajectories to first hit a target node that is sampled in proportion to the stationary distribution.[137] This quantity is independent of the initial condition[138–140] and is known as the Kemeny constant,[1,118] given by[112]

$$\begin{aligned}\zeta_{\mathrm{K}} &= \sum_{\gamma \in \mathcal{S}} \mathcal{T}_{\gamma j}\pi_{\gamma} \;\forall\, j \in \mathcal{S} \\ &= \mathrm{Tr}(\mathbf{Z})\tau = \Big(\mathrm{Tr}(\mathbf{A}^{\#}) + 1\Big)\tau.\end{aligned} \tag{1.15}$$

Higher moments of the mixing time distribution are dependent on the initial state, but can nonetheless be derived from a generalized inverse. Let the variance of the mixing time distribution when the relaxation to equilibrium is initialized from the j-th node be denoted by ν_j. These variances are given by

$$\nu_j = \sum_{\gamma \in \mathcal{S}} \mathcal{T}^{(2)}_{\gamma j}\pi_{\gamma} - \zeta_{\mathrm{K}}^2, \tag{1.16}$$

and are the elements of the vector[117] (using Eqs. 1.13 and 1.14)

$$\begin{aligned}\boldsymbol{\nu} =&\Big(2[\zeta_{\mathrm{K}}(\mathbf{1}_{\mathcal{S}}^{\top}\mathbf{G})^{\top} - \mathrm{Tr}(\mathbf{LG})\mathbf{1}_{\mathcal{S}}] \\ &+ 2(\boldsymbol{\alpha}^{\top}[\mathbf{I} - \mathbf{G} + \mathbf{E}\mathbf{G}_{\mathrm{d}}])^{\top} - \zeta_{\mathrm{K}}\mathbf{1}_{\mathcal{S}}\Big)\tau - \zeta_{\mathrm{K}}^{2}\mathbf{1}_{\mathcal{S}},\end{aligned} \tag{1.17}$$

where $\mathbf{L} = \mathbf{D}\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_{\mathrm{d}}^{-1}\boldsymbol{\mathcal{T}}$ is the mixing matrix,[138] and $\boldsymbol{\alpha}$ is the vector with elements $\alpha_i = \sum_{\gamma} \mathcal{T}_{i\gamma}\pi_{\gamma}$. The Kemeny constant effectively quantifies the extent to which all of the nodes in the state space of a Markov chain are mutually reachable.[137] When the dynamics are diffusive, the average mixing time is relatively small, and the variances of the mixing time distributions are likewise small and fairly uniform. In contrast, when there are metastable states, relaxation to the stationary distribution is a slow process, and the mixing time distributions are fat-tailed, with large means and variances, owing to the existence of rare event transitions.

1.2.3 Eigendecomposition analysis

It is sometimes more convenient, for reasons of numerical stability or efficiency, to compute properties of a Markov chain via an eigendecomposition operation, instead of a matrix inversion operation required to compute the group inverse (Sec. 1.2.2). Similar to the utility of generalized inverses, many dynamical quantities can be expressed in terms of the eigenspectrum of a Markovian network. For instance, the average mixing time (Eq. 1.15) is given directly by the eigenvalues of an irreducible and reversible Markov chain. In discrete-time,[108]

$$\zeta_{\mathrm{K}} = \Big(1 + \sum_{k} \frac{1}{1-\lambda_k}\Big)\tau. \tag{1.18}$$

The MFPTs for internode transitions can also be written in terms of the eigenspectrum of a reversible Markov chain.[133] In continuous-time, the $i \leftarrow j$ MFPT is given by[108,141]

$$\mathcal{T}_{ij} = \frac{1}{\pi_j}\sum_{k>1}^{|\mathcal{S}|} \frac{1}{-\gamma_k}\psi_j^{(k)}\Big[\phi_j^{(k)} - \phi_i^{(k)}\Big]. \tag{1.19}$$

The overall $\mathcal{A} \leftarrow \mathcal{B}$ MFPT, for a transition to an arbitrary absorbing macrostate $\mathcal{A}$ from a set of initial nodes $\mathcal{B}$, is a sum of MFPTs for transitions from each of the nodes comprising $\mathcal{B}$, where each term is weighted by the initial occupation probability of the starting node,

$$\mathcal{T}_{\mathcal{A}\mathcal{B}} = \sum_{b\in\mathcal{B}} p_b(0)\mathcal{T}_{\mathcal{A}b}. \tag{1.20}$$

Although the MFPT is the usual dynamical observable,[119,125–136] the complete FPT distribution

contains valuable dynamical information that is not captured in the average. For instance, the width of the FPT distribution characterizes the heterogeneity of the TPE. When the transition probabilities or rates of the Markov chain depend on an external parameter, such as the temperature in physical systems,[88] the form of the FPT distribution may change dramatically and with a relatively well-defined threshold.[142] The existence of multiple peaks in the FPT distribution suggests the presence of competing transition mechanisms that each make a non-negligible contribution to the MFPT.[101] To justify 'lumping' a subset of nodes of a Markov chain, and thus obtain a reduced Markovian representation of the dynamics,[1,143–153] the FPT distribution for escape from the community of nodes ought to be an exponential distribution, which has the memoryless property.[51,110]

In discrete-time, the FPT distribution can be computed from eigendecomposition of the substochastic transition probability matrix $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$, comprising only transient nodes of the set $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$. The equivalent formulation in continuous-time uses the corresponding rate matrix $\mathbf{K}_{\mathcal{Q}\mathcal{Q}}$, for which the diagonal elements include contributions from transitions to absorbing nodes of the set $\mathcal{A}$, so that the columns of $\mathbf{K}_{\mathcal{Q}\mathcal{Q}}$ do not necessarily sum to zero. $\mathbf{K}_{\mathcal{Q}\mathcal{Q}}$ does not have a zero eigenvalue and associated stationary distribution, and instead all eigenvalues have a negative real component. Since the dynamics within the transient set of nodes $\mathcal{Q}$ are unchanged prior to absorption, we can still write a linear master equation (*cf.* Eq. 1.2) for the dynamics within this subnetwork. There is a probability flux out of the macrostate $\mathcal{Q}$, which defines the probability distribution $p(t_{\mathrm{FPT}})$ for the $\mathcal{A} \leftarrow \mathcal{Q}$ first passage times t_{FPT}. Noting that the propagation of an initial probability distribution $\mathbf{p}(0)$ according to the linear master equation can be written in terms of the eigenspectrum of the rate matrix (*cf.* Eq. 1.3),[8]

$$\mathbf{p}(t) = \sum_{k=1}^{|\mathcal{S}|} \Big(\boldsymbol{\psi}^{(k)} \otimes \boldsymbol{\phi}^{(k)} \Big) \mathbf{p}(0) e^{\gamma_k t}, \tag{1.21}$$

where $\otimes$ denotes the outer product, we obtain the properly normalized FPT distribution[52]

$$p(t_{\mathrm{FPT}}) = -\mathbf{1}_{\mathcal{Q}}^{\top} \Big(\sum_k \gamma_k^{\mathcal{Q}} \boldsymbol{\psi}_{\mathcal{Q}}^{(k)} \otimes \boldsymbol{\phi}_{\mathcal{Q}}^{(k)} \Big) \Big(\sum_l e^{\gamma_l^{\mathcal{Q}} t} \boldsymbol{\psi}_{\mathcal{Q}}^{(l)} \otimes \boldsymbol{\phi}_{\mathcal{Q}}^{(l)} \Big) \mathbf{p}_{\mathcal{Q}}(0) \tag{1.22}$$

$$= -\sum_k \gamma_k^{\mathcal{Q}} e^{\gamma_k^{\mathcal{Q}} t} \mathbf{1}_{\mathcal{Q}}^{\top} \Big(\boldsymbol{\psi}_{\mathcal{Q}}^{(k)} \otimes \boldsymbol{\phi}_{\mathcal{Q}}^{(k)} \Big) \mathbf{p}_{\mathcal{Q}}(0). \tag{1.23}$$

Here, $\gamma_k^{\mathcal{Q}}$ is the k-th eigenvalue of $\mathbf{K}_{\mathcal{Q}\mathcal{Q}}$, with $\boldsymbol{\phi}_{\mathcal{Q}}^{(k)}$ and $\boldsymbol{\psi}_{\mathcal{Q}}^{(k)}$ the corresponding left and right eigenvectors, respectively, and $\mathbf{p}_{\mathcal{Q}}(0)$ is the vector containing the initial occupation probabilities for the transient nodes. In writing Eq. 1.23 from Eq. 1.22, we have assumed orthonormality of the eigenvectors (*cf.* Eq. 1.5). The eigendecomposition of the FPT distribution is particularly insightful, since it separates the distribution into contributions from individual

dynamical eigenmodes. The k-th orthonormal eigenmode makes a dominant contribution to the FPT distribution when the product $\mathbf{1}_{\mathcal{Q}}^{\top}[\boldsymbol{\psi}_{\mathcal{Q}}^{(k)} \otimes \boldsymbol{\phi}_{\mathcal{Q}}^{(k)}]\mathbf{p}_{\mathcal{Q}}(0)$ is close to unity. The n-th moment of the FPT distribution (Eq. 1.23) is given by[52,154]

$$\begin{aligned}\langle t_{\mathrm{FPT}}^{n} \rangle &= \int_{0}^{\infty} t_{\mathrm{FPT}}^{n} p(t_{\mathrm{FPT}}) dt_{\mathrm{FPT}} \\ &= n! \sum_{k} \frac{1}{|\gamma_{k}^{\mathcal{Q}}|^{n}} \mathbf{1}_{\mathcal{Q}}^{\top} \Big(\boldsymbol{\psi}_{\mathcal{Q}}^{(k)} \otimes \boldsymbol{\phi}_{\mathcal{Q}}^{(k)} \Big) \mathbf{p}_{\mathcal{Q}}(0). \end{aligned} \tag{1.24}$$

1.2.4 Numerical considerations for linear algebra methods

Many properties of an irreducible Markov chain can be directly expressed in terms of a fundamental matrix, obtained via a matrix inversion operation (Sec. 1.2.2). In Sec. 1.2.3, we noted that several key dynamical quantities characterizing the dynamics of a Markov chain can also be computed straightforwardly from its eigenspectrum, including MFPTs (Eq. 1.19), the FPT distribution (Eq. 1.22), the time-dependent occupation probability distribution (Eq. 1.21), and the average mixing time (Eq. 1.18). Typical algorithms for eigendecomposition, and for matrix inversion or diagonalization, have time complexity $\mathcal{O}(|\mathcal{S}|^3)$, which is comparable to the time complexity of the state reduction algorithms described in Secs. 1.3 and 1.4. However, dense linear algebra methods are afflicted by the propagation of roundoff error in the finite precision arithmetic when applied to nearly reducible Markov chains.[90,92,155] The extent to which a Markov chain with metastable states is ill-conditioned is essentially independent of its dimensionality.[140] Hence, for Markov chains featuring a rare event, numerical error can prohibit the use of conventional dense methods, such as LU decomposition to solve linear systems of equations,[35] or the QR algorithm to perform an eigendecomposition,[156] even when the network comprises just a few nodes.[52,93]

For reversible Markov chains,[2] marginal improvements in the numerical stability of linear algebra methods can be gained by employing the symmetrized transition rate matrix (in the continuous-time case), with elements $\widetilde{K}_{ij} = (K_{ij}K_{ji})^{1/2}$, or the symmetrized transition probability matrix (in the discrete-time case), with elements $\widetilde{T}_{ij}(\tau) = (T_{ij}(\tau)T_{ji}(\tau))^{1/2}$.[52] These symmetrized matrices have the same eigenvalues as their unsymmetrized counterparts, and the elements of the right and left eigenvectors for the two formulations are related via $\psi_{j}^{(k)} = \pi_{j}^{1/2}\widetilde{\psi}_{j}^{(k)}$ and $\phi_{j}^{(k)} = \pi_{j}^{-1/2}\widetilde{\phi}_{j}^{(k)}$, respectively.[8]

An alternative framework to solve linear systems of equations or to determine the eigenspectrum of a Markov chain is provided by iterative methods,[157–160] which can be optimized to treat Markov chains featuring a spectral gap via preconditioning.[161] Unlike direct solution methods, iterative methods preserve the sparsity of a stochastic matrix, and are therefore well-suited to treating high-dimensional structured systems,[162] such as those that arise in population

dynamics models.[37] Below, we briefly outline the basis of these iterative methods and preconditioning strategies to aid their convergence. A detailed review of both direct and iterative solution methods in the context of Markov chains can be found in Ref. 77.

As an illustrative example, we consider application of the simplest iterative solution method, namely the power method, to compute the stationary distribution of a Markov chain, i.e. to solve $\mathbf{K}\boldsymbol{\pi} = \mathbf{0}$ (in continuous-time) or $\mathbf{T}(\tau)\boldsymbol{\pi} = \boldsymbol{\pi}$ (in discrete-time). In the power method, the solution vector $\mathbf{x}$ is repeatedly updated according to

$$\mathbf{x}' = \mathbf{T}(\tau)\mathbf{x} = (\mathbf{I} - \mathbf{K})\mathbf{x}. \tag{1.25}$$

The convergence rate for Eq. 1.25 is λ_2.[77] Thus when the Markov chain is nearly reducible, the convergence $\mathbf{x} \to \boldsymbol{\pi}$ is exceedingly slow. To remedy this problem, we may introduce a preconditioning matrix $\mathbf{M}$ that is readily invertible, ideally such that $\mathbf{M} \approx \mathbf{K}$ and the inverse $\mathbf{M}^{-1}$ yields a matrix $(\mathbf{I} - \mathbf{M}^{-1}\mathbf{K})$ that has no subdominant eigenvalues close to the unique unit eigenvalue. The rate of convergence of the preconditioned version of Eq. 1.25 is consequently fast. In general, preconditioning refers to any method to modify a system of linear equations by premultiplication. That is, $\mathbf{A}\mathbf{x} = \mathbf{b} \Rightarrow \mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$, where $\mathbf{M}\mathbf{c} = \mathbf{y}$ can be solved efficiently with any $\mathbf{y}$, and this transformation favourably alters the distribution of eigenvalues, thereby aiding the convergence of iterative methods.

The success of iterative solution methods applied to ill-conditioned Markov chains exhibiting rare event dynamics is clearly strongly dependent on the choice of $\mathbf{M}$. Some iterative methods simplify the additional input required from the user by implicitly incorporating a preconditioning matrix. Successive overrelaxation[163] (SOR) to solve the linear problem $\mathbf{A}\mathbf{x} = \mathbf{b}$, which can be thought of as a generalization of Gauss-Seidel iteration, splits the relevant matrix $\mathbf{A}$ as

$$\omega\mathbf{A} = (\mathbf{D} - \omega\mathbf{L}) - (\omega\mathbf{U} + (1 - \omega)\mathbf{D}), \tag{1.26}$$

with $\mathbf{D}$ a diagonal matrix, $\mathbf{L}$ and $\mathbf{U}$ strictly lower- and upper-triangular matrices, respectively, and $\omega > 0$ is the relaxation factor. A solution vector $\mathbf{x}$ is then iterated according to

$$\mathbf{x}' = (\mathbf{D} - \omega\mathbf{L})^{-1}(\omega\mathbf{U} + (1 - \omega)\mathbf{D})\mathbf{x} + \omega(\mathbf{D} - \omega\mathbf{L})^{-1}\mathbf{b}. \tag{1.27}$$

The matrix $\omega^{-1}(\mathbf{D} - \omega\mathbf{L})$ effectively acts as a preconditioner in SOR.[163] Hence, the problem of selecting an appropriate preconditioning matrix is simplified to the problem of determining a suitable value for the scalar parameter ω. In practice, there is very little guidance on an appropriate choice of relaxation factor for solving a given system of linear equations if the matrix $\mathbf{A}$ is unstructured, and the possible gains in efficiency may be quite limited.[93]

Krylov subspace methods are a class of sparse iterative procedures that can be used to perform an eigendecomposition of a matrix $\mathbf{A}$.[155,156,164,165] The idea is to generate a sequence of m vectors, $\{\mathbf{v}_k\}\ \forall\, k = 1, \ldots, m$, which form an orthonormal basis of the m-th order *Krylov subspace* spanned by the set of vectors $(\mathbf{v}_1, \mathbf{A}\mathbf{x}, \ldots, \mathbf{A}^{m-1}\mathbf{v}_1)$, where $\mathbf{v}_1$ is an arbitrary normalized vector.[166] The Arnoldi algorithm[167] provides an iterative method to produce a sequence of orthonormal vectors $\mathbf{V}_m = (\mathbf{v}_1, \ldots, \mathbf{v}_m)$ spanning the Krylov subspace.[168,169] The procedure yields a $m \times m$-dimensional upper Hessenberg matrix $\mathbf{H}_m$ that satisfies $\mathbf{H}_m = \mathbf{V}_m^\top \mathbf{A}\mathbf{V}_m$. For a sufficiently large number of iterations m, the dominant eigenvalues of $\mathbf{H}$ converge to those of $\mathbf{A}$. Additionally, if $\boldsymbol{\varphi}_k^{(m)}$ denotes the k-th dominant eigenvector of $\mathbf{H}_m$, then the so-called Ritz vector $\mathbf{V}_m \boldsymbol{\varphi}_k^{(m)}$ approximates the k-th eigenvector of $\mathbf{A}$.[170] The generalized minimal residual (GMRES) algorithm[171] adapts this concept to solve the linear problem $\mathbf{A}\mathbf{x} = \mathbf{b}$.

For well-conditioned matrices $\mathbf{A}$, the eigenvalues of $\mathbf{H}_m$ converge rapidy to those of $\mathbf{A}$, so that the former matrix is comparatively low-dimensional and its eigenpairs can be found efficiently by direct solution methods. For more computationally challenging problems, the required number of iterations m is large, so that the storage requirements and operation counts of the procedure become prohibitively large. Typical implementations of the Arnoldi and related algorithms incorporate implicit restarting to improve efficiency and avoid "wasteful" memory usage.[172,173] However, numerical problems remain pervasive for pathological systems such as nearly reducible Markov chains.[174] In this ill-conditioned regime, the Arnoldi and GMRES methods can be made more effective by preconditioning in the usual way,[161] i.e. replacing the matrix $\mathbf{A}$ with $\mathbf{M}^{-1}\mathbf{A}$, where $\mathbf{M}$ is similar to $\mathbf{A}$ and $\mathbf{M}^{-1}\mathbf{y}$ can be evaluated efficiently and reliably for any vector $\mathbf{y}$.

Similar strategies can also be used to solve systems of ODEs. A numerically stable approach to solving the master equation for the time-dependent occupation probability distribution $\mathbf{p}(t)$ (Eq. 1.2) uses a stiff ODE[175,176] integrator,[177–181] employing GMRES iterations[171] preconditioned with an appropriate matrix to solve the relevant system of linear equations.[182]

In principle, preconditioned iterative methods provide stable algorithms to analyze Markov chains that are scalable in terms of both operation counts and memory usage. However, this approach is of limited use in practice, owing to nontrivial and system-specific considerations that arise. That is, the rate of convergence of iterative procedures is strongly dependent on the choice of preconditioning matrix and on the initial guess for the solution vector. Without appropriate preconditioning, iterative solvers may converge unfeasibly slowly for linear problems involving nearly reducible Markov chains.[93] For arbitrary Markov chains, there is limited *a priori* information on appropriate input to sparse procedures, and strategies are not generalizable. Hence, the additional parameters must usually be explored empirically

in a trial-and-error fashion.

1.2.5 Transition path theory

The theory presented in Secs. 1.2.2 and 1.2.3 was concerned with global dynamical properties of Markov chains. The role of individual nodes in determining macroscopic behaviour is elucidated by transition path theory (TPT).[95–99] TPT provides a theoretical framework to analyze the transition path ensemble[46,47,101] (TPE) of reactive paths,[100,183] which proceed directly from an initial macrostate $\mathcal{B}$ to an absorbing macrostate $\mathcal{A}$, without revisiting $\mathcal{B}$. The central object of TPT is the vector of committor probabilities[184–187] for nodes. We denote by q_j^+ the forward $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for the j-th node. q_j^+ is the probability that a trajectory occupying node j will visit the absorbing macrostate $\mathcal{A}$ before the initial macrostate $\mathcal{B}$. It is in this sense that the committor probability represents an idealized reaction coordinate quantifying the progress of a $\mathcal{A} \leftarrow \mathcal{B}$ transition, onto which state variables can be projected.[111]

Formally, the forward committor probability for the j-th node is defined as[98]

$$q_j^+ \equiv \mathbb{P}_j(h_{\mathcal{B}} < h_{\mathcal{A}}), \tag{1.28}$$

where $\mathbb{P}_j$ denotes the probability considering all trajectories $\xi(t)$ initialized from the node j, and the random variable $h_{\mathcal{B}} = \inf\{t > 0 : \xi(t) \in \mathcal{B}\}$ is the first hitting time for the macrostate $\mathcal{B}$.[2] In the context of Markovian networks, a trajectory $\xi(t)$ is a sequence of visited nodes and associated times. For reversible Markov chains, the backward committor probabilities, for the $\mathcal{B} \leftarrow \mathcal{A}$ direction, which are defined analogously to Eq. 1.28 but for the time-reversed dynamics,[98] are related to the forward committor probabilities simply by $q_j^- = 1 - q_j^+$.[111] The committor probabilities can be written as the solution of a series of linear equations obtained by a first-step analysis[4]

$$q_j^+ = \sum_{i \notin \mathcal{A}} T_{ij} q_i^+, \tag{1.29}$$

where we have noted that, in this definition, $q_{a\in\mathcal{A}}^+ = 1$ and $q_{b\in\mathcal{B}}^+ = 0$. See Ref. 98 for a detailed proof. Eq. 1.29 can be solved using a variety of methods,[111,188] such as Gauss-Seidel iteration,[189,190] successive over-relaxation,[92,165] or robustly by state reduction (Sec. 1.5.1).[191]

The committor probabilities can also be computed from the second dominant (i.e. first nontrivial) right eigenvector of the modified stochastic matrix $\overline{\mathbf{T}}$ for which nodes of the sets $\mathcal{B}$ and $\mathcal{A}$ are subsumed into single nodes b and a, respectively, and where both of these supernodes are considered to be absorbing. That is, $\overline{T}_{aj} = 0 \; \forall \, j \neq a$ and $\overline{T}_{bj} = 0 \; \forall \, j \neq b$.

This transition matrix has two unit eigenvalues, one of which is associated with the unit vector, and the second is associated with an eigenvector that we shall denote by $\overline{\boldsymbol{\psi}}^{(2)}$. The committor probability for the j-th node is then[111]

$$q_j^+ = \frac{\overline{\psi}_j^{(2)} - \overline{\psi}_a^{(2)}}{\overline{\psi}_b^{(2)} - \overline{\psi}_a^{(2)}}. \tag{1.30}$$

Obtaining the committor probabilities using Eq. 1.30 is the most scalable approach, since the Lanczos algorithm[169] can be used to compute $\overline{\psi}^{(2)}$ efficiently for sparse systems.[162] Moreover, this formulation is readily extended to the first hitting problem (*cf.* Eq. 1.28) for an arbitrary number of target states.[111]

Several quantities characterizing the $\mathcal{A} \leftarrow \mathcal{B}$ TPE at a nodewise level of detail can be obtained from the committor probabilities.[95–99] One such dynamical property of interest is the probability distribution of reactive trajectories m_j^{R}, defined as the probability that the j-th node is occupied at equilibrium by a trajectory that is reactive.[98] Intuitively, this quantity is a product of the probabilities that the trajectory last visited $\mathcal{B}$ before $\mathcal{A}$ and will next visit $\mathcal{A}$ before $\mathcal{B}$, and the stationary probability of the j-th node. Therefore, from the definition of the committor probabilities (Eq. 1.28), if the detailed balance condition holds, this probability is given by

$$m_j^{\mathrm{R}} = \pi_j q_j^+ (1 - q_j^+). \tag{1.31}$$

The probability that any given trajectory at equilibrium is reactive is equal to the normalization factor $Z_{\mathcal{AB}} = \sum_{j \in \mathcal{S}} m_j^{\mathrm{R}}$. The normalized distribution of reactive trajectories, $m_j = m_j^{\mathrm{R}} / Z_{\mathcal{AB}}$, is the probability to observe a trajectory at node j, conditional on the trajectory being reactive.

In the continuous-time case,[98] the net probability flux of reactive trajectories along the $i \leftarrow j$ edge of the network is

$$f_{ij}^+ = \begin{cases} \pi_j K_{ij} (q_i^+ - q_j^+), & \text{if} \quad q_i^+ > q_j^+, \\ 0, & \text{otherwise.} \end{cases} \tag{1.32}$$

In discrete-time, T_{ij} replaces K_{ij} in Eq. 1.32.[111] The $\mathcal{A} \leftarrow \mathcal{B}$ steady state rate constant,[192] $k_{\mathcal{AB}}^{\mathrm{SS}}$, can be calculated by defining an isocommittor cut[183] Σ_α, which partitions the network into two sets $\mathcal{A}^* \supset \mathcal{A}$ and $\mathcal{B}^* \supset \mathcal{B}$ such that $q_{a \in \mathcal{A}^*}^+ > \alpha > q_{b \in \mathcal{B}^*}^+$, and summing over the net probability fluxes associated with the edges of the cut[193,194]

$$k_{\mathcal{AB}}^{\mathrm{SS}} = \frac{1}{\pi_{\mathcal{B}}} \sum_{i \in \mathcal{A}^*, j \in \mathcal{B}^*} f_{ij}^+ = \frac{J_{\mathcal{AB}}}{\pi_{\mathcal{B}}}. \tag{1.33}$$

Here, we have denoted the steady state reactive $\mathcal{A} \leftarrow \mathcal{B}$ flux[43] as $J_{\mathcal{AB}}$, and $\pi_{\mathcal{B}} = \sum_{b \in \mathcal{B}} \pi_b$. Of particular interest is the isocommittor cut $\Sigma_{\alpha=0.5}$, which defines the edges that constitute the transition state ensemble (TSE). The TSE essentially characterizes the boundary between the effective basins of attraction associated with $\mathcal{A}$ and $\mathcal{B}$.[195–199] Alternative approaches to characterize the TSE are based on Bayesian path statistics[200] and the emission-absorption cut defined in terms of the eigenvectors.[64]

Eq. 1.33 shows that the steady state reactive flux, $\mathcal{J}_{\mathcal{AB}}$, can be exactly decomposed into additive contributions from individual edges. Hence, the local states that constitute the dominant channels for the productive pathways can be readily identified. Alternatively, $\mathcal{J}_{\mathcal{AB}}$ can be exactly decomposed into a finite set of contributions from transition flux-paths,[193,201] as we show in Chapter 2. This procedure allows for a pathwise analysis to quantify the relative importance of competing mechanisms for the transition process.

The expressions for the probability distribution of reactive trajectories for nodes (Eq. 1.31), and the net reactive fluxes for edges (Eq. 1.32), correspond to the equilibrium TPE. That is, these quantities are concerned with the situation where the system has relaxed to a steady state. Formally, this analysis involves considering a trajectory of infinite length in time, which continually transitions between the $\mathcal{A}$ and $\mathcal{B}$ states.[98] The dynamical obserable for the equilibrium path ensemble is $k^{\mathrm{SS}}_{\mathcal{AB}}$ (Eq. 1.33). In Ref. 100, a theory of transition paths was developed for the nonequilibrium path ensemble, i.e. considering the first hitting problem where trajectories are absorbed at the $\mathcal{A}$ state. In particular, expressions were derived for an analogue of the net reactive flux (*cf.* Eq. 1.32), and for the expected numbers of times that nodes are visited along reactive paths. In Chapter 3, expressions for the probabilities that nodes are visited along reactive paths are derived for both the nonequilibrium and equilibrium cases.[191] Together with the committor probabilities for nodes, the reactive visitation probabilities provide a rigorous metric to identify the local states that have a dominant influence on the productive $\mathcal{A} \leftarrow \mathcal{B}$ dynamics.[101]

1.3 Graph transformation method to calculate MFPTs

We now proceed to describe numerically stable *state reduction* methods to compute the properties of a Markov chain, including moments of the first passage time distribution (Eq. 1.24), the average mixing time (Eq. 1.15), and other quantities introduced in Sec. 1.2. The state reduction approach is exact and requires no additional knowledge of the Markov chain besides the transition probability or rate matrix. Hence, state reduction procedures are of greater general utility than sparse linear algebra algorithms in application to nearly reducible Markov chains, since convergence of the latter methods in the metastable regime

is strongly dependent on the careful choice, and moreover existence, of a suitable auxiliary matrix in the preconditioning scheme (Sec. 1.2.4). In this section, we introduce the concept of a renormalized Markov chain, which is central to the state reduction methodology, and prove that renormalization can be used to robustly compute the MFPT for a transition. In Sec. 1.4, we describe further closely related state reduction methods for the exact analysis of Markovian network dynamics.

1.3.1 Graph transformation algorithm

The graph transformation[89–93] (GT) algorithm is a procedure for the iterative elimination of nodes in an arbitrary Markov chain, while preserving the $\mathcal{A} \leftarrow \mathcal{B}$ mean first passage time (MFPT) for the transition from an initial node $\{b\} \equiv \mathcal{B}$ to an absorbing macrostate $\mathcal{A}$.[94] We denote the set of intervening nodes by $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$. The GT method uses renormalization of transition probabilities, and of mean waiting times (in the continuous-time case) or lag times (in discrete-time) for transitions from nodes, to preserve individual path probabilities and the average time associated with the ensemble of paths to the absorbing state.

At each step of the nodewise iterative GT algorithm, the n-th node, $n \in \mathcal{I}$, is eliminated from the network, and the probabilities for internode $i \leftarrow j$ transitions on the remaining network are renormalized according to

$$\begin{aligned} T'_{ij} &= T_{ij} + T_{in} T_{nj} \sum_{m=0}^{\infty} T_{nn}^{m} \\ &= T_{ij} + \frac{T_{in} T_{nj}}{1 - T_{nn}}. \end{aligned} \tag{1.34}$$

The waiting or lag times of nodes j, which we denote by τ_j, are likewise updated according to

$$\begin{aligned} \tau'_j &= \sum_{\gamma \neq n} \left\{ T_{\gamma j} \tau_j + T_{\gamma n} T_{nj} \sum_{m=0}^{\infty} \left(\tau_j + (m+1) \tau_n T_{nn}^{m} \right) \right\} \\ &= \tau_j + \frac{T_{nj} \tau_n}{1 - T_{nn}}. \end{aligned} \tag{1.35}$$

The transition probabilities for pairs of nodes i and j, where one or both of the nodes are not directly connected to n, are unaffected by the renormalization (Eq. 1.34). If i and j were not directly connected in the untransformed network, but both nodes were directly connected to n, i.e. the sequence of direct transitions $i \leftarrow n \leftarrow j$ existed prior to renormalization, then a new $i \leftarrow j$ transition connects the pair of nodes in the renormalized network. Hence, the renormalized network at the n-th iteration of the GT algorithm is less sparse than the

network at the $(n-1)$-th iteration, and self-loop $(j \leftarrow j)$ transtions, if not initially present, are introduced into the successive Markov chains in the course of the algorithm. The mean waiting or lag time for a transition from the j-th node, τ_j, increases upon elimination of the n-th node if the $n \leftarrow j$ transition exists, and remains unchanged by the renormalization otherwise (Eq. 1.35). Hence, if the τ_j are initially uniform, as is the case for the lag times of a DTMC, then they become nonuniform in the renormalized network. The effect of the renormalization procedure to eliminate a single node (Eqs. 1.34 and 1.35) is illustrated in Fig. 1.1.

Eq. 1.34 conserves the probability flow out of all noneliminated nodes, i.e. $\sum_\gamma T'_{\gamma j} = 1 \; \forall \, j \neq n$.[92] The renormalized transition probabilities T'_{ij} subsume all $i \leftarrow j$ transitions that occurred indirectly, i.e. via intervening n, with an arbitrary number of self-loop transitions of n, on the original network.[84] That is, the transition probabilities for the renormalized network not only account for direct $i \leftarrow j$ transitions, but also 'round-trip' transitions $i \leftarrow n \leftarrow j$, $i \leftarrow n \leftarrow n \leftarrow j$, etc. The renormalized mean waiting or lag times τ'_j (Eq. 1.35) have a similar probabilistic interpretation. Namely, τ'_j represents the expected time for a transition from the j-th node, accounting for the average contribution from deviations via the eliminated node n.[140] That is, τ'_j includes the average time attributable to $n \leftarrow \ldots \leftarrow n \leftarrow j$ loops before proceeding to escape from $\{j\} \cup \{n\}$.

Eqs. 1.34 and 1.35 exactly preserve the MFPTs from any given transient node in the network to the set of absorbing nodes $\mathcal{A}$. The renormalization of transition probabilities (Eq. 1.34) also preserves the probabilities associated with individual paths (in a renormalized representation) from transient to absorbing nodes. Hence, the elements B_{ab} of the absorption matrix $\mathbf{B}$, where $a \in \mathcal{A}$ and $b \in \mathcal{B}$, are given by the renormalized transition probabilities T'_{ab} of the network where all nodes of the set $(\mathcal{A} \cup \{b\})^\mathsf{c}$ have been eliminated using Eq. 1.34. A formal proof of these statements is the subject of Sec. 1.3.2. If the original Markov chain is irreducible, then the renormalized Markov chain is also irreducible, and an expression can be derived for the stationary distribution $\boldsymbol{\pi}$ of the transformed network.[74] This theorem forms the basis of state reduction methods to compute $\boldsymbol{\pi}$ (Secs. 1.4.1-1.4.3).

The GT method for the computation of $\mathcal{A} \leftarrow \mathcal{B}$ MFPTs is significantly more numerically stable than linear algebra methods.[92,93] The GT procedure can retain numerical precision even when a node n to be eliminated is associated with a dominant probability for the self-loop transition, $T_{nn} \to 1$, because in this case the equivalence $1 - T_{nn} \equiv \sum_{\gamma \neq n} T_{\gamma n}$ can be exploited to avoid performing the subtraction operation.[202,203] Using this numerical trick minimizes error in the finite precision arithmetic that is subsequently propagated,[75–80] which for nearly reducible Markov chains is otherwise prohibitively severe.[73,81]

The GT algorithm has a time complexity that is strongly dependent on the average

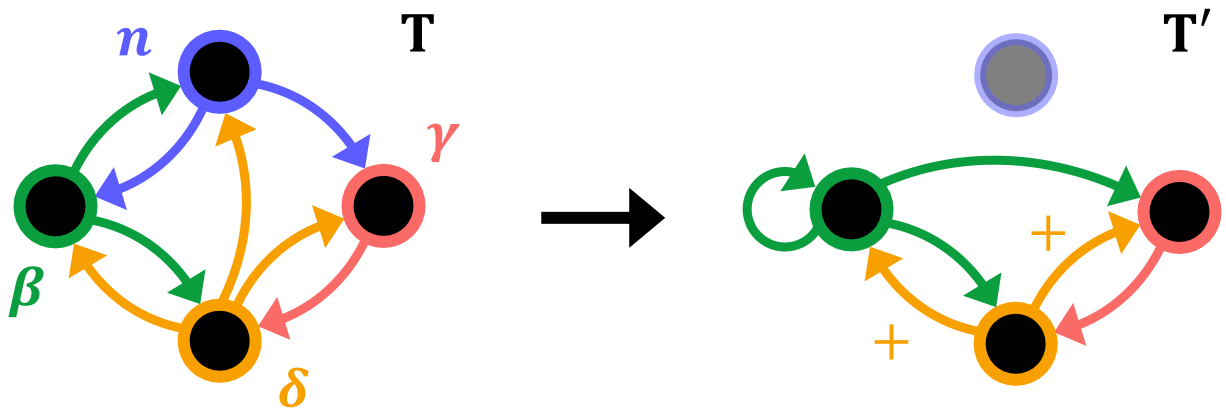


Figure 1.1: Schematic illustration of renormalization (Eq. 1.34) to eliminate a single node n from a Markov chain parameterized by the transition probability matrix $\mathbf{T}$. The transition probabilities $\mathbf{T}'$ of the renormalized Markov chain account for transitions that occur indirectly, via the "censored" state (here, the n-th node). Thus, the reduced model features a $\gamma \leftarrow \beta$ transition that is not present in the original network, which corresponds to the family of transitions $\gamma \leftarrow n \leftarrow \ldots \leftarrow n \leftarrow \beta$, where an arbitrary number of $n \leftarrow n$ transitions occur. Similarly, the reduced Markov chain contains a $\beta \leftarrow \beta$ transition, and the probabilities of the $\beta \leftarrow \delta$ and $\gamma \leftarrow \delta$ transitions have increased (indicated by $+$) to account for paths that proceed via the eliminated node n.

degree of nodes and on the heterogeneity of the node degree distribution.[90] Empirically, the nodewise iterative renormalization procedure has been observed to scale as $\mathcal{O}(|\mathcal{S}|^4)$ for sparse random networks and as $\mathcal{O}(|\mathcal{S}|^3)$ for some other classes of structured network.[84] Since a DTMC is less sparse than the corresponding CTMC, there is no advantage to converting from a continuous- to a discrete-time formulation (Eq. 1.3), as the state reduction computation will then be less efficient. The reverse operation (formally given by Eq. 1.4) is highly nontrivial to perform in practice,[140] and in any case an equivalent continuous-time representation of a DTMC does not necessarily exist.[10,204] The performance considerations for state reduction methods are complementary to those for matrix inversion and diagonalization methods (Sec. 1.2.4), which have time complexity $\mathcal{O}(|\mathcal{S}|^3)$,[205] but frequently fail when the Markov chain features a spectral gap.[81] GT is therefore the method of choice to compute MFPTs between two subsets of nodes in a Markov chain featuring a rare event, and likewise the state reduction procedures outlined in Sec. 1.4 allow for robust computation of further dynamical quantities that are otherwise challenging to obtain for nearly reducible Markov chains. There are numerous tricks for performance optimization of state reduction methods, for example prioritizing the elimination of nodes with a low degree,[92] switching from sparse to dense storage when the number of remaining nodes in the network falls below a threshold,[90] and eliminating multiple nodes at once via a matrix inversion operation (Sec. 1.4.1).[94,192]

1.3.2 Graph transformation proof

We wish to prove that the GT algorithm correctly preserves the MFPT from any given node in the network to a set of absorbing nodes $\mathcal{A}$. Recall that the overall $\mathcal{A} \leftarrow \mathcal{B}$ MFPT, $\mathcal{T}_{\mathcal{AB}}$, is an average of MFPTs $\mathcal{T}_{\mathcal{A}b}$ for transitions from a node $b \in \mathcal{B}$ of the initial set to any node of the absorbing set $\mathcal{A}$, for a specified initial occupation probability distribution (Eq. 1.20). The following argument demonstrates that the individual MFPTs $\mathcal{T}_{\mathcal{A}b}$, from which $\mathcal{T}_{\mathcal{AB}}$ is derived, can be computed exactly using the renormalized transition probabilities (Eq. 1.34) and waiting times (Eq. 1.35) that are obtained after eliminating all nodes of the set $(\mathcal{A} \cup b)^{\mathsf{c}}$, with $\mathcal{A}$ and b chosen arbitrarily.

First, we derive the renormalized mean waiting or lag times (Eq. 1.35) more explicitly. To do this, we introduce the reweighted transition probabilities $\widetilde{T}_{ij} = T_{ij} e^{\zeta \tau_j}$. Consider the n-step discrete path ξ specified as a sequence of visited nodes, $\xi = \{i_n \leftarrow i_{n-1} \leftarrow \ldots \leftarrow i_1\}$. The probability of this path from i_1 is $\mathcal{W}_\xi = \prod_{(i \leftarrow j) \in \xi} T_{ij}$, where the product includes all $i \leftarrow j$ transitions in the path ξ, with the correct multiplicities. The product of reweighted transition probabilities along the path, $\widetilde{\mathcal{W}}_\xi$, is defined similarly. The reweighted transition probabilities have the convenient property of satisfying

$$\left[\frac{d}{d\zeta}\widetilde{\mathcal{W}}_\xi\right]_{\zeta=0} = \mathcal{W}_\xi \sum_{k=1}^{n-1} \tau_{i_k}. \tag{1.36}$$

Hence, this derivative yields the product of the path probability and the average waiting time associated with the path. For an $a \in \mathcal{A} \leftarrow b \in \mathcal{B}$ path $\xi^{(a,b)}$, this quantity is the contribution of the path to the overall $\mathcal{A} \leftarrow b$ MFPT. Therefore to correctly preserve the $\mathcal{A} \leftarrow b$ MFPT by renormalization of the waiting times for nodes j that are directly connected to the n-th (eliminated) node, each renormalized waiting time τ'_j must be equal to the sum of derivatives (Eq. 1.36) of each of the reweighted transition probabilities for transitions from j to the set of nodes directly connected to j or n, excluding n. We will denote this set of adjacent nodes by Γ. If we use the GT relation for the reweighted transition probabilities (Eq. 1.34) in this expression, then we recover Eq. 1.35,[92]

$$\begin{aligned}\tau'_j &= \sum_{\gamma \in \Gamma} \left[\frac{d}{d\zeta}\widetilde{T}'_{\gamma j}\right]_{\zeta=0} = \sum_{\gamma \in \Gamma} \left[\frac{d}{d\zeta}\left(\widetilde{T}_{\gamma j} + \frac{\widetilde{T}_{\gamma n}\widetilde{T}_{nj}}{1 - \widetilde{T}_{nn}}\right)\right]_{\zeta=0} \\ &= \tau_j + \frac{T_{nj}\tau_n}{1 - T_{nn}}.\end{aligned} \tag{1.37}$$

While Eq. 1.34 preserves the probabilities associated with *individual* $\xi^{(a,q)}$ first passage paths (in their resulting reduced representation) from any transient node $q \in \mathcal{Q}$ to any

absorbing node $a \in \mathcal{A}$, Eq. 1.35 does *not* preserve the expected first passage times for individual paths to the absorbing state, but instead preserves the $\mathcal{A} \leftarrow q$ MFPTs for all transient nodes $q \in \mathcal{Q}$. That is, Eq. 1.35 preserves the *path ensemble average* time for the transition from a transient node to the *set* of absorbing nodes. To understand this result, note that the formula for the renormalized waiting time τ'_j (Eq. 1.35) is an average over all $\gamma \in \Gamma \leftarrow j$ transitions (*cf.* Eq. 1.37) and is associated with each of the renormalized probabilities for these transitions. We now proceed to provide a formal proof that Eq. 1.35 not only preserves the probability and mean time for the local $\gamma \in \Gamma \leftarrow j$ transitions, but also preserves the $\mathcal{A} \leftarrow q$ MFPTs for all transient nodes $q \in \mathcal{Q}$ that remain noneliminated.

The overall probability associated with a $a \leftarrow q$ first passage path that proceeds via at least one transition between nodes of the set Γ, on a network where the n-th node has been eliminated, can be factorized into probabilities for segments of the path divided as follows: the portion of the path from the starting transient node $q \in \mathcal{Q}$ to a node j of the set Γ (denoted $\xi^{(j,q)}$), the portion of the path from a node γ of the set Γ to the absorbing node $a \in \mathcal{A}$ (denoted $\xi^{(a,\gamma)}$), and transitions within nodes of the set Γ. The probability for any $\gamma \in \Gamma \leftarrow j$ transition on the transformed network is simply $T'_{\gamma j}$, and the path segments $\xi^{(j,q)}$ and $\xi^{(a,\gamma)}$ are associated with products of transition probabilities $\mathcal{W}_{\xi^{(j,q)}}$ and $\mathcal{W}_{\xi^{(a,\gamma)}}$, respectively, which do not involve renormalized transition probabilities. Using the reweighted transition probabilities and the convenient property of their derivatives (Eq. 1.36), we can hence write the contribution from the family of paths starting from node q and ending in node a, with a single transition $\gamma \leftarrow j$ between nodes of the set Γ, to the total $\mathcal{A} \leftarrow \mathcal{B}$ MFPT as follows:[92]

$$\begin{aligned}\left[\frac{d}{d\zeta}\sum_{\xi^{(j,q)}}\widetilde{\mathcal{W}}_{\xi^{(j,q)}}\sum_{\gamma\in\Gamma}\widetilde{T}'_{\gamma j}\sum_{\xi^{(a,\gamma)}}\widetilde{\mathcal{W}}_{\xi^{(a,\gamma)}}\right]_{\zeta=0} &= \sum_{\xi^{(j,q)}}\left[\frac{d\widetilde{\mathcal{W}}_{\xi^{(j,q)}}}{d\zeta}\right]_{\zeta=0}\sum_{\gamma\in\Gamma}\widetilde{T}'_{\gamma j}\sum_{\xi^{(a,\gamma)}}\widetilde{\mathcal{W}}_{\xi^{(a,\gamma)}} \\ &\quad + \sum_{\xi^{(j,q)}}\widetilde{\mathcal{W}}_{\xi^{(j,q)}}\sum_{\gamma\in\Gamma}\left[\frac{d\widetilde{T}'_{\gamma j}}{d\zeta}\right]_{\zeta=0}\sum_{\xi^{(a,\gamma)}}\widetilde{\mathcal{W}}_{\xi^{(a,\gamma)}} \\ &\quad + \sum_{\xi^{(j,q)}}\mathcal{W}_{\xi^{(j,q)}}\sum_{\gamma\in\Gamma}T'_{\gamma j}\sum_{\xi^{(a,\gamma)}}\left[\frac{d\widetilde{\mathcal{W}}_{\xi^{(a,\gamma)}}}{d\zeta}\right]_{\zeta=0}. \qquad (1.38)\end{aligned}$$

In this expression, there are sums over all path segments $\xi^{(j,q)}$ initialized from a particular transient node $q \in \mathcal{Q}$ and ending at a node $j \in \Gamma$, and over all path segments $\xi^{(a,\gamma)}$ beginning at a node $\gamma \in \Gamma$ and terminating at a particular absorbing node $a \in \mathcal{A}$. Only the second term in Eq. 1.38 is affected by the renormalization, and the derivative in this term is the same as the local term in Eq. 1.37. The contribution of the described family of paths (for a particular absorbing node a) to the overall $\mathcal{A} \leftarrow q$ MFPT is *not* preserved by the graph

transformation, owing to the fact that each $\gamma \in \Gamma \leftarrow j$ step is associated with the same averaged τ'_j, as stated previously. To obtain the result for all $\mathcal{A} \leftarrow q$ first passage paths, we sum over absorbing nodes $a \in \mathcal{A}$.[92,140] The contribution of this set of paths to the $\mathcal{A} \leftarrow q$ MFPT *is* conserved, $\sum_{a\in\mathcal{A}} \sum_{\xi^{(a,\gamma)}} \mathcal{W}_{\xi^{(a,\gamma)}} = 1 \; \forall \, \gamma \in \Gamma$. Hence, the renormalization equations (Eqs. 1.34 and 1.35), applied any number of times to the Markov chain, preserve the MFPTs $\mathcal{T}_{\mathcal{A}q}$ for transitions from all transient nodes $q \in \mathcal{Q}$ that remain noneliminated.

The vector of MFPTs for transitions from all transient nodes, of the set $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$, can be obtained by solving the following system of linear equations obtained from a first-step analysis:[4]

$$\mathcal{T}_{\mathcal{A}j} = \tau_j + \sum_{\gamma\notin\mathcal{A}} T_{\gamma j} \mathcal{T}_{\mathcal{A}\gamma}. \tag{1.39}$$

Recall that we wish to find the MFPT for the transition from a particular initial node b. When all nodes of the set $(\mathcal{A} \cup b)^{\mathsf{c}}$ have been eliminated from the Markov chain according to Eqs. 1.34 and 1.35, the only remaining edges in the network represent transitions from the b-th node to nodes of the absorbing macrostate $\mathcal{A}$, and the $b \leftarrow b$ self-loop. The first-step relation for the MFPTs (Eq. 1.39) therefore reduces to a direct solution for the $\mathcal{A} \leftarrow b$ MFPT,

$$\mathcal{T}_{\mathcal{A}b} = \frac{\tau'_b}{1 - T'_{bb}}. \tag{1.40}$$

To compute the overall $\mathcal{A} \leftarrow \mathcal{B}$ MFPT (Eq. 1.20), for an initial macrostate $\mathcal{B} \subseteq \mathcal{Q}$, the MFPTs for transitions from each node of the set $\mathcal{B}$ to the absorbing state $\mathcal{A}$ are required. When the initial set of nodes $\mathcal{B}$ is small, this can be achieved easily in practice as follows. Each individual MFPT, $\mathcal{T}_{\mathcal{A}b}$ for nodes $b \in \mathcal{B}$, is determined via Eq. 1.40 by eliminating all nodes of the set $\mathcal{B} \setminus b$, after first eliminating all nodes of the set $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$ and storing the resulting renormalized network. Then only the former computation needs to be repeated for each initial node $b \in \mathcal{B}$, in order to obtain $\mathcal{T}_{\mathcal{A}\mathcal{B}}$. When the absorbing state $\mathcal{A}$ is small, the MFPT for the reverse ($\mathcal{B} \leftarrow \mathcal{A}$) direction can be computed similarly with comparatively little additional computational effort, by eliminating all nodes of the set $\mathcal{A} \setminus a$ from the renormalized network resulting from elimination of all nodes of the set $\mathcal{I}$, for each node $a \in \mathcal{A}$ in turn.

1.4 Further state reduction algorithms

The renormalization of transition probabilities (Eq. 1.34) forms the basis for the family of state reduction methods to robustly compute the properties of a Markov chain. In this section, we extend the theory of renormalization to eliminate blocks of nodes simultaneously,

and describe numerically stable state reduction procedures to compute many of the dynamical quantities introduced in Sec. 1.2, including uncoupling-coupling methods to determine the stationary distribution (Secs. 1.4.1 and 1.4.2), and the REFUND algorithm to compute the group inverse (Sec. 1.4.4).

1.4.1 Exact uncoupling-coupling via stochastic complements

In Sec. 1.3, we argued that renormalization of the transition probabilities upon eliminating a single node n in the network (Eq. 1.34) preserves the probabilities of individual paths, in their renormalized representation, on the resulting reduced Markov chain. This theory can be extended to allow for elimination of a block of nodes $\mathcal{N}$ in a single step, using a matrix inversion operation. The state space of the Markov chain is partitioned as $\mathcal{S} \equiv \mathcal{N} \cup \mathcal{Z}$, so that we write the transition probability matrix in block form as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathcal{N}\mathcal{N}} & \mathbf{T}_{\mathcal{N}\mathcal{Z}} \\ \mathbf{T}_{\mathcal{Z}\mathcal{N}} & \mathbf{T}_{\mathcal{Z}\mathcal{Z}} \end{bmatrix}, \tag{1.41}$$

where $\mathbf{T}_{\mathcal{N}\mathcal{Z}}$ contains the $n \in \mathcal{N} \leftarrow z \in \mathcal{Z}$ transition probabilities, and the other blocks are defined similarly. Hence, $\mathbf{T}_{\mathcal{N}\mathcal{N}}$ is the substochastic matrix for transitions within the subset of nodes to be eliminated, $\mathcal{N}$, and $\mathbf{T}_{\mathcal{Z}\mathcal{Z}}$ likewise corresponds to the subnetwork comprising the nodes of the macrostate to be retained, $\mathcal{Z} \equiv \mathcal{N}^{\mathsf{c}}$. The renormalized Markov chain consisting only of the nodes within the set $\mathcal{Z}$ is associated with the transition probability matrix[203]

$$\mathbf{T}'_{\mathcal{Z}\mathcal{Z}} \leftarrow \mathbf{T}_{\mathcal{Z}\mathcal{Z}} + \mathbf{T}_{\mathcal{Z}\mathcal{N}}(\mathbf{I} - \mathbf{T}_{\mathcal{N}\mathcal{N}})^{-1}\mathbf{T}_{\mathcal{N}\mathcal{Z}}. \tag{1.42}$$

Eq. 1.42 is referred to as a *stochastic complement* by Meyer.[74] The renormalized transition probabilities account for the average behaviour of paths that visit $\mathcal{N}$, including transitions within $\mathcal{N}$, analogous to the case of eliminating a single node (Eq. 1.34).[206] Therefore, if there was no direct $i \leftarrow j$ transition in the original network, but there were $i \leftarrow \mathcal{N} \leftarrow j$ paths, then a direct $i \leftarrow j$ transition is present in the renormalized network for $i, j \in \mathcal{Z}$. Similarly, it can be shown that the $|\mathcal{Z}|$-dimensional vector of renormalized mean waiting or lag times for the remaining network is given by[94]

$$\boldsymbol{\tau}'_{\mathcal{Z}} \leftarrow \boldsymbol{\tau}_{\mathcal{Z}} + \boldsymbol{\tau}_{\mathcal{N}}(\mathbf{I} - \mathbf{T}_{\mathcal{N}\mathcal{N}})^{-1}\mathbf{T}_{\mathcal{N}\mathcal{Z}}. \tag{1.43}$$

Eqs. 1.42 and 1.43 constitute a block formulation of the GT algorithm (Sec. 1.3) that remains numerically stable when each of the blocks of nodes to be eliminated $\mathcal{N}$ constitute a metastable state, so that the $\mathbf{T}_{\mathcal{N}\mathcal{N}}$ matrices do not feature a spectral gap, and hence

the Markovian kernel ($\mathbf{I} - \mathbf{T}_{\mathcal{N}\mathcal{N}}$) can be safely inverted by dense linear algebra methods. Simultaneous elimination of blocks of nodes in a Markov chain can also lead to improved time complexity of the GT algorithm.[192]

It can be shown that if a Markov chain is irreducible, then any stochastic complement (Eq. 1.42) also has a well-defined stationary distribution,[1] and hence, so do any stochastic complements of that stochastic complement, and so on. Moreover, if a transition probability matrix is partitioned into N communities, $\mathcal{C} = \{1, 2, \ldots, N\}$, then the stochastic complements (i.e. renormalized Markov chains) corresponding to each of the communities have independent stationary distributions, from which the stationary distribution of the original Markov chain can be inferred.[207] Specifically, the stationary distribution of the original Markov chain partitioned according to

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \cdots & \mathbf{T}_{1N} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \cdots & \mathbf{T}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_{N1} & \mathbf{T}_{N2} & \cdots & \mathbf{T}_{NN} \end{bmatrix}, \tag{1.44}$$

is given by

$$\boldsymbol{\pi} = (\zeta_1 \boldsymbol{\pi}'_1, \ldots, \zeta_{N-1} \boldsymbol{\pi}'_{N-1}, \zeta_N \boldsymbol{\pi}'_N)^\top, \tag{1.45}$$

where $\zeta_{\mathcal{Y}} = \sum_{y \in \mathcal{Y}} \pi_y$ is the coupling factor corresponding to the stochastic complement (Eq. 1.42) comprising the nodes of the set $\mathcal{Y}$, which has a stationary distribution vector $\boldsymbol{\pi}'_{\mathcal{Y}}$, and $\sum_{\mathcal{Y}} \zeta_{\mathcal{Y}} = 1$. The N-dimensional vector of coupling factors,

$$\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_{N-1}, \zeta_N)^\top, \tag{1.46}$$

is the stationary distribution for the stochastic matrix $\mathbf{C}$ with elements $C_{\mathcal{X}\mathcal{Y}} = \mathbf{1}_{\mathcal{X}}^\top \mathbf{T}_{\mathcal{X}\mathcal{Y}} \boldsymbol{\pi}'_{\mathcal{Y}}$, for all communities $\mathcal{X}, \mathcal{Y} \in \mathcal{C}$.[207] $\mathbf{C}$ is referred to as the aggregation matrix, since its elements are simply the intercommunity transition probabilities when a local equilibrium[6] is established within each of the separate macrostates. If the community structure $\mathcal{C}$ appropriately characterizes the metastable communities of nodes, then $\mathbf{C}$ is well-conditioned, and hence $\boldsymbol{\zeta}$ can be determined accurately with relatively little computational effort, since the number of communities N is typically small. The coupling factors take a particularly simple form in the case of a two-level partition into sets $\mathcal{N}$ and $\mathcal{Z} \equiv \mathcal{N}^{\mathsf{c}}$ (Eq. 1.41), namely[74]

$$\zeta_{\mathcal{Z}} = \frac{1 - \mathbf{1}_{\mathcal{N}}^\top \mathbf{T}_{\mathcal{N}\mathcal{N}} \boldsymbol{\pi}'_{\mathcal{N}}}{2 - \mathbf{1}_{\mathcal{Z}\mathcal{Z}}^\top \mathbf{T}_{\mathcal{Z}\mathcal{Z}} \boldsymbol{\pi}'_{\mathcal{Z}} - \mathbf{1}_{\mathcal{N}}^\top \mathbf{T}_{\mathcal{N}\mathcal{N}} \boldsymbol{\pi}'_{\mathcal{N}}}, \tag{1.47}$$

and $\zeta_{\mathcal{N}} = 1 - \zeta_{\mathcal{Z}}$.

These observations suggest the following uncoupling-coupling algorithm to compute the stationary distribution vector. In the uncoupling phase, an irreducible Markov chain and its derived stochastic complements are repeatedly decomposed into two or more renormalized Markov chains of reduced dimensionality, based on a determined partitioning $\mathcal{C}$ at each iteration. For the set of stochastic complements resulting from the final iteration of the uncoupling phase, linear algebra methods (Sec. 1.2) or the GTH algorithm[125,208] (Sec. 1.4.3) can be used to compute the independent stationary distributions for each of the renormalized Markov chains with state space $\mathcal{Y}$, $\boldsymbol{\pi}'_{\mathcal{Y}} \, \forall \, \mathcal{Y} \in \mathcal{C}$. Then the aggregation matrix $\mathbf{C}$ is constructed from this information, and the coupling factors $\zeta_{\mathcal{Y}}$ associated with each of the stochastic complements are obtained as the stationary distribution of $\mathbf{C}$. The vectors $\{\boldsymbol{\pi}'_{\mathcal{Y}}\}$ and corresponding coupling factors $\{\zeta_{\mathcal{Y}}\}$ yield the stationary distribution of the parent Markov chain, comprising the nodes in *all* communities $\mathcal{Y} \in \mathcal{C}$ (Eq. 1.45). That is, the final iteration of the uncoupling stage has been undone. Repeated coupling of the stationary distributions for the stochastic complements at each level of the hierarchical partitioning that was performed in the uncoupling phase eventually recovers the stationary distribution of the original Markov chain. This uncoupling-coupling procedure based on stochastic complementation (Eq. 1.42) is illustrated in Fig. 1.2.

In practice, this exact uncoupling-coupling algorithm has been found to be highly effective for determining the stationary distribution of a nearly reducible Markov chain.[209] To understand this observation, consider an iteration of the uncoupling phase, where the Markov chain $\mathbf{T}$ has N metastable macrostates, and is partitioned into the set of N communities $\mathcal{C}$, which accurately characterizes the metastable sets of nodes. For each community $\mathcal{Z} \in \mathcal{C}$, the renormalized transition matrix $\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}$ (Eq. 1.42) is then close to the (nearly stochastic) block $\mathbf{T}_{\mathcal{Z}\mathcal{Z}}$ (Eq. 1.41) of the parent transition matrix.[74] Since the dynamics within the state space $\mathcal{Z}$ are fast compared to escape from this subnetwork, neither the absorbing Markov chain $\mathbf{T}_{\mathcal{Z}\mathcal{Z}}$ nor its derived stochastic complement $\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}$ have any subdominant eigenvalues close to unity, and the renormalized Markov chain $\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}$ is therefore well-conditioned. Hence, inversion of the Markovian kernel to determine the corresponding stochastic complement (Eq. 1.42), and computation of the stationary distribution for this stochastic complement, is numerically stable.[210]

More formally, since each of the metastable macrostates gives rise to a slowly decaying dynamical eigenmode, the nearly reducible Markov chain $\mathbf{T}$ necessarily has at least $N - 1$ eigenvalues λ_k that are close to unity, in addition to the eigenvalue $\lambda_1 = 1$ that is associated with the stationary distribution. Furthermore, each of the stochastic complements $\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}$, corresponding to communities $\mathcal{Z} \in \mathcal{C}$, necessarily has a unit eigenvalue.[207] Each of the unique unit eigenvalues for the N stochastic complements are associated with one of the N

dominant eigenvalues for the original Markov chain, and this mapping becomes exact in the limit where the original Markov chain is completely reducible, i.e. where the separate blocks $\mathcal{Z} \in \mathcal{C}$ are themselves irreducible.[73] Hence, by the continuity of the eigenspectrum, if there is a spectral gap after the N dominant eigenvalues of the original Markov chain, then each of the N stochastic complements must have a spectral gap after the unit eigenvalue.[74] In fact, it can be shown that there exists an upper bound on the second dominant eigenvalue for a stochastic complement of a reversible Markov chain.[211] Thus if the community structures used to partition the stochastic complements during the uncoupling phase appropriately characterize all metastable sets of nodes in the relevant Markov chains, so that the aggregation matrices are also well-conditioned, then the entire uncoupling-coupling procedure is numerically stable. Moreover, the uncoupling-coupling procedure is readily parallelizable, owing to the independence of the stationary distributions for the stochastic complements derived from a parent Markov chain.[207]

1.4.2 Iterative aggregation-disaggregation

Stochastic complementation is close in spirit to iterative aggregation-disaggregation (IAD) methods[212–215] to compute the stationary distribution $\boldsymbol{\pi}$. Both methods use an N-way partitioning of a Markov chain based on the community structure $\mathcal{C}$ (Eq. 1.44). Whereas the uncoupling-coupling in stochastic complementation is exact, IAD uses the substochastic blocks of a partitioned Markov chain directly, thereby avoiding the matrix inversion operations required to compute the stochastic complements (Eq. 1.42).

To infer an approximation to the stationary distribution of the parent Markov chain in IAD, the normalized right eigenvector $\boldsymbol{\psi}_{\mathcal{Y}}^{(1)}$ associated with the dominant eigenvalue (which is less than unity) is computed for each of the substochastic matrices $\mathbf{T}_{\mathcal{Y}\mathcal{Y}}$, corresponding to communities $\mathcal{Y} \in \mathcal{C}$. The vector of coupling factors $\boldsymbol{\zeta}^*$ (*cf.* Eq. 1.46) that are associated with these eigenvectors is determined as the stationary distribution for a $N \times N$-dimensional stochastic coupling matrix $\mathbf{C}^*$ with elements $C^*_{\mathcal{X}\mathcal{Y}} = \mathbf{1}_{\mathcal{X}}^\top \mathbf{T}_{\mathcal{X}\mathcal{Y}} \boldsymbol{\psi}_{\mathcal{Y}}^{(1)} \; \forall \, \mathcal{X}, \mathcal{Y} \in \mathcal{C}$. Note that thus far, the IAD procedure is analogous to the exact uncoupling-coupling method described in Sec. 1.4.1, except that the quantities are inexact since the substochastic blocks are used in place of the stochastic complements. The eigenvectors and associated coupling factors together yield an initial approximation to the stationary distribution of the original Markov chain (*cf.* Eq. 1.45), $\boldsymbol{\pi} \approx (\zeta_1^* \boldsymbol{\psi}_1^1, \ldots, \zeta_1^* \boldsymbol{\psi}_1^N)^\top$, which is iteratively refined as follows.

Let $\boldsymbol{\pi}^* = (\boldsymbol{\pi}_1^*, \ldots, \boldsymbol{\pi}_N^*)^\top$ denote the current estimate for the stationary distribution. First, the vectors $\boldsymbol{\pi}_{\mathcal{Y}} \; \forall \, \mathcal{Y} \in \mathcal{C}$ are normalized to yield the vectors $\{\bar{\boldsymbol{\pi}}_{\mathcal{Y}}\}$, and updated coupling factors $\{\zeta_{\mathcal{Y}}^*\}$ are computed as the stationary distribution of a new stochastic aggregation

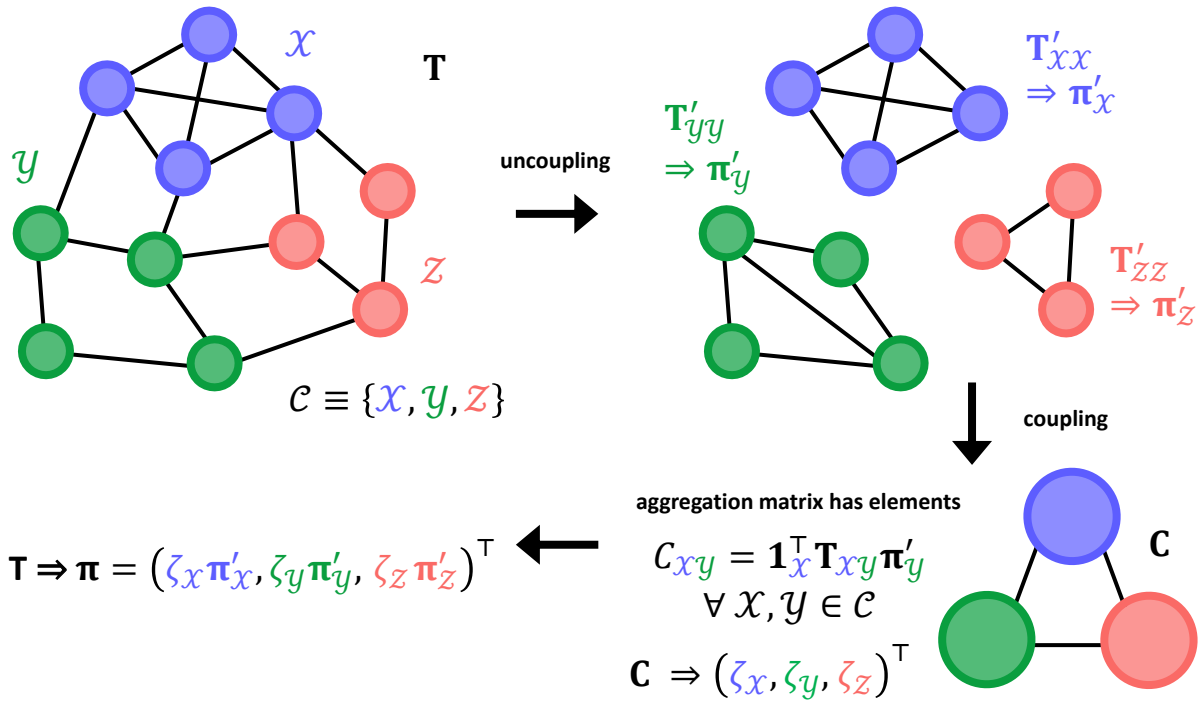


Figure 1.2: Schematic illustration of the exact uncoupling-coupling procedure (Sec. 1.4.1) to compute the stationary distribution $\boldsymbol{\pi}$ of an irreducible Markov chain parameterized by the transition probability matrix $\mathbf{T}$. The state space of the model network shown is partitioned into three communities, $\mathcal{S} \equiv \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$. The edges of the network are bidirectional. In the uncoupling step, the stochastic complement $\mathbf{T}'_{\mathcal{Y}\mathcal{Y}}$ (Eq. 1.42) is computed for each community $\mathcal{Y}$ of the set $\mathcal{C}$. Observe that the stochastic complements $\mathbf{T}'_{\mathcal{Y}\mathcal{Y}}$ contain additional edges compared to the corresponding substochastic blocks $\mathbf{T}_{\mathcal{Y}\mathcal{Y}}$ of the original Markov chain. For instance, the stochastic complement for the community $\mathcal{Y}$ has an additional edge between two nodes for which an indirect transition via the set $\mathcal{X} \cup \mathcal{Z}$ exists. After uncoupling, the stationary distribution is computed for each of the independent reduced Markov chains. In the figure, the notation $\mathbf{T} \Rightarrow \boldsymbol{\pi}$ indicates that the stochastic matrix $\mathbf{T}$ has stationary distribution $\boldsymbol{\pi}$. The independent stationary distributions are used to construct a stochastic aggregation (coupling) matrix $\mathbf{C}$ in which each community is represented by a single node. The stationary distribution $\boldsymbol{\pi}$ of the original Markov chain is constructed from the stationary distribution $\boldsymbol{\zeta}$ associated with $\mathbf{C}$, and from the independent stationary distributions of the separate stochastic complements, $\boldsymbol{\pi}'_{\mathcal{Y}}\ \forall\, \mathcal{Y} \in \mathcal{C}$ (coupling step). The algorithm illustrated above has a single uncoupling stage, but the independence of the stochastic complements can be exploited to recursively reduce the derived Markov chains in multiple uncoupling steps, and this algorithm is readily parallelizable. In iterative aggregation-disaggregation (IAD) procedures (Sec. 1.4.2), the substochastic blocks $\mathbf{T}_{\mathcal{Y}\mathcal{Y}}$ are used in place of the stochastic complements $\mathbf{T}'_{\mathcal{Y}\mathcal{Y}}$. Hence, the initial stationary distribution determined by the coupling step is inexact. New coupling factors $\zeta^*_{\mathcal{Y}}$ and corresponding approximate local stationary distributions $\boldsymbol{\pi}^*_{\mathcal{Y}}$ are iteratively updated by repeatedly forming a stochastic aggregation matrix and then solving a system of linear equations (Eq. 1.48).

matrix $\mathbf{C}^*$ with elements $C^*_{\mathcal{XY}} = \mathbf{1}_{\mathcal{X}}^\top \mathbf{T}_{\mathcal{XY}} \bar{\boldsymbol{\pi}}^*_{\mathcal{Y}} \; \forall \, \mathcal{X}, \mathcal{Y} \in \mathcal{C}$ (aggregation step). The new coupling factors yield the vector $\mathbf{z} = (\zeta_1^* \bar{\boldsymbol{\pi}}_1^*, \ldots, \zeta_N^* \bar{\boldsymbol{\pi}}_N^*)^\top$. Second, an updated estimate for $\boldsymbol{\pi}$ is obtained by solving the following N systems of linear equations (disaggregation step),[216]

$$\boldsymbol{\pi}^*_{\mathcal{X}} = \mathbf{T}_{\mathcal{XX}} \boldsymbol{\pi}^*_{\mathcal{X}} + \sum_{\mathcal{Y}<\mathcal{X}} \mathbf{T}_{\mathcal{XY}} \boldsymbol{\pi}^*_{\mathcal{Y}} + \sum_{\mathcal{Y}>\mathcal{X}} \mathbf{T}_{\mathcal{XY}} \mathbf{z}_{\mathcal{Y}}, \; \forall \, \mathcal{X} \in \mathcal{C}. \tag{1.48}$$

These two steps are repeated until the estimate for the stationary distribution of the parent Markov chain converges, $\boldsymbol{\pi}^* \to \boldsymbol{\pi}$. Further refinements to this procedure that improve numerical stability and convergence were reported in Ref. 217.

In practice, this procedure typically converges to the true stationary distribution rapidly.[210,218] IAD shares many of the same advantages as exact uncoupling-coupling, most notably the numerical stability conferred when the community structure reflects the nearly reducible structure of a Markov chain with metastable states, and the possibility of parallelization. The GTH algorithm (Sec. 1.4.3) can be used to solve the linear systems of equations that arise in both the aggregation and disaggregation steps of IAD, leading to improved numerical stability.[219]

1.4.3 Grassmann-Taksar-Heyman algorithm

The Grassmann-Taksar-Heyman (GTH) algorithm[125,208] is essentially a nodewise iterative formulation of exact uncoupling-coupling via stochastic complementation (Sec. 1.4.1). Consider the elimination of the n-th node in a Markov chain by renormalization (Eq. 1.34), as in the nodewise iterative formulation of the GT algorithm (Sec. 1.3). Since the equilibrium distribution is equal to the proportion of time spent at a node in the infinite time limit, the stationary distribution of the parent Markov chain, $\boldsymbol{\pi}$, must be proportional to that of the reduced model, $\boldsymbol{\pi}'$:

$$\pi_j = \alpha \pi'_j \; \forall \, j \in \mathcal{S} \setminus \{n\}, \tag{1.49a}$$

$$\pi_n = 1 - \alpha. \tag{1.49b}$$

From Eq. 1.49 and the global balance equation for the stationary distribution of the parent Markov chain at the n-th node, $\pi_n = \sum_\gamma \pi_\gamma T_{n\gamma}$, we have[123]

$$\alpha = \left(1 + \frac{\sum_{\gamma \neq n} \pi'_\gamma T_{n\gamma}}{1 - T_{nn}} \right)^{-1}, \tag{1.50}$$

where T_{ij} is the $i \leftarrow j$ transition probability for the original Markov chain.

Eqs. 1.49 and 1.50 suggest the following nodewise iterative procedure to compute the stationary distribution of an irreducible Markov chain. The GTH algorithm uses renormalization of transition probabilities (*cf.* Eq. 1.34) to eliminate all nodes n of the Markov chain for which $|\mathcal{S}| \geq n > 1$. The stationary probability for the reduced Markov chain comprising the single remaining node ($n = 1$) is, of course, known to be $\pi'_1 = 1$. The algorithm then employs a back substitution phase to "undo" the state reduction stage, thereby recovering the original Markov chain and associated stationary distribution.[40] The GTH algorithm, which is essentially equivalent to a Gaussian elimination,[40] is given as pseudocode in Algorithm 1. Like the GT algorithm, the GTH algorithm is numerically stable,[220] since the relation $1 - T_{nn} \equiv \sum_{\gamma \neq n} T_{\gamma n}$ can be exploited to avoid problematic subtraction operations when $T_{nn} \to 1$.[202]

input : transition probability matrix $\mathbf{T}$
set of nodes $n \in \mathcal{S} \setminus \{1\}$ to be eliminated from the state space $\mathcal{S}$
output: stationary probability distribution vector $\boldsymbol{\pi}$

`/* elimination phase */`
for $n = |\mathcal{S}|, |\mathcal{S}| - 1, \ldots, 2$ **do**
  $S_n = \sum_{\gamma < n} T_{\gamma n} \quad (\equiv 1 - T_{nn})$; `// confers numerical stability`
  for $j < n$ **do**
    $T_{nj} \leftarrow T_{nj}/S_n$;
    for $i < n$ **do**
      $T_{ij} \leftarrow T_{ij} + T_{in}T_{nj}$;
`/* back substitution phase */`
$\pi_1 \leftarrow 1$;
$\mu \leftarrow 1$;
for $n = 2, \ldots, |\mathcal{S}| - 1, |\mathcal{S}|$ **do**
  $\pi_n = T_{n1} + \sum_{\gamma=2}^{n-1} \pi_\gamma T_{n\gamma}$;
  $\mu \leftarrow \mu + \pi_n$;
`/* normalization */`
for $n = 1, \ldots, |\mathcal{S}| - 1, |\mathcal{S}|$ **do**
  $\pi_n \leftarrow \pi_n/\mu$;
return $\boldsymbol{\pi}$;

Algorithm 1: Grassmann-Taksar-Heyman (GTH) algorithm[125,208] to compute the stationary distribution $\boldsymbol{\pi}$ of a Markov chain.

1.4.4 FUND and REFUND algorithms

Recall that the key macroscopic dynamical properties of an irreducible Markov chain, including moments of the first passage time distributions for all pairwise transitions between nodes, and moments of the mixing time distributions for relaxation processes from alternative starting nodes, can be computed from any generalized matrix inverse (Sec. 1.2.2). The FUND[79,221,222]

and REFUND[123] algorithms provide numerically stable procedures to compute the group inverse $\mathbf{A}^{\#}$ (Eqs. 1.8-1.10) associated with an irreducible Markov chain, and can be readily adapted to obtain related generalized inverses such as the fundamental matrix $\mathbf{Z}$.

The FUND algorithm is based on the fact that a generalized inverse $\mathbf{X}$ satisfying[129]

$$\mathbf{X}(\mathbf{I}-\mathbf{T}) = \mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}, \tag{1.51}$$

can be obtained from the result of the first phase of the GTH algorithm (Algorithm 1), and that the group inverse can be written directly in terms of any generalized inverse $\mathbf{X}$ satisfying Eq. 1.51:[123]

$$\mathbf{A}^{\#} = \mathbf{X}(\mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}). \tag{1.52}$$

The first step of the FUND algorithm is to obtain a LU decomposition of the Markovian kernel; $\mathbf{I}-\mathbf{T} = \mathbf{UL}$, where $\mathbf{U}$ and $\mathbf{L}$ are upper- and lower-triangular matrices, respectively. To see that this factorization is achieved naturally and robustly by the elimination phase of the GTH algorithm (Sec. 1.4.3), consider that the elements of the stochastic matrix $\mathbf{T}$ are overwritten during the procedure, yielding the matrix $\mathbf{T}^{*}$. Let $\mathbf{F}$ denote the strictly lower-triangular matrix and $\mathbf{G}$ the upper-triangular matrix containing the corresponding elements of $\mathbf{T}^{*}$, so that $\mathbf{T}^{*} = \mathbf{F} + \mathbf{G} + (\mathbf{I}-\mathbf{S})$, where $\mathbf{S}$ is the diagonal matrix with nonzero elements $S_{nn} = S_n$ (*cf.* Algorithm 1). It can be shown that:[40,221]

$$\mathbf{I}-\mathbf{T} = (\mathbf{G}-\mathbf{S})(\mathbf{F}-\mathbf{I}) = \mathbf{UL}. \tag{1.53}$$

The $\mathbf{U}$ matrix has all column sums equal to zero. The diagonal elements of $\mathbf{U}$ can be computed by enforcing this constraint, which thereby provides an additional opportunity to enforce numerical stability.[79] The diagonal elements of $\mathbf{L}$ are all equal to -1, and hence $\mathbf{L}$ is non-singular.

The LU decomposition of Eq. 1.53 can be used to solve Eq. 1.51 in two stages. Firstly, the (unique) solution $\mathbf{Y}$ to the problem $\mathbf{YL} = \mathbf{I} - \boldsymbol{\pi}\mathbf{1}_{\mathcal{S}}^{\top}$ is determined by backward substitution. The second step is to solve $\mathbf{XU} = \mathbf{Y}$.[129] Since the first column of $\mathbf{U}$ is identically zero, the first column of $\mathbf{Y}$ is also necessarily the null vector, and therefore the generalized inverse $\mathbf{X}$ is not a unique solution to this equation.[221] The simplest choice is to set the first column of $\mathbf{X}$ to be the null vector, and the remaining component vectors can then be solved for by forward substitution.[222] The group inverse $\mathbf{A}^{\#}$ then follows straightforwardly from $\mathbf{X}$ (Eq. 1.52), where $\boldsymbol{\pi}$ is computed by following through the later stages of the GTH algorithm.

The FUND algorithm outlined above is usually stable since there is not significant numerical error associated with the forward and backward substitution procedures to determine

$\mathbf{Y}$ and $\mathbf{X}$. However, a difficulty may sometimes arise where $\mathbf{L}$ is ill-conditioned even though the Markovian kernel $\mathbf{I} - \mathbf{T}$ is not. A small refinement of the FUND algorithm proposed in Ref. 79 addresses this issue. The time complexity of either formulation of the FUND algorithm is $\mathcal{O}(|\mathcal{S}|^3)$.

An explicit formula relating the group inverses associated with a renormalized Markov chain where a single node has been eliminated and the corresponding parent Markov chain was derived in Ref. 123. This relation allows for a state reduction procedure to compute the group inverse that has a similar structure to the GTH algorithm, namely the REFUND algorithm. That is, the REFUND algorithm consists of an elimination phase to iteratively reduce the Markov chain by renormalization, trivial assignment of the group inverse for the Markov chain with only a single node remaining, and a backwards pass phase to compute the group inverses for the sequence of Markov chains resulting from the restoration of nodes in turn, given the group inverse for the reduced Markov chain where the node of the current iteration is eliminated. The recursive phase of the REFUND procedure also computes the stationary distribution of the reduced Markov chain at each iteration, which is required to compute the group inverse of the parent Markov chain. The n-dimensional stationary distribution vector for the reduced Markov chain with the n-th node restored, $\boldsymbol{\pi}_n$, is normalized at each iteration of the backwards pass phase. This differs from the GTH algorithm (Algorithm 1), where the stationary distribution is only normalized at the final step. REFUND has comparable stability to the refined FUND algorithm, and likewise has asymptotic time complexity $\mathcal{O}(|\mathcal{S}|^3)$. The REFUND algorithm is given as pseudocode in Algorithm 2.

1.4.5 Other state reduction methods

Many procedures based on the concept of renormalization have been proposed as numerically stable approaches to solve various other linear algebra problems for Markov chains and related systems.[126] In Ref. 130, an extension of the GTH algorithm was derived to compute the variance and higher moments of the FPT distributions for the transitions from each of the transient nodes of a reducible Markov chain to the absorbing state. In Chapter 3, we propose a state reduction procedure to determine the expected numbers of times that transient nodes are visited on first passage paths prior to absorption.[191] Together with state reduction methods to compute the stationary probability distribution (Secs. 1.4.1-1.4.3) and committor probabilities (Sec. 1.5.1), this method allows for robust computation of all microscopic properties characterizing the transition path ensemble (Sec. 1.2.5). Notably, it is possible to obtain the probabilities that nodes are visited along reactive paths, for both

nonequilibrium and equilibrium cases, which clearly identifies the kinetically relevant states with respect to the productive transition.

In Ref. 223, a state reduction framework was formulated for application to Markov decision processes (MDPs). A MDP augments the state space of a Markov chain with alternative sets of probabilities for transitions from the nodes, each associated with a different *action* that is available to the decision-making agent. A set of actions corresponding to each of the nodes of the MDP is a *policy*, which defines how the agent operates. The $i \leftarrow j$ transition under action a_j is associated with a reward $R_{ij}(a_j)$. The reward at the t-th step is usually weighted by γ^t, where γ is a discount factor to ensure that the reward for a trajectory converges, $0 < \gamma < 1$. The usual optimization problem is to determine the *optimal policy*, namely that which maximizes the expected reward when starting from a specified initial node. The algorithm of Ref. 223 is a robust method to find the optimal policy by policy iteration, using repeated application of the following two-stage procedure. Firstly, a state reduction algorithm is used to determine the vector of expected rewards starting from each node and when the agent obeys the current policy. Secondly, these expected rewards are used in an improvement step to obtain an update to a suboptimal policy. This numerically stable method is valuable in many practical applications, where nodes associated with large rewards may be rarely visited under an initial policy.

1.5 Algorithms for simulating pathways

The state reduction framework presented in Secs. 1.3 and 1.4 provides numerically stable procedures to compute almost all dynamical properties of a Markov chain. However, there are two notable quantities introduced in Sec. 1.2 that have not yet been obtained by state reduction methods. Firstly, there is no state reduction algorithm to exactly compute the time-dependent occupation probability distribution vector $\mathbf{p}(t)$, given a transition rate matrix $\mathbf{K}$. Instead, $\mathbf{p}(t)$ must be computed by matrix exponentiation (Eq. 1.3) or eigendecomposition (Eq. 1.21). Secondly, we desire a state reduction method to compute the committor probabilities for nodes (Eq. 1.29), which constitute the central object in transition path theory (Sec. 1.2.5).

In the present section, we describe the kinetic path sampling (kPS) method to efficiently simulate trajectories, and hence $\mathbf{p}(t)$, for a nearly reducible Markov chain. The kPS algorithm applies a state reduction procedure to the currently occupied metastable macrostate of the Markovian network, followed by a backwards pass phase to sample the numbers of internode transitions, and therefore the time, associated with a trajectory escaping from the subnetwork. In Chapter 4, we propose a workflow for kPS simulations based on obtaining an accurate partition of a Markov chain into metastable macrostates, and apply our methodology

to a peptide folding transition. The formulation of the graph transformation algorithm (Sec. 1.3) employed in kPS leads to an exact state reduction procedure to compute the committor probabilities for all nodes of a Markov chain (Sec. 1.5.1).

In principle, numerical estimates for the $\mathcal{A} \leftarrow \mathcal{B}$ FPT distribution (Eq. 1.22), and for the occupation probability distribution $\mathbf{p}(t)$, can be obtained by explicit simulation of $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths using any algorithm that samples the solution to the linear master equation (Eq. 1.1) exactly. The standard procedure to simulate trajectories on a Markovian network in continuous-time is rejection-free kinetic Monte Carlo (kMC).[82,83] In the the kMC algorithm, a trajectory that currently occupies the j-th node is advanced to the next node using two random numbers $r_1, r_2 \in (0, 1]$ drawn from a uniform distribution.[224] The first random number is used to sample an $i \leftarrow j$ transition in proportion to the branching probabilities P_{ij}, and the second is used to increment the simulation clock by $\Delta t = \tau_j \ln r_2$.[110] The probabilities of the sampled paths then agree exactly with the linear master equation (Eq. 1.1). By using the branching probability matrix, this procedure avoids self-loops, and so prevents the system from becoming trapped in any one node associated with small outgoing transition rates. However, for nearly reducible Markov chains, the trajectories exhibit a strong tendency to 'flicker' within metastable communities of nodes,[85–88] which can cause kMC simulation to be unfeasibly inefficient.[84,101]

For Markov chains featuring a rare event, large gains in efficiency can be achieved by defining the currently occupied metastable community of nodes (or basin) $\mathbb{B}$, sampling a node at the boundary $\partial\mathbb{A} \subseteq \mathbb{A}$ of the absorbing macrostate $\mathbb{A} \equiv \mathbb{B}^{\mathsf{c}}$, and sampling an escape time for the trajectory segment escaping to the absorbing boundary. The kinetic path sampling[84,85] (kPS) (Sec. 1.5.1) and Monte Carlo with absorbing Markov chains[102–105] (MCAMC) (Sec. 1.5.2) algorithms provide exact methods to do this. The kPS and MCAMC algorithms forfeit resolution of the trajectory within the metastable macrostates, which in any case ought to be unproductive and hence of little interest, and gain the desirable property that their efficiency is essentially independent of the metastability of the Markov chain.[84]

The kPS and MCAMC methods require a partitioning of the network into metastable communities. This clustering can be specified *a priori*, for example with a suitable community detection algorithm,[225] or constructed on-the-fly, for instance using a breadth-first search procedure with criteria for including nodes in the basin.[226] The algorithms can be used in conjunction with standard rejection-free kMC, automatically switching to the advanced method when flickering of a trajectory within a subset of nodes is detected.[84]

1.5.1 Kinetic path sampling

Renormalization phase

To simulate a trajectory segment in the kinetic path sampling[84,85] (kPS) algorithm, the state space $\mathcal{S} \equiv \mathbb{B} \cup \mathbb{A}$ is divided into the currently occupied community $\mathbb{B}$ and an absorbing macrostate $\mathbb{A}$. The nodes of the basin $\mathbb{B} \equiv \mathbb{E} \cup \mathbb{T}$ are further divided into the set of eliminated nodes $\mathbb{E} \subseteq \mathbb{B}$ and the set of retained nodes $\mathbb{T} \subset \mathbb{B}$. The set $\mathbb{T}$ may be empty, $\mathbb{T} = \emptyset$. We also define the absorbing boundary $\partial\mathbb{A} \subseteq \mathbb{A}$ as the subset of nodes of the absorbing macrostate that are directly connected to one or more nodes of the basin $\mathbb{B}$. The total number of nodes in the set $\mathbb{E} \cup \mathbb{T} \cup \partial\mathbb{A}$ is denoted N_c. The definitions of these states are illustrated in Fig. 1.3.

To simulate a trajectory escaping from the active community $\mathbb{B}$ to the absorbing boundary $\partial\mathbb{A}$, kPS uses a stochastic matrix corresponding to the relevant subnetwork $\mathbb{E} \cup \mathbb{T} \cup \partial\mathbb{A}$. In the first phase of the kPS algorithm, nodes $1, \ldots, |\mathbb{E}| \in \mathbb{E}$ are eliminated in turn by renormalization (Sec. 1.3), while retained nodes $|\mathbb{E}|+1, \ldots, |\mathbb{B}| \in \mathbb{T}$ and absorbing boundary nodes $|\mathbb{B}|+1, \ldots, N_\mathrm{c} \in \partial\mathbb{A}$ remain noneliminated. The input to the sampling stage of the kPS algorithm is the set of $|\mathbb{E}|+1$ transition probability matrices $\{\mathbf{T}^{(n)}\}$, $0 \leq n \leq |\mathbb{E}|$, with $\mathbf{T}^{(0)}$ corresponding to the transition probability matrix for the subnetwork $\mathbb{B} \cup \partial\mathbb{A}$ of the original (untransformed) network. $\mathbf{T}^{(0)}$ has dimensions $N_\mathrm{c} \times |\mathbb{B}|$, since the absorbing boundary nodes have no outgoing transitions. Successive stochastic matrices (with $n > 0$) are computed by the iterative elimination of the $|\mathbb{E}|$ nodes $n \in \mathbb{E}$ from this subnetwork using the GT algorithm (Sec. 1.3). In the renormalization, the transition probabilities for *all* pairs of nodes are updated according to

$$T_{ij}^{(n)} = T_{ij}^{(n-1)} + \frac{T_{nj}^{(n-1)}(T_{in}^{(n-1)} - \delta_{in})}{1 - T_{nn}^{(n-1)}}, \tag{1.54}$$

cf. Eq. 1.34. Transitions from eliminated to noneliminated nodes are preserved by Eq. 1.54, but transitions from noneliminated to eliminated nodes have zero probability. As in Eq. 1.34, only the probabilities for transitions between pairs of nodes that are both directly connected to the n-th node are affected by the renormalization. However, unlike renormalization using Eq. 1.34, connections involving eliminated nodes must also be considered. The mean waiting or lag times for transitions from nodes are *not* renormalized (*cf.* Eq. 1.35) in the kPS algorithm, since the state reduction procedure is reversed before sampling the time associated with a trajectory escaping from the basin to the absorbing boundary.

The formulation of GT expressed in Eq. 1.54 is analogous to a LU decomposition[210,227] of a stochastic matrix.[84] Exploiting this analogy reduces the memory requirements of the kPS algorithm, since the intermediate transition matrices arising from the iterative elimination

of nodes need not be stored. Instead, it is only necessary to store the original and final stochastic matrices, $\mathbf{T}^{(0)}$ and $\mathbf{T}^{(|\mathbb{E}|)}$, respectively, and the matrices $\mathbf{L}$ and $\mathbf{U}$ that contain the elements required to construct $\mathbf{T}^{(n-1)}$ from $\mathbf{T}^{(n)}$. Specifically, the $\mathbf{L}$ and $\mathbf{U}$ matrices have elements (*cf.* Eq. 1.54)

$$L_{nj} = \frac{T_{nj}^{(n-1)}}{1 - T_{nn}^{(n-1)}} \quad \text{and} \quad U_{in} = T_{in}^{(n-1)} - \delta_{in}. \tag{1.55}$$

Sampling dynamics on the renormalized network and iterative reverse randomization

Recall that renormalization preserves the path probabilities to individual absorbing nodes (Sec. 1.3.2).[92] This property of the state reduction framework is the basis for the kPS algorithm. To sample the absorbing boundary node $\alpha \in \partial\mathbb{A}$ at which the trajectory segment terminates, the following probability vectors are used to simulate successive $i \leftarrow j$ transitions on the renormalized subnetwork comprising the nodes of the set $\mathbb{B} \cup \partial\mathbb{A}$:

$$\mathbf{c}_j = \begin{cases} \mathbf{t}_{*,j}^{(|\mathbb{E}|)}, & \text{if} \quad j \leq |\mathbb{E}|, \\ \mathbf{t}_{*,j}^{(j)}, & \text{if} \quad j > |\mathbb{E}|, \end{cases} \tag{1.56}$$

starting from the currently occupied node $\epsilon \in \mathbb{B}$. Here, $\mathbf{t}_{*,j}^{(|\mathbb{E}|)}$ denotes the vector containing the elements of the j-th column of $\mathbf{T}^{(|\mathbb{E}|)}$, and $\mathbf{t}_{*,j}^{(j)}$ (for $j > |\mathbb{E}|$) denotes the j-th column of the transition matrix given by the elimination (Eq. 1.54) of node j from $\mathbf{T}^{(0)}$. In this sampling procedure, transitions between eliminated nodes are not permitted (*cf.* Eq. 1.54), but transitions from noneliminated transient nodes $j \in \mathbb{T}$ to eliminated nodes $i \in \mathbb{E}$ are possible. This setup greatly reduces the number of steps required to reach a node at the absorbing boundary. If $\mathbb{T} = \emptyset$, then the trajectory necessarily reaches the absorbing boundary in a single transition. If $\mathbb{E} = \emptyset$, then Eq. 1.56 reduces to the standard rejection-free kMC algorithm.[228]

Let $\mathbf{h}$ denote the $|\mathbb{E}|$-dimensional vector with elements h_j equal to the number of transitions from node $j \in \mathbb{E}$ to nodes $i > j$, i.e. transitions to nodes that are noneliminated in the stochastic matrix at the j-th iteration of the renormalization phase. Let $\boldsymbol{\eta}$ denote the $|\mathbb{B}|$-dimensional vector with elements η_j equal to the total number of transitions from the j-th node. We also define the set of $|\mathbb{E}| + 1$ hopping matrices $\{\mathbf{H}^{(n)}\}$, $0 \leq n \leq |\mathbb{E}|$, each of dimension $(N_\mathrm{c} - n) \times |\mathbb{E}|$. The element $H_{ij}^{(n)}$ corresponds to the number of $i \leftarrow j \in \mathbb{E}$ transitions on the renormalized stochastic matrix resulting from the elimination of n nodes (Eq. 1.54). Hence, $\eta_j = \sum_{\gamma \in \mathbb{B} \cup \partial\mathbb{A}} H_{\gamma j}^{(0)}$. Note that there are no transitions to eliminated nodes (with indices $j \leq n$) in the hopping matrix of the n-th iteration, hence the stated

dimensionality.

In the next stage of the kPS algorithm, an iterative reverse randomization procedure is used to sample the matrix $\mathbf{H}^{(0)}$ with elements $H_{ij}^{(0)}$ corresponding to the numbers of $i \leftarrow j$ transitions on the original (untransformed) subnetwork, with associated stochastic matrix $\mathbf{T}^{(0)}$. This procedure exploits the fact that $\mathbf{H}^{(n-1)}$ can be sampled directly from $\mathbf{H}^{(n)}$, $\mathbf{T}^{(n)}$, and $\mathbf{T}^{(n-1)}$ without requiring explicit simulation of the dynamics on the network corresponding to $\mathbf{T}^{(n-1)}$, in which the n-th node is not eliminated. In this sense, a single iteration of the reverse randomization procedure "undoes" a single iteration of the renormalization (Eq. 1.54). The relationship between renormalization and iterative reverse randomization is illustrated in Fig. 1.4. The sampling of successive hopping matrices is based on the set of $N_c \times n$-dimensional matrices $\{\mathbf{G}^{(n)}(\tau)\}$, for $0 < n \leq |\mathbb{E}|$, with elements $G_{ij}^{(n)}$ given by the ratio of $i \leftarrow j$ transition probabilities in Markov chains where the n-th node is the next to be eliminated in the renormalization phase, and where the n-th node has been eliminated:

$$G_{ij}^{(n)} = \frac{T_{ij}^{(n-1)}}{T_{ij}^{(n)}} \quad \forall\, i > n. \tag{1.57}$$

Hence, $G_{ij}^{(n)}$ is the fraction of $i \leftarrow j$ transitions in $\mathbf{T}^{(n-1)}$ that are direct, i.e. that do not proceed via the n-th node. If either i or j are not directly connected to n in $\mathbf{T}^{(n-1)}$, then $G_{ij}^{(n)} = 1$.

Since the total number of $i \leftarrow j$ kMC transitions along the $\alpha \leftarrow \epsilon$ trajectory on the original Markov chain is known from $\boldsymbol{\eta}$, which is derived from $\mathbf{H}^{(0)}$ and $\mathbf{h}$, the time $t_{\mathbb{A}}$ elapsed along the path for escape from the community $\mathbb{B}$ can be sampled. The pseudocode for the categorical sampling, iterative reverse randomization, and transition time sampling procedures detailed in this section is given in Algorithm 3. Here, $\sim$ denotes a random number drawn from a distribution, $\Gamma(\alpha, \beta)$ is the gamma distribution with shape parameter α and rate parameter β^{-1}, and $\mathrm{B}(h, p)$ and $\mathrm{NB}(r, p)$ are the binomial and negative binomial distributions, respectively, with trial number h, success number r, and success probability p.

The final section of Algorithm 3, in which $t_{\mathbb{A}}$ is sampled, assumes that the input transition probability matrices correspond to a CTMC. Recall that in the continuous-time case, the time for a $i \leftarrow j$ transition is sampled from an exponential distribution with rate parameter τ_j^{-1}.[110] If the linearized transition probability matrix (Eq. 1.57) is used instead of the branching probability matrix, then the mean waiting times for nodes on the untransformed subnetwork $\mathbf{T}^{(0)}$ are uniform, $\tau_j \equiv \tau\ \forall j$. Hence, the time $t_{\mathbb{A}}$ elapsed along the escape trajectory on the untransformed network can be drawn from a single gamma distribution, with shape parameter equal to the total number of internode transitions on $\mathbf{T}^{(0)}$ and rate

parameter τ^{-1}. That is, $t_{\mathbb{A}} \sim \Gamma(\sum_{j\in\mathbb{E}\cup\mathbb{T}} \eta_j, \tau)$. In the discrete-time case, the time step is fixed and uniform for all transitions, equal to the lag time τ, and so $t_{\mathbb{A}} = \tau \sum_{j\in\mathbb{E}\cup\mathbb{T}} \eta_j$.

Algorithm 3 describes the generation of a single escape trajectory from the currently occupied community $\mathbb{B}$, given the set of transition matrices $\{\mathbf{T}^{(n)}\}$ computed by renormalization (Sec. 1.5.1). In a kPS simulation to sample complete $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths, the main loop of the kPS algorithm (comprising renormalization, categorical sampling, iterative reverse randomization, and sampling of a transition time) is repeated many times to yield a stochastic trajectory that hits the target state $\mathcal{A}$.

Model example

For clarity, we illustrate the kPS algorithm for a model example. The network shown in Fig. 1.4 consists of four nodes. The set $\mathbb{E}$ comprises three nodes $\beta < n < \gamma$ that are to be eliminated by graph transformation, in order of increasing indices. The absorbing boundary comprises a single node, $\partial\mathbb{A} \equiv \{\alpha\}$, and there are no retained nodes, so that $\mathbb{E} \equiv \mathbb{B}$. Therefore, when all three nodes of the set $\mathbb{E}$ have been eliminated, the only available moves are to the absorbing node α. Hence, in the categorical sampling procedure based on Eq. 1.56, the escape trajectory reaches the absorbing boundary from the initial node ϵ in a single transition. The corresponding element $H^{(\gamma)}_{\alpha\epsilon}$ of the hopping matrix $\mathbf{H}^{(\gamma)}$, which contains the numbers of transitions on the network $\mathbf{T}^{(\gamma)}$ where all nodes of the set $\mathbb{E}$ are eliminated, is then incremented by one.

To understand the effect of renormalization, in the form employed within kPS (Eq. 1.54), consider the elimination of node n from the stochastic matrix $\mathbf{T}^{(\beta)}$ to give $\mathbf{T}^{(n)}$, as shown in Fig. 1.4. Prior to the elimination of node n, node β is already eliminated, and hence no edges to node β are included in the network $\mathbf{T}^{(\beta)}$. Upon the elimination of node n, edges to node n, including the $n \leftarrow n$ self-loop, are likewise removed from the network. To compensate for the removal of these transitions, the transition probabilities of edges from node n to neighbouring noneliminated nodes (i.e. γ and α) increase. The transition probabilities for edges between pairs of nodes that are both directly connected to node n, and for which the terminal node is noneliminated, similarly increase. This operation involves updating the $\gamma \leftarrow \gamma$ and $\alpha \leftarrow \alpha$ self-loops, as well as the $\alpha \leftarrow \gamma$, $\gamma \leftarrow \alpha$, and $\alpha \leftarrow \beta$ edges, which already exist in the network, and the addition of a $\gamma \leftarrow \beta$ edge (indicated by $*$ in Fig. 1.4) in the renormalized network, since nodes β and γ were not directly connected in $\mathbf{T}^{(\beta)}$. Fig. 1.4 also shows the reverse randomization procedure to sample $\mathbf{H}^{(\beta)}$ from $\mathbf{H}^{(n)}$, hopping matrices for which node n is noneliminated and eliminated (i.e. corresponding to the networks $\mathbf{T}^{(\beta)}$ and $\mathbf{T}^{(n)}$), respectively. The numbers of transitions along the edges to be

updated are generated randomly in a way that compensates for the changes in the transition probabilities resulting from the corresponding single iteration of the renormalization phase, as outlined in Algorithm 3.

A single iteration of the reverse randomization procedure, given as pseudocode in the third **for** loop of Algorithm 3, comprises four individual steps. Fig. 1.5 shows the effect of each of these individual steps, (i)-(iv), on the hopping matrix $\mathbf{H}^{(n)}$ to yield the matrix $\mathbf{H}^{(\beta)}$ (*cf.* Fig. 1.4). In step (i), the numbers of transitions along each edge from a node of the set $\mathbb{E}$ to a noneliminated node, excluding edges associated with node n, are updated by drawing new values from a binomial distribution. For each transition along an edge of the renormalized network $\mathbf{T}^{(n)}$, a Bernoulli trial is conducted with success probability equal to the ratio of transition probabilities for the edge in the networks $\mathbf{T}^{(\beta)}$ and $\mathbf{T}^{(n)}$, where node n is eliminated and noneliminated, respectively. It is not necessary to consider edges associated with a node that is not directly connected to node n in $\mathbf{T}^{(\beta)}$, because renormalization does not affect these edges, and hence the success probability in the Bernoulli trials is unity. Similarly, if two nodes are not directly connected to one another, but are both connected to node n in $\mathbf{T}^{(\beta)}$, then the ratio of transition probabilities associated with this edge in $\mathbf{T}^{(\beta)}$ and $\mathbf{T}^{(n)}$ is necessarily zero, and hence there are no transitions along this (nonexistent) edge in $\mathbf{H}^{(\beta)}$. Thus the $\gamma \leftarrow \beta$ edge in Fig. 1.5 is removed after step (i).

In step (ii), the numbers of transitions along edges from nodes of the set $\mathbb{E}$ to node n, excluding the $n \leftarrow n$ self-loop, increase to account for any decreases in the numbers of transitions from nodes of the set $\mathbb{E}$ in step (i). Similarly, in step (iii), the numbers of transitions from node n to noneliminated nodes increase to account for any decreases in the numbers of transitions to noneliminated nodes in step (i). Finally, in step (iv), the number of $n \leftarrow n$ self-loop transitions is drawn from a negative binomial distribution, where the number of trials is equal to the number of transitions from node n to noneliminated nodes. By repeated application of this reverse randomization procedure, the numbers of transitions from all eliminated (and any transient) nodes on the original subnetwork $\mathbf{T}^{(0)}$ are obtained, and hence a time for the trajectory can be sampled.

Committor and absorption probabilities

The formulation of the state reduction methodology employed in kPS (Eq. 1.54) provides a numerically stable procedure to compute committor probabilities (Sec. 1.2.5). Recall that the forward $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for the j-th node, q_j^+ (Eqs. 1.28 and 1.29), is defined as the probability that a trajectory at node j hits the absorbing macrostate $\mathcal{A}$ before hitting the initial macrostate $\mathcal{B}$.[98] By definition, $q_b^+ = 0$ for $b \in \mathcal{B}$ and $q_a^+ = 1$ for $a \in \mathcal{A}$.[111] The

committor probabilities for all nodes are obtained as a result of the renormalization phase of the kPS computation if the state space is divided so that $\mathbb{A} \equiv \mathcal{A} \cup \mathcal{B}$ and $\mathbb{E} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$. Then, after repeated application of Eq. 1.54 to the $|\mathbb{E}|$ nodes in the set $\mathbb{E}$, the only transitions from eliminated nodes in the transformed network $\mathbf{T}^{(|\mathbb{E}|)}$ are to nodes in either of the endpoint states $\mathcal{A}$ or $\mathcal{B}$. The forward committor probability for the j-th node, $j \in \mathbb{E}$, is therefore simply

$$q_j^+ = \frac{\sum_{a \in \mathcal{A}} T_{aj}^{(|\mathbb{E}|)}}{\sum_{\gamma} T_{\gamma j}^{(|\mathbb{E}|)}}. \tag{1.58}$$

We provide a detailed description of our proposed state reduction procedure to robustly compute the committor probabilities in Chapter 3.

This formulation of GT also allows for the robust computation of the absorption probabilities.[191] The probability B_{aj} that a trajectory initialized at the j-th node is absorbed at node $a \in \mathcal{A}$ is given straightforwardly from the transition probabilities of the renormalized network where only the nodes of the set $\mathcal{A}^{\mathsf{c}}$ remain noneliminated, i.e. using $\mathbb{A} \equiv \mathcal{A}$ and $\mathbb{E} \equiv \mathcal{A}^{\mathsf{c}}$, via $b_{a \in \mathcal{A}, j \notin \mathcal{A}} = T_{aj}^{(|\mathbb{E}|)}$.

1.5.2 Monte Carlo with absorbing Markov chains

The Monte Carlo with absorbing Markov chains[102–105] (MCAMC) algorithm provides an alternative approach to simulate pathways for a nearly reducible Markov chain, and is similar in spirit to kPS. In the first passage time analysis[229] (FPTA) variant of the MCAMC method, the problem of sampling a transition time $t_{\mathbb{A}}$ and exit node at the absorbing boundary $\alpha \in \partial\mathbb{A}$, for a trajectory escaping from an initial node of the currently occupied community $\epsilon \in \mathbb{B}$, is solved exactly using eigendecomposition (Sec. 1.2).

The probability that the trajectory has exited the community $\mathbb{B}$ at time t is equal to the sum of occupation probabilities for the absorbing boundary nodes at that time, which in the continuous-time case is given by (*cf.* Eq. 1.21)

$$p_{\partial\mathbb{A}}(t) = \sum_{\alpha' \in \partial\mathbb{A}} p_{\alpha'}(t) = \sum_{\alpha \in \partial\mathbb{A}} \sum_{k} \psi_{\alpha'}^{(k)} \phi_{\epsilon}^{(k)} e^{\gamma_k t}, \tag{1.59}$$

where we have used the fact that the initial probability distribution is localized at the ϵ-th node. The Markov chain for the subnetwork $\mathbb{B} \cup \partial\mathbb{A}$ is reducible because the nodes of the state $\partial\mathbb{A}$ are treated as absorbing, so that $p_{\partial\mathbb{A}}(t \to \infty) = 1$. An exit time $t_{\mathbb{A}}$ can therefore be sampled by drawing a random number $r_1 \in (0, 1]$ and solving for $p_{\partial\mathbb{A}}(t_{\mathbb{A}}) = r_1$ numerically using a bracketing and bisection method.[229] The iterative calculation to determine $t_{\mathbb{A}}$ in the

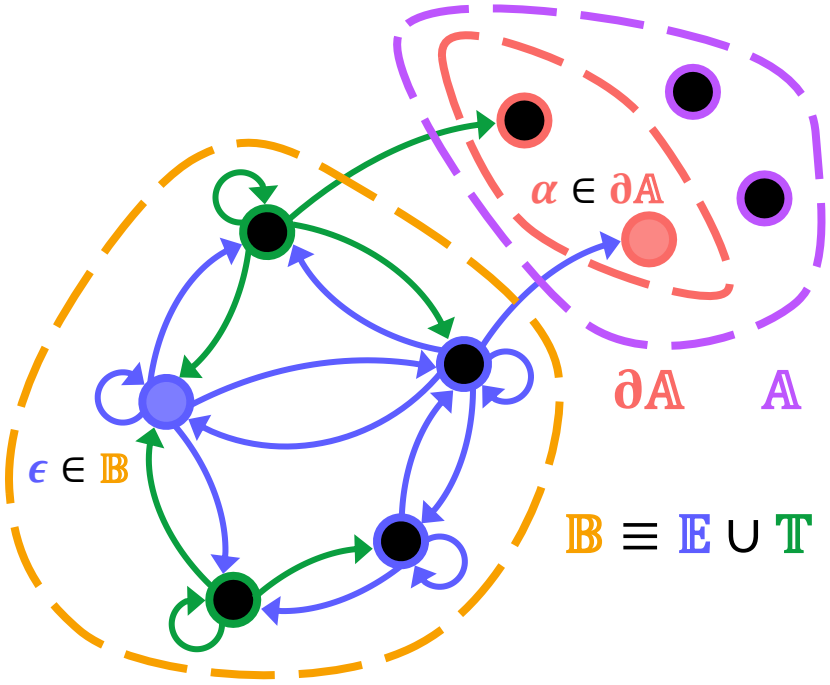


Figure 1.3: Formulation of the escape from a community $\mathbb{B}$ to the absorbing boundary $\partial\mathbb{A} \subseteq \mathbb{A}$ in the kPS algorithm. Given an initially occupied node ϵ, which in the above illustration belongs to the subset $\mathbb{E} \subseteq \mathbb{B}$ of nodes of the basin $\mathbb{B}$ that are to be eliminated by renormalization, the output of Algorithm 3 is a stochastically drawn escape time $t_{\mathbb{A}}$ for a sampled trajectory from ϵ to an absorbing node $\alpha \in \partial\mathbb{A}$. A subset $\mathbb{T} \subset \mathbb{B}$ of nodes of the basin may be retained in the renormalization stage of the kPS algorithm.

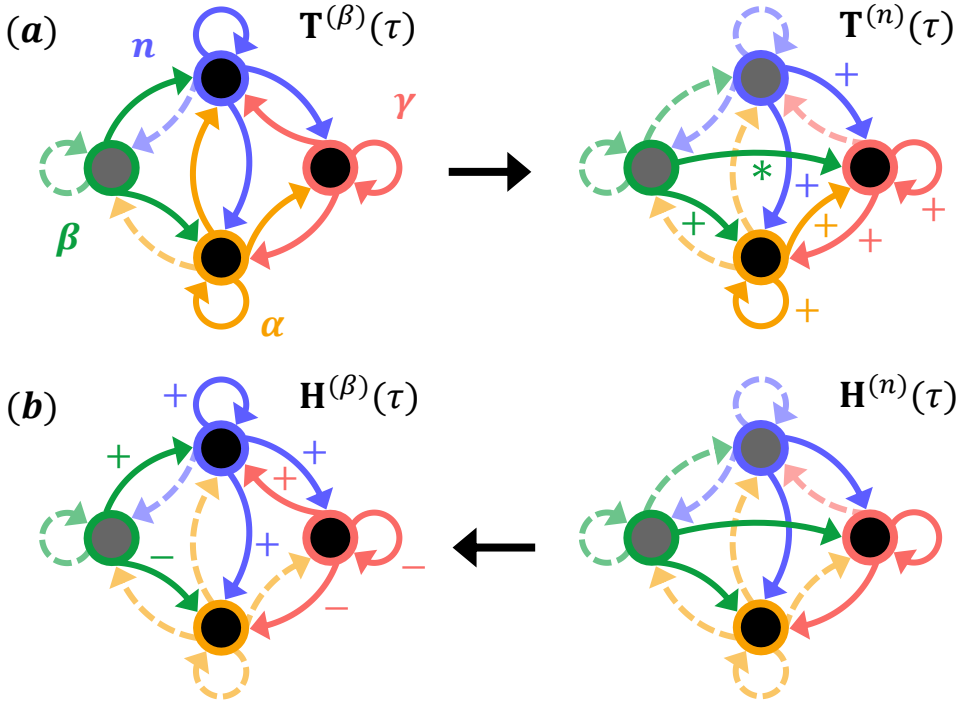


Figure 1.4: Illustration of the main idea of the kPS algorithm. The model network shown consists of four nodes and the state space is divided as follows: $\mathbb{E} = \{\beta, n, \gamma\}$, $\partial\mathbb{A} = \{\alpha\}$, and $\mathbb{T} = \emptyset$. The transition probability matrix $\mathbf{T}^{(\beta)}$ is given by the elimination of node β from the initial transition matrix $\mathbf{T}^{(0)}$. (a) Elimination of node n from $\mathbf{T}^{(\beta)}$ by renormalization gives the matrix $\mathbf{T}^{(n)}$. (b) The hopping matrix $\mathbf{H}^{(\beta)}$, containing the numbers of internode transitions on $\mathbf{T}^{(\beta)}$, can be generated randomly from the hopping matrix $\mathbf{H}^{(n)}$, where node n is eliminated, using $\mathbf{T}^{(\beta)}$ and $\mathbf{T}^{(n)}$. $+$ and $-$ indicate that the matrix element corresponding to the relevant edge must have increased or decreased, respectively. Note that the values of the elements of the hopping matrix may instead stay the same in the reverse randomization procedure. Broken arrows indicate that the edge does not exist or that the corresponding matrix element is zero. Eliminated nodes are shown as transparent. $*$ indicates a new edge resulting from renormalization. The illustration assumes that the stochastic matrix is the linearized or discrete-time transition matrix, so that both the transition and hopping matrices are dependent on the lag time τ, and each node has a self-loop transition.

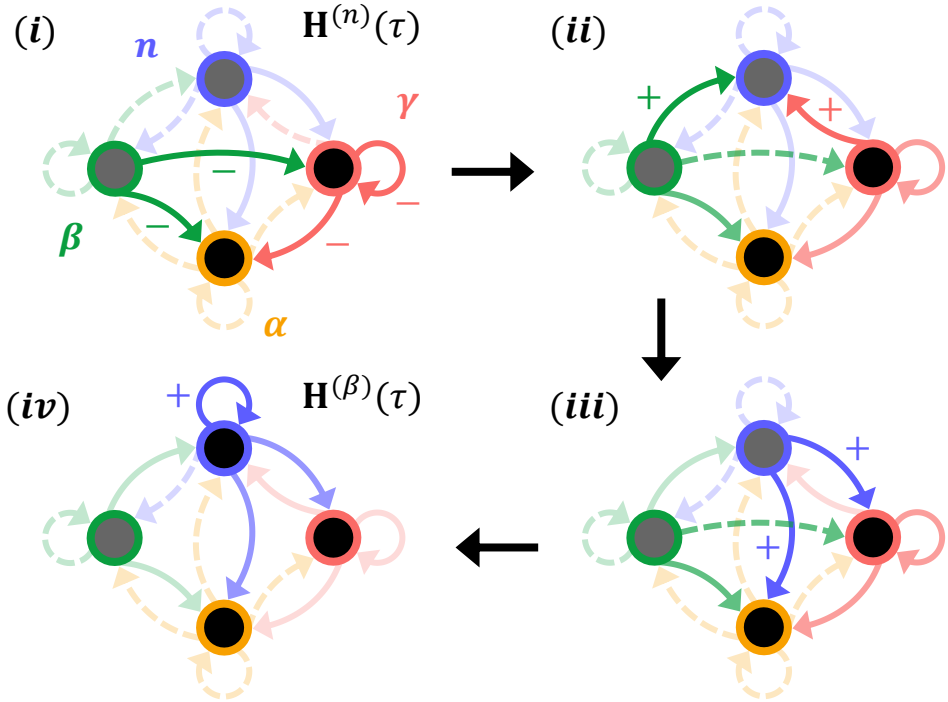


Figure 1.5: Illustration of a single iteration of the reverse randomization procedure (the third **for** loop of Algorithm 3) for the network depicted in Fig. 1.4. The hopping matrix $\mathbf{H}^{(\beta)}$, which contains the numbers of kMC moves on the network $\mathbf{T}^{(\beta)}$, where node n is noneliminated, is generated randomly from the hopping matrix $\mathbf{H}^{(n)}$, which contains the numbers of internode transitions on the network $\mathbf{T}^{(n)}$, where node n is eliminated. This procedure uses the transition probabilities that are the elements of the stochastic matrices $\mathbf{T}^{(\beta)}$ and $\mathbf{T}^{(n)}$, and is completed in four stages, (i)-(iv), as described in Sec. 1.5.1. $+$ and $-$ indicate that the hopping matrix element corresponding to the relevant edge must have increased or decreased, respectively, or else have remained the same. Broken arrows indicate that the edge does not exist or that the corresponding matrix element is zero. Eliminated nodes are shown as transparent. The highlighted edges are those that are updated in the indicated stage of the reverse randomization procedure. Edges used in this calculation are weakly transparent, and irrelevant edges are strongly transparent. The illustration assumes that the stochastic matrix is the linearized or discrete-time probability matrix, so that the elements of the hopping matrices are dependent on the lag time τ, and each node has a self-loop transition.

FPTA method is initialized from the mean exit time, which is given by

$$\langle t_{\mathbb{A}} \rangle = \frac{-\sum_{k\geq 2}^{|\mathcal{S}|} \sum_{\alpha' \in \partial\mathbb{A}} \psi_{\alpha'}^{(k)} \phi_{\epsilon}^{(k)} \gamma_k}{\sum_{k\geq 2}^{|\mathcal{S}|} \sum_{\alpha' \in \partial\mathbb{A}} \psi_{\alpha'}^{(k)} \phi_{\epsilon}^{(k)}}, \tag{1.60}$$

where we have used the Perron-Frobenius theorem, $\gamma_1 = 0$.[107] The approximate mean rate method avoids the iterative calculation to determine $t_{\mathbb{A}}$ by simply using $\langle t_{\mathbb{A}} \rangle$ to advance the simulation clock.[230] The relative occupation probability distribution of absorbing nodes at any given time is also known from the eigendecomposition (Eq. 1.59), and therefore an exit node α can be sampled by drawing a second random number $r_2 \in (0, 1]$ and comparing with the probability distribution $p_{\alpha'}(t_{\mathbb{A}})/p_{\partial\mathbb{A}}(t_{\mathbb{A}})$ $\forall \alpha' \in \partial\mathbb{A}$, analogous to the procedure for selecting a move in the standard kMC algorithm.[82,83]

1.5.3 Practical considerations for advanced simulation algorithms

The time complexity to simulate a trajectory segment escaping from the currently occupied basin $\mathbb{B}$ to the absorbing boundary $\partial\mathbb{A}$ is $\mathcal{O}(|\mathbb{B} \cup \partial\mathbb{A}|^3)$ for both the kPS[84] and MCAMC[93] algorithms. Importantly, the CPU time to execute a single iteration of the main loop in the kPS and MCAMC algorithms does not depend on the actual number of transitions along the path. This feature makes the methods extremely powerful when the basins are chosen to accurately reflect the metastable sets of nodes, so that the trajectory segments are sufficiently long that simulation of the path by standard kMC is unfeasible. Thus the computational overhead associated with the more advanced algorithms is then offset.[101] The choice of community structure that is leveraged in these advanced simulation algorithms is therefore crucial to their success in simulating trajectories on nearly reducible Markov chains. The partitioning is also a critical consideration in the block formulation of the GT algorithm to compute the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT, and in exact uncoupling-coupling to compute the stationary distribution (Sec. 1.4.1), where inversion of the Markovian kernel for a community of nodes is numerically unstable if the subnetwork encompasses a separation of characteristic timescales. Likewise, a condition for efficient convergence of iterative aggregation-disaggregation is that the diagonal blocks of the partitioned Markov chain are nearly stochastic (Sec. 1.4.2).

A discussion of appropriate community detection procedures to identify metastable macrostates is beyond the scope of the present review. However, we briefly outline a framework to refine an initial partitioning, which is useful in practical applications of the aforementioned methods, where the efficiency and/or numerical stability is highly sensitive even to slight perturbations of the community structure. The central theorem of this approach is that, for a given partitioning of a CTMC into a set of N communities, there exists an upper bound on

the second dominant eigenvalue of the $N \times N$-dimensional lumped rate matrix given by the local equilibrium approximation (LEA).[141,153,231] An analogous theorem applies in discrete-time. This result suggests a variational optimization procedure to refine an initial clustering, where the assigned macrostates for one or more nodes at the intercommunity boundaries are randomly switched to that of a neighbouring community. The second dominant eigenvalue $\gamma_2^{\mathcal{C}}$ of the updated lumped Markov chain given by the LEA[6,140] is then computed. The latter operation is efficient if N is not large, especially since the full eigenspectrum is not required and therefore Krylov subpsace methods (Sec. 1.2.4) can be employed. Moreover, the eigendecomposition is numerically stable if the lumped Markov chain does not encompass a separation of characteristic timescales, which in any case is required for the Markovian approximation to the coarse-grained dynamics to be valid.

An increase in $\gamma_2^{\mathcal{C}}$ essentially guarantees that the reduced Markovian network better approximates the slowest relaxation process of the original Markov chain.[232,233] This eigenvalue therefore provides a rigorous metric to improve an initial clustering in an interpretable way. Recently, it was shown that there exists a lower bound for the Kemeny constant $\zeta_{\mathrm{K}}^{\mathcal{C}}$ (Eq. 1.15) of a reduced Markov chain lumped according to the community structure $\mathcal{C}$, and with coarse-grained intercommunity transition probabilities or rates given by the LEA.[108] $\zeta_{\mathrm{K}}^{\mathcal{C}}$ therefore provides an alternative objective function for the variational optimization procedure, which may be preferable to using the second dominant eigenvalue $\gamma_2^{\mathcal{C}}$. The former is a sum of eigenvalues (Eq. 1.18), and therefore quantifies the extent to which the coarse-grained Markov chain reproduces *all* relaxation processes of the original Markov chain, with slower dynamical eigenmodes receiving a larger weighting. In the simplest possible implementation, the variational optimization is performed using a greedy approach, where node-switching moves that increase $\gamma_2^{\mathcal{C}}$ (or decrease $\zeta_{\mathrm{K}}^{\mathcal{C}}$) are always accepted. More sophisticated stochastic optimization approaches calculate an acceptance probability for a proposed move. Our proposed procedure to refine a community structure $\mathcal{C}$ by variational optimization is discussed in more detail in Chapter 4.

The computational overhead associated with the kPS and MCAMC algorithms may become cumbersome if a metastable community comprises a large number of nodes. It is not an option to split a large basin into separate macrostates, because then escape of a trajectory to an absorbing boundary does not constitute a long-timescale process. The only way to reduce the computational expense for a given iteration of either simulation algorithm is to reduce the dimensionality of the relevant subnetwork. A simple way to achieve this goal is recursive regrouping[6] to subsume sets of nodes that are interconnected by transition rates that exceed a specified threshold,[52] with the new intergroup transition rates given by the local equilibrium approximation.[153] If the basin is metastable, then the error resulting from this

procedure ought to be small, since the regrouping of nodes that are interconnected by fast transition rates should not have a significant effect on the slow dynamics for escape from the community. A more advanced pre-processing strategy is partial graph transformation,[234] where renormalization (Eqs. 1.34 and 1.35) is used to eliminate a subset of chosen nodes within the predefined communities.[87] This framework exploits the fact that, for a typical Markov chain, it is likely that the path probability distribution for the ensemble of escape trajectories from a given basin is localized.[101] For instance, the probability distribution of reactive trajectories (Eq. 1.31) may be concentrated in a small fraction of nodes, or escape to a particular node at the absorbing boundary may be strongly favoured. Therefore, in principle, it should be possible to identify the subset of basin nodes to be eliminated that minimizes the information loss on the slow dynamics for escape from the community. Empirically, we find that it is favourable to retain nodes at the intercommunity boundary, as well as nodes with large stationary probabilities and small mean waiting times.[234]

1.6 Conclusions

We have provided an overview of linear algebra (Sec. 1.2) and state reduction (Secs. 1.3-1.5.1) methods for the exact numerical analysis of Markovian network dynamics. In Sec. 1.2, we surveyed expressions for properties characterizing both the global and local dynamical behaviour of finite Markov chains. We began by noting that macroscopic quantities, such as moments of the first passage and mixing time distributions, can be computed from a fundamental matrix associated with an irreducible Markov chain (Sec. 1.2.2), or using eigendecomposition (Sec. 1.2.3). We also defined quantities that characterize an $\mathcal{A} \leftarrow \mathcal{B}$ transition from an initial ($\mathcal{B}$) to an absorbing ($\mathcal{A}$) state at a microscopic level of detail, including the committor probabilities for nodes, which are the central object in analyzing the features of the productive transition (Sec. 1.2.5). For nearly reducible Markov chains, which feature a separation of characteristic timescales, ill-conditioning typically prohibits the solution of a given system of linear equations by conventional algorithms. While preconditioning schemes can be employed to aid the convergence of sparse linear algebra methods applied to Markov chains exhibiting metastability, this framework is not readily generalizable, since nontrivial and system-specific considerations arise (Sec. 1.2.4).

We therefore focused on state reduction algorithms, which have inherent numerical stability, to analyze arbitrary discrete- and continuous-time Markov chains. State reduction methods are based on renormalization (Eq. 1.34) to sequentially eliminate nodes or blocks thereof. Some state reduction algorithms (such as the GTH[125,208] and REFUND[123] algorithms) also incorporate a backward pass phase to restore nodes in turn, recursively computing the

properties of the successive Markov chains having assigned the trivial solution to the simple one-node reduced system. Sec. 1.3 detailed the graph transformation algorithm to robustly compute the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT. In Sec. 1.4, state reduction procedures were described that enable computation of all dynamical properties outlined in Sec. 1.2 for nearly reducible Markov chains. The stationary distribution can be computed by exact uncoupling-coupling (Sec. 1.4.1), iterative aggregation-disaggregation (Sec. 1.4.2), or the Grassmann-Taksar-Heyman algorithm (Sec. 1.4.3). The fundamental matrix of an irreducible Markov chain can be determined by the REFUND algorithm (Sec. 1.4.4). Other state reduction methods were summarized in Sec. 1.4.5. Finally, we noted that sampling trajectories using kinetic Monte Carlo is unfeasibly inefficient for systems exhibiting rare event dynamics, and presented an account of kinetic path sampling, which extends the state reduction methodology to sample the numbers of internode transitions along pathways (Sec. 1.5).

In this thesis, we extend the scope of the state reduction methodology with new procedures, propose novel quantitative analyses to extract global and local dynamical information from finite Markov chains (that are applicable to nearly reducible models when the necessary quantities are obtained robustly), and address workflow and implementation issues to optimize the efficiency of state reduction algorithms. In Chapter 2, we extend the theory of eliminating nodes from a Markov chain by renormalization (Sec. 1.3.1) to compute the expectation of any first passage path property that is a sum of contributions from individual transitions. We also propose a method to obtain a finite set of simple transition flux-paths that correspond to a factorization of the total reactive $\mathcal{A} \leftarrow \mathcal{B}$ flux (Eq. 1.33). This approach provides a pathwise analysis of the reactive flux that is complementary to the usual analysis based on a $\mathcal{A}$-$\mathcal{B}$ cut set of edges (Sec. 1.2.5). Our analysis uses a shortest paths algorithm with edge weights that depend on the stationary distribution and committor probabilities, which can be obtained robustly via state reduction (see Secs. 1.4.1-1.4.3 and Sec. 1.5.1, respectively). In Chapter 3, we extend the GT algorithm of Sec. 1.3.1 with a backwards pass phase to compute the MFPTs for transitions from all transient nodes in a single computation. In addition, we report state reduction algorithms to compute the committor probabilities and the expected numbers of times that nodes are visited on first passage and transition paths. We also derive an expression for the probability that a node is visited on a transition path, which we refer to as the *reactive visitation probability*, and which can be evaluated straightforwardly from this information. Hence, we can assess the importance of individual nodes in facilitating the dominant $\mathcal{A} \leftarrow \mathcal{B}$ pathways. In Chapter 4, we propose a workflow for the unsupervised simulation of complete $\mathcal{A} \leftarrow \mathcal{B}$ pathways using kPS (Sec. 1.5.1), based on obtaining an initial approximate partition of the Markov chain into metastable macrostates, and subsequent refinement of this community structure by a variational optimization procedure (Sec. 1.5.3).

We show that our approach provides a powerful framework to obtain simulation estimates for the committor and reactive visitation probabilities, as well as first passage time distributions.

We demonstrate our methodology with applications to realistic and computationally challenging systems that are relevant to the physical sciences, namely: a model landscape with metastable states (Chapter 2), a structural transition in an atomic cluster (Chapter 3), and the folding transition of a peptide (Chapter 4). Each of these systems features a separation of characteristic timescales, and therefore corresponds to a nearly reducible Markov chain. Metastability is crucially important to consider, since realistic dynamical models typically exhibit a particular rare event that is the process of interest.[52–66] Thus we show that the state reduction procedures reviewed herein allow for comprehensive numerical analysis of nearly reducible Markov chains, which would otherwise be intractable, and hence are valuable in many practical applications. The ability to extract observable quantities and perform detailed analyses of numerically challenging models will lead to new insights into the dynamical behaviour of complex systems. We discuss possibilities for future work in Chapter 5. In particular, we can probe the relationship between the local features of a Markovian network and the slow global dynamics, which is typically influenced strongly by a small number of states that facilitate the dominant transition mechanisms.[88,101,191,201]

input : transition probability matrix $\mathbf{T}$
set of nodes $n \in \mathcal{S} \setminus \{1\}$ to be eliminated from the state space $\mathcal{S}$
output: group inverse $\mathbf{A}^{\#}$
stationary distribution $\boldsymbol{\pi}$

/* elimination phase. Note that nodes are eliminated in reverse order, thereby effectively performing the triangular decomposition of the Markovian kernel: $\mathbf{I} - \mathbf{T} = \mathbf{UL}$ */
for $n = |\mathcal{S}|, |\mathcal{S}| - 1, \ldots, 2$ **do**
/* $\mathbf{p}_{:n}$ is the column and $\mathbf{q}_{:n}^{\top}$ the row vector, respectively, corresponding to the node to be eliminated (n) in the stochastic matrix for the reduced Markov chain at the current iteration, not including the diagonal element. Both vectors are of dimension $(n-1)$ */
$\mathbf{p}_{:n}, \mathbf{q}_{:n}^{\top} \leftarrow$ GetBlock($\mathbf{T}$,n);
$S_n \leftarrow \sum_{\gamma<n} T_{\gamma n} \quad (\equiv 1 - T_{nn})$; // factors $S_n \ \forall 1 < n < |\mathcal{S}|$ are stored in a vector
$\mathbf{T}'_{:n,:n} \leftarrow \mathbf{T}_{:n,:n} + S_n^{-1} \mathbf{p}_{:n} \mathbf{q}_{:n}^{\top}$; // GTH elimination of the n-th node
$\mathbf{T}_{:n+1,:n+1} \leftarrow \begin{pmatrix} \mathbf{T}'_{:n,:n} & \mathbf{p}_{:n} \\ S_n^{-1} \mathbf{q}_{:n}^{\top} & T_{nn} \end{pmatrix}$; // overwrite the elements of the transition matrix
/* the stationary distribution and group inverse for the reduced Markov chain with only one node remaining are trivial */
$\boldsymbol{\pi}_1 \leftarrow (1)$;
$\mathbf{A}_1^{\#} \leftarrow (0)$;
/* backwards pass phase. At this stage, the first diagonal element of $\mathbf{T}$, $T_{11} = 1$, corresponds to the stochastic matrix for a reduced Markov chain comprising only a single node. The other off-diagonal elements of $\mathbf{T}$ are the vectors $\mathbf{p}_{:n}$ and $\mathbf{q}_{:n}^{\top}$ for $1 < n \leq |\mathcal{S}|$ */
for $n = 2, \ldots, |\mathcal{S}| - 1, |\mathcal{S}|$ **do**
/* $\mathbf{p}_{:n}^{\top}$ and $\mathbf{q}_{:n}^{\top}$ are defined above, but $\mathbf{q}_{:n}^{\top}$ is now scaled by S_n^{-1} compared to the $\mathbf{q}_{:n}^{\top}$ vector returned by GetBlock during the elimination phase */
$\mathbf{p}_{:n}, \mathbf{q}_{:n}^{\top} \leftarrow$ GetBlock($\mathbf{T}$,n);
$\alpha \leftarrow (1 + \boldsymbol{\pi}_{n-1}^{\top} \mathbf{q}_{:n})^{-1}$;
$\beta \leftarrow \alpha \boldsymbol{\pi}_{n-1}^{\top} \mathbf{q}_{:n} / S_n \quad (\equiv (1-\alpha)/S_n)$; // confers numerical stability
$\mathbf{r} \leftarrow \alpha \mathbf{A}_{n-1}^{\#} \mathbf{q}_{:n}$;
$\mathbf{t}^{\top} \leftarrow \beta \mathbf{p}_{:n}^{\top} \mathbf{A}_{n-1}^{\#}$;
$c \leftarrow \beta(\alpha + \mathbf{p}_{:n}^{\top} \mathbf{r})$;
$\mathbf{c} \leftarrow c \boldsymbol{\pi}_{n-1} - \mathbf{t}$;
/* stationary distribution for the reduced Markov chain with the n-th node restored */
$\boldsymbol{\pi}_n \leftarrow \alpha \begin{pmatrix} \boldsymbol{\pi}_{n-1}^{\top} \mathbf{q}_{:n} \\ \boldsymbol{\pi}_{n-1} \end{pmatrix}$; // this stationary vector is normalized
$\delta \leftarrow (\boldsymbol{\pi}_{n-1}^{\top} \mathbf{q}_{:n})^{-1} \quad (\equiv \alpha/(1-\alpha))$; // confers numerical stability
/* the group inverse for the parent Markov chain, for which the n-th node is restored, is recovered by an explicit formula */
$\mathbf{A}_n^{\#} \leftarrow \begin{pmatrix} \mathbf{A}_{n-1}^{\#} - [\mathbf{r}\boldsymbol{\pi}_{n-1}^{\top} + \mathbf{1}_{n-1}\mathbf{c}^{\top}]^{\top} & -\delta\mathbf{c} \\ \mathbf{r}^{\top} - c\mathbf{1}_{n-1}^{\top} & \delta c \end{pmatrix}$; // $\mathbf{1}_{n-1}$ is the $(n-1)$-dimensional unit vector
$\mathbf{A}^{\#} \leftarrow \mathbf{A}_{|\mathcal{S}|}^{\#}; \quad \boldsymbol{\pi} \leftarrow \boldsymbol{\pi}_{|\mathcal{S}|}$;
return $\mathbf{A}^{\#}, \boldsymbol{\pi}$;

/* function to partition the relevant block of the matrix $\mathbf{T}$. During the elimination phase, this block corresponds to the stochastic matrix for the reduced Markov chain at the current iteration (the other elements are overwritten during this phase) */
function GetBlock($\mathbf{T}$,n)
let $\mathbf{T}_{:n+1,:n+1} = \begin{pmatrix} \mathbf{T}_{:n,:n} & \mathbf{p}_{:n} \\ \mathbf{q}_{:n}^{\top} & T_{nn} \end{pmatrix}$; // reduced Markov chain partitioned into blocks
return $\mathbf{p}_{:n}, \mathbf{q}_{:n}^{\top}$;

Algorithm 2: REFUND algorithm[123] to compute the group inverse $\mathbf{A}^{\#}$ of a Markov chain (Eqs. 1.8-1.10). The stationary distribution $\boldsymbol{\pi}$ is computed concomitantly. In the above, $\mathbf{T}_{:i,:j}$ denotes the $(i-1) \times (j-1)$-dimensional block of the stochastic matrix $\mathbf{T}$ comprising elements with row index less than i and column index less than j. Similarly, $\mathbf{p}_{:n}$ is the $(n-1)$-dimensional column vector containing the elements of $\mathbf{p}$ with indices $1 \leq \gamma < n$.

input : sets of nodes $\mathbb{E} \neq \emptyset$, $\mathbb{T}$, $\mathbb{B} \equiv \mathbb{E} \cup \mathbb{T}$, $\mathbb{A} \equiv \mathbb{B}^{\mathsf{c}}$ and $\partial\mathbb{A} \subseteq \mathbb{A}$ (Fig. 1.3)
$N_{\mathrm{c}} = |\mathbb{E} \cup \mathbb{T} \cup \partial\mathbb{A}|$
initially occupied node $\epsilon \in \mathbb{B}$
set of $|\mathbb{E}|+1$ stochastic matrices $\{\mathbf{T}^{(n)}\}$, $0 \leq n \leq |\mathbb{E}|$, from renormalization (Eq. 1.54) of nodes in $\mathbb{E}$
set of $|\mathbb{E}|$ matrices $\{\mathbf{G}^{(n)}\}$, for $0 < n \leq |\mathbb{E}|$, derived from the $\{\mathbf{T}^{(n)}\}$ matrices via Eq. 1.57
output: absorbing node $\alpha \in \partial\mathbb{A}$
hopping matrix $\mathbf{H}^{(0)}$ with elements $H_{ij}^{(0)}$ equal to the numbers of $i \leftarrow j \in \mathbb{E}$ internode transitions along the sampled $\alpha \leftarrow \epsilon$ path on $\mathbf{T}^{(0)}$
vector $\boldsymbol{\eta}$ with elements η_j equal to the total number of transitions from node $j \in \mathbb{B}$
time elapsed $t_{\mathbb{A}}$ for the $\alpha \leftarrow \epsilon$ trajectory for escape from $\mathbb{B}$

initialize $\mathbf{H}^{(|\mathbb{E}|)}$ (dimension $(N_{\mathrm{c}} - |\mathbb{E}|) \times |\mathbb{E}|$);
initialize $\mathbf{h}$ (dimension $|\mathbb{E}|$);
initialize $\boldsymbol{\eta}$ (dimension $|\mathbb{B}|$);
$\beta \leftarrow \epsilon$;
/* Categorical sampling procedure to sample an absorbing boundary node $\alpha \in \partial\mathbb{A}$ and numbers of internode transitions on the renormalized subnetwork $\mathbb{B} \cup \partial\mathbb{A}$ */
while $\beta \notin \partial\mathbb{A}$ **do**
  $\gamma \sim \mathbf{c}_\beta$ (Eq. 1.56);
  if $\beta \in \mathbb{E}$ **then**
    $H_{\gamma\beta}^{(|\mathbb{E}|)} \leftarrow H_{\gamma\beta}^{(|\mathbb{E}|)} + 1$;
  else
    $\eta_\beta \leftarrow \eta_\beta + 1$;
  $\beta \leftarrow \gamma$;
$\alpha \leftarrow \beta$;
/* Sample numbers of transitions from noneliminated nodes */
for $\delta \in \mathbb{T}$ **do**
  $f_\delta \sim \mathrm{NB}(\eta_\delta, 1 - T_{\delta\delta}^{(0)})$;
  $\eta_\delta \leftarrow \eta_\delta + f_\delta$;
/* Iterative reverse randomization to generate the numbers of internode transitions on the original subnetwork $\mathbb{B} \cup \partial\mathbb{A}$ */
for $n \leftarrow |\mathbb{E}|$ **to** 1 **do**
  initialize $\mathbf{H}^{(n-1)}$ (dimension $(N_{\mathrm{c}} - (n-1)) \times |\mathbb{E}|$);
  for $\beta \in \mathbb{E} \setminus \{n\}$ **and** $\gamma \in \{n+1, \ldots, N_{\mathrm{c}}\}$ **do**
    $H_{\gamma\beta}^{(n-1)} \sim \mathrm{B}(H_{\gamma\beta}^{(n)}, G_{\gamma\beta}^{(n)})$;
  for $\beta \in \mathbb{E} \setminus \{n\}$ **do**
    $H_{n\beta}^{(n-1)} \leftarrow \sum_{\gamma \in \{n+1,\ldots,N_{\mathrm{c}}\}} H_{\gamma\beta}^{(n)} - H_{\gamma\beta}^{(n-1)}$;
  for $\gamma \in \{n+1, \ldots, N_{\mathrm{c}}\}$ **do**
    $H_{\gamma n}^{(n-1)} \leftarrow H_{\gamma n}^{(n)} + \sum_{\beta \in \mathbb{E} \setminus \{n\}} H_{\gamma\beta}^{(n)} - H_{\gamma\beta}^{(n-1)}$;
  $h_n \leftarrow \sum_{\gamma \in \{n+1,\ldots,N_{\mathrm{c}}\}} H_{\gamma n}^{(n-1)}$;
  $H_{nn}^{(n-1)} \sim \mathrm{NB}(h_n, 1 - T_{nn}^{(n-1)})$;
  $\eta_n \leftarrow h_n + H_{nn}^{(n-1)}$;
  deallocate $\mathbf{H}^{(n)}$, $\mathbf{T}^{(n)}$;
/* sample the transition time for the $\alpha \in \partial\mathbb{A} \leftarrow \epsilon \in \mathbb{B}$ trajectory */
$t_{\mathbb{A}} \leftarrow 0$;
for $j \in \mathbb{B}$ **do**
  $\Delta_j \sim \Gamma(\eta_j, \tau_j)$;
  $t_{\mathbb{A}} \leftarrow t_{\mathbb{A}} + \Delta_j$;
deallocate $\mathbf{H}^{(0)}$, $\mathbf{T}^{(0)}$, $\mathbf{h}$, $\boldsymbol{\eta}$;
return $t_{\mathbb{A}}$, α;

Algorithm 3: The categorical sampling, iterative reverse randomization, and transition time sampling procedures that constitute the remainder of the main loop of the kinetic path sampling (kPS) algorithm, after determination of the set of transition probability matrices $\{\mathbf{T}^{(n)}\}$, $0 \leq n \leq |\mathbb{E}|$ by renormalization. Note that the final **for** loop, in which a transition time is sampled, corresponds to the continuous-time formulation.

Bibliography

[1] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand, New Jersey, USA, 1960.

[2] J. R. Norris. *Markov Chains.* Cambridge University Press, New York, USA, 1997.

[3] C. M. Grinstead and J. L. Snell. *Introduction to Probability.* American Mathematical Society, Providence, Rhode Island, 1997.

[4] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, London, UK, third edition, 1998.

[5] D. J. Wales. *Energy Landscapes.* Cambridge University Press, Cambridge, UK, 2003.

[6] D. J. Wales and P. Salamon. *Proc. Natl. Acad. Sci. USA*, 111:617–622, 2014.

[7] S. Sriraman, I. G. Kevrekidis, and G. Hummer. *J. Phys. Chem. B*, 109:6479–6484, 2005.

[8] N.-V. Buchete and G. Hummer. *J. Phys. Chem. B*, 112:6057–6069, 2008.

[9] N.-V. Buchete and G. Hummer. *Phys. Rev. E*, 77:030902, 2008.

[10] P. Metzner, E. Dittmer, T. Jahnke, and C. Schütte. *J. Comput. Phys.*, 227:353–375, 2007.

[11] R. T. McGibbon and V. S. Pande. *J. Chem. Phys.*, 143:034109, 2015.

[12] P. Metzner, F. Noé, and C. Schütte. *Phys. Rev. E*, 80:021106, 2009.

[13] B. Trendelkamp-Schroer and F. Noé. *J. Chem. Phys.*, 138:164113, 2013.

[14] B. Trendelkamp-Schroer, H. Wu, F. Paul, and F. Noé. *J. Chem. Phys.*, 143:174101, 2015.

[15] B. Trendelkamp-Schroer and F. Noé. *Phys. Rev. X*, 6:011009, 2016.

[16] G. R. Bowman, V. S. Pande, and F. Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation.* Springer, Netherlands, first edition, 2014.

[17] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. *J. Chem. Phys.*, 134:174105, 2011.

[18] V. S. Pande, K. Beauchamp, and G. R. Bowman. *Methods*, 52:99–105, 2010.

[19] B. E. Husic and V. S. Pande. *J. Am. Chem. Soc.*, 140:2386–2896, 2018.

[20] A. Mardt, L. Pasquali, H. Wu, and F. Noé. *Nat. Commun.*, 9:5, 2018.

[21] R. G. Mantell, C. E. Pitt, and D. J. Wales. *J. Chem. Theory Comput.*, 12:6182–6191, 2016.

[22] D. J. Wales. *Mol. Phys.*, 100:3285–3305, 2002.

[23] D. J. Wales. *Mol. Phys.*, 102:891–908, 2004.

[24] F. Noé, D. Krachtus, J. C. Smith, and S. Fischer. *J. Chem. Theory Comput.*, 2:840–857, 2006.

[25] F. Noé and J. C. Smith. In A. Deutsch, L. Brusch, J. Byrne, G. de Vries, and H.-P. Herzel, editors, *Mathematical Modeling of Biological Systems, Volume I.*, pages 125–144. Birkhäuser, Boston, 2007.

[26] F. Noé and S. Fischer. *Curr. Op. Struct. Biol.*, 18:154–162, 2008.

[27] D. J. Wales. *Annu. Rev. Phys. Chem.*, 69:401–425, 2018.

[28] J. A. Joseph, K. Röder, D. Chakraborty, R. G. Mantell, and D. J. Wales. *Chem. Commun.*, 53:6974–6988, 2017.

[29] K. Röder, J. A. Joseph, B. E. Husic, and D. J. Wales. *Adv. Theory Simul.*, 2:1800175, 2019.

[30] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier, Amsterdam, Netherlands, 1992.

[31] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists.* Academic Press, New York, USA, 1992.

[32] R. Zwanzig. *J. Stat. Phys.*, 30:255–262, 1983.

[33] N. Masuda, M. A. Porter, and R. Lambiotte. *Phys. Rep.*, 716-717:1–58, 2017.

[34] E. Seneta. *Linear Algebra Appl.*, 34:259–267, 1980.

[35] D. P. Heyman. *J. Appl. Probab.*, 32:893–901, 1995.

[36] B. Munsky and M. Khammash. *J. Chem. Phys.*, 124:044104, 2006.

[37] K. N. Dinh and R. B. Sidje. *Phys. Biol.*, 13:035003, 2016.

[38] A. S. Novozhilov, G. P. Karev, and E. V. Koonin. *Brief. Bioinformatics*, 7:70–85, 2006.

[39] L. M. Riciardi. In T. G. Hallam and S. A. Levin, editors, *Mathematical Ecology*, pages 155–190. Springer Berlin, Heidelberg, 1986.

[40] C. D. Meyer Jr and R. J. Plemmons. *Linear algebra, Markov chains, and queuing models.* Springer-Verlag, New York, 1993.

[41] T. Schmiedl and U. Seifert. *J. Chem. Phys.*, 126:044101, 2007.

[42] L. B. Newcomb, M. Alaghemandi, and J. R. Green. *J. Chem. Phys.*, 147:034108, 2017.

[43] T. L. Hill. *Free Energy Transduction and Biochemical Cycle Kinetics.* Springer-Verlag, New York, NY, USA, 1989.

[44] H. Ge. *J. Phys. A: Math. Theor.*, 45:215002, 2012.

[45] H. Ge, M. Qian, and H. Qian. *Phys. Rep.*, 510:87–118, 2012.

[46] R. J. Allen, P. B. Warren, and P. R. ten Wolde. *Phys. Rev. Lett.*, 94:018104, 2005.

[47] R. J. Allen, D. Frenkel, and P. R. ten Wolde. *J. Chem. Phys.*, 124:024102, 2006.

[48] B. K. Chu, M. J. Tse, R. R. Sato, and E. L. Read. *BMC Syst. Biol.*, 11:14, 2017.

[49] M. J. Tse, B. K. Chu, C. P. Gallivan, and E. L. Read. *PLoS Comput. Biol.*, 14:e1006336, 2018.

[50] L. J. S. Allen. In F. Brauer, P. van den Driessche, and J. Wu, editors, *Mathematical Epidemiology*, pages 81–130. Springer-Verlag, Berlin, 2008.

[51] L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology.* Prentice Hall, Upper Saddle River, New Jersey, 2003.

[52] T. D. Swinburne, D. Kannan, D. J. Sharpe, and D. J. Wales. *J. Chem. Phys.*, 153:134115, 2020.

[53] D. J. Aldous and M. Brown. In M. Shaked and Y. L. Tong, editors, *IMS Lecture Notes in Statistics, Vol. 22: Stochastic Inequalities*, pages 1–16. Institute of Mathematical Statistics, Ohio, USA, 1992.

[54] P. Heidelberger. *ACM Trans. Model. Comput. Simul.*, 5:43–85, 1995.

[55] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. *Oper. Res.*, 47:495–645, 1999.

[56] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. *J. Phys. A.: Math. Gen.*, 33:L447–L451, 2000.

[57] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. *Commun. Math. Phys.*, 228:219–255, 2002.

[58] S. Juneja and P. Shahabuddin. *Manage. Sci.*, 47:547–562, 2001.

[59] J. Beltrán and C. Landim. *J. Stat. Phys.*, 140:1065–1114, 2010.

[60] E. Vanden-Eijnden and J. Weare. *Commun. Pure Appl. Math*, 65:1770–1803, 2012.

[61] O. Benois and M. Mourragui. *J. Stat. Phys.*, 153:967–990, 2013.

[62] C. Hartmann, R. Banisch, M. Sarich, T. Badowski, and C. Schütte. *Entropy*, 16:350–376, 2014.

[63] M. Sarich, R. Banishc, C. Hartmann, and C. Schütte. *Entropy*, 16:258–286, 2014.

[64] M. K. Cameron. *J. Chem. Phys.*, 141:184113, 2014.

[65] T. Gan and M. Cameron. *J. Nonlinear Sci.*, 27:927–972, 2017.

[66] C. Pérez-Espigares and P. I. Hurtado. *Chaos*, 29:083106, 2019.

[67] D. Helbing. *Quantitative Sociodynamics*. Springer-Verlag, Berlin, second edition, 2010.

[68] F. Noé, I. Horenko, C. Schütte, and J. C. Smith. *J. Chem. Phys.*, 126:155102, 2007.

[69] F. Noé. *J. Chem. Phys.*, 128:244103, 2008.

[70] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. *Proc. Natl. Acad. Sci. USA*, 106:19011–19016, 2009.

[71] J.-H. Prinz, B. Keller, and F. Noé. *Phys. Chem. Chem. Phys.*, 13:16912–16927, 2011.

[72] J. D. Chodera and F. Noé. *Curr. Op. Struct. Biol.*, 25:135–144, 2014.

[73] D. J. Hartfiel and C. D. Meyer Jr. *Linear Algebra Appl.*, 272:193–203, 1998.

[74] C. D. Meyer Jr. *SIAM Rev.*, 31:240–272, 1989.

[75] D. P. Heyman and A. Reeves. *ORSA J. Comp.*, 1:52–60, 1989.

[76] G. W. Stewart and G. Zhang. *Numer. Math.*, 59:1–11, 1991.

[77] B. Philippe, Y. Saad, and W. J. Stewart. *Oper. Res.*, 40:1156–1179, 1992.

[78] C. D. Meyer Jr. *SIAM J. Matrix Anal. Appl.*, 15:715–728, 1994.

[79] D. P. Heyman and D. P. O'Leary. *SIAM J. Matrix Anal. Appl.*, 19:534–540, 1998.

[80] J. L. Barlow. *SIAM J. Matrix Anal. Appl.*, 22:230–241, 2000.

[81] T. J. Frankcombe and S. C. Smith. *Theor. Chem. Acc.*, 124:303–317, 2009.

[82] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. *J. Comput. Phys.*, 17:10–18, 1975.

[83] D. T. Gillespie. *J. Comput. Phys.*, 28:395–407, 1978.

[84] M. Athènes and V. V. Bulatov. *Phys. Rev. Lett.*, 113:230601, 2014.

[85] M. Athènes, S. Kaur, G. Adjanor, T. Vanacker, and T. Jourdan. *Phys. Rev. Materials*, 3:103802, 2019.

[86] D. R. Mason, R. E. Rudd, and A. P. Sutton. *Comput. Phys. Comm.*, 160:140–157, 2004.

[87] V. V. Bulatov, T. Oppelstrup, and M. Athènes. Technical Report LLNL-TR-517795, Lawrence Livermore National Laboratory, 2011.

[88] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 151:124101, 2019.

[89] S. A. Trygubenko and D. J. Wales. *Mol. Phys.*, 104:1497–1507, 2006.

[90] S. A. Trygubenko and D. J. Wales. *J. Chem. Phys.*, 124:234110, 2006.

[91] D. J. Wales. *Int. Rev. Phys. Chem.*, 25:237–282, 2006.

[92] D. J. Wales. *J. Chem. Phys.*, 130:204111, 2009.

[93] J. D. Stevenson and D. J. Wales. *J. Chem. Phys.*, 141:041104, 2014.

[94] R. S. MacKay and J. D. Robinson. *Phil. Trans. Roy. Soc. A*, 376:20170232, 2018.

[95] E. Vanden-Eijnden. In M. Ferrario, G. Ciccotti, and K. Binder, editors, *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, pages 453–493. Springer Berlin, Heidelberg, 2006.

[96] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *J. Chem. Phys.*, 125:084110, 2006.

[97] W. E and E. Vanden-Eijnden. *J. Stat. Phys.*, 123:503–523, 2006.

[98] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *Multiscale Model. Simul.*, 7:1192–1219, 2009.

[99] W. E and E. Vanden-Eijnden. *Annu. Rev. Phys. Chem.*, 61:391–420, 2010.

[100] M. von Kleist, C. Schütte, and W. Zhang. *J. Stat. Phys.*, 170:809–843, 2018.

[101] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 153:024121, 2020.

[102] M. A. Novotny. *Phys. Rev. Lett.*, 74:1–5, 1995.

[103] M. A. Novotny. In D. P. Landau, K. K. Mon, and H.-B. Schüttler, editors, *Computer Simulation Studies in Condensed-Matter Physics VII*, pages 161–165. Springer-Verlag, Berlin, 1994.

[104] M. A. Novotny. In D. Stauffer, editor, *Annual Reviews of Computational Physics: Vol. 9*, pages 153–210. World Scientific, Singapore, 2001.

[105] M. A. Novotny. *Comput. Phys. Commun.*, 147:659–664, 2002.

[106] J. Goutsias and G. Jenkinson. *Phys. Rep.*, 529:199–264, 2013.

[107] C. R. MacCluer. *SIAM Rev.*, 42:487–498, 2000.

[108] A. Kells, V. Koskin, E. Rosta, and A. Annibale. *J. Chem. Phys.*, 152:104108, 2020.

[109] J. G. Kemeny and J. L. Snell. *Theory Prob. Its Appl.*, 6:101–105, 1961.

[110] S. A. Serebrinsky. *Phys. Rev. E*, 83:037701, 2011.

[111] J.-H. Prinz, M. Held, J. C. Smith, and F. Noé. *Multiscale Model. Simul.*, 9:545–567, 2011.

[112] J. J. Hunter. *Linear Algebra Appl.*, 447:38–55, 2014.

[113] J. J. Hunter. *Adv. Appl. Probab.*, 1:188–210, 1969.

[114] C. D. Meyer. *Linear Algebra Appl.*, 22:41–47, 1978.

[115] J. J. Hunter. *Linear Algebra Appl.*, 45:157–198, 1982.

[116] J. J. Hunter. *Linear Algebra Appl.*, 127:71–84, 1990.

[117] J. J. Hunter. *Linear Algebra Appl.*, 429:1135–1162, 2008.

[118] J. G. Kemeny. *Linear Algebra Appl.*, 38:193–206, 1981.

[119] C. D. Meyer Jr. *SIAM Rev.*, 17:443–464, 1975.

[120] P. Coolen-Schrijner and E. A. van Doorn. *Probab. Eng. Inf. Sci.*, 16:351–366, 2002.

[121] B. F. Lamond and M. L. Puterman. *SIAM J. Matrix Anal. Appl.*, 10:118–134, 1989.

[122] J. J. Hunter. *Linear Algebra Appl.*, 102:121–142, 1988.

[123] I. Sonin and J. Thornton. *SIAM J. Matrix Anal. Appl.*, 23:209–224, 2001.

[124] I. Sonin. *Adv. Math.*, 145:159–188, 1999.

[125] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. *Oper. Res.*, 33:1107–1116, 1985.

[126] J. Kohlas. *Zeit. Oper. Res.*, 30:197–207, 1986.

[127] J. J. Hunter. *Spec. Matrices*, 4:151–175, 2016.

[128] J. J. Hunter. *Linear Algebra Appl.*, 511:176–202, 2016.

[129] J. J. Hunter. *Linear Algebra Appl.*, 549:100–122, 2018.

[130] T. Dayar and N. Akar. *SIAM J. Matrix Anal. Appl.*, 27:396–412, 2005.

[131] Z. Zhang, A. Julaiti, B. Hou, H. Zhang, and G. Chen. *Eur. Phys. J. B*, 84:691–697, 2011.

[132] Z. Zhang, Y. Sheng, Z. Hu, and G. Chen. *Chaos*, 22:043129, 2012.

[133] Z. Zhang, T. Shan, and G. Chen. *Phys. Rev. E*, 87:012112, 2013.

[134] N. F. Polizzi, M. J. Therien, and D. N. Beratan. *Isr. J. Chem.*, 56:816–824, 2016.

[135] M. Torchala, P. Chelminiak, M. Kurzynski, and P. A. Bates. *BMC Systems Biology*, 7:130, 2013.

[136] D. J. Bicout and A. Szabo. *J. Chem. Phys.*, 106:10292–10298, 1997.

[137] J. J. Hunter. *Linear Algebra Appl.*, 430:2607–2621, 2009.

[138] J. J. Hunter. *Linear Algebra Appl.*, 417:108–123, 2006.

[139] J. J. Hunter. *Commun. Stat. - Theory Methods*, 43:1309–1321, 2014.

[140] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. *J. Chem. Phys.*, 153:244108, 2020.

[141] A. Kells, Z. E. Mihálka, A. Annibale, and E. Rosta. *J. Chem. Phys.*, 150:134107, 2019.

[142] W. Zheng, E. Gallicchio, N. Deng, M. Andrec, and R. M. Levy. *J. Phys. Chem. B*, 115:1512–1523, 2011.

[143] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. *Linear Algebra Appl.*, 315:39–59, 2000.

[144] P. Deuflhard and M. Weber. *Linear Algebra Appl.*, 398:161–184, 2005.

[145] S. Kube and M. Weber. *J. Chem. Phys.*, 126:024103, 2007.

[146] B. Reuter, K. Fackeldey, and M. Weber. *J. Chem. Phys.*, 150:174103, 2019.

[147] G. R. Bowman. *J. Chem. Phys.*, 137:134111, 2012.

[148] A. Jain and G. J. Stock. *J. Chem. Theory Comput.*, 8:3810–3819, 2012.

[149] F. Noé, H. Wu, J.-H. Prinz, and N. Plattner. *J. Chem. Phys.*, 139:184114, 2013.

[150] G. R. Bowman, L. Meng, and X. Huang. *J. Chem. Phys.*, 139:121905, 2013.

[151] B. E. Husic, K. A. McKiernan, H. K. Wayment-Steele, M. M. Sultan, and V. S. Pande. *J. Chem. Theory Comput.*, 14:1071–1082, 2018.

[152] J. A. Ward and M. López-García. *Appl. Netw. Sci.*, 4:108, 2019.

[153] G. Hummer and A. Szabo. *J. Phys. Chem. B*, 119:9029–9037, 2015.

[154] G. H. Weiss. *Adv. Chem. Phys.*, 13:1–18, 1967.

[155] Y. Saad. *Numerical methods for large eigenvalue problems.* SIAM, Philadelphia, PA, 2011.

[156] D. S. Watkins. *The matrix eigenvalue problem: GR and Krylov subspace methods.* SIAM, Philadelphia, PA, second edition, 2007.

[157] Y. Saad. *Math. Comput.*, 42:567–588, 1984.

[158] A. L. Hageman and D. M. Young. *Applied iterative methods.* Academic Press, New York, 1981.

[159] C. C. Paige and M. A. Saunders. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[160] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. *SIAM J. Numer. Anal.*, 20:345–357, 1983.

[161] C. Vuik. *ESAIM Proc.*, 63:1–43, 2018.

[162] R. G. Grimes, J. G. Lewis, and H. D. Simon. *SIAM J. Matrix Anal. Appl.*, 15:228–272, 1994.

[163] A. Hadjidimos. *J. Comput. Appl. Math*, 123:177–199, 2000.

[164] Y. Saad. *Math. Comput.*, 37:105–126, 1981.

[165] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, PA, second edition, 2003.

[166] G. W. Stewart. *Numer. Math.*, 25:123–136, 1976.

[167] W. E. Arnoldi. *Quart. Appl. Math.*, 9:17–29, 1951.

[168] Y. Saad. *Linear Algebra Appl.*, 34:269–295, 1980.

[169] C. C. Paige. *Linear Algebra Appl.*, 34:235–258, 1980.

[170] Z.-X. Jia and L. Elsner. *J. Comput. Math.*, 18:265–276, 2000.

[171] Y. Saad and M. H. Schultz. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[172] R. B. Lehoucq and D. C. Sorensen. *SIAM J. Matrix Anal. Appl.*, 17:789–821, 1996.

[173] R. B. Morgan. *SIAM J. Sci. Comput.*, 24:20–37, 2002.

[174] H. D. Simon. *Linear Algebra Appl.*, 61:101–131, 1984.

[175] L. Shampine and C. Gear. *SIAM Rev.*, 21:1–17, 1979.

[176] C. W. Gear and I. G. Kevrekidis. *SIAM J. Sci. Comput.*, 24:1091–1106, 2003.

[177] T. J. Frankcombe and S. C. Smith. *J. Theor. Comput. Chem.*, 2:171–191, 2003.

[178] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh. *SIAM J. Sci. Stat. Comput.*, 10:1038–1051, 1989.

[179] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold. *SIAM J. Sci. Stat. Comput.*, 15:1467–1488, 1994.

[180] C. W. Gear and Y. Saad. *SIAM J. Sci. Stat. Comput.*, 4, 1983.

[181] P. N. Brown and A. C. Hindmarsh. *Appl. Math. Comput.*, 31:40–91, 1989.

[182] T. J. Frankcombe and S. C. Smith. *J. Chem. Phys.*, 119:12741–12748, 2003.

[183] M. K. Cameron and E. Vanden-Eijnden. *J. Stat. Phys.*, 156:427–454, 2014.

[184] C. Dellago, P. G. Bolhuis, and P. L. Geissler. *Adv. Chem. Phys.*, 123:1–78, 2002.

[185] W. E, W. Ren, and E. Vanden-Eijnden. *Chem. Phys. Lett.*, 413:242–247, 2005.

[186] A. M. Berezhkovskii and A. Szabo. *J. Chem. Phys.*, 150:054106, 2019.

[187] Q. Li, B. Lin, and W. Ren. *J. Chem. Phys.*, 151:054112, 2019.

[188] P. Lenz, B. Zagrovic, J. Shapiro, and V. Pande. *J. Chem. Phys.*, 120:6769–6778, 2004.

[189] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma. *J. Comput. Biol.*, 10:257–281, 2003.

[190] N. Singhal, C. D. Snow, and V. S. Pande. *J. Chem. Phys.*, 121:415–425, 2004.

[191] D. J. Sharpe and D. J. Wales. *Phys. Rev. E*, 2021. (in press).

[192] T. D. Swinburne and D. J. Wales. *J. Chem. Theory Comput.*, 16:2661–2679, 2020.

[193] A. Berezhkovski, G. Hummer, and A. Szabo. *J. Chem. Phys.*, 130:205102, 2009.

[194] A. M. Berezhkovskii, R. D. Murphy, and N.-V. Buchete. *J. Chem. Phys.*, 138:036101, 2013.

[195] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. S. Shakhnovich. *J. Chem. Phys.*, 108:334–350, 1998.

[196] G. Hummer. *J. Chem. Phys.*, 120:516–523, 2004.

[197] A. Berezhkovskii and A. Szabo. *J. Chem. Phys.*, 125:104902, 2006.

[198] C. D. Snow, Y. M. Rhee, and V. S. Pande. *Biophys. J.*, 91:14–24, 2006.

[199] D. Antoniou and S. D. Schwatz. *J. Chem. Phys.*, 130:151103, 2009.

[200] R. B. Best and G. Hummer. *Proc. Natl. Acad. Sci. USA*, 102:6732–6737, 2005.

[201] D. J. Sharpe and D. J. Wales. *Phys. Rev. E*, 2021. (in press).

[202] W. Grassmann and D. A. Stanford. In W. Grassmann, editor, *Computational Probability*, pages 153–203. Springer, New York, 2000.

[203] E. Seneta. *SIAM J. Matrix Anal. Appl.*, 19:556–563, 1998.

[204] D. T. Crommelin and E. Vanden-Eijnden. *J. Comput. Phys.*, 217:782–805, 2006.

[205] T. J. Frankcombe and S. C. Smith. *J. Chem. Phys.*, 119:12729–12740, 2003.

[206] A. Milias-Argeitis and J. Lygeros. *J. Chem. Phys.*, 138:184109, 2013.

[207] C. D. Meyer. *Linear Algebra Appl.*, 114-115:69–94, 1989.

[208] T. J. Sheskin. *Oper. Res.*, 33:228–235, 1985.

[209] R. B. Mattingly. *ORSA J. Comp.*, 7:117–124, 1995.

[210] P. J. Courtouis and S. P. *Linear Algebra Appl.*, 76:59–70, 1986.

[211] E. Meerbach, C. Schütte, and A. Fischer. *Linear Algebra Appl.*, 398:141–160, 2005.

[212] W.-L. Cao and W. J. Stewart. *J. Assoc. Comp. Mach.*, 32:702–719, 1985.

[213] P. J. Schweitzer. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 63–87. Marcel Dekker, New York, 1991.

[214] M. Haviv. *SIAM J. Numer. Anal.*, 22:952–966, 1987.

[215] W. J. Stewart and W. Wu. *ORSA J. Comp.*, 4:336–350, 1992.

[216] J. R. Koury, D. F. McAllister, and W. J. Stewart. *SIAM J. Alg. Discr. Meth.*, 5:164–186, 1984.

[217] W. J. Stewart and A. Touzene. Technical Report RR 921I, Institute IMAG, Grenoble, France, 1993.

[218] P. J. Schweitzer and K. W. Kindle. *Appl. Math. Comput.*, 18:313–353, 1986.

[219] T. Dayar and W. J. Stewart. *SIAM J. Sci. Comput.*, 17:287–303, 1996.

[220] D. P. Heyman. *SIAM J. Alg. Discr. Meth.*, 8:226–232, 1987.

[221] D. P. Heyman. *SIAM J. Matrix Anal. Appl.*, 16:954–963, 1995.

[222] D. P. Heyman and D. P. O'Leary. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 151–161. Springer, New York, 1995.

[223] T. J. Sheskin. *Int. J. Math. Educ. Sci. Technol.*, 30:167–185, 1999.

[224] K. A. Fichthorn and W. H. Weinberg. *J. Chem. Phys.*, 95:1090–1096, 1991.

[225] S. Fortunato. *Phys. Rep.*, 486:75–174, 2010.

[226] A. Chatterjee and A. F. Voter. *J. Chem. Phys.*, 132:194101, 2010.

[227] J. W. Demmel, N. J. Higham, and R. S. Schrieber. *Num. Linear Algebr. Appl.*, 2:173–190, 1995.

[228] J. D. Muñoz, M. A. Novotny, and S. J. Mitchell. *Phys. Rev. E*, 67:026101, 2003.

[229] B. Puchala, M. L. Falk, and K. Garikipati. *J. Chem. Phys.*, 132:134104, 2010.

[230] K. A. Fichthorn and Y. Lin. *J. Chem. Phys.*, 138:164104, 2013.

[231] L. Martini, A. Kells, R. Covino, G. Hummer, N.-V. Buchete, and E. Rosta. *Phys. Rev. X*, 7:031060, 2017.

[232] F. Noé and F. Nüske. *Multiscale Model. Simul.*, 11:635–655, 2013.

[233] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé. *J. Chem. Theory Comput.*, 10:1739–1752, 2014.

[234] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. (unpublished), 2020.

Chapter 2

Graph transformation and shortest paths algorithms for finite Markov chains

The graph transformation (GT) algorithm robustly computes the mean first passage time to an absorbing state in a finite Markov chain. Here we present a concise overview of the iterative and block formulations of the GT procedure and generalize the GT formalism to the case of any path property that is a sum of contributions from individual transitions. In particular, we examine the path action, which directly relates to the path probability, and analyze the first passage path ensemble for a model Markov chain that is metastable and therefore numerically challenging. We compare the mean first passage path action, obtained using GT, with the full path action probability distribution simulated efficiently using kinetic path sampling, and with values for the highest-probability paths determined by the recursive enumeration algorithm (REA). In Markov chains representing realistic dynamical processes, the probability distributions of first passage path properties are typically fat-tailed and therefore difficult to converge by sampling, which motivates the use of exact and numerically stable approaches to compute the expectation. We find that the kinetic relevance of the set of highest-probability paths depends strongly on the metastability of the Markov chain, and so the properties of the dominant first passage paths may be unrepresentative of the global dynamics. Use of a global measure for edge costs in the REA, based on net productive fluxes, allows the total reactive flux to be decomposed into a finite set of contributions from simple flux-paths. By considering transition flux-paths, a detailed quantitative analysis of the relative importance of competing dynamical processes is possible even in the metastable regime.

2.1 Introduction

Diverse stochastic phenomena are conveniently represented by finite Markov chains;[1] probabilistic network models for which the future dynamics depend only on the currently occupied state and not on the prior history of the trajectory.[2] Discrete-time Markov chains[3] (DTMCs) are commonly estimated from trajectory data on a continuous potential energy landscape in the Markov State Model (MSM) framework.[4–8] In a complementary approach, continuous-time Markov chains[1] (CTMCs) can be mapped from a potential energy landscape by geometry optimization[9] of local stationary points in the discrete path sampling (DPS) framework.[10–13] CTMCs with a countably-infinite state space[14,15] are widely used to represent the number of each species in population dynamics[16–18] processes such as chemical and biochemical reaction cycles,[19–24] and can be transformed to finite Markov chains with negligible error by truncating the state space.[25,26]

In previous work we have considered a discrete-state Markov reward process[27] on a finite state space $\mathcal{S}$, where individual $i \leftarrow j$ transitions in the Markov chain are associated with a reward R_{ij} that depends only on the identity of the currently occupied node j and not the next node i (i.e. $R_{ij} \equiv R_j \; \forall \, i$). The graph transformation (GT) algorithm[28–32] can be used to compute the average reward along first passage[33,34] trajectories from an initial set of nodes $\mathcal{B}$ to an absorbing set of nodes $\mathcal{A}$ in this case. An important example of a path property of this kind is the path time. The mean time elapsed along an $\mathcal{A} \leftarrow \mathcal{B}$ first passage path[33,34] in a CTMC[2] is a sum of mean waiting times τ_j for transitions from nodes j in the path.[30] For a DTMC,[3] the fixed lag time associated with transitions is uniform for all nodes, $\tau_j \equiv \tau \; \forall \, j$.[35] The $\mathcal{A} \leftarrow \mathcal{B}$ mean first passage time (MFPT),[36] $\mathcal{T}_{\mathcal{AB}}$, is a sum of path times, taken over all possible first passage paths, weighted by the associated path probabilities.[35]

The GT algorithm is numerically stable, and therefore valuable in many practical applications.[31] Markov chains representing realistic dynamical processes are frequently observed to encompass a separation of characteristic timescales, and the corresponding transition probability or rate matrix is therefore ill-conditioned.[37–48] This feature arises in Markovian networks constructed using the MSM and DPS frameworks because of the exponential sensitivity of estimated transition probabilities or rates to the structure of the underlying energy landscape.[49,50] Metastability also emerges in reaction networks where the rate constants for alternative competing reactions are disparate.[19–24] Markov chains that harbour metastable communities of nodes pose numerical challenges, since dynamical simulations become unfeasibly inefficient[51,52] and conventional linear algebra methods lead to a severe propagation of numerical error. The GT algorithm provides a powerful alternative approach to compute MFPTs in high-dimensional and ill-conditioned Markov chains.[31]

In the present contribution we generalize the GT algorithm[28–32] to the case of rewards R_{ij} that are different for transitions to alternative destination nodes i from the currently occupied node j. Relevant examples of such rewards include the path action,[53] which directly relates to the path probability, and the entropy flow,[54] which quantifies the reversibility of a trajectory.[55,56] Although they are not dynamical observables, the average path action and entropy flow have rigorous interpretations, and the probability distributions for these path properties yield important insight into the characteristics of a Markov chain. For instance, the expectation of the path action is the Shannon entropy[57] associated with the ensemble of first passage paths.[58–62] A similar quantity is employed in the maximum caliber and maximum entropy frameworks as the objective function in a variational principle to estimate Markovian transition probabilities or rates for a discrete set of states, given constraints on the stationary distribution and additional global dynamical information.[63,64] The entropy flow is a central quantity in stochastic thermodynamics,[65] since the average entropy production is governed by an integral fluctuation theorem.[66] Previous analytical results considering paths in Markov chains weighted by arbitrary rewards are limited and do not lend themselves to the design of computational procedures that have the desirable scalability and stability of our generalized GT algorithm.[67]

Following derivations of the expected rewards for paths on renormalized Markov chains (Sec. 2.2.2), and of the iterative and block formulations for the generalized GT algorithm (Sec. 2.2.3), we compute the mean first passage path action for a model metastable Markov chain (Sec. 2.3). We compare the expectation for the path action with the full probability distribution simulated efficiently using kinetic path sampling,[68,69] and with the values for the highest-probability paths determined by the recursive enumeration algorithm (REA) (Sec. 2.2.4).[70] We demonstrate that the probability distributions of first passage path properties are typically fat-tailed, and that the fraction of the total probability flux to the absorbing state accounted for by the dominant first passage paths depends strongly on the metastability of the Markov chain. Hence, it is often challenging to obtain an accurate numerical estimate for the expectation of a first passage path property by sampling trajectories, and it may be unfeasible to converge the pathwise sum for the expectation using shortest paths algorithms. We propose an alternative shortest paths analysis to provide quantitative information on the relative importance of alternative $\mathcal{A} \leftarrow \mathcal{B}$ processes, using edge costs in the REA that are based on net reactive fluxes.[71] This formulation allows the total $\mathcal{A} \leftarrow \mathcal{B}$ reactive flux to be decomposed into a sum of contributions from a *finite* set of simple flux-paths (Sec. 2.2.5). We find that the total reactive flux becomes increasingly localized among a small subset of transition flux-paths with increasing metastability.

2.2 Theory

2.2.1 Mathematical definitions

We consider arbitrary discrete-[3] and continuous-time[72] finite Markov chains. A DTMC is parameterized by $i \leftarrow j$ transition probabilities $T_{ij}(\tau)$ for a fixed time step τ. A CTMC is parameterized by $i \leftarrow j \neq i$ transition rates K_{ij}.[1] Equivalently, a CTMC can be specified by a branching probability matrix[73] $\mathbf{P}$ with off-diagonal elements $P_{ij} = K_{ij}/\sum_{\gamma \neq j} K_{\gamma j}$ and diagonal elements $P_{jj} = 0$, and a vector of mean waiting times for transitions from nodes j, with elements $\tau_j = 1/\sum_{\gamma \neq j} K_{\gamma j}$. In the present work, we denote the stochastic matrix of a Markov chain ($\mathbf{T}(\tau)$ for a DTMC and $\mathbf{P}$ for a CTMC) by $\mathbf{T}$ for generality, as in Chapter 1. We denote the state space of the Markov chain (i.e. the complete set of nodes) as $\mathcal{S}$, and consider two disjoint sets of endpoint nodes $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A} \cup \mathcal{B} \subseteq \mathcal{S}$, which are the target and initial states, respectively.

Let the $i \leftarrow j$ transition be associated with a reward R_{ij}, which does not modify the dynamics but instead is used to assign a weight $\mathcal{R}[\xi]$ to paths ξ. The total reward along a particular $\mathcal{A} \leftarrow \mathcal{B}$ first passage path $\xi \equiv \{a \in \mathcal{A} \leftarrow i_n \leftarrow i_{n-1} \leftarrow \cdots \leftarrow i_1 \leftarrow b \in \mathcal{B}\}$, where $i_1, \ldots, i_n \notin \mathcal{A}$, is a sum of contributions from individual transitions along ξ, $\mathcal{R}[\xi] = \sum_{(i \leftarrow j) \in \xi} R_{ij}$. An important example of a path property of this type is the path action,[57] $-\ln \mathcal{W}[\xi] = -\sum_{(i \leftarrow j) \in \xi} \ln T_{ij}$. Here, $\mathcal{W}[\xi]$ denotes the product of transition probabilities along the path ξ, i.e. the path weight.[30] The path probability $\mathcal{P}[\xi]$ is equal to this probability weighted by the probability $p_b(0)$ of starting at the initial node $b \in \mathcal{B}$ of the path ξ, $\mathcal{P}[\xi] = p_b(0)\mathcal{W}[\xi]$.[74] Another tangible example of a reward is the entropy flow.[54] In discrete time, the path entropy flow is[75] $\mathcal{S}[\xi] = \sum_{(i \leftarrow j) \in \xi} \ln(T_{ji}/T_{ij})$ (in units of the Boltzmann constant), and in continuous time[76] $\mathcal{S}[\xi] = \sum_{(i \leftarrow j) \in \xi} \ln(K_{ji}/K_{ij})$. The numerical results presented in Sec. 2.3 are concerned with the path action.

In addition to rewards $\mathcal{R}[\xi]$ along individual trajectories ξ, we are interested in the ensemble average reward $\mathcal{R}_{\mathcal{AB}}$, considering all trajectories that start in the state $\mathcal{B}$ and are absorbed upon hitting the state $\mathcal{A}$, including revisits to $\mathcal{B}$.[30] We refer to this set of trajectories as the first passage path ensemble[58–62] (FPPE) and $\mathcal{R}_{\mathcal{AB}}$ as the mean first passage reward (MFPR).

2.2.2 Expected rewards for individual paths on censored Markov chains

Our generalized GT algorithm to calculate the MFPR (Sec. 2.2.3) utilizes the concept of a *censored Markov chain*,[16,77–82] introduced in Chapter 1. We begin by considering the effect of renormalization to eliminate a single node[83] n on the rewards associated with paths on

the resulting censored network. For pairs of nodes i and j for which there exist $i \leftarrow n$ and $n \leftarrow j$ transitions, it is possible to define renormalized transition probabilities T'_{ij} that account for the average contribution of $i \leftarrow j$ transitions proceeding via the eliminated node n. Specifically, the total probability of the $i \leftarrow j$ transition in the renormalized Markov chain is[30]

$$T'_{ij} = T_{ij} + \frac{T_{in}T_{nj}}{1 - T_{nn}}. \tag{2.1}$$

Here, the first contribution corresponds to direct $i \leftarrow j$ transitions on the original network, and the second contribution corresponds to indirect ('round-trip') transitions, $i \leftarrow n \leftarrow \ldots \leftarrow n \leftarrow j$, where the eliminated node n is visited an arbitrary number of times.[68] The updated transition probabilities of Eq. 2.1 naturally yield a new stochastic matrix without requiring explicit normalization.[30]

We wish to derive the renormalized reward R'_{ij} associated with the $i \leftarrow j$ transition in the censored Markov chain for which the n-th node is eliminated. We must account for the fact that the expected reward associated with $i \leftarrow j$ transitions proceeding indirectly, via the eliminated (censored) node n, is different from the reward for the direct $i \leftarrow j$ transition, which does not involve the censored node. The conditional probability that a $i \leftarrow j$ transition is direct is given by T_{ij}/T'_{ij} (*cf.* Eq. 2.1), and the reward for the direct transition is simply R_{ij}. The contribution to the renormalized $i \leftarrow j$ reward arising from indirect transitions via node n is more complicated. On average, a trajectory at node n will transition from n a total of $(1 - T_{nn})^{-1}$ times before leaving n, including the final transition to escape from n.[3] Thus the expected reward for an indirect $i \leftarrow j$ transition is

$$\langle R_{in}^{\text{indir}} \rangle = R_{in} + R_{nj} + R_{nn}\left(\frac{1}{1 - T_{nn}} - 1\right). \tag{2.2}$$

The average reward associated with the $i \leftarrow j$ transition for the renormalized (censored) network is the sum of direct and average indirect rewards weighted by the conditional probabilities of direct and indirect $i \leftarrow j$ transitions, respectively,

$$R'_{ij} = \frac{1}{T'_{ij}}\left(T_{ij}R_{ij} + \frac{T_{in}T_{nj}}{1 - T_{nn}}\langle R_{in}^{\text{indir}} \rangle\right). \tag{2.3}$$

Here T'_{ij} is given by Eq. 2.1 and $\langle R_{in}^{\text{indir}} \rangle$ by Eq. 2.2. Eq. 2.3 conserves the average rewards $\mathcal{R}[\xi]$ for *all* individual paths ξ, with arbitrary initial and final nodes, on a censored Markov chain.[77] The rewards associated with trajectories on the censored Markov chain are strictly an *expectation* with respect to the contributions of path segments that visit censored nodes.[77] Reducing the dimensionality of Markov chains by renormalization provides a strategy to

facilitate the sampling of trajectories,[84] and Eq. 2.3 allows for the probability distributions of path rewards on the transformed network to be estimated within this framework.

Our result in Eq. 2.3 can also be exploited to compute the overall $\mathcal{A} \leftarrow \mathcal{B}$ MFPR. Following elimination of all nodes of the set $(\mathcal{A} \cup b)^{\mathsf{c}}$ using renormalization of the transition probabilities and rewards (Eqs. 2.1 and 2.3, respectively), where $b \in \mathcal{B}$ is a single node of the initial state, the average reward associated with the ensemble of $\mathcal{A} \leftarrow b$ trajectories is

$$\mathcal{R}_{\mathcal{A}b} = \left(\frac{1}{1 - T'_{bb}} - 1\right) R'_{bb} + \sum_{a \in \mathcal{A}} \frac{T'_{ab} R'_{ab}}{1 - T'_{bb}}. \tag{2.4}$$

Here, we have again used the result that the expected number of transitions from node b before hitting a different node is $(1 - T'_{bb})^{-1}$, all of which except the final transition are $b \leftarrow b$ self-loop transitions, and the probability that the node $a \in \mathcal{A}$ is hit upon leaving b is $T'_{ab}/(1 - T'_{bb})$. The average reward for paths of the $\mathcal{A} \leftarrow \mathcal{B}$ FPPE,[58–62] $\mathcal{R}_{\mathcal{A}\mathcal{B}}$, is simply a weighted average of rewards $\mathcal{R}_{\mathcal{A}b}$ (Eq. 2.4) with respect to the initial occupation probability distribution $p_b(0)$ for nodes $b \in \mathcal{B}$.[31]

2.2.3 Mean first passage reward computed using a generalized graph transformation procedure

Let the set of transient (nonabsorbing) nodes of the Markov chain be denoted $\mathcal{Q}$, and the complete set of nodes as $\mathcal{S} \equiv \mathcal{Q} \cup \mathcal{A}$. The results of Sec. 2.2.2 demonstrate that the $\mathcal{A} \leftarrow \mathcal{B}$ MFPR can be computed by iteratively renormalizing the elements of a reward matrix $\mathbf{R}$ that is initially of dimensions $|\mathcal{Q}| \times |\mathcal{Q}|$, and from which the n-th row and column are removed when eliminating node n. In fact, it is only necessary to consider an initial $|\mathcal{Q}|$-dimensional vector of mean rewards for transitions from the transient nodes in order to compute the MFPR to the absorbing state, as we now show.

For a general Markov chain, the sum of path probabilities to the absorbing state $\mathcal{A}$ from a transient node $q \in \mathcal{Q}$ is given by the component $\left[\mathbf{1}_{\mathcal{A}}^{\top} \mathbf{T}_{\mathcal{A}\mathcal{Q}} \mathbf{N}_{\mathcal{Q}\mathcal{Q}}\right]_q$, and is unity for all q.[85] Here, $\mathbf{N}_{\mathcal{Q}\mathcal{Q}} = (\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \mathbf{T}_{\mathcal{Q}\mathcal{Q}})^{-1}$ is the fundamental matrix[3] associated with the absorbing Markov chain parameterized by the substochastic matrix $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$, for transitions between nodes of the set $\mathcal{Q}$, $\mathbf{1}_{\mathcal{A}}$ is a column vector of dimension $|\mathcal{A}|$ with unit entries, and $\mathbf{I}_{\mathcal{Q}\mathcal{Q}}$ denotes the $|\mathcal{Q}|$-dimensional identity matrix.

To produce a general formula for the MFPR from the set of transient nodes to the absorbing state, $\mathcal{R}_{\mathcal{A}\mathcal{Q}}$, we introduce the reweighted $i \leftarrow j$ transition probabilities $\widehat{T}_{ij} = T_{ij} \exp(\zeta R_{ij})$. This is the same mathematical trick that we used to prove the GT algorithm

in Chapter 1. In component form we have $\partial \widehat{T}_{ij}/\partial\zeta\big|_{\zeta=0} = T_{ij}R_{ij}$, and so

$$\left.\frac{\partial \widehat{\mathbf{T}}_{\mathcal{A}\mathcal{Q}}}{\partial\zeta}\right|_{\zeta=0} = \mathbf{T}_{\mathcal{A}\mathcal{Q}} \circ \mathbf{R}_{\mathcal{A}\mathcal{Q}}, \tag{2.5}$$

where $\circ$ denotes the elementwise (Hadamard) product, which we write as $\mathbf{C}_{\mathcal{A}\mathcal{Q}}$. Here, $\mathbf{T}_{\mathcal{A}\mathcal{Q}}$ is the substochastic matrix for transitions from transient to absorbing nodes, with dimensions $|\mathcal{A}| \times |\mathcal{Q}|$, and $\mathbf{R}_{\mathcal{A}\mathcal{Q}}$ is the corresponding matrix of associated rewards for the transitions.

The $\mathcal{A} \leftarrow \mathcal{Q}$ MFPR can be computed from the fundamental matrix $\mathbf{N}_{\mathcal{Q}\mathcal{Q}}$ of the absorbing Markov chain as

$$\mathcal{R}_{\mathcal{A}\mathcal{Q}} = \frac{\partial}{\partial\zeta}\mathbf{1}_{\mathcal{A}}^{\top}\widehat{\mathbf{T}}_{\mathcal{A}\mathcal{Q}}\widehat{\mathbf{N}}_{\mathcal{Q}\mathcal{Q}}\Big|_{\zeta=0}\mathbf{p}_{\mathcal{Q}}(0), \tag{2.6}$$

where $\mathbf{p}_{\mathcal{Q}}(0)$ is the initial occupation probability distribution within $\mathcal{Q}$. We require the derivative

$$\begin{aligned}\left.\frac{\partial \widehat{\mathbf{N}}_{\mathcal{Q}\mathcal{Q}}}{\partial\zeta}\right|_{\zeta=0} &= \mathbf{N}_{\mathcal{Q}\mathcal{Q}}\left.\frac{\partial \widehat{\mathbf{T}}_{\mathcal{Q}\mathcal{Q}}}{\partial\zeta}\right|_{\zeta=0}\mathbf{N}_{\mathcal{Q}\mathcal{Q}} \\ &= \mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{C}_{\mathcal{Q}\mathcal{Q}}\mathbf{N}_{\mathcal{Q}\mathcal{Q}},\end{aligned} \tag{2.7}$$

which gives

$$\begin{aligned}\mathcal{R}_{\mathcal{A}\mathcal{Q}} &= \mathbf{1}_{\mathcal{A}}^{\top}\left(\mathbf{C}_{\mathcal{A}\mathcal{Q}} + \mathbf{T}_{\mathcal{A}\mathcal{Q}}\mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{C}_{\mathcal{Q}\mathcal{Q}}\right)\mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{p}_{\mathcal{Q}}(0) \\ &= \left(\mathbf{1}_{\mathcal{A}}^{\top}\mathbf{C}_{\mathcal{A}\mathcal{Q}} + \mathbf{1}_{\mathcal{Q}}^{\top}\mathbf{C}_{\mathcal{Q}\mathcal{Q}}\right)\mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{p}_{\mathcal{Q}}(0) \\ &= \mathbf{r}_{\mathcal{Q}}^{\top}\mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{p}_{\mathcal{Q}}(0),\end{aligned} \tag{2.8}$$

where the q-th component of the column vector $\mathbf{r}_{\mathcal{Q}}$ is the average reward for transitions from node q:

$$[\mathbf{r}_{\mathcal{Q}}]_q = \sum_{\gamma} T_{\gamma q}R_{\gamma q}. \tag{2.9}$$

For comparison, the corresponding formula for the MFPT (derived in Chapter 1) is[85] $\mathcal{T}_{\mathcal{A}\mathcal{Q}} = \boldsymbol{\tau}^{\top}\mathbf{N}_{\mathcal{Q}\mathcal{Q}}\mathbf{p}_{\mathcal{Q}}(0)$, where $\boldsymbol{\tau}$ is the vector of mean waiting times (for a CTMC) or lag times (for a DTMC) for transitions from the nodes. Eq. 2.8 also demonstrates that $[\mathbf{N}_{\mathcal{Q}\mathcal{Q}}]_{ij}$ is the expected number of times the i-th node is visited prior to absorption for first passage paths initialized from the j-th node.[3] We consider the matrix $\mathbf{N}_{\mathcal{Q}\mathcal{Q}}$ in detail in Chapter 3. Using Eqs. 2.8 and 2.9, the $\mathcal{A} \leftarrow q$ MFPRs for all transient nodes $q \in \mathcal{Q}$ can be computed

simultaneously by inversion of a matrix with dimensions $|\mathcal{Q}| \times |\mathcal{Q}|$.

For Markov chains exhibiting metastability, the matrix inversion operation to compute $\mathbf{N}_{\mathcal{Q}\mathcal{Q}}$ is numerically unstable. We can instead iteratively eliminate blocks of one or more nodes to compute $\mathcal{A} \leftarrow \mathcal{B}$ MFPRs by renormalization of an average reward vector (Eq. 2.9) and a transition probability matrix. The set of initial nodes $\mathcal{B}$ forms a subset of the transient state, with the set of other (intervening) nodes denoted $\mathcal{I}$, i.e. $\mathcal{Q} \equiv \mathcal{B} \cup \mathcal{I}$. After eliminating nodes of the state $\mathcal{I}$, so that the Markov chain comprises only nodes of the set $\mathcal{B} \cup \mathcal{A}$, the corresponding path probabilities can be written as $\mathbf{1}_{\mathcal{A}}^{\top} \mathbf{T}_{\mathcal{A}\mathcal{B}}^{\mathcal{I}} \mathbf{N}_{\mathcal{B}\mathcal{B}}^{\mathcal{I}}$, where[32,77,85]

$$\mathbf{T}_{\mathcal{A}\mathcal{B}}^{\mathcal{I}} = \mathbf{T}_{\mathcal{A}\mathcal{B}} + \mathbf{T}_{\mathcal{A}\mathcal{I}} \mathbf{N}_{\mathcal{I}\mathcal{I}} \mathbf{T}_{\mathcal{I}\mathcal{B}}, \tag{2.10a}$$

$$\mathbf{N}_{\mathcal{B}\mathcal{B}}^{\mathcal{I}} = (\mathbf{I}_{\mathcal{B}\mathcal{B}} - \mathbf{T}_{\mathcal{B}\mathcal{B}}^{\mathcal{I}})^{-1}. \tag{2.10b}$$

Here, we have used the superscript $\mathcal{I}$ to indicate that nodes of the set $\mathcal{I}$ have been eliminated by renormalization. Introducing the reweighted transition probabilities $\widehat{T}_{ij}$ and following a derivation analogous to that for Eq. 2.8 yields the following expression for the $\mathcal{A} \leftarrow \mathcal{B}$ MFPR:

$$\mathcal{R}_{\mathcal{A}\mathcal{B}} = \left(\mathbf{r}_{\mathcal{B}}^{\top} + \mathbf{r}_{\mathcal{I}}^{\top} \mathbf{N}_{\mathcal{I}\mathcal{I}} \mathbf{T}_{\mathcal{I}\mathcal{B}} \right) \mathbf{N}_{\mathcal{B}\mathcal{B}}^{\mathcal{I}} \mathbf{p}_{\mathcal{B}}(0). \tag{2.11}$$

Since the initial occupation probability distribution is localized within $\mathcal{B}$, the MFPR will be conserved if we iteratively eliminate blocks of nodes $\mathcal{N} \subseteq \mathcal{I}$, renormalizing the probabilities for transitions from nodes in the set $\mathcal{Q}' \equiv \mathcal{Q} \setminus \mathcal{N}$ according to the usual GT formula (*cf.* Eq. 2.10a),

$$\mathbf{T}_{\mathcal{S}\mathcal{Q}'}^{\mathcal{N}} = \mathbf{T}_{\mathcal{S}\mathcal{Q}'} + \mathbf{T}_{\mathcal{S}\mathcal{N}} \mathbf{N}_{\mathcal{N}\mathcal{N}} \mathbf{T}_{\mathcal{N}\mathcal{Q}'}, \tag{2.12}$$

and updating the average rewards according to

$$\mathbf{r}_{\mathcal{Q}'}^{\mathcal{N}\,\top} = \mathbf{r}_{\mathcal{Q}'}^{\top} + \mathbf{r}_{\mathcal{N}}^{\top} \mathbf{N}_{\mathcal{N}\mathcal{N}} \mathbf{T}_{\mathcal{N}\mathcal{Q}'}. \tag{2.13}$$

Eq. 2.12 is the block analogue of Eq. 2.1. That is, Eq. 2.12 yields the same renormalized stochastic matrix as the repeated application of Eq. 2.1 to iteratively eliminate the nodes of the set $\mathcal{N}$ in any order. The generalized GT procedure to compute the $\mathcal{A} \leftarrow \mathcal{B}$ MFPR based on Eqs. 2.12 and 2.13 is both numerically stable and efficient if nodes in blocks $\mathcal{N}$ to be eliminated simultaneously belong to the same metastable community.[32,85] The communities can be determined *a priori* by an appropriate clustering algorithm.[52,86]

Eq. 2.13 is analogous to the result for the renormalized waiting times that preserve the MFPT,[30,85] with the mean rewards for transitions from transient nodes $q \in \mathcal{Q}$ in place of the mean waiting (or lag) times. Eliminating a single node $n \in \mathcal{I}$ by renormalization, Eq. 2.12

reduces to Eq. 2.1, and Eq. 2.13 reduces to

$$[\mathbf{r}^n_{\mathcal{Q}'}]_q = [\mathbf{r}_{\mathcal{Q}'}]_q + \frac{[\mathbf{r}_{\mathcal{Q}'}]_n T_{nq}}{1 - T_{nn}}. \tag{2.14}$$

Exploiting the relation $1 - T_{nn} = \sum_{\gamma \neq n} T_{\gamma n}$ when $T_{nn} \to 1$ avoids the propagation of significant roundoff error in the finite precision arithmetic, and this algorithm is numerically stable.[37–47] The time complexity of the iterative procedure depends on the average degree of nodes and on the heterogeneity of the degree distribution,[29,68] and varies between $\mathcal{O}(|\mathcal{Q}|^3)$ and $\mathcal{O}(|\mathcal{Q}|^4)$.

2.2.4 Recursive enumeration algorithm

Formally, the expected $\mathcal{A} \leftarrow \mathcal{B}$ reward is a sum of contributions $\mathcal{R}[\xi]$ from all paths ξ of the first passage path ensemble[35,57,62]

$$\mathcal{R}_{\mathcal{A}\mathcal{B}} = \sum_{\xi \in \{\mathcal{A} \leftarrow \mathcal{B}\}} p_b(0) \mathcal{W}[\xi] \mathcal{R}[\xi]. \tag{2.15}$$

The weighted sum in Eq. 2.15 has an infinite number of terms for Markov chains featuring loops, but the contributions to the sum from paths related by additional traversals of a particular loop converge. The highest-probability first passage paths,[87] and their contribution to the $\mathcal{A} \leftarrow \mathcal{B}$ reward sum in Eq. 2.15, can be determined by a k shortest paths algorithm[88–90] where the $i \leftarrow j$ edge cost is $-\ln T_{ij}$, i.e. the contribution of the transition to the path action.[74,91] For DTMCs, self-loop transitions for nodes can be eliminated by renormalization using $T_{ij} \Rightarrow T_{ij}/(1 - T_{jj})\ \forall\, i \neq j$,[30] while preserving the MFPR using $R_{ij} \Rightarrow R_{ij} + R_{jj}[(1 - T_{jj})^{-1} - 1]$ (see Sec. 2.2.2).

For Markov chains with metastable states, successive shortest paths tend to differ by small modifications, such as a single additional loop traversal or a small segment of the path proceeding via a few alternative nodes.[74] We therefore choose to employ the recursive enumeration algorithm (REA) of Jiménez and Marzal,[70] which is particularly efficient in cases where the set of k shortest paths share most of their nodes in common, and consist of a small fraction of the total number of nodes in the network.[70] The REA has worst-case time complexity $\mathcal{O}(E + kV \log(E/V))$ for a network comprising V nodes and E edges. The algorithm is empirically observed to outperform alternative general k shortest paths algorithms that have superior asymptotic time complexity, such as those of Eppstein [time complexity $\mathcal{O}(E+V+k \log k)$],[92,93] Azevedo *et al.* [time complexity $\mathcal{O}(kE)$],[94,95] and Martins, Pascoal, and dos Santos [time complexity $\mathcal{O}(kV \log V)$],[96–98] because the REA is associated with a comparatively small computational overhead.[70] In the following informal derivation

of the REA, we assume that all nodes are reachable from all nonabsorbing nodes, i.e. the set $\mathcal{S} \setminus \mathcal{A}$ is transient.

The REA formulates the general single-source node, single-sink node $a \in \mathcal{A} \leftarrow b \in \mathcal{B}$ k shortest paths problem as a set of Bellman equations,[99] which are solved recursively.[100] Let the k-th shortest path to node j be denoted $\xi^k(j)$, with associated cost $\mathcal{R}[\xi^k(j)]$, and the set of nodes with direct transitions to node j be denoted $\mathcal{D}(j)$. The first stage of the REA constructs the shortest path tree for the transitions from the single initial node to all alternative nodes using any appropriate procedure, such as Dijkstra's algorithm [worst case time complexity $\mathcal{O}(E + V \log V)$].[101–103] The REA exploits the fact that the k-th shortest path to node j can be written in the form $\xi^k(j) \equiv \xi^{k'}(i) \cup \{j \leftarrow i\}$, where $i \in \mathcal{D}(j)$ and $k' \leq k$. At the $(k-1)$-th iteration of the REA, the next (k-th) shortest path to the absorbing node a can therefore be selected from a list $\mathcal{M}(a)$ of such candidate paths. For each node $i \in \mathcal{D}(a)$, only the candidate path $\xi^{k'}(i) \cup \{a \leftarrow i\}$ for which $\xi^{k'}(i)$ has the lowest cost, and which has not already been chosen as a previous shortest path to the a-th node, needs to be considered. Hence, there are at most $|\mathcal{D}(a)|$ candidates for the next shortest path to node a; one for each node with a transition to a. Here, we have noted that ties may be broken arbitrarily, and that there is no more than one edge connecting any pair of nodes in a Markovian network. The REA maintains an array of candidate paths $\mathcal{M}(j)$ for all nodes j of the network, and an array of the k-th shortest paths to each node. At each iteration of the REA, a function is called to determine the next shortest path to the target node a. Recursive calls to this function are used to ensure that candidate paths are assigned, and that with lowest cost selected, for the shortest paths to preceding nodes, as required. The pseudocode for this procedure applied to determine the highest-probability first passage paths in a finite Markov chain, employing path costs $\mathcal{R}[\xi] \equiv -\ln \mathcal{W}[\xi]$, is presented in Algorithm 4.

2.2.5 Transition flux-paths

The transition probabilities are a local measure of the probability flux, and $-\ln T_{ij}$ represents only one possible choice for the $i \leftarrow j$ edge costs to extract dynamical information from shortest paths algorithms.[87] As we show in Sec. 2.3, the k shortest paths for this choice of edge costs are very closely related for Markov chains exhibiting metastability, and may together account for only a small proportion of the total $\mathcal{A} \leftarrow \mathcal{B}$ probability flux. Hence, for Markov chains exhibiting metastability, the number of shortest paths that can be feasibly determined by the REA is typically insufficient to converge the pathwise sum for the MFPR (Eq. 2.15).

An alternative choice, which may be especially useful in the metastable regime, is to use

edge costs that represent a global measure of the probability flux. The contribution of a transition path $\xi \equiv \{a \in \mathcal{A} \equiv i_{n+1} \leftarrow i_n \leftarrow \ldots \leftarrow i_1 \leftarrow b \in \mathcal{B}\}$, where $i_1, \ldots, i_n \notin \mathcal{A} \cup \mathcal{B}$, to the total reactive steady-state[104] flux $\mathcal{J}_{\mathcal{AB}}$ is[105]

$$\mathcal{J}[\xi] = f^+_{i_1 b} \prod_{k=1}^{n} \frac{f^+_{i_{k+1} i_k}}{f^+_{i_k}}. \tag{2.16}$$

Here,

$$f^+_{ij} = \begin{cases} \pi_j T_{ij}(q^+_i - q^+_j), & \text{if} \quad q^+_i > q^+_j, \\ 0, & \text{otherwise}, \end{cases} \tag{2.17}$$

is the net $\mathcal{A} \leftarrow \mathcal{B}$ reactive flux along the $i \leftarrow j$ edge,[71] $f^+_j = \sum_\gamma f^+_{\gamma j} = \sum_\gamma f^+_{j\gamma}$, π_j is the stationary probability for the j-th node,[83,106] and q^+_j is the forward committor probability for the j-th node,[107–109] i.e. the probability that a trajectory initialized at node j hits the target set of nodes $\mathcal{A}$ before hitting the initial state $\mathcal{B}$.[31] Eq. 2.16 implies the following definition for the $i \leftarrow j$ edge costs:

$$\mathcal{R}[\{i \leftarrow j\}] = \begin{cases} -\ln \frac{f^+_{ij}}{f^+_j}, & \text{if} \quad j \notin \mathcal{B}, \\ -\ln f^+_{ij}, & \text{otherwise}. \end{cases} \tag{2.18}$$

Unlike the local edge costs based on transition probabilities, i.e. $-\ln T_{ij}$, the global edge costs based on reactive fluxes (Eq. 2.18) represent the $\mathcal{A} \leftarrow \mathcal{B}$ transition mechanism when the system has reached a steady-state.[71] That is, f^+_{ij} (Eq. 2.17) is the net productive flux along the $i \leftarrow j$ edge for the *equilibrium* FPPE,[104] and therefore $\mathcal{J}_{\mathcal{AB}}$ is the total *steady-state* reactive $\mathcal{A} \leftarrow \mathcal{B}$ flux.[105] This stationary flux directly relates to the steady-state rate constant, which is the dynamical observable associated with the equilibrium FPPE.[85] Note that it is also possible to define net reactive fluxes along individual edges, and hence edge costs to determine flux-paths (*cf.* Eq. 2.18), for the *nonequilibrium* FPPE,[62,104] for which the MFPT is the associated dynamical observable. The two FPPEs are discussed in more detail in Chapter 3.

When using edge costs given by Eq. 2.18, the weighted network representing the Markov chain in the shortest paths algorithm has *unidirectional* edges, which are directed such that paths are forced to proceed productively through a series of isocommittor cuts in the network.[110] Hence, there are no loops in the network (i.e. all paths based on the choice of edge costs representing the net reactive flux are *simple*[89]), and the sum over transition flux-paths to obtain the total reactive flux, $\mathcal{J}_{\mathcal{AB}} = \sum_{\xi \in \{\mathcal{A} \leftarrow \mathcal{B}\}} \mathcal{J}[\xi]$, is finite. Since the set of nodes $\mathcal{S} \setminus \mathcal{A}$ is not transient when the edge costs are given by Eq. 2.18, the REA as presented

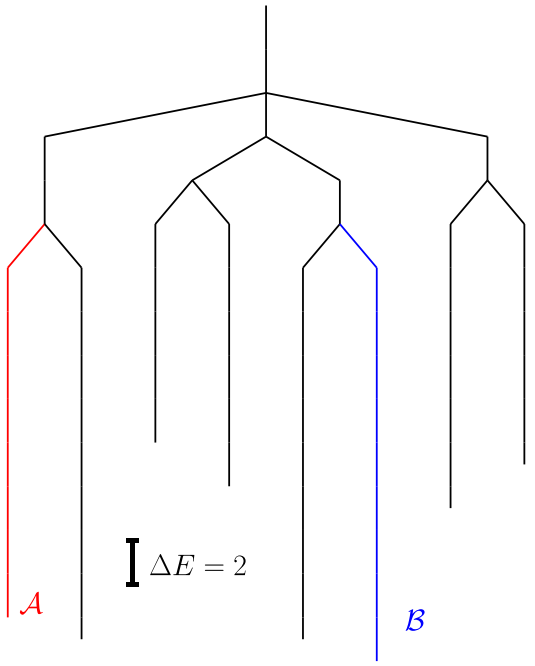


Figure 2.1: Disconnectivity graph[114] representing the energy landscape of the model eight-state CTMC, at a threshold energy increment of $\Delta E = 2$. The branches of the tree terminate at the energies of the corresponding nodes. A fork indicates that there exists a path between the corresponding sets of nodes via a highest-energy transition state that lies in between the neighbouring energy thresholds. The branches corresponding to the absorbing and initial nodes, which constitute the sets $\mathcal{A}$ and $\mathcal{B}$, are colored red and blue, respectively.

in Algorithm 4 must be adapted to account for the situation where a candidate path does not exist, as outlined in the original description of the REA (see Ref. 70). With this minor modification, which is also required when using local edge costs $-\ln T_{ij}$ for Markov chains that are reducible,[3] the REA can be used to obtain the complete set of reactive $\mathcal{A} \leftarrow \mathcal{B}$ flux-paths and their contributions to the total reactive flux.

For ill-conditioned Markov chains, evaluation of the edge costs in Eq. 2.18 is highly susceptible to numerical error.[37–47] Hence, in the metastable regime, the stationary probabilities $\{\pi_j\}$ and committor probabilities $\{q_j^+\}$ required in Eq. 2.17 should be determined by a numerically stable method. The $\{\pi_j\}$ can be computed robustly by the GTH algorithm[83,106] or an uncoupling-coupling procedure,[77,111–113] as described in Chapter 1. We report a state reduction algorithm for the computation of the $\{q_j^+\}$ in Chapter 3.[104]

2.3 Numerical results

We illustrate our methodology with results for the model eight-state CTMC considered in Ref. 115, for which the disconnectivity graph[114] is shown in Fig. 2.1. The system corresponds

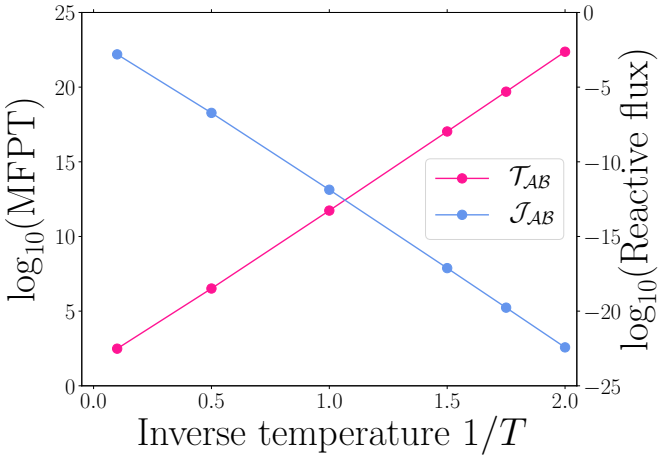


Figure 2.2: Variation in the $\mathcal{A} \leftarrow \mathcal{B}$ mean first passage time, $\mathcal{T}_{\mathcal{AB}}$, and the steady-state $\mathcal{A} \leftarrow \mathcal{B}$ reactive flux, $\mathcal{J}_{\mathcal{AB}}$, with inverse temperature, for the eight-state CTMC (Fig. 2.1). The CTMC is an effective two-state system, and hence the MFPT is dominated by the time to transition via the single largest energy barrier. Thus the MFPT approximately follows an Arrhenius law, and the height of the energy barrier associated with the slow bottleneck transition, which can be discerned from the disconnectivity graph in Fig. 2.1, can be inferred from the gradient of the above linear plot.

to a coarse-grained representation of an energy landscape,[57] with a discrete set of states connected via energy barriers. The internode transition rates have an Arrhenius form,[116] dependent on the temperature T,[35] and characterize the Markovian network dynamics in terms of branching probabilities[73] and mean waiting times for transitions from nodes.[30] A complete specification of this model system is given in Appendix 2.A.

The variation in the heights of energy barriers for transitions in the system induces a separation of timescales that increases with decreasing temperature. Both the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT and the steady-state $\mathcal{A} \leftarrow \mathcal{B}$ reactive flux vary by around twenty orders of magnitude in the range of inverse temperature $1/T$ from 0.1 to 2 (Fig. 2.2). At low temperatures, conventional linear algebra methods to determine MFPTs fail owing to numerical instability.[36] This CTMC therefore provides a useful benchmark problem, since Markov chains representing realistic dynamical processes are frequently metastable[117–124] and therefore ill-conditioned.[37–48] We consider a single source node and a single sink node. Thus there is no contribution to the path probability from the initial node occupation probability distribution, and therefore $\mathcal{W}[\xi] \equiv \mathcal{P}[\xi]$.

Fig. 2.3 shows the probability distribution for the path action in the FPPE obtained from kinetic path sampling[68] (kPS) simulations, and the mean path action computed by the nodewise iterative formulation of the generalized GT algorithm (Eqs. 2.1 and 2.14), for the eight-state CTMC at an inverse temperature of $1/T = 2$. At low temperatures,

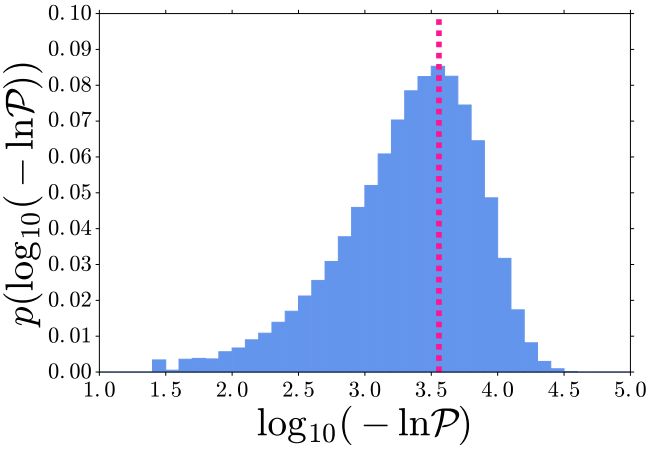


Figure 2.3: Probability distribution of the path action for the ensemble of $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths, obtained from 100000 kinetic path sampling[68] iterations, and mean path action obtained using the generalized GT algorithm (pink), for the eight-state CTMC (Fig. 2.1) at an inverse temperature of $1/T = 2$. At this temperature, the CTMC is strongly metastable.

where the model Markov chain is metastable, the use of the standard kinetic Monte Carlo algorithm[125] to sample $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths is unfeasibly inefficient.[52] Kinetic path sampling[68,69] (kPS), described in Chapter 1, can instead be used to sample the numbers of individual $i \leftarrow j$ transitions along $\mathcal{A} \leftarrow \mathcal{B}$ paths, and hence the probability distribution for first passage path rewards. The path action distribution at this low value of the temperature is fat-tailed.[126–128] That is, there is a small but appreciable proportion of probability mass at extreme values, which makes a substantial contribution to the mean, and thus the second and higher central moments of the distribution are significant. Hence, reliable estimation of the mean path action in the metastable regime by sampling paths requires a very large number of observations, even for this low-dimensional system.

It is common in dynamical models of realistic systems for first passage time distributions associated with transitions between two endpoint states to be fat-tailed.[52,126–128] One approach to examine this phenomenon is to compute the proportion of the $\mathcal{A} \leftarrow \mathcal{B}$ probability flux that can be attributed to the dominant first passage paths, and to examine the convergence of the sum for the expectation of the first passage time (*cf.* Eq. 2.15) when an increasing number of paths are included.

In Fig. 2.4, we compare the mean path action computed using GT to the values associated with the highest-probability paths determined by the REA,[70] for the eight-state CTMC (Fig. 2.1) at an inverse temperature of $1/T = 2$. Fig. 2.4 also illustrates the convergence of the pathwise sum (Eq. 2.15) for the MFPT, $\mathcal{T}_{\mathcal{AB}}$. The dominant first passage paths are highly atypical, and are associated with values for the path action and time that are several standard deviations smaller than the means of the respective distributions. Because these distributions

are fat-tailed, the 100000 highest-probability paths account for a fraction of only around 5×10^{-7} of the total $\mathcal{A} \leftarrow \mathcal{B}$ probability flux, and the pathwise sum for the MFPT is far from converged. The probabilities associated with the 100000 dominant first passage paths are close to uniform at this low temperature, suggesting that the paths determined by the REA are all closely related. Indeed, the small number of paths (around 5000) with the very highest probabilities are very similar, involving a small number of transitions via the lowest energy barriers. However, subsequent shortest paths can be divided into two families: longer paths involving only the most favourable transitions, and short paths proceeding via one or more alternative, less favourable, transitions. Our analysis of this simple model demonstrates that, while examination of the properties of the highest-probability first passage paths is insightful, this analysis alone may be misleading, and it is crucial to calculate the expectation for the path property of interest.

The extent to which the first passage time and path action distributions are fat-tailed depends strongly on the metastability of the Markov chain. Fig. 2.5 shows the evolution of the cumulative sum of probabilities for the 100000 dominant $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths at varying temperature. Notably, the convergence of the path probability sum follows the same pattern at all temperatures. There are a very small number (roughly 10-100) of first passage paths with relative probabilities that are particularly high, and the profile for the cumulative $\mathcal{A} \leftarrow \mathcal{B}$ path probability then reaches a plateau. Thus, even for the model system with a small state space and in the high-temperature limit, it is unfeasible to obtain the set of paths that account for the majority (say, $> 90\%$) of the $\mathcal{A} \leftarrow \mathcal{B}$ first passage path probability by shortest paths algorithms, which would require an exceptionally large number (more than 10^{10}) of paths to be determined. Nonetheless, at high temperatures, where there is no significant separation of characteristic timescales, almost 50% of the total $\mathcal{A} \leftarrow \mathcal{B}$ probability flux is accounted for by the 100000 dominant paths, and therefore it is feasible to obtain a representative picture of the global dynamics using the REA with edge costs that represent a local measure of the probability flux. At low temperatures, however, the set of highest-probability first passage paths accounts for a negligible proportion of the total probability flux, and is therefore not kinetically relevant.

To provide quantitative information on the relative importance of alternative families of first passage paths, we use the REA employing the edge costs given in Eq. 2.18, which are based on the net reactive flux along individual edges (Eq. 2.17). The decomposition of the total reactive $\mathcal{A} \leftarrow \mathcal{B}$ steady-state flux $\mathcal{J}_{\mathcal{AB}}$ (Eq. 2.16) into contributions from individual simple transition flux-paths, at varying temperature, is shown in Fig. 2.6. The observed order of committor probabilities for nodes, which does not change with temperature for this system, yields a pattern of unidirectional net reactive fluxes associated with a total of 36

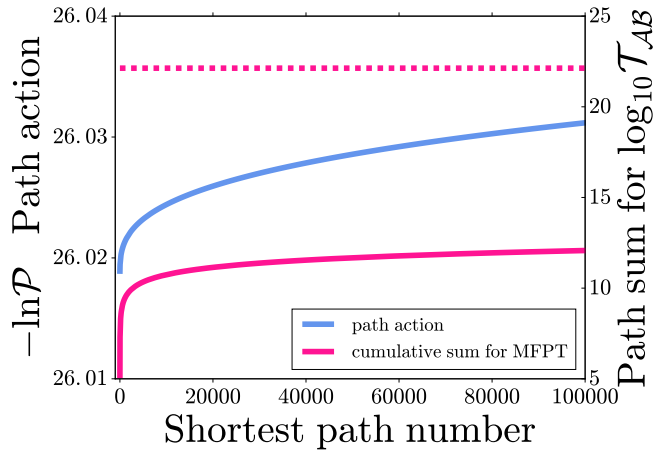


Figure 2.4: Values of the path action (blue) and cumulative pathwise sum (Eq. 2.15) for the MFPT (pink) for the 100000 highest-probability $\mathcal{A} \leftarrow \mathcal{B}$ first passage paths in the eight-state CTMC (Fig. 2.1) at an inverse temperature of $1/T = 2$. At this temperature, the CTMC is strongly metastable. The value of the MFPT, computed using GT, is indicated by a dashed pink line. The paths were determined using the recursive enumeration algorithm (Algorithm 4).[70] Note the small scale for the path action axis.

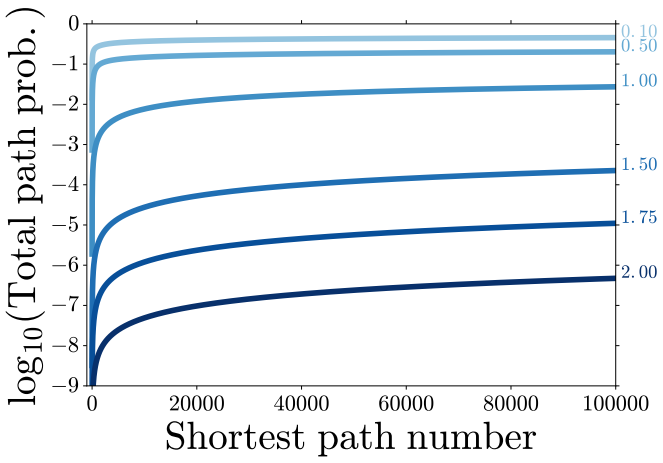


Figure 2.5: Cumulative sum of $\mathcal{A} \leftarrow \mathcal{B}$ first passage path probabilities at varying temperature for the eight-state CTMC (Fig. 2.1), accounting for an increasing number of the highest-probability paths determined by the recursive enumeration algorithm (Algorithm 4).[70] The annotations denote the value of the inverse temperature, i.e. $1/T$.

$\mathcal{A} \leftarrow \mathcal{B}$ simple flux-paths. Evidently, the reactive flux becomes increasingly localized among a small subset of transition flux-paths with decreasing temperature (increasing metastability). In the high-temperature regime ($1/T = 0.1$), the single flux-path associated with the largest contribution to the pathwise sum for $\mathcal{J}_{\mathcal{AB}}$ contributes around a third of the total reactive flux, and the 15 dominant simple paths are required to account for almost all ($> 99\%$) of the total reactive flux. Conversely, in the low-temperature regime ($1/T = 2$), the highest-flux simple path contributes more than half of the total reactive flux, and the vast majority of the total $\mathcal{A} \leftarrow \mathcal{B}$ flux is associated with the four dominant flux-paths.

Because the set of simple flux-paths is finite, decomposition of the total reactive flux $\mathcal{J}_{\mathcal{AB}}$ into additive contributions from transition flux-paths (*cf.* Eq. 2.16) provides a representative picture of the global dynamics even in the metastable regime, where the set of highest-probability first passage paths accounts for a negligible proportion of the total path probability (Fig. 2.5). By effectively grouping together paths that are related by unproductive flickering,[52] a quantitative comparison of the kinetic relevance of different competing $\mathcal{A} \leftarrow \mathcal{B}$ transition mechanisms is recovered. This shortest paths analysis is an alternative to the augmenting paths algorithm of Ref. 71, which distinguishes families of simple flux-paths on the basis of their associated dynamical bottleneck edges.[105] Moreover, computation of the committor probabilities[109] by a robust state reduction algorithm[104] or alternative linear algebra methods,[108] and determination of the shortest paths using the REA, scales favourably with dimensionality of the Markov chain. Hence, the complete set of simple flux-paths can be feasibly computed using the REA for sparse networks comprising several tens of thousands of nodes.

2.4 Conclusions

In this chapter, we have derived a general expression for renormalized rewards (Eq. 2.3) associated with arbitrary paths on a censored Markov chain.[16,77–82] We have also derived numerically stable iterative (Eqs. 2.1 and 2.14) and block (Eqs. 2.12 and 2.13) graph transformation[28–32] (GT) procedures to compute the mean reward for the ensemble of first passage paths,[57,62] i.e. the MFPR for a transition from an initial set of nodes $\mathcal{B}$ to an absorbing set of nodes $\mathcal{A}$. These formulations are applicable to both discrete- and continuous-time finite Markov chains.[35] If the system is not metastable, so that the transition probability matrix is well-conditioned, then MFPRs for transitions from all nonabsorbing nodes can be computed simultaneously using a single matrix inversion operation (Eq. 2.8).[85]

Knowledge of the expectation for the probability distributions of path properties in the FPPE is useful for assessing the convergence when sampling these distributions, which are frequently observed to be fat-tailed[126–128] owing to the existence of rare events in dynamical

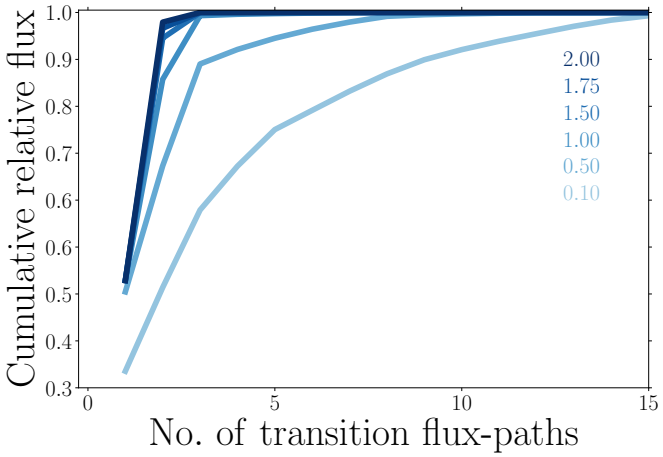


Figure 2.6: Cumulative sum of relative contributions (*cf.* Eq. 2.16) to the total $\mathcal{A} \leftarrow \mathcal{B}$ steady-state reactive flux, $\mathcal{J}_{\mathcal{AB}}$, from alternative simple flux-paths, for the eight-state CTMC (Fig. 2.1) at varying temperature. There are 36 simple flux-paths in total, but no more than 15 transition flux-paths are required to account for the vast majority ($> 99\%$) of the total reactive flux for all temperatures shown. The annotations denote the value of the inverse temperature, i.e. $1/T$. The flux-paths were determined using the recursive enumeration algorithm[70] (Algorithm 4) with edge costs based on net reactive fluxes (Eq. 2.18).

models for realistic systems.[124]

The mean values for first passage path properties can also be compared to the values associated with the highest-probability paths[74,91] determined by the recursive enumeration algorithm (REA),[70] to assess the extent to which the characteristics of the dominant first passage paths[87] are typical or otherwise. The shortest paths analysis allows us to evaluate the dominant terms in the pathwise sum (Eq. 2.15) for the expectation of a first passage path property, such as the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT.[36] Even for low-dimensional Markov chains that do not feature a separation of characteristic timescales, a substantial proportion of the $\mathcal{A} \leftarrow \mathcal{B}$ probability flux is attributable to an exceptionally large number of paths each associated with a very small probability. Hence, the set of shortest paths alone typically accounts for only a fraction of the pathwise sum for the MFPR.

In the metastable regime, low-probability paths comprising a very large number of transitions account for the overwhelming majority of the total first passage path probability, and the set of shortest paths alone is therefore not kinetically relevant. Alternative edge costs reflecting the global dynamics (Eq. 2.18) can be employed to exactly decompose the reactive steady-state $\mathcal{A} \leftarrow \mathcal{B}$ flux, $\mathcal{J}_{\mathcal{AB}}$, into a sum of contributions from simple flux-paths. This formulation is exact,[105] and provides a complementary viewpoint to the typical approach of decomposing $\mathcal{J}_{\mathcal{AB}}$ into additive contributions (*cf.* Eq. 2.17) from members of a set of edges that together constitute an $\mathcal{A}$-$\mathcal{B}$ cut in the network.[71] The latter framework, described in Chapter 1,

effectively groups together transition paths that share the same dynamical bottleneck edge of the cut set, and is therefore best suited to compare the relative importance of individual edges comprising a chosen $\mathcal{A}$-$\mathcal{B}$ cut.[129] The flux-pathwise analysis proposed in the current work provides a more detailed analysis that can be used to quantitatively understand the characteristic features of the whole $\mathcal{A} \leftarrow \mathcal{B}$ transition mechanism.

The GT and REA procedures scale favourably and can be applied to complex networks with state spaces comprising several hundred thousand nodes.[31,91] The methodology described herein will therefore provide fundamental insight into a variety of first passage processes in stochastic models. For instance: what is the single most probable route for the extinction of a species in a population dynamics[16–18] process? What are the most probable paths that together account for a specified proportion of the first passage probability flux, and what is the collective contribution of these paths to the MFPT?

2.A Description of the model system

Here we provide a complete specification of the model eight-state CTMC (for which the disconnectivity graph is shown in Fig. 2.1) that was employed to demonstrate our proposed methodology in Sec. 2.3. Let the diagonal element E_{jj} of the matrix $\mathbf{E}$ represent the energy of the j-th node of the Markov chain, and the off-diagonal element E_{ij} (for $i \neq j$) represent the energy of the transition state connecting nodes i and j, so that $E_{ij} - E_{jj}$ is the energy barrier for the $i \leftarrow j$ transition. The matrix $\mathbf{E}$ is:

$$\mathbf{E} = \begin{pmatrix} 0 & 28 & 103 & \infty & \infty & \infty & 18 & \infty \\ \vdots & 8 & 20 & 25 & 30 & \infty & 22 & \infty \\ \vdots & \ddots & 10 & 35 & 25 & \infty & 83 & \infty \\ \vdots & \ddots & \ddots & 9 & 20 & 125 & \infty & 24 \\ \vdots & \ddots & \ddots & \ddots & 7 & 26 & \infty & 36 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 1 & \infty & 19 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 & \infty \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \end{pmatrix}, \tag{2.19}$$

where off-diagonal entries E_{ij} equal to ∞ indicate that a direct connection between the i and j nodes does not exist. Note that the matrix $\mathbf{E}$ is symmetric about the diagonal, hence only the upper triangular elements are specified in Eq. 2.19. In the numerical results of Sec. 2.3, we defined the initial state to be $\mathcal{B} = \{1\}$ and the absorbing state to be $\mathcal{A} = \{8\}$.

The $i \leftarrow j$ transition rate is then given by the Arrhenius expression[116]

$$K_{ij} = \exp\left(- \frac{E_{ij} - E_{jj}}{T} \right) \quad \forall\, i \neq j, \tag{2.20}$$

where T is an effective temperature and we have set all the pre-exponential factors to unity for simplicity. The calculations in Sec. 2.3 used the Markov chain parameterized by the branching probability matrix as defined in Sec. 2.2.1. The eight-state model system provides an ideal benchmark to test the numerical stability of algorithms, since the extent to which the Markov chain is more or less ill-conditioned can be tuned by the value of the parameter T (*cf.* Fig. 2.2).

input : network $\mathcal{G}$ representing a finite Markov chain with state space $\mathcal{S}$, and $i \leftarrow j$ edge costs $-\ln T_{ij}\ \forall i,j \in \mathcal{S}$
initial (source) node $b \in \mathcal{B}$ and absorbing (sink) node $a \in \mathcal{A}$
total number of highest-probability paths to compute, k_{tot}
output: k_{tot} highest-probability $a \leftarrow b$ first passage paths; $\xi^k(a)$ for $1 \le k \le k_{\text{tot}}$

$\xi^k(j) \leftarrow \emptyset \ \ \forall\, 1 \le k \le k_{\text{tot}}, j \in \mathcal{S}$;
$\mathcal{M}(j) \leftarrow \emptyset \ \ \forall\, j$;
$\xi^1(j)\ \forall\, j \neq b \leftarrow$ `Dijkstra(`$\mathcal{G}$`)`;
`/* main loop of REA */`
for $k = 2, \ldots, k_{\text{tot}}$ **do**
    $\xi^k(a) \leftarrow$ `NextPath(`k,a`)`;
return $\xi^k(a)$ for $1 \le k \le k_{\text{tot}}$;

`/* one-to-all shortest path algorithm */`
function `Dijkstra(`$\mathcal{G}$`)`
  return $\xi^1(j) \leftarrow \arg\min_{\xi(j)} \mathcal{R}[\xi(j) \equiv \xi^1(i) \cup \{j \leftarrow i\}] \quad \forall\, j \in \mathcal{S} \setminus b$;

function `Pred(`k,j`)`
  return k', i where $\xi^k(j) \equiv \xi^{k'}(i) \cup \{j \leftarrow i\}$;

`/* function to determine the` k`-th most probable first passage path to node` j`, given that the` $1, \ldots, (k-1)$`-th most probable first passage paths are known */`
function `NextPath(`k,j`)`
  if $k == 2$ **then**
    `/* Initialize set of candidates for the next most probable path to node` j `*/`
    $\mathcal{M}(j) \leftarrow \{\xi^1(i) \cup \{j \leftarrow i\} \ \ \forall\, i \in \mathcal{D}(j)$ and $1, i \neq$ `Pred(1,`j`)` $\}$;
    if $j == b$ **then**
      goto selectpath;
  k',$i \leftarrow$ `Pred(`$k-1$,j`)`;
  `/* the` $(k'+1)$`-th shortest path to node` i `is a viable parent segment of a candidate path to node` j`. If unknown, compute this path with a recursive call to the NextPath function */`
  if $\xi^{(k'+1)}(i) \equiv \emptyset$ **then**
    $\xi^{(k'+1)}(i) \leftarrow$ `NextPath(`$k'+1$,i`)`;
  $\mathcal{M}(j) \leftarrow \mathcal{M}(j) \cup \{\xi^{(k'+1)}(i) \cup \{j \leftarrow i\}\}$; `// add candidate path to list`
  selectpath:
    $\xi^k(j) \leftarrow \arg\min_{\xi \in \mathcal{M}(j)} \mathcal{R}[\xi]$; `// assign candidate path with lowest cost`
    $\mathcal{M}(j) \leftarrow \mathcal{M}(j) \setminus \xi^k(j)$; `// remove assigned candidate path from list`
  return $\xi^k(j)$;

Algorithm 4: Recursive enumeration algorithm[70] (REA) to compute the k highest-probability first passage paths from an initial node b to an absorbing node a in an irreducible finite Markov chain. $\xi^k(j)$ denotes the k-th most probable $a \leftarrow b$ path to node j. The cost associated with the path ξ is $\mathcal{R}[\xi] = -\sum_{(i \leftarrow j) \in \xi} \ln T_{ij}$. $\mathcal{D}(j)$ denotes the set of nodes for which a direct $i \leftarrow j$ connection exists. $\mathcal{M}(j)$ denotes the set of candidates for the next most probable path to the j-th node. The ordered sequence of transitions along the k-th highest-probability path can be obtained by tracing the shortest paths array using the `Pred` function. For reducible Markov chains, or when using edge costs based on net reactive fluxes (Eq. 2.18), the `NextPath` function may encounter the situation where a candidate path to a node cannot be found. In this case, the next shortest path to the node does not exist. If the node in question is the target node a, then the main loop of the REA is exited, the complete set of $a \leftarrow b$ paths having been found.

Bibliography

[1] J. R. Norris. *Markov Chains*. Cambridge University Press, New York, USA, 1997.

[2] N. Masuda, M. A. Porter, and R. Lambiotte. *Phys. Rep.*, 716-717:1–58, 2017.

[3] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, New Jersey, USA, 1960.

[4] G. R. Bowman, V. S. Pande, and F. Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Springer, Netherlands, first edition, 2014.

[5] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. *Proc. Natl. Acad. Sci. USA*, 106:19011–19016, 2009.

[6] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. *J. Chem. Phys.*, 134:174105, 2011.

[7] B. E. Husic and V. S. Pande. *J. Am. Chem. Soc.*, 140:2386–2896, 2018.

[8] A. Mardt, L. Pasquali, H. Wu, and F. Noé. *Nat. Commun.*, 9:5, 2018.

[9] R. G. Mantell, C. E. Pitt, and D. J. Wales. *J. Chem. Theory Comput.*, 12:6182–6191, 2016.

[10] D. J. Wales. *Mol. Phys.*, 100:3285–3305, 2002.

[11] D. J. Wales. *Int. Rev. Phys. Chem.*, 25:237–282, 2006.

[12] J. A. Joseph, K. Röder, D. Chakraborty, R. G. Mantell, and D. J. Wales. *Chem. Commun.*, 53:6974–6988, 2017.

[13] D. J. Wales. *Annu. Rev. Phys. Chem.*, 69:401–425, 2018.

[14] E. Seneta. *Linear Algebra Appl.*, 34:259–267, 1980.

[15] D. P. Heyman. *J. Appl. Probab.*, 32:893–901, 1995.

[16] T. Dayar, H. Hermanns, D. Spieler, and V. Wolf. *Numer. Linear Algebra Appl.*, 18:931–946, 2011.

[17] E. Renshaw. *Modelling Biological Populations in Space and Time*. Cambridge University Press, Cambridge, UK, 1991.

[18] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK, 1998.

[19] J. Goutsias. *J. Chem. Phys.*, 122:184102, 2005.

[20] Y. Cao, D. T. Gillespie, and L. R. Petzold. *J. Chem. Phys.*, 122:014116, 2005.

[21] W. E, D. Liu, and E. Vanden-Eijnden. *J. Chem. Phys.*, 123:194107, 2005.

[22] W. E, D. Liu, and E. Vanden-Eijnden. *J. Comput. Phys.*, 221:158–180, 2007.

[23] H.-W. Kang and T. G. Kurtz. *Ann. Appl. Probab.*, 23:529–593, 2013.

[24] J. Goutsias and G. Jenkinson. *Phys. Rep.*, 529:199–264, 2013.

[25] B. Munsky and M. Khammash. *J. Chem. Phys.*, 124:044104, 2006.

[26] K. N. Dinh and R. B. Sidje. *Phys. Biol.*, 13:035003, 2016.

[27] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Massachusetts, second edition, 2018.

[28] S. A. Trygubenko and D. J. Wales. *Mol. Phys.*, 104:1497–1507, 2006.

[29] S. A. Trygubenko and D. J. Wales. *J. Chem. Phys.*, 124:234110, 2006.

[30] D. J. Wales. *J. Chem. Phys.*, 130:204111, 2009.

[31] J. D. Stevenson and D. J. Wales. *J. Chem. Phys.*, 141:041104, 2014.

[32] R. S. MacKay and J. D. Robinson. *Phil. Trans. Roy. Soc. A*, 376:20170232, 2018.

[33] S. Redner. *A Guide to First-Passage Processes*. Cambridge University Press, Cambridge, UK, 2012.

[34] R. Metzler, G. Oshanin, and S. Redner. *First-Passage Phenomena and Their Applications*. World Scientific, Singapore, 2014.

[35] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. *J. Chem. Phys.*, 153:244108, 2020.

[36] J. J. Hunter. *Linear Algebra Appl.*, 549:100–122, 2018.

[37] W. Grassmann and D. A. Stanford. In W. Grassmann, editor, *Computational Probability*, pages 153–203. Springer, New York, 2000.

[38] J. R. Koury, D. F. McAllister, and W. J. Stewart. *SIAM J. Alg. Discr. Meth.*, 5:164–186, 1984.

[39] D. P. Heyman. *SIAM J. Alg. Discr. Meth.*, 8:226–232, 1987.

[40] D. P. Heyman and A. Reeves. *ORSA J. Comp.*, 1:52–60, 1989.

[41] B. Philippe, Y. Saad, and W. J. Stewart. *Oper. Res.*, 40:1156–1179, 1992.

[42] C. A. O'Cinneide. *Numer. Math.*, 65:109–120, 1993.

[43] C. D. Meyer Jr. *SIAM J. Matrix Anal. Appl.*, 15:715–728, 1994.

[44] C. A. O'Cinneide. *Numer. Math.*, 73:507–519, 1996.

[45] D. P. O'Leary and Y.-J. J. Wu. *SIAM J. Matrix Anal. Appl.*, 17:470–488, 1996.

[46] D. J. Hartfiel and C. D. Meyer Jr. *Linear Algebra Appl.*, 272:193–203, 1998.

[47] J. L. Barlow. *SIAM J. Matrix Anal. Appl.*, 22:230–241, 2000.

[48] M. Benzi. *Linear Algebra Appl.*, 386:27–49, 2004.

[49] O. Valsson, P. Tiwary, and M. Parrinello. *Annu. Rev. Phys. Chem.*, 67:159–184, 2016.

[50] B. Trendelkamp-Schroer and F. Noé. *Phys. Rev. X*, 6:011009, 2016.

[51] M. A. Novotny. *Phys. Rev. Lett.*, 74:1–5, 1995.

[52] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 153:024121, 2020.

[53] M. Manhart and A. V. Morozov. *Phys. Rev. Lett.*, 111:088102, 2013.

[54] U. Seifert. *Rep. Prog. Phys.*, 75:126001, 2012.

[55] M. V. S. Bonança and C. Jarzynski. *Phys. Rev. E*, 93:022101, 2016.

[56] C. Maes and K. Netočný. *J. Stat. Phys.*, 110:269–310, 2003.

[57] M. Manhart, W. Kion-Crosby, and A. V. Morozov. *J. Chem. Phys.*, 143:214106, 2015.

[58] S. X. Sun. *Phys. Rev. Lett.*, 96:210602, 2006.

[59] B. Harland and S. X. Sun. *J. Chem. Phys.*, 127:104103, 2007.

[60] T. Oppelstrup, V. V. Bulatov, A. Donev, M. H. Kalos, G. H. Gilmer, and B. Sadigh. *Phys. Rev. E*, 80:066701, 2009.

[61] S. Hwang, D.-S. Lee, and B. Kahng. *Phys. Rev. Lett.*, 109:088701, 2012.

[62] M. von Kleist, C. Schütte, and W. Zhang. *J. Stat. Phys.*, 170:809–843, 2018.

[63] P. Dixit, A. Jain, G. Stock, and K. A. Dill. *J. Chem. Theory Comput.*, 11:5464–5472, 2015.

[64] P. Dixit, J. Wagoner, C. Weistuch, S. Pressé, K. Ghosh, and K. A. Dill. *J. Chem. Phys.*, 148:010901, 2018.

[65] C. Van den Broeck. In C. Bechinger, F. Sciortino, and P. Ziherl, editors, *Proceedings of the International School of Physics "Enrico Fermi" Vol. 184: Physics of Complex Colloids*, pages 155–193. IOS Press, Amsterdam, 2013.

[66] U. Seifert. *Phys. Rev. Lett.*, 95:040602, 2005.

[67] M. J. Goldberg and S. Kim. *Appl. Comput. Harmon. Anal.*, 30:37–46, 2011.

[68] M. Athènes and V. V. Bulatov. *Phys. Rev. Lett.*, 113:230601, 2014.

[69] M. Athènes, S. Kaur, G. Adjanor, T. Vanacker, and T. Jourdan. *Phys. Rev. Materials*, 3:103802, 2019.

[70] V. M. Jiménez and A. Marzal. In J. S. Vitter and C. D. Zaroliagis, editors, *Algorithm Engineering: 3rd International Workshop, WAE'99, London, UK*, pages 15–29. Springer Berlin, Heidelberg, 1999.

[71] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *Multiscale Model. Simul.*, 7:1192–1219, 2009.

[72] J. G. Kemeny and J. L. Snell. *Theory Prob. Its Appl.*, 6:101–105, 1961.

[73] S. A. Serebrinsky. *Phys. Rev. E*, 83:037701, 2011.

[74] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 151:124101, 2019.

[75] J. K. Weber, D. Shukla, and V. S. Pande. *Proc. Natl. Acad. Sci. USA*, 112:10377–10382, 2015.

[76] S. Vaikuntanathan, T. R. Gingrich, and P. L. Geissler. *Phys. Rev. E*, 89:062108, 2014.

[77] C. D. Meyer Jr. *SIAM Rev.*, 31:240–272, 1989.

[78] Y. Q. Zhao and D. Liu. *J. Appl. Probab.*, 33:623–629, 1996.

[79] E. Meerbach, C. Schütte, and A. Fischer. *Linear Algebra Appl.*, 398:141–160, 2005.

[80] N. Pekergin, T. Dayar, and D. N. Alparslan. *Eur. J. Oper. Res.*, 165:810–825, 2005.

[81] J.-M. Fourneau, N. Pekergin, and S. Younès. In K. Wolter, editor, *EPEW 2007: Formal Methods and Stochastic Models for Performance Evaluation*, pages 213–227. Springer Berlin, Heidelberg, 2007.

[82] A. Milias-Argeitis and J. Lygeros. *J. Chem. Phys.*, 138:184109, 2013.

[83] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. *Oper. Res.*, 33:1107–1116, 1985.

[84] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. (unpublished), 2020.

[85] T. D. Swinburne and D. J. Wales. *J. Chem. Theory Comput.*, 16:2661–2679, 2020.

[86] G. R. Bowman, L. Meng, and X. Huang. *J. Chem. Phys.*, 139:121905, 2013.

[87] S. Viswanath, S. M. Kreuzer, A. E. Cardenas, and R. Elber. *J. Chem. Phys.*, 139:174105, 2013.

[88] D. R. Shier. *Networks*, 9:195–214, 1979.

[89] E. Q. V. Martins. *Eur. J. Op. Res.*, 18:123–130, 1984.

[90] E. Q. V. Martins and M. M. B. Pascoal. *4OR*, 1:121–133, 2003.

[91] J. M. Carr and D. J. Wales. In A. Solov'yov and J.-P. Connerade, editors, *Latest Advances in Atomic Cluster Collisions: Structure and Dynamics from the Nuclear to the Biological Scale*, pages 321–330. Imperial College Press, London, 2008.

[92] D. Eppstein. *SIAM J. Comput.*, 28:652–673, 1999.

[93] V. M. Jiménez and A. Marzal. In K. Jansen, M. Margraf, M. Mastrolilli, and J. D. P. Rolim, editors, *International Workshop on Experimental and Efficient Algorithms, WEA 2003*, pages 179–191. Springer Berlin, Heidelberg, 2003.

[94] J. A. Azevedo, M. E. O. S. Costa, J. J. R. E. S. Madeira, and E. Q. V. Martins. *Eur. J. Op. Res.*, 69:97–106, 1993.

[95] J. A. Azevedo, J. J. R. E. S. Madeira, E. Q. V. Martins, and F. M. A. Pires. *Eur. J. Op. Res.*, 73:188–191, 1994.

[96] E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. dos Santos. *Int. J. Found. Comp. Sci.*, 10:247–261, 1999.

[97] E. Q. V. Martins and J. L. E. dos Santos. *Investigação Operacional*, 20:47–62, 2000.

[98] E. Q. V. Martins, M. M. B. Pascoal, and J. L. E. dos Santos. *Investigação Operacional*, 21:47–60, 2001.

[99] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, New Jersey, USA, 1962.

[100] S. E. Dreyfus. *Oper. Res.*, 17:395–412, 1969.

[101] E. W. Dijkstra. *Numer. Math.*, 1:269–271, 1959.

[102] M. L. Fredman and R. E. Tarjan. *J. Assoc. Comput. Mach.*, 34:596–615, 1987.

[103] R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan. *J. Assoc. Comput. Mach.*, 37:213–223, 1990.

[104] D. J. Sharpe and D. J. Wales. (unpublished), 2020.

[105] A. Berezhkovski, G. Hummer, and A. Szabo. *J. Chem. Phys.*, 130:205102, 2009.

[106] T. J. Sheskin. *Oper. Res.*, 33:228–235, 1985.

[107] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *J. Chem. Phys.*, 125:084110, 2006.

[108] J.-H. Prinz, M. Held, J. C. Smith, and F. Noé. *Multiscale Model. Simul.*, 9:545–567, 2011.

[109] A. M. Berezhkovskii and A. Szabo. *J. Chem. Phys.*, 150:054106, 2019.

[110] M. K. Cameron and E. Vanden-Eijnden. *J. Stat. Phys.*, 156:427–454, 2014.

[111] M. Haviv. *SIAM J. Numer. Anal.*, 22:952–966, 1987.

[112] T. Dayar and W. J. Stewart. *SIAM J. Sci. Comput.*, 17:287–303, 1996.

[113] Y.-J. J. Wu. Technical Report TR-3347, University of Maryland, 1994.

[114] O. M. Becker and M. Karplus. *J. Chem. Phys.*, 106:1495–1517, 1997.

[115] T. D. Swinburne, D. Kannan, D. J. Sharpe, and D. J. Wales. *J. Chem. Phys.*, 153:134115, 2020.

[116] A. Kells, V. Koskin, E. Rosta, and A. Annibale. *J. Chem. Phys.*, 152:104108, 2020.

[117] M. Weber, S. Kube, L. Walter, and P. Deuflhard. *Multiscale Model. Simul.*, 6:396–416, 2007.

[118] A. Dickson and A. R. Dinner. *Annu. Rev. Phys. Chem.*, 61:441–459, 2010.

[119] J. T. Berryman and T. Schilling. *J. Chem. Phys.*, 133:244101, 2010.

[120] C. Giardina, J. Kurchan, V. Lecomte, and J. Tailleur. *J. Stat. Phys.*, 145:787–811, 2011.

[121] C. Hartmann, R. Banisch, M. Sarich, T. Badowski, and C. Schütte. *Entropy*, 16:350–376, 2014.

[122] M. Sarich, R. Banishc, C. Hartmann, and C. Schütte. *Entropy*, 16:258–286, 2014.

[123] M. K. Cameron. *J. Chem. Phys.*, 141:184113, 2014.

[124] C. Pérez-Espigares and P. I. Hurtado. *Chaos*, 29:083106, 2019.

[125] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. *J. Comput. Phys.*, 17:10–18, 1975.

[126] T. Hoffmann, M. A. Porter, and R. Lambiotte. *Phys. Rev. E*, 86:046102, 2012.

[127] J.-C. Delvenne, R. Lambiotte, and L. E. C. Rocha. *Nat. Commun.*, 6:7366, 2015.

[128] E. F. dos Reis, A. Li, and N. Masuda. *Phys. Rev. E*, 102:052303, 2020.

[129] D. Nagel, A. Weber, and G. Stock. *J. Chem. Theory Comput.*, 16:7874–7882, 2020.

Chapter 3

Numerical analysis of first passage processes in finite Markov chains exhibiting metastability

We describe state reduction algorithms for the analysis of first passage processes in discrete- and continuous-time finite Markov chains. We present a formulation of the graph transformation (GT) algorithm that allows for the evaluation of exact mean first passage times (MFPTs), stationary probabilities, and committor probabilities for all nonabsorbing nodes of a Markov chain in a single computation. Calculation of the committor probabilities within the state reduction formalism is readily generalizable to the first hitting problem for any number of alternative target states. We then show that a state reduction algorithm can be formulated to compute the expected number of times that each node is visited along a first passage path. Hence, all properties required to analyze the first passage path ensemble (FPPE) at both a microscopic and macroscopic level of detail, including the mean and variance of the FPT distribution, can be computed using state reduction methods. In particular, we derive expressions for the probability that a node is visited along a direct transition path, which proceeds without returning to the initial state, considering both the nonequilibrium and equilibrium (steady-state) FPPEs. The reactive visitation probability provides a rigorous metric to quantify the dynamical importance of a node for the productive transition between two endpoint states, and thus allows the local states that facilitate the dominant transition mechanisms to be readily identified. The state reduction procedures remain numerically stable even for Markov chains exhibiting metastability, which can be severely ill-conditioned. The rare event regime is frequently encountered in realistic models of dynamical processes, and our methodology therefore provides valuable tools for the analysis of Markov chains in practical applications. We illustrate our approach with numerical results for a kinetic network representing a structural transition in an atomic cluster.

3.1 Introduction

The analysis of first passage processes,[1–4] concerning the evolution of a system until a specified target state is hit, is of fundamental interest in the theory of stochastic dynamics. The usual dynamical observable is the mean first passage time (MFPT), defined as the expected time for trajectories to hit the target state.[5–8] The set of possible paths and their associated probabilities, for a given initial occupation probability distribution, defines the (nonequilibrium) first passage path ensemble (FPPE).[9–17] Finite Markov chains,[18–24] in which a dynamical process is modeled as a sequence of memoryless jumps between the nodes of a network,[25] are a class of discrete-state stochastic model that have been widely adopted in diverse disciplines. First passage processes in Markov chains can be used to model stochastic phenomena as varied as biomolecular conformational transitions,[26–33] animal movement to a foraging site within an ecosystem,[34] and the sequence of events leading to a stock market crash in economics.[35]

The dynamical properties of interest characterizing the FPPE for a Markov chain can in principle be computed by solving a corresponding system of linear equations (Chapter 1).[36,37] However, the required computations, including eigendecomposition or matrix inversion operations, are liable to encounter numerical issues arising from finite precision when the Markov chain exhibits metastability.[38] For Markov chains featuring a separation of characteristic timescales, the subdominant eigenvalue of the underlying transition probability (rate) matrix approaches unity (zero),[39,40] respectively, and the system is therefore severely ill-conditioned.[41–44] Moreover, in general, it is nontrivial to apply preconditioning techniques to improve the numerical stability of sparse linear algebra methods.[42,45] In realistic applications, there is typically a rare event that is of particular interest, which is the first passage process that we wish to analyze.[38,46–59] This situation motivates the development of alternative procedures that have inherent numerical stability, so that the fundamental dynamical properties of a Markov chain can be computed robustly.

In this chapter we focus on state reduction methods[60] to derive numerically stable algorithms for the analysis of Markov chain dynamics.[61–64] These methods proceed via elimination of the nodes in a Markov chain, while preserving averages for the dynamical properties of interest, and may also employ a back substitution phase to restore the eliminated nodes in turn.[65] State reduction algorithms have been formulated to compute the stationary distribution,[66,67] MFPTs,[68] moments of the FPT distribution,[69] and the group inverse[36,37] of an irreducible Markov chain.[70–72]

We present a convenient formulation of the graph transformation (GT) algorithm[73–78] that allows for the simultaneous determination of the MFPTs, stationary probabilities,

and committor probabilities[79–82] for *all* nonabsorbing nodes of a Markov chain in a single computation (Sec. 3.2), as well as the absorption probabilities.[18] We also show that a state reduction algorithm can be designed to compute the expected number of times that nodes are visited along first passage paths (Sec. 3.3.2). Other quantities that characterize the global and local properties of the FPPE, such as the variance of the FPT distribution and the variances in the number of times that nodes are visited, can be determined from this information (Sec. 3.3). We derive expressions for the probabilities that nodes are visited along reactive transition paths, which proceed directly to the absorbing state without returning to the initial state, considering both the nonequilibrium and equilibrium (i.e. steady-state) FPPEs (Sec. 3.3.4). The reactive visitation probabilities quantify the dynamical relevance of individual nodes, and therefore allow us to identify the key mechanisms for productive transitions and the bottleneck nodes that mediate the dominant pathways. The theory of committor and reactive visitation probabilities, and of reactive fluxes along individual edges of the network, is generalizable to the case where there are multiple *taboo* states.[83–85] Separation of the dynamics into competing first passage processes associated with alternative target states is frequently of interest in models featuring several attractors.

The use of the state reduction algorithms is illustrated with numerical results for a kinetic network representing a solid-solid structural transition in a model atomic cluster, for which standard linear algebra methods are unable to compute any of the aforementioned dynamical quantities (Sec. 3.4). Hence, our methodology provides a viable means to analyze first passage processes in Markov chains exhibiting rare event dynamics, at both a microscopic and macroscopic level of detail. The computations were performed using our `DISCOTRESS` software.[86]

3.2 LU decomposition formulation of graph transformation

3.2.1 Markov chain dynamics

As in the previous chapters, we consider discrete-time Markov chains (DTMCs) parameterized by a transition probability matrix $\mathbf{T}(\tau)$, where $i \leftarrow j$ transitions have probabilities $T_{ij}(\tau)$ and are associated with a fixed lag time τ,[18] and continuous-time Markov chains (CTMCs) parameterized by a transition rate matrix $\mathbf{K}$.[20] The off-diagonal elements of $\mathbf{K}$ are the $i \leftarrow j$ transition probabilities per unit time in the limit of an infinitesimally small time step, and the diagonal elements are $K_{jj} = -\sum_{\gamma \neq j} K_{\gamma j}$, so that the columns of the matrix sum to zero.[87] The $i \leftarrow j$ transition probabilities for a CTMC are the elements $P_{ij} = K_{ij} / \sum_{\gamma \neq j} K_{\gamma j}$ of the branching probability matrix $\mathbf{P}$, and the waiting time for the $i \leftarrow j$ transition is

drawn from an exponential distribution with mean $\tau_j = 1/\sum_{\gamma \neq j} K_{\gamma j}$.[88] We shall denote the transition probability matrix, either $\mathbf{T}(\tau)$ or $\mathbf{P}$, by $\mathbf{T}$ for generality. We consider a Markov chain with state space $\mathcal{S}$ partitioned into the set of absorbing nodes $\mathcal{A}$ and the set of *transient* (nonabsorbing) nodes $\mathcal{Q}$, so that $\mathcal{S} \equiv \mathcal{Q} \cup \mathcal{A}$.[89] That is, we consider Markov chains where the set $\mathcal{A}$ must eventually be reached when the process is initialized at any node of the set $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$.[18] If the nodes of the set $\mathcal{A}$ are not absorbing in the underlying model, so that it is possible to reach any node of the network from any other node, then the Markov chain is said to be *irreducible*.[36,90–92] We may then define the stationary probability distribution (column) vector $\boldsymbol{\pi}$, which satisfies the global balance equations $\mathbf{T}(\tau)\boldsymbol{\pi} = \boldsymbol{\pi}$ and $\mathbf{K}\boldsymbol{\pi} = \mathbf{0}$.[87]

A key dynamical quantity characterizing a first passage process is the $\mathcal{A} \leftarrow j$ mean first passage time (MFPT) $\mathcal{T}_{\mathcal{A}j}$, defined as the expected time at which a trajectory first hits the state $\mathcal{A}$, given that it was initialized at node j.[20,93,94] Let the time associated with a particular $\mathcal{A} \leftarrow j$ first passage trajectory be denoted by t_{FPT}. Then the MFPT is defined as

$$\mathcal{T}_{\mathcal{A}j} = \langle t_{\mathrm{FPT}} \rangle = \int_0^\infty t_{\mathrm{FPT}} p(t_{\mathrm{FPT}}) dt_{\mathrm{FPT}}. \tag{3.1}$$

Here, $p(t_{\mathrm{FPT}})$ is the first passage time (FPT) distribution[15]

$$\begin{aligned} p(t_{\mathrm{FPT}}) = \Pr\{\xi(t_{\mathrm{FPT}}) \in \mathcal{A}, \xi(0 \leq t < t_{\mathrm{FPT}}) \notin \mathcal{A} \\ \mid \xi(t=0) = j\}, \end{aligned} \tag{3.2}$$

where $\xi(t)$ denotes the node of the Markov chain that is occupied for the first passage path $\xi = (\mathcal{A} \leftarrow i_n \leftarrow \ldots \leftarrow i_1 \leftarrow j)$ at time t, where $j, i_1, \ldots, i_n \notin \mathcal{A}$.[95] The MFPTs satisfy a first-step relation,[22]

$$\mathcal{T}_{\mathcal{A}j} = \tau_j + \sum_{\gamma \notin \mathcal{A}} T_{\gamma j} \mathcal{T}_{\mathcal{A}\gamma}. \tag{3.3}$$

Therefore, in principle, the MFPTs $\mathcal{T}_{\mathcal{A}j}$ $\forall j \notin \mathcal{A}$ can be determined by solving Eq. 3.3 using any appropriate linear algebra method, such as Gauss-Seidel iteration[96] or successive over-relaxation.[97] The MFPT for a transition from an initial set $\mathcal{B} \subseteq \mathcal{A}^{\mathsf{c}}$, associated with a specified initial occupation probability distribution vector $\mathbf{p}(0)$, is then obtained simply as a weighted average,

$$\mathcal{T}_{\mathcal{A}\mathcal{B}} = \sum_{b \in \mathcal{B}} p_b(0) \mathcal{T}_{\mathcal{A}b}. \tag{3.4}$$

However, for Markov chains exhibiting metastability, the linear system of equations in Eq. 3.3 can be severely ill-conditioned, so that standard dense linear algebra methods may experience a severe propagation of numerical error arising from finite precision,[76,77] and Krylov subspace methods[45,98] may fail to converge.[42,99]

3.2.2 Stochastic complements and the graph transformation algorithm

In this section, we summarize relevant theory of censored Markov chains and the GT algorithm that was introduced in Chapter 1. The $\mathcal{A} \leftarrow j$ MFPT for a transition from a particular node $j \notin \mathcal{A}$ can be computed robustly using stochastic complementation.[100–102] Let us partition the transition probability matrix as

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{\mathcal{Z}\mathcal{Z}} & \mathbf{T}_{\mathcal{Z}\mathcal{N}} \\ \mathbf{T}_{\mathcal{N}\mathcal{Z}} & \mathbf{T}_{\mathcal{N}\mathcal{N}} \end{bmatrix}, \tag{3.5}$$

where $\mathcal{Z} \equiv \mathcal{A} \cup \{j\}$ and $\mathcal{N} \equiv \mathcal{Z}^{\mathsf{c}} \equiv \mathcal{Q} \setminus \{j\}$. In Eq. 3.5, $\mathbf{T}_{\mathcal{Z}\mathcal{N}}$ contains the probabilities for transitions from nodes of the set $\mathcal{N}$ to the set $\mathcal{Z}$, and the other blocks are defined similarly. The diagonal blocks $\mathbf{T}_{\mathcal{Z}\mathcal{Z}}$ and $\mathbf{T}_{\mathcal{N}\mathcal{N}}$ are therefore square substochastic matrices.[91,103] The stochastic complement for the nodes in $\mathcal{Z}$ is defined as[100–102]

$$\mathbf{T}'_{\mathcal{Z}\mathcal{Z}} \leftarrow \mathbf{T}_{\mathcal{Z}\mathcal{Z}} + \mathbf{T}_{\mathcal{Z}\mathcal{N}}(\mathbf{I}_{\mathcal{N}\mathcal{N}} - \mathbf{T}_{\mathcal{N}\mathcal{N}})^{-1}\mathbf{T}_{\mathcal{N}\mathcal{Z}}, \tag{3.6}$$

where $\mathbf{I}_{\mathcal{N}\mathcal{N}}$ is the $|\mathcal{N}| \times |\mathcal{N}|$-dimensional identity matrix. Eq. 3.6 defines renormalized transition probabilities for the nodes in $\mathcal{Z}$. This transformed system is sometimes referred to as a *censored* Markov chain, because the renormalized probabilities correspond to the values that would be observed if the transitions within $\mathcal{N}$ were obscured (see Fig. 3.1).[104–106] With $\mathcal{Z} \equiv \mathcal{A} \cup \{j\}$, the only remaining transitions associated with node j in the renormalized network are to nodes of the set $\mathcal{A}$, and the $j \leftarrow j$ self loop. If the waiting times of the nodes in the set $\mathcal{Z}$, contained in the $|\mathcal{Z}|$-dimensional vector $\boldsymbol{\tau}_{\mathcal{Z}}$, are renormalized according to[78,107]

$$\boldsymbol{\tau}'_{\mathcal{Z}} \leftarrow \boldsymbol{\tau}_{\mathcal{Z}} + \boldsymbol{\tau}_{\mathcal{N}}(\mathbf{I}_{\mathcal{N}\mathcal{N}} - \mathbf{T}_{\mathcal{N}\mathcal{N}})^{-1}\mathbf{T}_{\mathcal{N}\mathcal{Z}}, \tag{3.7}$$

then the MFPT for the $\mathcal{A} \leftarrow j$ transition after eliminating the $|\mathcal{Q}| - 1$ nodes of set $\mathcal{N}$ is given by[76]

$$\begin{aligned} \mathcal{T}_{\mathcal{A}j} &= [\boldsymbol{\tau}'_{\mathcal{Z}}]_j \left(1 - [\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}]_{jj}\right) \sum_{n=1}^{\infty} n [\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}]_{jj}^{n-1} \\ &= \frac{[\boldsymbol{\tau}'_{\mathcal{Z}}]_j}{1 - [\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}]_{jj}}. \end{aligned} \tag{3.8}$$

Here, we have used the fact that $(1 - [\mathbf{T}'_{\mathcal{Z}\mathcal{Z}}]_{jj})^{-1}$ is the expected number of $j \leftarrow j$ transitions before node j is exited plus one for the final escape step.[18]

The renormalization of the mean waiting (or lag) times (Eq. 3.7) accounts for the expected number of transitions within the set of nodes to be eliminated, $\mathcal{N}$. That is, the updated

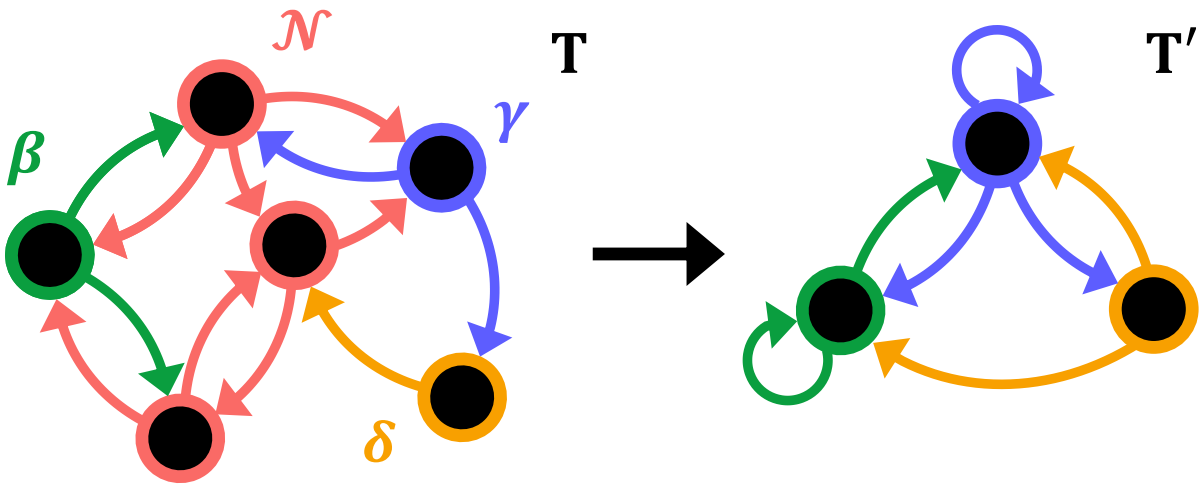


Figure 3.1: Illustration of a renormalization operation performed on the stochastic matrix $\mathbf{T}$ to yield the censored Markov chain $\mathbf{T}'$. To eliminate the block of nodes $\mathcal{N}$, the renormalized transition probabilities (Eq. 3.6) in the resulting censored Markov chain (or *stochastic complement*[100]) must account for transitions that proceed via $\mathcal{N}$ in the original network. Hence, the renormalized stochastic matrix includes a direct $\gamma \leftarrow \beta$ transition that is not present in the original network, which corresponds to the collective probabilities of all possible $\gamma \leftarrow \mathcal{N} \leftarrow \beta$ paths. By the same reasoning, the stochastic complement contains nonzero probabilities for $\beta \leftarrow \beta$, $\beta \leftarrow \gamma$, $\gamma \leftarrow \gamma$, and $\gamma \leftarrow \delta$ transitions. The probability for the $\delta \leftarrow \gamma$ transition does not increase with renormalization, and likewise there is no $\delta \leftarrow \beta$ transition in the derived stochastic complement, since there are no such indirect transitions proceeding via $\mathcal{N}$ in the original Markov chain. The waiting times associated with all three of the retained nodes are increased in the censored Markov chain (Eq. 3.7), since the transition probabilities to $\mathcal{N}$ are nonzero for each of these nodes.

waiting time for the j-th node, $j \notin \mathcal{N}$, includes a contribution corresponding to the average of all $\mathcal{N}^{\mathsf{c}} \leftarrow \mathcal{N} \leftarrow j$ paths that leave j, enter $\mathcal{N}$, and exit to $\mathcal{N}^{\mathsf{c}}$. Because this contribution is not specific to which node $\mathcal{N}^{\mathsf{c}}$ is hit upon leaving $\mathcal{N}$, the GT algorithm (Eqs. 3.6 and 3.7) preserves the *average* MFPT to the *set* of absorbing nodes $\mathcal{A}$, and not the individual MFPTs to particular absorbing nodes $a \in \mathcal{A}$.[76,107] The renormalized $i \leftarrow j$ transition probabilities do preserve the probabilities of the (censored) paths to individual absorbing nodes, because the updated transition probabilities exactly account for the probability to transition from j to i via $\mathcal{N}$.[73–75,100]

The mean first passage path length for the $\mathcal{A} \leftarrow j \notin \mathcal{A}$ transition can be derived by direct analogy to the MFPT. In general, the two are not related for a CTMC parameterized by the branching probability matrix $\mathbf{P}$, because the mean waiting times for nodes are nonuniform.[88] For a path on the original Markov chain, an $i \leftarrow j$ transition increments the path length by one, and so the initial mean number of steps to exit node j, ℓ_j, are all unity. When nodes of the set $\mathcal{N}$ are eliminated, an $i \leftarrow j$ transition on the renormalized network also includes

steps taken within the censored region $\mathcal{N}$. Hence, the $\{\ell_j\}$ can be renormalized by analogy with Eq. 3.7, and the mean $\mathcal{A} \leftarrow j$ path length is then given by a relation analogous to Eq. 3.8, again with ℓ_j replacing τ_j.

We can also apply GT renormalization analogous to Eqs. 3.6 and 3.7 to eliminate nodes $n = 1, 2, \ldots, |\mathcal{Q}|-1$ one at a time, where $n \notin \mathcal{A}$.[73–78] In a variation of this iterative procedure, where all $|\mathcal{Q}|$ transient nodes are now to be eliminated, the network of the $(n-1)$-th iteration can be related to the network of the n-th iteration via

$$T_{ij}^{(n-1)} = T_{ij}^{(n)} - L_{nj}U_{in}, \tag{3.9}$$

where the matrix $\mathbf{L}$ has elements

$$L_{nj} = T_{nj}^{(n-1)}/(1 - T_{nn}^{(n-1)}), \tag{3.10}$$

and the matrix $\mathbf{U}$ has elements

$$U_{in} = T_{in}^{(n-1)} - \delta_{in}, \tag{3.11}$$

with δ_{in} the Kroenecker delta. In practice, the equivalence $1 - T_{nn} \equiv \sum_{\gamma \neq n} T_{\gamma n}$ is exploited to avoid subtraction operations and thus maintain numerical stability.[61–64] This iterative version of the GT algorithm can be thought of as a LU decomposition of a stochastic matrix.[90,108] The LU decomposition formulation of the GT algorithm (Eqs. 3.9-3.11) gives $T_{nj}^{(n)} = 0$ and $T_{in}^{(n)} = T_{in}^{(n-1)}/(1-T_{nn}^{(n-1)})$, thereby removing transitions to the eliminated node, n, and renormalizing the $i \leftarrow n$ transition probability to account for self-transitions. Hence, renormalization using Eqs. 3.9-3.11 preserves transitions from eliminated to noneliminated nodes, but not *vice versa*. Comparing Eqs. 3.7 and 3.10, the renormalized waiting time for the j-th node in the censored Markov chain at the n-th iteration can be written as

$$\tau_j^{(n)} = \tau_j^{(n-1)} + \tau_n^{(n-1)} L_{nj}. \tag{3.12}$$

The $\mathcal{A} \leftarrow j$ MFPT for the $j \equiv |\mathcal{Q}|$-th (transient) node, i.e. the last node to be eliminated, is therefore obtained straightforwardly as the associated renormalized waiting time in the censored Markov chain for which only the $|\mathcal{A}|$ absorbing nodes remain noneliminated. Although the preservation of transitions from eliminated nodes is not necessary to compute the $\mathcal{A} \leftarrow j$ MFPT, the formulation of the GT algorithm in Eqs. 3.9-3.11 can be used to compute other dynamical quantities such as committor probabilities, as we show in the following section.

3.2.3 Committor and absorption probabilities from graph transformation

If we define an initial macrostate $\mathcal{B} \subset \mathcal{A}^{\mathsf{c}}$, the transition probabilities for the renormalized Markov chain where all nodes of the intervening set $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$ have been eliminated by the LU decomposition formulation of the GT algorithm relate straightforwardly to the committor probabilities. Recall from Chapter 1 that the $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for the j-th node, q_j^+, is defined as the probability that a trajectory at node j first hits the target macrostate $\mathcal{A}$ before returning to the initial macrostate $\mathcal{B}$.[79–82] By definition, $q^+_{a\in\mathcal{A}} = 1$ and $q^+_{b\in\mathcal{B}} = 0$.[93] The committor probabilities of all other nodes satisfy a first-step relation,[77,96]

$$q_j^+ = \sum_{\gamma \notin \mathcal{B}} T_{\gamma j} q_\gamma^+. \tag{3.13}$$

For the renormalized Markov chain where all nodes of the set $\mathcal{I}$ have been eliminated according to Eqs. 3.9-3.11, with transition probabilities T'_{ij}, the only transitions from nodes of the set $\mathcal{I}$ are directly to either of the endpoint macrostates $\mathcal{A}$ or $\mathcal{B}$. The $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for the j-th node is therefore given straightforwardly by

$$q_j^+ = \sum_{a\in\mathcal{A}} T'_{aj} \equiv T'_{\mathcal{A}j} = 1 - T'_{\mathcal{B}j}. \tag{3.14}$$

An analogous expression yields the committor probabilities for the reverse ($\mathcal{B} \leftarrow \mathcal{A}$) direction.

The definition of the committor probability in Eq. 3.14 is readily extended to the case where there are multiple *taboo* macrostates that are forbidden to be visited.[83,84] Let us define the set of macrostates $\mathcal{H} \equiv \{\mathcal{H}_1 \cup \ldots \cup \mathcal{H}_N\} \subset \mathcal{S}$, which forms a subset of the complete state space. We wish to determine the probability of hitting a particular target macrostate $\mathcal{H}_k$ before hitting any of the taboo nodes of the set $\mathcal{H} \backslash \mathcal{H}_k$, when the process is initialized at node $j \in \mathcal{H}^{\mathsf{c}}$. The j-th node is associated with separate committor probabilities corresponding to each of the first passage processes defined by permuting the state $\mathcal{H}_k$ that is considered to be the target. We denote by $q_j^{\mathcal{H}_k}$ the committor probability for the $\mathcal{H}_k \leftarrow \mathcal{H}^{\mathsf{c}}$ transition, with all nodes of the set $\mathcal{H} \setminus \mathcal{H}_k$ considered taboo. These committor probabilities satisfy $q_j^{\mathcal{H}_1} + \ldots + q_j^{\mathcal{H}_N} = 1 \; \forall \, j$. The committor probabilities of all nodes with respect to all first passage processes can be determined efficiently by solving a single system of linear equations using the GT approach. That is, the transition probabilities of the renormalized network for which all nodes of the set $\mathcal{H}^{\mathsf{c}}$ have been eliminated using the LU decomposition formulation of the GT algorithm (Eqs. 3.9-3.11) yield the various committor probabilities via Eq. 3.14. We can then define a net reactive flux to the target state along an $i \leftarrow j$ edge for each first passage process.[109–113] Specifically, for an irreducible Markov chain at equilibrium, the $i \leftarrow j$

net flux to the target macrostate $\mathcal{H}_k$ when all nodes of the set $\mathcal{H} \setminus \mathcal{H}_k$ are taboo is

$$f_{ij}^{\mathcal{H}_k} = \begin{cases} \pi_j T_{ij}(q_i^{\mathcal{H}_k} - q_j^{\mathcal{H}_k}), & \text{if} \quad q_i^{\mathcal{H}_k} > q_j^{\mathcal{H}_k}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.15}$$

where π_j is the stationary (equilibrium occupation) probability of node j.

For an $\mathcal{A} \leftarrow \mathcal{B}$ transition, if we also eliminate the nodes $b \in \mathcal{B}$ of the initial macrostate according to Eqs. 3.9-3.11, thus leaving only the nodes of the absorbing macrostate $\mathcal{A}$, then the final renormalized transition probabilities are

$$T''_{aj} = T'_{aj} + T'_{aj}T'_{jj}/(1 - T'_{jj}) = T'_{aj}/T'_{\mathcal{A}j} \equiv B_{aj}. \tag{3.16}$$

Here, we have denoted the absorption (hitting) probability, i.e. the probability that trajectories initialized at the j-th transient node, $j \in \mathcal{Q}$, will be absorbed at the a-th absorbing node, $a \in \mathcal{A}$, by B_{aj}. The sum over absorbing nodes $a \in \mathcal{A}$ for B_{aj} is unity for all nodes j.

3.2.4 Extension of graph transformation with a backward pass phase

Following $|\mathcal{Q}| - 1$ renormalization steps of the standard formulation of the GT algorithm (Eqs. 3.6 and 3.7) to eliminate a single node at each iteration, the network only has a single noneliminated transient node $j \equiv |\mathcal{Q}| \notin \mathcal{A}$, and the $\mathcal{A} \leftarrow j$ MFPT is given by

$$\mathcal{T}_{\mathcal{A}j} = \frac{\tau_j^{(j-1)}}{1 - T_{jj}^{(j-1)}}. \tag{3.17}$$

Working backwards to undo the GT procedure, in the previous iteration the first-step relation (Eq. 3.3) gives

$$\begin{aligned} & \mathcal{T}_{\mathcal{A},j-1} = \tau_{j-1}^{(j-2)} + \mathcal{T}_{\mathcal{A},j-1}T_{j-1,j-1}^{(j-2)} + \mathcal{T}_{\mathcal{A}j}T_{j,j-1}^{(j-2)}, \\ \text{so} \quad & \mathcal{T}_{\mathcal{A},j-1} = \frac{\tau_{j-1}^{(j-2)} + \mathcal{T}_{\mathcal{A}j}T_{j,j-1}^{(j-2)}}{1 - T_{j-1,j-1}^{(j-2)}}, \end{aligned} \tag{3.18}$$

and we can therefore determine $\mathcal{T}_{\mathcal{A},j-1}$ from $\mathcal{T}_{\mathcal{A}j}$ if we save the necessary quantities from iteration $j - 2$. In general, we have the following expression to compute the MFPT for the $\mathcal{A} \leftarrow n$ transition, where n is the node that was eliminated at the n-th iteration of the

forward pass phase:

$$\mathcal{T}_{\mathcal{A}n} = \tau_n^{(n-1)} + \sum_{n \leq \gamma \leq |\mathcal{Q}|} \mathcal{T}_{\mathcal{A}\gamma} T_{\gamma n}^{(n-1)},$$

$$\text{or} \quad \mathcal{T}_{\mathcal{A}n} = \frac{\tau_n^{(n-1)} + \sum_{n < \gamma \leq |\mathcal{Q}|} \mathcal{T}_{\mathcal{A}\gamma} T_{\gamma n}^{(n-1)}}{1 - T_{nn}^{(n-1)}}. \tag{3.19}$$

Hence, we can work backwards and compute $\mathcal{T}_{\mathcal{A}n}$ from $\mathcal{T}_{\mathcal{A}\gamma}$ with $\gamma = n+1, n+2, \ldots, |\mathcal{Q}|$ if we save $\tau_n^{(n-1)}$ and $T_{\gamma n}^{(n-1)}$ for $n \leq \gamma \leq |\mathcal{Q}|$ during the forward pass phase. An analogous scheme can be written for the committor probabilities, which obey a first-step relation (Eq. 3.13) of the same form as that for MFPTs (Eq. 3.3).[93]

Using the LU decomposition formulation of the GT algorithm (Eqs. 3.9-3.11), we can write a more concise expression for the $\mathcal{A} \leftarrow n$ MFPT, where node n was eliminated at the n-th iteration of the forward pass phase. This expression requires the renormalized probabilities for transitions from, and waiting time for, the n-th node at the iteration where this node was eliminated, as well as the MFPTs for transitions from nodes that were eliminated after node n in the forward pass phase:

$$\mathcal{T}_{\mathcal{A}n} = \tau_n^{(n)} + \sum_{\gamma \notin \mathcal{A}} \mathcal{T}_{\mathcal{A}\gamma} T_{\gamma n}^{(n)}. \tag{3.20}$$

This equation follows from the fact that the $\gamma \leftarrow n$ transition probabilities for eliminated nodes $\gamma \leq n$ vanish in $\mathbf{T}^{(n)}$ by construction from Eq. 3.9.

The above derivation shows that we can calculate the MFPTs for *all* nonabsorbing nodes in a backward pass phase of the GT algorithm, by iteratively undoing the steps of the renormalization procedure and computing the MFPT for the newly restored node. This is the idea behind the extended GTH (EGTH) algorithm of Hunter.[94,114] As we have shown, it is not necessary to store the complete transition matrices at each iteration in the course of the forward (elimination) phase of the algorithm, and instead only a subset of waiting times and transition probabilities are required. Another convenient way to implement the backward pass phase of the algorithm that avoids excessive memory usage is to exploit the analogy between the GT algorithm and LU decomposition (Eq. 3.9), in which case the MFPTs for restored nodes are computed via Eq. 3.20. This procedure has the advantage of simultaneously yielding the committor and absorption probabilities via Eqs. 3.14 and 3.16, respectively. The overall procedure is given as pseudocode in Algorithm 5 and illustrated in Fig. 3.2. The steps of the GTH algorithm[66,67] to compute the stationary distribution of an irreducible Markov chain can also be readily incorporated into this procedure.

There are numerous practical factors to consider in optimizing the efficiency and memory usage of state reduction algorithms. Prioritizing renormalization of nodes with the lowest number of connections can speed up the calculation significantly.[76] The GT procedure in our `PATHSAMPLE` program uses a compressed row storage scheme at the start of a calculation for the MFPT, when the transition matrix is sparse. The GT renormalization adds non-zero probabilities as nodes are eliminated, and the program switches to dense storage when more than 2% of the elements became non-zero, if there are fewer than 11,000 remaining nodes. Our `DISCOTRESS` software[86] is designed similarly, employing a sparse data structure to keep memory requirements manageable and avoid unnecessary floating point operations when the network is large.

3.3 Expected number of node visits and node visitation probabilities for first passage and transition paths

3.3.1 Fundamental matrix of an absorbing Markov chain

Consider the substochastic $|\mathcal{Q}|\times|\mathcal{Q}|$-dimensional matrix $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$ whose elements are the probabilities for transitions within the set $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$. All nodes represented in $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$ must be transient. That is, it must be possible to reach the absorbing set of nodes $\mathcal{A}$ from any node in $\mathcal{Q}$. Then the inverse $\mathbf{N}_{\mathcal{Q}\mathcal{Q}} = \mathbf{I}_{\mathcal{Q}\mathcal{Q}} + \mathbf{T}_{\mathcal{Q}\mathcal{Q}} + \mathbf{T}^2_{\mathcal{Q}\mathcal{Q}} + \ldots = (\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \mathbf{T}_{\mathcal{Q}\mathcal{Q}})^{-1}$ exists, and is called the *fundamental matrix* of the absorbing Markov chain.[21] Since the fundamental matrix of a reducible Markov chain with $|\mathcal{Q}|$ transient nodes is always a $|\mathcal{Q}| \times |\mathcal{Q}|$-dimensional square matrix, in the following we will use the notation $\mathbf{N}$ for brevity. Note that the fundamental matrix $\mathbf{N}_{\mathcal{Q}\mathcal{Q}}$ is distinct from the $|\mathcal{S}| \times |\mathcal{S}|$-dimensional fundamental matrix of an irreducible Markov chain, $\mathbf{Z}$, introduced in Chapter 1, and that the interpretations of the elements of these matrices are not the same.

The element N_{ij} of the fundamental matrix is the expected number of times that the i-th node is visited along a first passage path initialized from node j.[115] Many more dynamical properties of interest can be written straightforwardly in terms of $\mathbf{N}$.[18] For example, the variance in the number of times that node i is visited prior to absorption when trajectories are initialized from node j is given by the relevant element of the matrix[18]

$$\mathbf{N}^{(2)} = \mathbf{N}(2\mathbf{N}_{\mathrm{d}} - \mathbf{I}) - (\mathbf{N} \circ \mathbf{N}), \tag{3.21}$$

where $\circ$ again denotes the Hadamard (i.e. element-wise) product, and $\mathbf{N}_{\mathrm{d}}$ is the matrix whose only nonzero elements are the diagonal elements of $\mathbf{N}$. A general expression for the n-th moment of this distribution, $\mathbf{N}^{(n)}$, is derived in Ref. 18. The probabilities H_{ij} that the

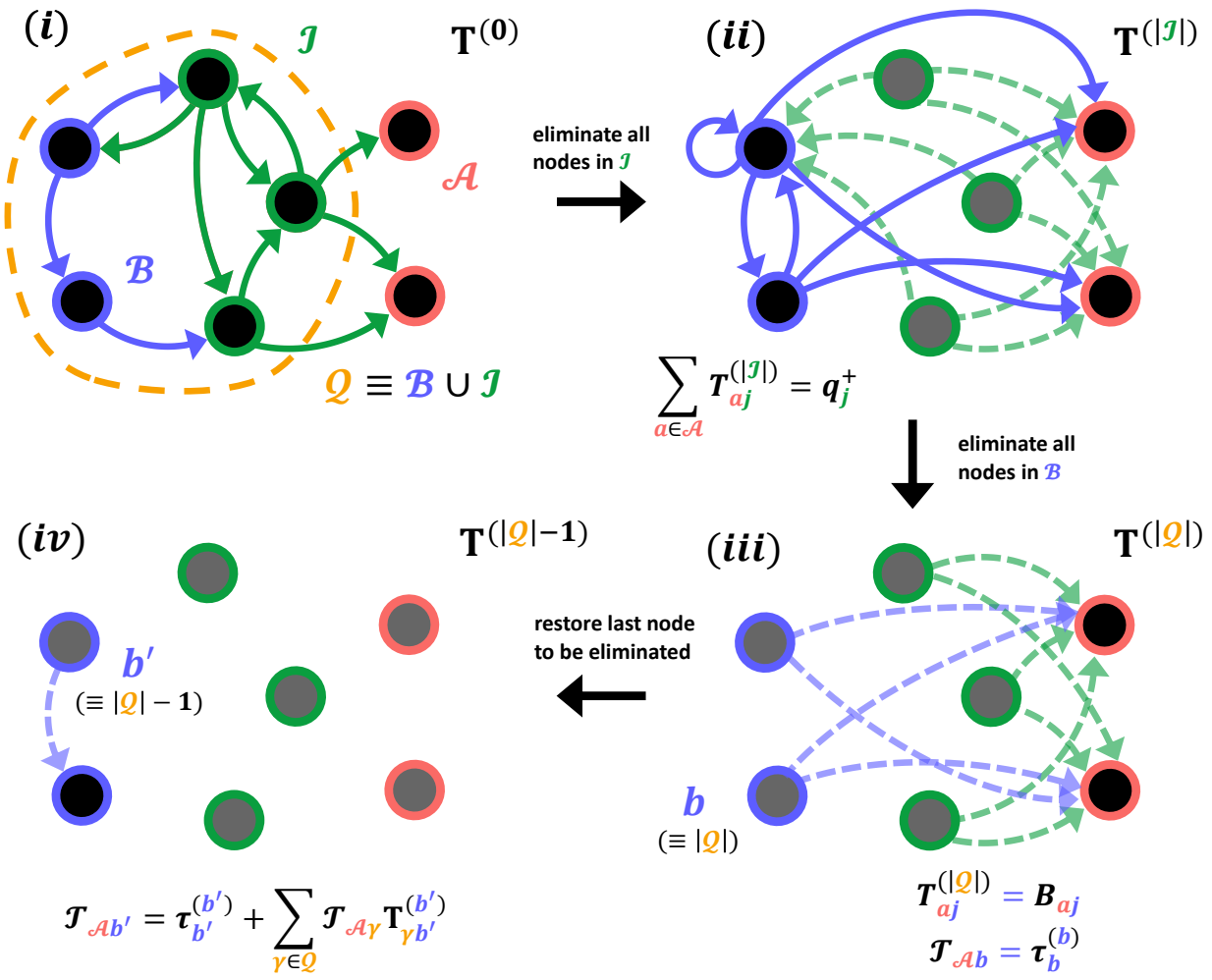


Figure 3.2: Illustration of the LU decomposition formulation of the graph transformation (GT) algorithm with a backwards pass phase (Algorithm 5), which computes the committor probabilities and MFPTs for all transient nodes, as well as the absorption probabilities. The steps of the GTH algorithm[66,67] to compute the stationary distribution can also be incorporated into this procedure. Nodes that have been eliminated by renormalization (see Fig. 3.1) are shown as transparent. LU decomposition (Eqs. 3.9-3.11) is used to renormalize transition probabilities, so that transitions from eliminated to noneliminated nodes are preserved (such connections are indicated by a transparent, dashed line). (i) A Markov chain for which the state space $\mathcal{S}$ is divided into the set of absorbing nodes $\mathcal{A}$ and the set of transient nodes $\mathcal{Q}$. The set of transient nodes is further divided into an initial macrostate, $\mathcal{B}$, and the set of intervening nodes, $\mathcal{I}$. (ii) In the first stage of the forward pass phase of the algorithm, all nodes of the state $\mathcal{I}$ are iteratively eliminated by renormalization (Eqs. 3.9-3.11), with the mean waiting (or lag) times for nodes renormalized according to Eq. 3.12. In the censored Markov chain where only nodes of the set $\mathcal{A} \cup \mathcal{B}$ remain noneliminated, the sum of transition probabilities from the j-th transient node to absorbing nodes is the $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for node j, q_j^+ (Eq. 3.13). (iii) In the remainder of the forward pass phase, the nodes of the initial state $\mathcal{B}$ are iteratively eliminated. In the censored network where only absorbing nodes remain noneliminated, the renormalized $i \leftarrow j$ transition probabilities from transient to absorbing nodes are the absorption probabilities B_{ij} (Eq. 3.16). The MFPT for the $\mathcal{A} \leftarrow b$ transition, where $b \in \mathcal{B}$ was the last node to be eliminated, is equal to the renormalized waiting time for the b-th node in the final censored Markov chain. (iv) In the backwards pass phase, eliminated nodes are iteratively restored using the $\mathbf{L}$ and $\mathbf{U}$ matrices that were constructed during the forward pass phase, and the MFPTs for transitions from transient nodes are computed by a recursive formula (Eq. 3.20). The figure shows the first step of this phase, in which the final node to be eliminated, b, for which the $\mathcal{A} \leftarrow b$ MFPT has previously been calculated, is restored. The $\mathcal{A} \leftarrow b'$ MFPT, where b' is the node that was eliminated before b, has two contributions. The first term is the renormalized mean waiting (or lag) time for the b'-th node in the censored Markov chain of the b'-th iteration. The second contribution corresponds to transitions to noneliminated transient nodes (here, $b \leftarrow b'$) of the relevant renormalized network.

i-th node is visited along first passage paths initialized from the j-th node, excluding the initial occupation of j, also follow directly from the N_{ij} elements. The mean number of visits to node i for such first passage paths, N_{ij}, must be equal to the probability of hitting node i, multiplied by the mean number of visits to i prior to absorption for paths starting from i, plus one if i is the initial node:

$$N_{ij} = \delta_{ij} + H_{ij}N_{ii} \quad \Rightarrow \quad H_{ij} = (N_{ij} - \delta_{ij})/N_{ii}. \tag{3.22}$$

In matrix form, the above condition is

$$\mathbf{H} = \mathbf{N}_{\mathrm{d}}^{-1}(\mathbf{N} - \mathbf{I}). \tag{3.23}$$

The absorption probabilities B_{ij} are the elements of the matrix $\mathbf{B} = \mathbf{T}_{\mathcal{A}\mathcal{Q}}\mathbf{N}$, where $\mathbf{T}_{\mathcal{A}\mathcal{Q}}$ is the matrix of probabilities for transitions from $\mathcal{Q}$ to $\mathcal{A}$. The absorption probabilities are nonzero only for $i \in \mathcal{A}, j \notin \mathcal{A}$.

The $\mathcal{A} \leftarrow \mathcal{B}$ MFPT can be obtained from $\mathbf{N}$ via

$$\mathcal{T}_{\mathcal{A}\mathcal{B}} = \sum_{j \in \mathcal{Q}} \sum_{b \in \mathcal{B}} p_b(0) N_{jb} \tau_j, \tag{3.24}$$

for an initial probability distribution $\mathbf{p}(0)$ localized in $\mathcal{B}$, $\sum_{b\in\mathcal{B}} p_b(0) = 1$. Higher moments of the FPT distribution can also be determined given the elements of the fundamental matrix. For a DTMC, the waiting times for nodes are fixed and equal to the lag time τ. It can be shown that the vector with elements $\mathcal{V}_{\mathcal{A}j}$, i.e. the variance of the FPT distribution for transitions from the j-th (transient) node, is[69]

$$\boldsymbol{\mathcal{V}}_{\mathcal{A}} = \Big((2\mathbf{N}^{\top} - \mathbf{I})\boldsymbol{\ell} - (\boldsymbol{\ell} \circ \boldsymbol{\ell})\Big)\tau^2, \tag{3.25}$$

where $\boldsymbol{\ell} = \mathbf{N}^{\top}\mathbf{1}_{\mathcal{Q}}$ is the vector of mean first passage path lengths, with $\mathbf{1}_{\mathcal{Q}}$ the $|\mathcal{Q}|$-dimensional column vector with all elements equal to unity. See Ref. 18 for a derivation. In the continuous-time case, Eq. 3.25 gives the variances of the FPT distributions for transitions from nodes of the Markov chain parameterized by the linearized transition probability matrix[108] $\mathbf{T}_{\mathrm{lin}}(\tau) = \mathbf{I} + \tau\mathbf{K}$, where $\tau \leq \min\{-K_{jj}^{-1} : \forall\, j\}$, for which the mean waiting times are uniform, $\tau_j \equiv \tau\ \forall\, j$.[115]

3.3.2 Fundamental matrix of an absorbing Markov chain computed using state reduction

Inversion of the Markovian kernel $\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \mathbf{T}_{\mathcal{Q}\mathcal{Q}}$, which is required to compute the fundamental matrix $\mathbf{N}$ of an absorbing Markov chain, is numerically unstable when the transition matrix features metastable macrostates. Therefore, as for the computation of MFPTs and committor probabilities in Markov chains exhibiting rare event dynamics (Sec. 3.2), we wish to devise an inherently stable algorithm to robustly compute $\mathbf{N}$, and hence many additional dynamical properties of interest. To this end, we define the augmented matrix

$$\mathbf{N}^* = \begin{pmatrix} \mathbf{T}_{\mathcal{Q}\mathcal{Q}} & \mathbf{I}_{\mathcal{Q}\mathcal{Q}} \\ \mathbf{I}_{\mathcal{Q}\mathcal{Q}} & \mathbf{0}_{\mathcal{Q}\mathcal{Q}} \\ \mathbf{T}_{\mathcal{A}\mathcal{Q}} & \mathbf{0}_{\mathcal{A}\mathcal{Q}} \end{pmatrix}, \tag{3.26}$$

where $\mathbf{0}_{\mathcal{Q}\mathcal{Q}}$ is the $|\mathcal{Q}| \times |\mathcal{Q}|$-dimensional null matrix. Evidently, $\mathbf{N}^*$ does not relate to a stochastic matrix, since the column sums corresponding to transition probabilities from transient nodes necessarily exceed unity. Nonetheless, if we proceed to compute the analogue of the stochastic complement (Eq. 3.6) corresponding to the remaining network when all nodes represented in $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$ are eliminated from $\mathbf{N}^*$, then we obtain the fundamental matrix $\mathbf{N}$.

Specifically, the proposed state reduction algorithm to compute the fundamental matrix $\mathbf{N}$ of an absorbing Markov chain is as follows. For each of the transient nodes in the network, of the set $\mathcal{Q}$, we introduce a dummy partner node. Thus we have the augmented state space $\mathcal{S}^* \equiv \mathcal{S} \cup \mathcal{Q}^*$, where $\mathcal{Q}^*$ denotes the set of dummy nodes, with $|\mathcal{Q}^*| \equiv |\mathcal{Q}|$. Each dummy node is connected to its transient partner by forward and reverse edges, both with weights equal to unity (*cf.* Eq. 3.26). When all of the transient nodes have been eliminated, via the analogue of a stochastic complement (Eq. 3.6), the weights of the $i \leftarrow j$ edges in the remaining network that correspond to transitions between dummy nodes are the elements N_{ij} of the fundamental matrix $\mathbf{N}$. In the nodewise iterative formulation of this procedure, upon eliminating a single transient node n, the edge weights for transitions between all remaining nodes in the augmented network are updated according to

$$N^*_{ij} \leftarrow N^*_{ij} + \frac{N^*_{in} N^*_{nj}}{\sum_{\gamma \in \mathcal{S} \setminus \{n\}} N^*_{\gamma n}} \; \forall \, i, j \in \mathcal{S}^* \setminus \{n\}. \tag{3.27}$$

This state reduction procedure is illustrated in Fig. 3.3 and given as pseudocode in Algorithm 6. Eq. 3.27 shows the advantage of including the absorbing nodes in the augmented state space. In particular, if the probabilities for transitions from transient to absorbing nodes are

renormalized in the course of the algorithm, then the total probabilities for transitions from transient to non-dummy nodes remain conserved, and equal to unity. Hence, since only the transient nodes are eliminated, all subtraction operations can be avoided by exploiting the relation $1 - T_{nn} = \sum_{\gamma \neq n} T_{\gamma n}$, where $\mathbf{T}$ denotes the stochastic matrix of the censored Markov chain comprised by the noneliminated nodes of the state space $\mathcal{S}$. Using this trick, the state reduction algorithm described above is numerically stable.[61–64]

The theory of stochastic complements presented in Sec. 3.2.2 and reviewed in Chapter 1 can be leveraged to design a block formulation of this state reduction algorithm, wherein multiple nodes are eliminated simultaneously.[100] Let us consider the elimination of a set of transient nodes $\mathcal{N} \subseteq \mathcal{Q}$, and denote the set of all remaining nonabsorbing nodes in the augmented state space as $\mathcal{Z} \equiv \mathcal{S}^* \setminus (\mathcal{A} \cup \mathcal{N})$. The augmented matrix (Eq. 3.26) is then updated according to (*cf.* Eq. 3.6)

$$\mathbf{N}^*_{\mathcal{Z}\mathcal{Z}} \leftarrow \mathbf{N}^*_{\mathcal{Z}\mathcal{Z}} + \mathbf{N}^*_{\mathcal{Z}\mathcal{N}} (\mathbf{I}_{\mathcal{N}\mathcal{N}} - \mathbf{T}_{\mathcal{N}\mathcal{N}})^{-1} \mathbf{N}^*_{\mathcal{N}\mathcal{Z}}, \tag{3.28}$$

where we have used the notation $\mathbf{N}^*_{\mathcal{Z}\mathcal{Z}}$ to explicitly indicate the dimensionality of the augmented network. For $\mathcal{N} \equiv \{n\}$, Eq. 3.28 reduces to Eq. 3.27. After eliminating all transient nodes, the resulting matrix $\mathbf{N}^*_{\mathcal{Z}\mathcal{Z}}$ is the fundamental matrix $\mathbf{N}$. This procedure is numerically stable if the blocks of nodes to be eliminated, $\mathcal{N}$, correspond to metastable macrostates, so that the matrix inversion operations for the Markovian kernels $\mathbf{I}_{\mathcal{N}\mathcal{N}} - \mathbf{T}_{\mathcal{N}\mathcal{N}}$ are not associated with significant numerical error. This formulation of the algorithm therefore requires a careful partitioning of the nodes into appropriate communities, but leads to improved time complexity.[107,116]

3.3.3 Reactive and nonreactive segments of the first passage path ensemble

Knowledge of the committor probabilities, $\{q_j^+\}$ (Eq. 3.14), and the fundamental matrix of the absorbing Markov chain, $\mathbf{N}$ (Sec. 3.3.1), can be exploited to divide the $\mathcal{A} \leftarrow \mathcal{B}$ first passage path ensemble (FPPE) into nonreactive ($\mathcal{B} \leftarrow \mathcal{B}$) and reactive (direct $\mathcal{A} \leftarrow \mathcal{B}$) segments (Fig. 3.4a).[17] This division allows for a more detailed analysis of the FPPE at a nodewise level of detail, beyond the results outlined in Sec. 3.3.1. The reactive segments of the FPPE, which correspond to the transition path ensemble (TPE),[109–113] are particularly insightful to understand the characteristics of the productive $\mathcal{A} \leftarrow \mathcal{B}$ process. In Sec. 3.3.4, we will derive novel analytical results for key dynamical properties characterizing the influence of individual nodes on direct $\mathcal{A} \leftarrow \mathcal{B}$ transitions, such as the probability that a particular node is visited along a reactive (transition) path (Eq. 3.44). Since we have shown that both the $\{q_j^+\}$ and $\mathbf{N}$ can be computed robustly by state reduction methods (Secs. 3.2.3 and

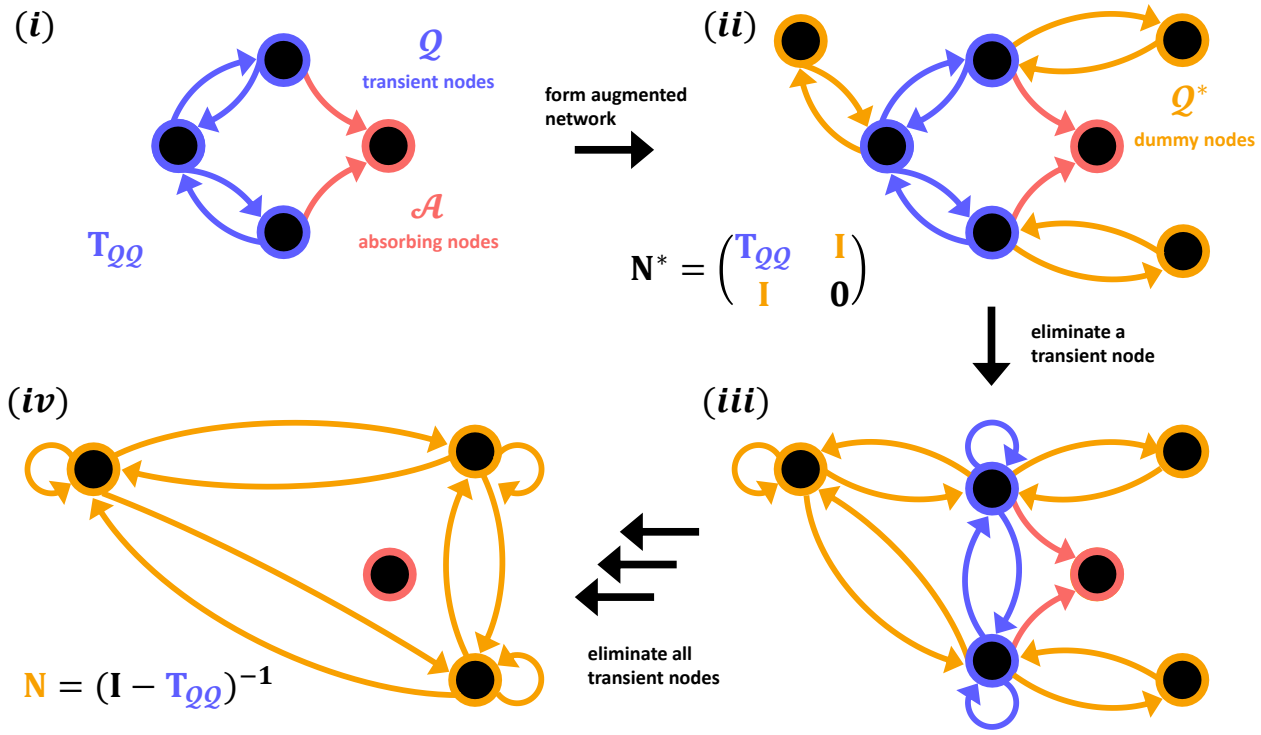


Figure 3.3: Illustration of the numerically stable state reduction procedure to compute the fundamental matrix $\mathbf{N}$ for an absorbing Markov chain, effectively performing a matrix inversion operation on the Markovian kernel $\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \mathbf{T}_{\mathcal{Q}\mathcal{Q}}$. (i) The state space $\mathcal{S} \equiv \mathcal{Q} \cup \mathcal{A}$ of the Markov chain is divided into sets of transient and absorbing nodes, denoted $\mathcal{Q}$ and $\mathcal{A}$, respectively. The substochastic matrix $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$ only includes transition probabilities between transient nodes (blue), and does not include absorbing nodes (red). (ii) Dummy nodes (yellow), of the set $\mathcal{Q}^*$, are partnered with transient nodes via forward and reverse edges with weights equal to unity. (iii) State reduction (*cf.* Eq. 3.6) is used to eliminate transient nodes either iteratively (Eq. 3.27) or in blocks (Eq. 3.28). The updated $i \leftarrow j$ edge weights account for paths that proceed via the eliminated nodes. Transitions from dummy to absorbing nodes do not have a meaningful interpretation and are not required in the algorithm, so can be ignored. (iv) The $i \leftarrow j$ edge weights in the network where only the dummy nodes remain are the elements N_{ij} of the fundamental matrix $\mathbf{N}$.

3.3.2, respectively), we can likewise compute the derived properties by a numerically stable route. In the remainder of the current section, we formally introduce the factorization of first passage paths into reactive and nonreactive trajectory segments.

The expected number of times that a nonabsorbing node is visited along a first passage path is simply an average of the mean number of visits when starting from the initial state $\mathcal{B} \subseteq \mathcal{Q}$, taken over the initial occupation probability distribution localized within this set:

$$\theta_j = \sum_{b \in \mathcal{B}} p_b(0) N_{jb} \quad \forall\, j \in \mathcal{Q}, \tag{3.29}$$

with $\sum_{b \in \mathcal{B}} p_b(0) = 1$. Absorbing nodes can only be visited once along a particular first passage path, with probability B_{aj}, and the average over the initial distribution is

$$\theta_a = \sum_{j \notin \mathcal{A}} p_j(0) B_{aj} \quad \forall\, a \in \mathcal{A}. \tag{3.30}$$

The first-step relation for the elements of the fundamental matrix of the absorbing Markov chain, which includes only transient nodes of the set $\mathcal{Q}$, is[18]

$$N_{ij} = \delta_{ij} + \sum_{\gamma \in \mathcal{Q}} [\mathbf{T}_{\mathcal{Q}\mathcal{Q}}]_{\gamma j} N_{i\gamma}. \tag{3.31}$$

Note that this expression does not have the same form as the first-step relations for the MFPTs, committor probabilities, or absorption probabilities (*cf.* Eq. 3.3), and therefore the state reduction algorithms presented in Secs. 3.2.2-3.2.4 cannot be used to compute the $\{\theta_j\}$. Since $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}\mathbf{N} = \mathbf{N}\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$, which follows from writing $\mathbf{N}$ as a geometric progression in $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$, we can rewrite the first-step relation (Eq. 3.31) and sum over the initial distribution within $\mathcal{B}$ to obtain

$$\begin{aligned} \sum_{b \in \mathcal{B}} p_b(0) N_{jb} &= \sum_{b \notin \mathcal{B}} p_b(0) \delta_{jb} + \sum_{\gamma \in \mathcal{Q}} [\mathbf{T}_{\mathcal{Q}\mathcal{Q}}]_{j\gamma} \sum_{b \in \mathcal{B}} p_b(0) N_{\gamma b} \\ \text{so} \quad \theta_j &= p_j(0) + \sum_{\gamma \in \mathcal{Q}} [\mathbf{T}_{\mathcal{Q}\mathcal{Q}}]_{j\gamma} \theta_\gamma \quad \forall\, j \in \mathcal{Q}. \end{aligned} \tag{3.32}$$

The absorption probability matrix is $\mathbf{B} = \mathbf{T}_{\mathcal{A}\mathcal{Q}}\mathbf{N}$, so for absorbing nodes we have

$$\begin{aligned} \theta_a &= \sum_{b \in \mathcal{B}} p_b(0) B_{ab} = \sum_{\gamma \in \mathcal{Q}} [\mathbf{T}_{\mathcal{A}\mathcal{Q}}]_{a\gamma} \sum_{b \in \mathcal{B}} p_b(0) N_{\gamma b} \\ &= \sum_{\gamma \in \mathcal{Q}} [\mathbf{T}_{\mathcal{A}\mathcal{Q}}]_{a\gamma} \theta_\gamma \quad \forall\, a \in \mathcal{A}. \end{aligned} \tag{3.33}$$

Hence, the $\{\theta_j\}$ satisfy the following system of linear equations

$$\theta_j = p_j(0) + \sum_{\gamma \in \mathcal{Q}} T_{j\gamma}\theta_\gamma \quad \forall\, j \in \mathcal{S}, \tag{3.34}$$

where $p_j(0) = 0$ for $j \notin \mathcal{B}$. Eq. 3.34 can be solved directly by standard linear algebra methods, but the $\{\theta_j\}$ are most robustly determined via Eqs. 3.29 and 3.30 when $\mathbf{N}$ is computed using the state reduction algorithm described in Sec. 3.3.2.

We can now break down properties of the FPPE into contributions from reactive and nonreactive path segments.[17,112,117] Recall that nodes not belonging to endpoint states are members of the intervening set $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$. A reactive path from $\mathcal{B}$ is one that leaves $\mathcal{B}$ and reaches $\mathcal{A}$ without returning to $\mathcal{B}$.[110] Nonreactive paths contain nodes from $\mathcal{B} \cup \mathcal{I}$ in first passage path segments starting in $\mathcal{B}$ up to the final escape from $\mathcal{B}$ before reaching $\mathcal{A}$. The average numbers of visits to the j-th node along nonreactive and reactive paths, $\bar{\theta}_j$ and $\widetilde{\theta}_j$, respectively, are given by[17]

$$\bar{\theta}_j = \theta_j(1 - q_j^+), \tag{3.35a}$$

$$\widetilde{\theta}_j = \mu_j + \theta_j q_j^+, \tag{3.35b}$$

where

$$\mu_j = 1_{\mathcal{B}}(j)\bar{\theta}_j \sum_\gamma T_{\gamma j} q_\gamma^+, \tag{3.36}$$

is the probability that a reactive path left the initial state $\mathcal{B}$ from node $j \in \mathcal{B}$. Here, $1_{\mathcal{B}}(j)$ is the indicator function for the initial region, equal to unity for $j \in \mathcal{B}$ and zero otherwise, which ensures that the initial probability distribution in Eq. 3.36 is contained within $\mathcal{B}$. Let $\partial\mathcal{B} \subseteq \mathcal{B}$ denote the boundary nodes of the initial set, i.e. nodes of the initial set for which a direct transition to a node of the set $\mathcal{B}^{\mathsf{c}}$ exists. Then $\sum_{b \in \partial\mathcal{B}} \mu_b = 1$ and $\mu_j = 0\ \forall\, j \notin \partial\mathcal{B}$. Eq. 3.35b simply states that the expected number of times that a *reactive* trajectory, beginning at the boundary $\partial\mathcal{B}$ of the initial state $\mathcal{B}$, visits a node $j \notin \mathcal{B}$ is the product of the expected number of times that any first passage trajectory visits the j-th node and the probability that a trajectory initialized from node j is reactive.

The decomposition of the FPPE into reactive and nonreactive segments allows for analysis of the individual nodes and edges of the Markovian network that make significant and productive contributions to the $\mathcal{A} \leftarrow \mathcal{B}$ process. The flux along the $i \leftarrow j$ edge of the network is defined as $J_{ij} = \theta_j T_{ij}$.[17] This flux can also be split into nonreactive and reactive

contributions $\bar{J}_{ij}$ and $\widetilde{J}_{ij}$, respectively,

$$J_{ij} \equiv \theta_j T_{ij} = \bar{J}_{ij} + \widetilde{J}_{ij}, \tag{3.37}$$

where the reactive flux along the $i \leftarrow j$ edge is given by[17]

$$\widetilde{J}_{ij} = \frac{\widetilde{\theta}_j T_{ij} q_i^+}{\sum_\gamma T_{\gamma j} q_\gamma^+}, \tag{3.38}$$

for $i \in \mathcal{I} \cup \mathcal{A}, j \in \mathcal{I}$, and when the set $\mathcal{A}$ is reachable from both nodes i and j. Here, 'reachable' means that a path to $\mathcal{A}$ exists that passes through the $\mathcal{I}$ set without hitting $\mathcal{B}$. Other than this condition, the derivation of Eqs. 3.34-3.38 does not assume that the Markov chain is ergodic.[17] $\widetilde{J}_{ij}$ is essentially the nonequilibrium analogue of the stationary (i.e. equilibrium) reactive flux f_{ij}^+ (Eq. 3.15) that we first introduced in Chapter 1. In Chapter 2, we factorized the total reactive flux in the equilibrium TPE into contributions from simple transition flux-paths, and determined the dominant flux-paths using a shortest paths algorithm with edge weights based on the $\{f_{ij}^+\}$. We can perform the same analysis for the nonequilibrium case using edge weights based on the $\{\widetilde{J}_{ij}\}$ (Eq. 3.38).

3.3.4 Analysis of reactive paths

A further key dynamical property characterizing the ensemble of $\mathcal{A} \leftarrow \mathcal{B}$ transition (i.e. reactive) paths[112] is the conditional probability that the i-th node is visited along a trajectory initialized from node j when the trajectory is reactive. We shall denote this quantity by $\widetilde{H}_{ij}$. To simplify the notation in deriving this probability, we assume that there are no nodes in the set $\mathcal{I}$ from which the absorbing macrostate $\mathcal{A}$ is not reachable, since such nodes do not contribute to the reactive segment of the FPPE. Similarly, it is not necessary to consider nodes of the set $\mathcal{B} \setminus \partial\mathcal{B}$. If there are no such internal initial nodes, so that $\partial\mathcal{B} \equiv \mathcal{B}$, then we have $|Q|$ nodes in the relevant set of transient nodes $\partial\mathcal{B} \cup \mathcal{I} \subseteq \mathcal{Q}$. For brevity, we shall assume this to be the case, and we therefore consider the $|Q| \times |Q|$-dimensional substochastic matrix $\mathbf{T}_{\mathcal{Q}\mathcal{Q}}$.

To derive the $\widetilde{H}_{ij}$ probabilities, we introduce the substochastic transition probability matrix for the reactive process on the set of (relevant) transient nodes, $\widetilde{\mathbf{T}}_{\mathcal{Q}\mathcal{Q}}$.[117] We also define the $|Q|$-dimensional vector of committor probabilities for the relevant transient nodes, $\mathbf{q}_\mathcal{Q}^+$, and the modified committor probability vector $\mathbf{q}_\mathcal{Q}^{+\prime}$, for which the elements corresponding to initial boundary nodes are non-zero, equal to $q_{b\in\partial\mathcal{B}}^{+\prime} = \sum_{\gamma\notin\mathcal{B}} T_{\gamma b} q_\gamma^+$. This probability is the probability that a trajectory is absorbed before *hitting* any node of the set $\mathcal{B}$ (*cf.* Eq. 3.13).[76]

Then the reactive transition probability matrix for transient nodes is[17]

$$\widetilde{\mathbf{T}}_{\mathcal{Q}\mathcal{Q}} = \mathrm{diag}(\mathbf{q}_{\mathcal{Q}}^{+})\mathbf{T}_{\mathcal{Q}\mathcal{Q}}\mathrm{diag}(\mathbf{q}_{\mathcal{Q}}^{+\prime})^{-1}, \tag{3.39}$$

and can be evaluated robustly by using a state reduction algorithm to compute the committor probabilities (Sec. 3.2.3). We note again that internal initial nodes are discarded in this representation, since such nodes do not contribute to the reactive segment of the FPPE, i.e. $q^{+\prime}_{b\in\mathcal{B}\setminus\partial\mathcal{B}} = 0$. The corresponding fundamental matrix for the reactive process is

$$\widetilde{\mathbf{N}}_{\mathcal{Q}\mathcal{Q}} = (\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \widetilde{\mathbf{T}}_{\mathcal{Q}\mathcal{Q}})^{-1}, \tag{3.40}$$

and can be computed using the numerically stable state reduction algorithm derived in Sec. 3.3.2. We shall henceforth omit the dimensionality subscripts from $\widetilde{\mathbf{N}}$ for notational simplicity. The fundamental matrix for the reactive process provides a natural means to express the expected number of visits to a transient node along a reactive path:

$$\widetilde{\theta}_j = \sum_{b\in\partial\mathcal{B}} \mu_b \widetilde{N}_{jb} \quad \forall\, j \in \mathcal{Q}. \tag{3.41}$$

Recall that for absorbing nodes we simply have $\widetilde{\theta}_a = \theta_a \;\forall\, a \in \mathcal{A}$. By analogy with the visitation probability matrix $\mathbf{H}$ associated with the FPPE (Eq. 3.23), we can calculate $\widetilde{H}_{ij}$, the probability that a *reactive* trajectory will ever visit node i if it starts at node j for $i, j \in \mathcal{Q}$, not counting the occupancy of the initial node:

$$\widetilde{N}_{ij} = \delta_{ij} + \widetilde{H}_{ij}\widetilde{N}_{ii}. \tag{3.42}$$

Hence, we obtain the matrix $\widetilde{\mathbf{H}}$:

$$\widetilde{\mathbf{H}} = \widetilde{\mathbf{N}}_{\mathrm{d}}^{-1}(\widetilde{\mathbf{N}} - \mathbf{I}). \tag{3.43}$$

Thus the reactive visitation probabilities $\widetilde{H}_{ij}$ can be determined robustly by using state reduction algorithms to compute the committor probability vector, $\mathbf{q}_{\mathcal{Q}}^{+}$, and the fundamental matrix for the reactive process, $\widetilde{\mathbf{N}}$.

We can similarly define a substochastic transition matrix corresponding to the nonreactive process on the set of transient nodes. This Markov chain is constructed so that 'absorption' corresponds to the first passage trajectory hitting a node at the boundary of the initial state for the final time, after which point the trajectory proceeds to be reactive.[17] Fundamental and visitation probability matrices for the nonreactive segment of the FPPE then follow by

an analogous argument to the reactive case.

The probability that the j-th non-initial transient node is visited along a reactive $\mathcal{A} \leftarrow \mathcal{B}$ transition path,[118] r_j^+, is an average of the $\widetilde{H}_{jb}$ elements with respect to the initial occupation probability distribution for reactive trajectories $\boldsymbol{\mu}$ (Eq. 3.36),

$$r_j^+ = \sum_{b \in \partial\mathcal{B}} \mu_b \widetilde{H}_{jb} \quad \forall\, j \in \mathcal{Q} \setminus \partial\mathcal{B}. \tag{3.44}$$

For initial boundary nodes $b \in \partial\mathcal{B}$, $\widetilde{H}_{bj} = 0\ \forall\, j$, since initial nodes cannot be revisited along reactive paths by definition. The probability that a node at the boundary of the initial state appears along a reactive path is therefore simply $r_b^+ = \mu_b$. The probability that an absorbing node $a \in \mathcal{A}$ appears along a reactive path is an average of the elements of the absorption probability matrix (Eq. 3.16) weighted by the $\boldsymbol{\mu}$ distribution; $r_a^+ = \sum_{b \in \partial\mathcal{B}} \mu_b B_{ab}$.

The reactive visitation probability r_j^+ provides detailed characterization of the FPPE at a microscopic level of detail. Nodes that have a high probability of being visited along reactive trajectories are those that mediate the dominant pathways for the overall productive transition. For an effective two-state system, nodes that are associated with a high r_j^+ probability, and which also have values for the committor probability q_j^+ close to 0.5, represent the dynamical bottleneck region of the network.[119–121] That is, these nodes constitute the transition state ensemble (TSE).[122] Global dynamical quantities, including the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT,[123] are most sensitive to perturbations in the transition probabilities associated with these bottleneck nodes.[124,125] Therefore the reactive visitation and committor probabilities are the central objects in understanding how the local dynamics at a small subset of nodes, namely the TSE, modulate the slow, macroscopic dynamics. In general, for systems exhibiting multiple metastable macrostates, there are multiple TSEs that are the boundary regions between the metastable states, across which the committor probability changes sharply.

The visitation probability of the j-th node along reactive trajectories in Eq. 3.44 corresponds to the *nonequilibrium* TPE.[17] If the set of initial boundary nodes $\partial\mathcal{B}$ contains more than one node, then Eq. 3.44 differs from the result when the system is at a steady state, i.e. corresponding to the *equilibrium* TPE.[112] The two path ensembles are illustrated schematically in Fig. 3.4. The reactive component of the steady state TPE is the object of study in transition path theory (Chapter 1). In the steady state regime, which exists if the Markov chain is irreducible,[100] the $\mathcal{A} \leftarrow \mathcal{B}$ path ensemble has relaxed to equilibrium. The probability μ_j^{SS} that the reactive portion of trajectories began after the nonreactive trajectory segment

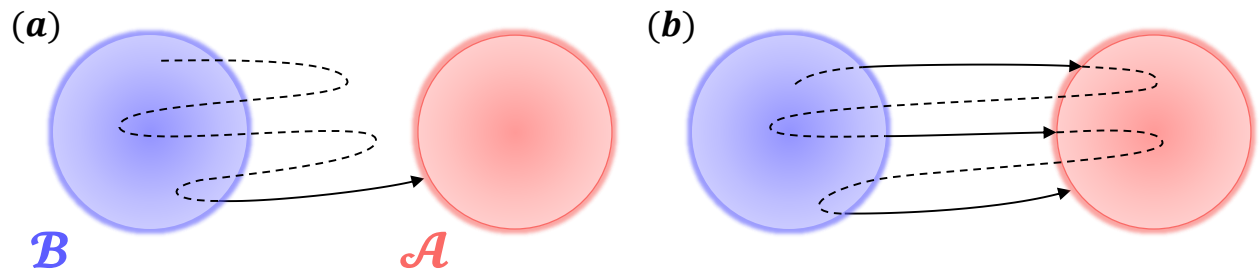


Figure 3.4: Schematic depiction of the nonequilibrium and equilibrium (i.e. steady state) $\mathcal{A} \leftarrow \mathcal{B}$ first passage and transition path ensembles (FPPE and TPE, respectively). (a) Trajectories of the nonequilibrium FPPE start within the initial state $\mathcal{B}$ and are absorbed upon hitting the target state $\mathcal{A}$. The TPE (solid line) is the portion of the FPPE that transitions directly to $\mathcal{A}$ from $\mathcal{B}$ without revisiting $\mathcal{B}$. (b) For an irreducible Markov chain, we can also consider the equilibrium FPPE and TPE, which result from considering an infinitely long trajectory that continually transitions between the $\mathcal{B}$ and $\mathcal{A}$ states. The steady state $\mathcal{A} \leftarrow \mathcal{B}$ TPE is the set of path segments that transition directly from $\mathcal{B}$ to $\mathcal{A}$ at equilibrium. The steady state MFPT is the inverse of the rate at which trajectories that last visited the initial state hit the target state.

hit node j is therefore dependent on the stationary distribution,[17]

$$\mu_j^{\mathrm{SS}} \propto 1_{\mathcal{B}}(j)\pi_j \sum_{\gamma} T_{\gamma j} q_{\gamma}^{+}. \tag{3.45}$$

Similar to the nonequilibrium case (Eq. 3.36), this initial distribution for reactive trajectories at steady state satisfies $\sum_{b\in\partial\mathcal{B}} \mu_b^{\mathrm{SS}} = 1$ and $\mu_j^{\mathrm{SS}} = 0 \; \forall \, j \notin \partial\mathcal{B}$. That is, this distribution is localized at the boundary of the initial state. The visitation probability of the j-th non-initial transient node along reactive trajectories for the equilibrium TPE is a weighted average of the elements of the $\widetilde{\mathbf{H}}$ matrix (Eq. 3.43) with respect to this initial occupation probability distribution,

$$r_j^{+,\mathrm{SS}} = \sum_{b\in\partial\mathcal{B}} \mu_b^{\mathrm{SS}} \widetilde{H}_{jb} \quad \forall \, j \in \mathcal{Q} \setminus \partial\mathcal{B}. \tag{3.46}$$

In addition, $r_b^{+,\mathrm{SS}} = \mu_b^{\mathrm{SS}} \; \forall \, b \in \partial\mathcal{B}$ and $r_a^{+,\mathrm{SS}} = \sum_{b\in\partial\mathcal{B}} \mu_b^{\mathrm{SS}} B_{ab} \; \forall \, a \in \mathcal{A}$. Similarly, the average number of times that the transient node j is visited along reactive trajectories at steady state is a weighted average of the elements of the $\widetilde{\mathbf{N}}$ matrix (Eq. 3.40) with respect to the $\boldsymbol{\mu}^{\mathrm{SS}}$ distribution,

$$\widetilde{\theta}_j^{\mathrm{SS}} = \sum_{b\in\partial\mathcal{B}} \mu_b^{\mathrm{SS}} \widetilde{N}_{jb} \quad \forall j \in \mathcal{Q}, \tag{3.47}$$

and for absorbing nodes we have $\widetilde{\theta}_a^{\mathrm{SS}} = \sum_{b\in\partial\mathcal{B}} \mu_b^{\mathrm{SS}} B_{ab} \; \forall \, a \in \mathcal{A}$.

Recall that within the state reduction formalism, a single linear system of equations can be solved to obtain the set of committor probabilities $\{\mathbf{q}^{\mathcal{H}_1}, \ldots, \mathbf{q}^{\mathcal{H}_N}\}$ robustly, where each committor probability vector $\mathbf{q}^{\mathcal{H}_k}$ is associated with a different target macrostate $\mathcal{H}_k \in \mathcal{H}$,

conditioned on all nodes of the set $\mathcal{H} \setminus \mathcal{H}_k$ being taboo (Sec. 3.2.3). This formulation allows us to compute all the dynamical properties that we have derived relating to the reactive segments of the FPPEs for alternative $\mathcal{H}_k \leftarrow \mathcal{H}^c$ transitions. This result is useful if, for example, we want to analyze transition paths associated with a particular sequence of events, which form a subset of the ensemble of all paths transitioning to the target state. This analysis can be achieved by setting states that are not involved in the paths of interest to be taboo. For instance, a Markov chain representing the folding transition of a protein may feature several competing mechanisms that can be distinguished on the basis of the intermediate metastable states that are visited.[27,126–128] By designating a particular intermediate state to be taboo, we can investigate the TPE specifically for transitions that proceed via alternative intermediate states.

3.4 Numerical results

We demonstrate the methodology outlined in Secs. 3.2 and 3.3 with numerical results for a kinetic network representing a structural transition for a cluster of 38 atoms bound by the Lennard-Jones potential (LJ_{38}).[129,130] Specifically, we consider the transition from a structure based on an incomplete Mackay icosahedron (I_h) to a face-centered cubic (F) geometry, which was also analyzed in Ref. 38. The network model was constructed by mapping the local minima and transition states of the underlying potential energy landscape to the nodes and edges of a CTMC, which consists of 885 nodes and 1126 bidirectional edges. The face-centered cubic (F) state is represented by the single node of the Markov chain with the largest stationary probability (lowest free energy), and the icosahedral (I_h) state is represented by the single node with lowest free energy belonging to a separate funnel on the landscape. Because these two competing low-energy nodes are separated by a large energy barrier, the $F \leftarrow I_h$ solid-solid transition becomes an increasingly rare event[59] with decreasing temperature. We employ standard reduced units for the LJ potential in the following analysis.[129,130]

We analyze the Markov chain parameterized at a temperature of $T = 0.12$, which approximately coincides with the start of the regime where the kinetic network exhibits significant metastability. At this temperature, the number of internode transitions in $F \leftarrow I_h$ first passage paths is typically 10^8 or 10^9, precluding the use of the standard kinetic Monte Carlo[131] (kMC) algorithm to sample the FPPE (see also Chapter 4). Moreover, dense linear algebra methods to perform an eigendecomposition of the Markov chain, to invert the Markovian kernel $\mathbf{I}_{\mathcal{Q}\mathcal{Q}} - \mathbf{T}_{\mathcal{Q}\mathcal{Q}}$ (required to compute the fundamental matrix $\mathbf{N}$ and the stochastic complement (Eq. 3.6)), or to solve relevant linear systems of equations (Eqs. 3.3,

3.13, and 3.34) suffer from a severe propagation of numerical error arising from finite precision.[116] Similarly, iterative sparse linear algebra methods[42] fail to converge, as discussed in Chapter 1. Computational analysis of this Markov chain is therefore intractable without employing the state reduction algorithms described in Secs. 3.2 and 3.3.

Fig. 3.5 shows the results of various state reduction calculations to robustly compute the salient dynamical properties associated with individual nodes of the kinetic network, for the $F \leftarrow I_h$ transition. The Markov chain is visualized as a disconnectivity graph,[132] where the interconvertibility of sets of nodes in the network is considered at decreasing threshold increments representing the available energy. A fork in the graph indicates that a transition between the sets of nodes requires energy exceeding the threshold, and the branches terminate at the energies of the corresponding nodes. MFPTs to the F state and committor probabilities were computed by the iterative LU decomposition formulation of the GT algorithm with a backward pass phase (Algorithm 5). The expected number of node visits along reactive paths, $\widetilde{\theta}_j$ (Eq. 3.41), were obtained from the fundamental matrix for the reactive process (Eq. 3.40), computed using the state reduction procedure given in Algorithm 6. The reactive visitation probabilities for nodes were determined from the elements of the reactive fundamental matrix and from the committor probabilities via Eqs. 3.39-3.44. The exact results from the state reduction calculations were verified numerically by comparison with kinetic path sampling (kPS) simulations,[108,133] an advanced method to sample the numbers of internode transitions along trajectories that is unaffected by metastability.[118] A detailed account of kPS was given in Chapter 1.

Inspection of the committor probabilities (Fig. 3.5a) reveals that the network is effectively a two-state system, with the I_h and F nodes representing strong attractors that characterize the respective regions of the state space. That is, there are relatively few nodes with intermediate values for the committor probability ($q_j^+ \approx 0.5$), and instead the vast majority of nodes are strongly associated with relaxation to either the I_h or the F state (indicated by committor probabilities $q_j^+ \approx 0$ and $q_j^+ \approx 1$, respectively). The MFPTs to the F state are $\mathcal{T}_{F \leftarrow j} \approx 10^9$ for most nodes j, although there are a small number of nodes associated with extreme values for $\mathcal{T}_{F \leftarrow j}$. In particular, nodes separated from the F state by small energy barriers relax to the F state comparatively rapidly ($\mathcal{T}_{F \leftarrow j} \approx 10^4$), but there are nodes that constitute kinetic traps, for which transitions to F correspond to very long timescales ($\mathcal{T}_{F \leftarrow j} \approx 10^{18}$).

A striking feature of the network is the localization of the reactive dynamics to a small subset of nodes, demonstrating that there are strongly preferred pathways for the $F \leftarrow I_h$ transition at this temperature. On average, only around 10 % of nodes are visited more than once along a reactive $F \leftarrow I_h$ path, and the average number of visits is $\widetilde{\theta}_j < 10^{-4}$ for around

half of the nodes j (Fig. 3.5c). The localization of the transition path ensemble is also evident from the reactive visitation probabilities r_j^+ for nodes (Fig. 3.5d): less than 10 % of nodes are associated with values $r_j^+ > 0.1$, and only around half of the nodes have $r_j^+ > 10^{-5}$. The reactive visitation probabilities for the 10 % of nodes with the highest stationary probabilities are essentially negligible ($r_j^+ < 10^{-10}$). With decreasing temperature, the number of nodes associated with non-negligible values for the reactive visitation probability becomes even smaller.[118] Moreover, the expected number of times that nodes are visited along $F \leftarrow I_h$ transition paths represents only a small fraction of the expected number of times that nodes are visited along first passage paths. This result confirms that the majority of the MFPT is accounted for by unproductive 'flickering'[108] within nodes that have a strong tendency to relax back to the I_h attractor node.

It is often insightful to closely examine the properties of specific nodes that play a critical role in the reactive dynamics. In Fig. 3.5, we highlight two metastable intermediate structures, M_1 and M_2, that are particularly relevant to the $F \leftarrow I_h$ transition. The M_1 state is a somewhat disordered structure that is highly likely to be visited along a reactive $F \leftarrow I_h$ transition path ($r_{M_1}^+ \approx 0.9$), although trajectories at this node have a high probability of returning to the initial I_h state ($q_{M_1}^+ \approx 10^{-3}$). The M_1 state therefore represents a structure that (usually) must be located to successfully transition to the F from the I_h state, but this is an early step that does not modulate the slow dynamics. Large perturbations to the transition probabilities associated with the M_1 node would significantly affect the MFPT, but the global dynamics are not overly sensitive to small perturbations of this node, since the state does not constitute a limiting step in the rare event.[124,125] The M_2 state, a configuration that retains some of the symmetry of the incomplete icosahedral I_h state, is a true dynamical bottleneck node in the network. Around half of the reactive trajectories proceed to F via M_2 ($r_{M_2}^+ \approx 0.47$). Furthermore, the M_2 node is a member of the transition state ensemble[122] (TSE) of nodes dividing the effective regions of attraction characterized by the I_h and F states. That is, trajectories reaching the M_2 state then have an approximately equal probability of first hitting either I_h or F, with the latter state slightly favoured ($q_{M_2}^+ \approx 0.63$). Since the M_2 node is likely to be visited along reactive paths and corresponds to a limiting step of the overall slow transition, the global dynamics, including the MFPT, are highly sensitive even to small perturbations of the transition probabilities corresponding to this node.[123]

The principles that we have used in our analysis of the LJ_{38} system can be applied to yield insight into the dynamics of an abritrary discrete- or continuous-time Markov chain. It is particularly useful to identify the nodes that comprise the TSE, for which the local dynamics have a critical effect in determining the global dynamics,[118] and to identify the favoured nodes

that mediate the dominant pathways for the productive transition between two endpoint states. For Markov chains where the transition probabilities or rates depend on an external parameter,[134] such as the temperature in physical systems, perturbations may significantly alter the dynamical behaviour. For instance, a switching effect may be observed in systems with alternative competing mechanisms for a given $\mathcal{A} \leftarrow \mathcal{B}$ transition, with different reactive pathways and dynamical bottlenecks being favoured in separate parameter regimes. When the system exhibits rare event dynamics, the origins of switching behaviour can likely be traced to a small number of influential states. The quantities discussed in this chapter, and especially the reactive visitation probability[118] derived herein, provide a convenient means to rigorously assess which regions of the state space are kinetically relevant with respect to a particular $\mathcal{A} \leftarrow \mathcal{B}$ process of interest. Our proposed methodology, which allows for the treatment of models with metastable states, is therefore essential for analyzing the features of a general Markov chain, and for understanding differences in the dynamical behaviour of related models.

3.5 Conclusions

In this chapter, we have described state reduction algorithms for the numerically stable analysis of first passage processes in finite discrete- and continuous-time Markov chains exhibiting metastability, for which the systems of linear equations to be solved are severely ill-conditioned.[41–44] Since a separation of characteristic timescales is a ubiquitous feature of Markov chains representing realistic dynamical processes,[38,46–59] our methodology provides a valuable approach to analyze complex systems in practical applications. The limiting factor affecting the viability of the state reduction procedures presented here is the available computer memory. Nonetheless, the methodology remains feasible for sparse networks comprising several thousand nodes.[108] For larger networks, metastability can be exploited to lump[18,135–137] the nodes of the Markov chain without introducing significant error in the representation of the slow dynamics.[38] We have illustrated our approach with numerical results for a CTMC representing a structural transition in a model atomic cluster at low temperature, which is difficult to analyze by standard linear algebra methods.[38]

We have presented an iterative formulation of the graph transformation (GT) algorithm[73–78] (Sec. 3.2.2) that incorporates a backward pass phase, which enables the MFPTs for transitions from all nonabsorbing nodes to the absorbing state to be determined simultaneously (Sec. 3.2.4). The procedure requires storing a subset of elements of the transition probability matrix, and (optionally) waiting times for nodes, during the forward pass phase. If the MFPTs for transitions from all nonabsorbing nodes of the Markov chain are of interest, then our proposed

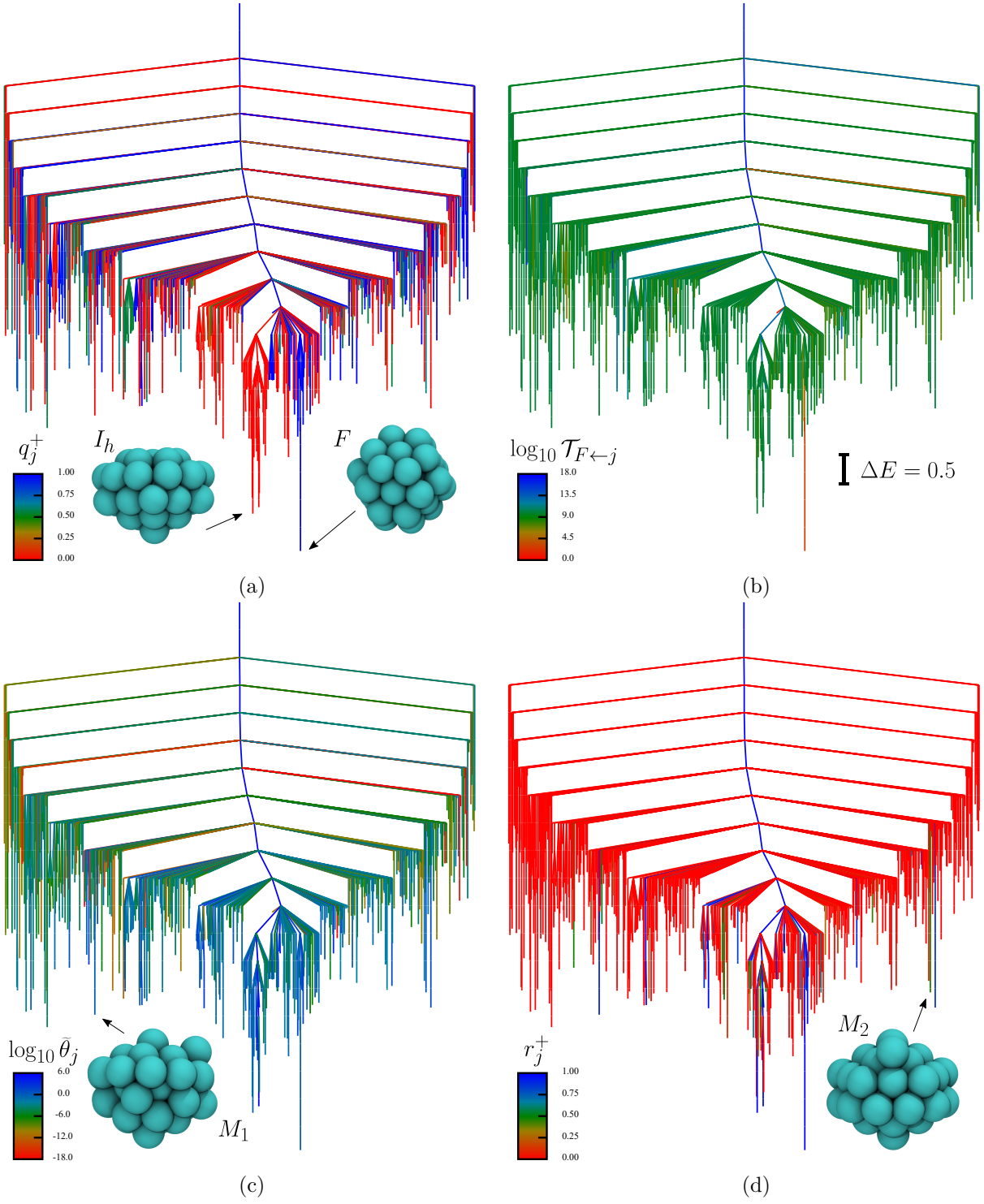


Figure 3.5: Disconnectivity graphs[132] showing the dynamical properties of nodes in the Markov chain for the transition of the LJ_{38} cluster from an incomplete icosahedron (I_h) to a face-centered cubic (F) structure, computed using state reduction algorithms as described in Sec. 3.4. The vertical axis represents the potential energy, and the threshold increment is $\Delta E = 0.5$ (in reduced units). (a) Committor probabilities q_j^+ for nodes j. (b) MFPTs $\mathcal{T}_{F \leftarrow j}$ for transitions to the F state. (c) Expected numbers of node visits along reactive paths that leave I_h and reach F without returning to I_h, $\widetilde{\theta}_j$ (Eq. 3.41). (d) Reactive visitation probabilities r_j^+ (Eq. 3.44). Nodes M_1 and M_2 both have high visitation probabilities, but only the latter has a committor probability close to 0.5. The M_2 structure is therefore a dynamical bottleneck that has a critical role in modulating the overall transition.

variation of the GT algorithm (Algorithm 5) is preferable to previous formulations[76,107] that compute the MFPT for a single transition from a particular node. For example, the new version is particularly advantageous when computing the optimal coarse-grained transition probabilities or rates for a given partitioning of the Markov chain,[8,138,139] which requires the matrix of MFPTs for all pairwise transitions between nodes.[116] We have also proposed a numerically stable state reduction algorithm to compute the committor probabilities[79–82] (Sec. 3.2.3), which can be incorporated into the above procedure for computing MFPTs. Accurate calculation of the committor probabilities is required to determine the edge weights in the shortest paths analysis proposed in Chapter 2, where the total productive flux is decomposed into contributions from simple transition flux-paths.

We then derived a state reduction algorithm to compute the fundamental matrix of an absorbing Markov chain (Sec. 3.3.2), the elements of which are the expected number of node visits along first passage paths. This procedure provides a numerically stable route to compute the variance of the FPT distribution (Eq. 3.25), a key global dynamical property that is otherwise challenging to obtain in a robust manner.[69] Together with the committor probabilities, the expected number of node visits allows for the straightforward evaluation of key dynamical properties that characterize the direct transition process to the absorbing state at a nodewise level of detail (Sec. 3.3.3). In particular, we have derived expressions for the reactive visitation probabilities of nodes (Sec. 3.3.4), that is, the probability that a node is visited along a trajectory that hits the absorbing state without first re-entering the initial state (*cf.* Fig. 3.4). We considered reactive visitation probabilities for both the nonequilibrium[17] and equilibrium[112] (i.e. steady state) path ensembles (Eqs. 3.44 and 3.46, respectively), thus extending the results of transition path theory outlined in Chapter 1. The expected number of times that nodes are visited along reactive paths (Eqs. 3.41 and 3.47) can be obtained similarly.

The methodology presented herein can be used to gain fundamental insight into dynamical processes on finite Markov chains, including when there are metastable states. The separation of the first passage path ensemble into nonreactive and reactive components[17] allows for the individual nodes and edges that are critical in facilitating the productive transition process to be readily identified. This nodewise analysis complements the flux-pathwise analysis proposed in Chapter 2. In particular, nodes with intermediate values for the committor probability and large reactive visitation probabilities constitute the dynamical bottlenecks for the dominant transition pathways. We utilize the definition of a reactive visitation probability again in Chapter 4, where we show that kPS can be used to efficiently obtain simulation estimates for the reactive visitation and committor probabilities associated with nodes or groups thereof.

input : discrete-or continuous-time transition probability matrix $\mathbf{T}$ with state space $\mathcal{S} \equiv \mathcal{Q} \cup \mathcal{A}$
set of absorbing nodes $\mathcal{A}$ and set of transient nodes $\mathcal{Q} \equiv \mathcal{A}^{\mathsf{c}}$
set of initial nodes $\mathcal{B} \subseteq \mathcal{A}$ (note that nodes of the set $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$ are prioritized for elimination)
$|\mathcal{Q}|$-dimensional vector of mean waiting (or lag) times $\boldsymbol{\tau}$ for nodes $j \in \mathcal{Q}$
output: $|\mathcal{Q}|$-dimensional vector of $\mathcal{A} \leftarrow j$ MFPTs $\boldsymbol{\mathcal{T}}_{\mathcal{A}}$ for nodes $j \in \mathcal{Q}$
$|\mathcal{Q}|$-dimensional vector of $\mathcal{A} \leftarrow \mathcal{B}$ committor probabilities $\mathbf{q}^+$ for nodes $j \in \mathcal{Q}$
$|\mathcal{A}| \times |\mathcal{Q}|$-dimensional matrix $\mathbf{B}$ of $i \in \mathcal{A} \leftarrow j \notin \mathcal{A}$ absorption probabilities B_{ij}
$|\mathcal{S}|$-dimensional stationary distribution vector $\boldsymbol{\pi}$ for all nodes (exists if the chain is irreducible)

initialize $\boldsymbol{\mathcal{T}}_{\mathcal{A}}, \mathbf{q}^+, \mathbf{B}, \boldsymbol{\pi}, \mathbf{L}, \mathbf{U}$;
$\mathbf{T}^{(0)} \leftarrow \mathbf{T}$, $n \leftarrow 1$;
`/* forward pass phase to eliminate all transient nodes by renormalization */`
while $n \leq |\mathcal{Q}|$ (i.e. $n \notin \mathcal{A}$) **do**
  for $i \in \mathcal{S}, j \notin \mathcal{A}$ **do**
    $L_{nj} \leftarrow T_{nj}^{(n-1)}/(1 - T_{nn}^{(n-1)})$, $\quad U_{in} \leftarrow T_{in}^{(n-1)} - \delta_{in}$; `// LU decomposition of transition matrix`
    $T_{ij}^{(n)} \leftarrow T_{ij}^{(n-1)} + L_{nj} U_{in}$; `// eliminate node by graph transformation`
    $\tau_j^{(n)} \leftarrow \tau_j^{(n-1)} + \tau_n^{(n-1)} L_{nj}$; `// renormalize waiting times`
    $n \leftarrow n + 1$;
  if all nodes of the set $\mathcal{I} \equiv (\mathcal{A} \cup \mathcal{B})^{\mathsf{c}}$ have been eliminated with this iteration **then**
    $q_b^+ \leftarrow 0 \; \forall b \in \mathcal{B}$;
    `/* compute committor probabilities for all intermediate nodes */`
    for $j \leftarrow n, n+1, \ldots, |\mathcal{Q}|$ (i.e. $j \in \mathcal{I}$) **do**
      $q_j^+ \leftarrow \sum_{a \in \mathcal{A}} T_{aj}^{(n)}$;
$B_{ij} \leftarrow T_{ij}^{(n)} \; \forall i \in \mathcal{A}, j \notin \mathcal{A}$; `// compute absorption probabilities`
`/* If the Markov chain is irreducible, eliminate and then restore all but one of the absorbing nodes, needed to compute the stationary distribution (GTH algorithm) */`
while $|\mathcal{Q}| < n < |\mathcal{S}|$ (i.e. $n \in \mathcal{A} \setminus |\mathcal{S}|$) **do**
  $L_{nj} \leftarrow T_{nj}^{(n-1)}/(1 - T_{nn}^{(n-1)})$, $\quad U_{in} \leftarrow T_{in}^{(n-1)} - \delta_{in}$;
  $T_{ij}^{(n)} \leftarrow T_{ij}^{(n-1)} + L_{nj} U_{in}$; `// eliminate node by graph transformation`
  $n \leftarrow n + 1$;
$\pi_n \leftarrow 1$, $\mu \leftarrow 1$; `// at this point, only the` $|\mathcal{S}|$`-th node remains`
while $|\mathcal{Q}| < n < |\mathcal{S}|$ (i.e. $n \in \mathcal{A} \setminus |\mathcal{S}|$) **do**
  $T_{ij}^{(n-1)} \leftarrow T_{ij}^{(n)} - L_{nj} U_{in}$; `// restore node (i.e. undo graph transformation)`
  $n \leftarrow n - 1$;
  $\pi_n \leftarrow L_{n,|\mathcal{S}|} + \sum_{k=n+1}^{|\mathcal{S}|-1} \pi_k L_{nk}$, $\quad \mu \leftarrow \mu + \pi_n$; `// GTH step`
`/* compute MFPT and stationary probability for the` $|\mathcal{Q}|$`-th node, which was the last transient node to be eliminated */`
$\mathcal{T}_{\mathcal{A}n} \leftarrow \tau_n^{(n)}$;
`/* backward pass phase to compute MFPTs and stationary probabilities for all other transient nodes */`
while $n \geq 1$ (i.e. $n \notin \mathcal{A}$) **do**
  $T_{ij}^{(n-1)} \leftarrow T_{ij}^{(n)} - L_{nj} U_{in}$; `// restore node (i.e. undo graph transformation)`
  $\tau_j^{(n-1)} \leftarrow \tau_j^{(n)} - \tau_n^{(n-1)} L_{nj}$;
  $n \leftarrow n - 1$;
  $\mathcal{T}_{\mathcal{A}n} \leftarrow \tau_n^{(n)} + \sum_{\gamma \notin \mathcal{A}} \mathcal{T}_{\mathcal{A}\gamma} T_{\gamma n}^{(n)}$; `// compute MFPT for restored node`
  $\pi_n \leftarrow L_{n,|\mathcal{S}|} + \sum_{k=n+1}^{|\mathcal{S}|-1} \pi_k L_{nk}$, $\quad \mu \leftarrow \mu + \pi_n$; `// GTH step`
$\pi_j \leftarrow \pi_j/\mu \; \forall j$; `// renormalization of the stationary distribution`
deallocate $\mathbf{L}$, $\mathbf{U}$;
return $\boldsymbol{\mathcal{T}}_{\mathcal{A}}, \mathbf{q}^+, \mathbf{B}, \boldsymbol{\pi}$;

Algorithm 5: State reduction algorithm to simultaneously compute the MFPTs, committor probabilities, and absorption probabilities for all transient (i.e. nonabsorbing) nodes in a DTMC or CTMC. This algorithm is illustrated in Fig. 3.2. The steps of the Grassmann-Taksar-Heyman (GTH) algorithm[66,67] are also incorporated into this procedure, so that the stationary distribution is computed if the Markov chain is irreducible.

input : discrete-or continuous-time transition probability matrix $\mathbf{T}$ with state space $\mathcal{S} \equiv \mathcal{Q} \cup \mathcal{A}$
set of transient nodes $\mathcal{Q} \subset \mathcal{S}$
set of absorbing nodes $\mathcal{A} \subset \mathcal{S}$
set of dummy nodes $\mathcal{Q}^*$, where $|\mathcal{Q}^*| = |\mathcal{Q}|$
output: $|\mathcal{Q}| \times |\mathcal{Q}|$-dimensional fundamental matrix $\mathbf{N}$ associated with the absorbing Markov chain

`/* define a network with augmented state space` $\mathcal{S}^*$ `that includes transient, absorbing, and dummy nodes */`
$\mathcal{S}^* \leftarrow \mathcal{S} \cup \mathcal{Q}^*$;
`/* set the initial` $i \leftarrow j$ `edge weights,` N^*_{ij}`, of the augmented network with state space` $\mathcal{S}^*$ `*/`
$N^*_{ij} \leftarrow T_{ij} \quad \forall\, i \in \mathcal{S}, j \in \mathcal{Q}$;
$N^*_{ij} \leftarrow 1 \quad \forall\, i \in \mathcal{Q}^*, j \in \mathcal{Q}$;
$N^*_{ij} \leftarrow 1 \quad \forall\, i \in \mathcal{Q}, j \in \mathcal{Q}^*$;
$N^*_{ij} \leftarrow 0 \quad \forall\, i \in \mathcal{S}, j \in \mathcal{A}$;
$N^*_{ij} \leftarrow 0 \quad \forall\, i, j \in \mathcal{Q}^*$;
`/* eliminate all transient nodes of the augmented network by renormalization */`
$\mathcal{E} \leftarrow \emptyset$; `// set of nodes that have been eliminated (initially empty)`
for $n \in \mathcal{Q}$ **do**
  $\mathcal{E} \leftarrow \mathcal{E} \cup \{n\}$;
  $N^*_n \leftarrow \sum_{\gamma \in \mathcal{S} \setminus \mathcal{E}} N^*_{\gamma n} \quad (\equiv 1 - N^*_{nn})$; `// confers numerical stability`
  for $i, j \in \mathcal{S}^* \setminus \mathcal{E}$ **do**
    $N^*_{ij} \leftarrow N^*_{ij} + (N^*_{in} N^*_{nj} / N^*_n)$; `// renormalization preserves` $\sum_{\gamma \in \mathcal{S} \setminus \mathcal{E}} N^*_{\gamma n} = 1 \;\; \forall\, n \in \mathcal{Q} \setminus \mathcal{E}$
`/* once all transient nodes have been eliminated, the edge weights for transitions between dummy nodes in the remaining network are the elements of the fundamental matrix` $\mathbf{N}$ `*/`
initialize $\mathbf{N}$;
$N_{ij} \leftarrow N^*_{ij} \quad \forall\, i, j \in \mathcal{Q}^*$;
return $\mathbf{N}$;

Algorithm 6: State reduction algorithm to robustly compute elements N_{ij}, the expected number of times that the i-th node is visited along a first passage path initialized at node j prior to absorption, for all transient nodes $i, j \in \mathcal{Q}$. This procedure is illustrated in Fig. 3.3.

Bibliography

[1] S. Redner. *A Guide to First-Passage Processes.* Cambridge University Press, Cambridge, UK, 2012.

[2] R. Metzler, G. Oshanin, and S. Redner. *First-Passage Phenomena and Their Applications.* World Scientific, Singapore, 2014.

[3] S. Iyer-Biswas and A. Zilman. *Adv. Chem. Phys.*, 160:261–306, 2016.

[4] M. Castro, M. López-García, G. Lythe, and C. Molina-París. *Sci. Rep.*, 8:15054, 2018.

[5] Z. Zhang, A. Julaiti, B. Hou, H. Zhang, and G. Chen. *Eur. Phys. J. B*, 84:691–697, 2011.

[6] Z. Zhang, Y. Sheng, Z. Hu, and G. Chen. *Chaos*, 22:043129, 2012.

[7] Z. Zhang, T. Shan, and G. Chen. *Phys. Rev. E*, 87:012112, 2013.

[8] A. Kells, V. Koskin, E. Rosta, and A. Annibale. *J. Chem. Phys.*, 152:104108, 2020.

[9] S. Park, M. K. Sener, D. Lu, and K. Schulten. *J. Chem. Phys.*, 119:1313–1319, 2003.

[10] S. X. Sun. *Phys. Rev. Lett.*, 96:210602, 2006.

[11] B. Harland and S. X. Sun. *J. Chem. Phys.*, 127:104103, 2007.

[12] T. Oppelstrup, V. V. Bulatov, A. Donev, M. H. Kalos, G. H. Gilmer, and B. Sadigh. *Phys. Rev. E*, 80:066701, 2009.

[13] S. Hwang, D.-S. Lee, and B. Kahng. *Phys. Rev. Lett.*, 109:088701, 2012.

[14] M. Manhart and A. V. Morozov. *Phys. Rev. Lett.*, 111:088102, 2013.

[15] M. Manhart, W. Kion-Crosby, and A. V. Morozov. *J. Chem. Phys.*, 143:214106, 2015.

[16] M. Manhart and A. V. Morozov. *Proc. Natl. Acad. Sci. USA*, 112:1797–1802, 2015.

[17] M. von Kleist, C. Schütte, and W. Zhang. *J. Stat. Phys.*, 170:809–843, 2018.

[18] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand, New Jersey, USA, 1960.

[19] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier, Amsterdam, Netherlands, 1992.

[20] J. R. Norris. *Markov Chains.* Cambridge University Press, New York, USA, 1997.

[21] C. M. Grinstead and J. L. Snell. *Introduction to Probability.* American Mathematical Society, Providence, Rhode Island, 1997.

[22] H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, London, UK, third edition, 1998.

[23] L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology.* Prentice Hall, Upper Saddle River, New Jersey, 2003.

[24] J. Goutsias and G. Jenkinson. *Phys. Rep.*, 529:199–264, 2013.

[25] N. Masuda, M. A. Porter, and R. Lambiotte. *Phys. Rep.*, 716-717:1–58, 2017.

[26] G. R. Bowman, V. S. Pande, and F. Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation.* Springer, Netherlands, first edition, 2014.

[27] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. *Proc. Natl. Acad. Sci. USA*, 106:19011–19016, 2009.

[28] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. *J. Chem. Phys.*, 134:174105, 2011.

[29] V. S. Pande, K. Beauchamp, and G. R. Bowman. *Methods*, 52:99–105, 2010.

[30] B. E. Husic and V. S. Pande. *J. Am. Chem. Soc.*, 140:2386–2896, 2018.

[31] A. Mardt, L. Pasquali, H. Wu, and F. Noé. *Nat. Commun.*, 9:5, 2018.

[32] J. A. Joseph, K. Röder, D. Chakraborty, R. G. Mantell, and D. J. Wales. *Chem. Commun.*, 53:6974–6988, 2017.

[33] K. Röder, J. A. Joseph, B. E. Husic, and D. J. Wales. *Adv. Theory Simul.*, 2:1800175, 2019.

[34] E. M. Hanks, M. B. Hooten, and M. W. Alldredge. *Ann. Appl. Stat.*, 9:145–165, 2015.

[35] H. A. Simon and A. Ando. *Econometrica*, 29:111–138, 1961.

[36] C. D. Meyer Jr. *SIAM Rev.*, 17:443–464, 1975.

[37] D. P. Heyman and D. P. O'Leary. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 151–161. Springer, New York, 1995.

[38] T. D. Swinburne, D. Kannan, D. J. Sharpe, and D. J. Wales. *J. Chem. Phys.*, 153:134115, 2020.

[39] D. J. Hartfiel and C. D. Meyer Jr. *Linear Algebra Appl.*, 272:193–203, 1998.

[40] C. R. MacCluer. *SIAM Rev.*, 42:487–498, 2000.

[41] D. P. Heyman and A. Reeves. *ORSA J. Comp.*, 1:52–60, 1989.

[42] B. Philippe, Y. Saad, and W. J. Stewart. *Oper. Res.*, 40:1156–1179, 1992.

[43] C. D. Meyer Jr. *SIAM J. Matrix Anal. Appl.*, 15:715–728, 1994.

[44] J. L. Barlow. *SIAM J. Matrix Anal. Appl.*, 22:230–241, 2000.

[45] Y. Saad. *Numerical methods for large eigenvalue problems.* SIAM, Philadelphia, PA, 2011.

[46] D. J. Aldous and M. Brown. In M. Shaked and Y. L. Tong, editors, *IMS Lecture Notes in Statistics, Vol. 22: Stochastic Inequalities*, pages 1–16. Institute of Mathematical Statistics, Ohio, USA, 1992.

[47] P. Heidelberger. *ACM Trans. Model. Comput. Simul.*, 5:43–85, 1995.

[48] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. *Oper. Res.*, 47:495–645, 1999.

[49] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. *J. Phys. A.: Math. Gen.*, 33:L447–L451, 2000.

[50] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. *Commun. Math. Phys.*, 228:219–255, 2002.

[51] S. Juneja and P. Shahabuddin. *Manage. Sci.*, 47:547–562, 2001.

[52] J. Beltrán and C. Landim. *J. Stat. Phys.*, 140:1065–1114, 2010.

[53] E. Vanden-Eijnden and J. Weare. *Commun. Pure Appl. Math*, 65:1770–1803, 2012.

[54] O. Benois and M. Mourragui. *J. Stat. Phys.*, 153:967–990, 2013.

[55] C. Hartmann, R. Banisch, M. Sarich, T. Badowski, and C. Schütte. *Entropy*, 16:350–376, 2014.

[56] M. Sarich, R. Banishc, C. Hartmann, and C. Schütte. *Entropy*, 16:258–286, 2014.

[57] M. K. Cameron. *J. Chem. Phys.*, 141:184113, 2014.

[58] T. Gan and M. Cameron. *J. Nonlinear Sci.*, 27:927–972, 2017.

[59] C. Pérez-Espigares and P. I. Hurtado. *Chaos*, 29:083106, 2019.

[60] D. P. Heyman. *SIAM J. Alg. Discr. Meth.*, 8:226–232, 1987.

[61] W. Grassmann and D. A. Stanford. In W. Grassmann, editor, *Computational Probability*, pages 153–203. Springer, New York, 2000.

[62] C. A. O'Cinneide. *Numer. Math.*, 65:109–120, 1993.

[63] C. A. O'Cinneide. *Numer. Math.*, 73:507–519, 1996.

[64] D. P. O'Leary and Y.-J. J. Wu. *SIAM J. Matrix Anal. Appl.*, 17:470–488, 1996.

[65] I. Sonin. *Adv. Math.*, 145:159–188, 1999.

[66] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. *Oper. Res.*, 33:1107–1116, 1985.

[67] T. J. Sheskin. *Oper. Res.*, 33:228–235, 1985.

[68] J. Kohlas. *Zeit. Oper. Res.*, 30:197–207, 1986.

[69] T. Dayar and N. Akar. *SIAM J. Matrix Anal. Appl.*, 27:396–412, 2005.

[70] D. P. Heyman. *SIAM J. Matrix Anal. Appl.*, 16:954–963, 1995.

[71] D. P. Heyman and D. P. O'Leary. *SIAM J. Matrix Anal. Appl.*, 19:534–540, 1998.

[72] I. Sonin and J. Thornton. *SIAM J. Matrix Anal. Appl.*, 23:209–224, 2001.

[73] S. A. Trygubenko and D. J. Wales. *Mol. Phys.*, 104:1497–1507, 2006.

[74] S. A. Trygubenko and D. J. Wales. *J. Chem. Phys.*, 124:234110, 2006.

[75] D. J. Wales. *Int. Rev. Phys. Chem.*, 25:237–282, 2006.

[76] D. J. Wales. *J. Chem. Phys.*, 130:204111, 2009.

[77] J. D. Stevenson and D. J. Wales. *J. Chem. Phys.*, 141:041104, 2014.

[78] R. S. MacKay and J. D. Robinson. *Phil. Trans. Roy. Soc. A*, 376:20170232, 2018.

[79] C. Dellago, P. G. Bolhuis, and P. L. Geissler. *Adv. Chem. Phys.*, 123:1–78, 2002.

[80] W. E, W. Ren, and E. Vanden-Eijnden. *Chem. Phys. Lett.*, 413:242–247, 2005.

[81] A. M. Berezhkovskii and A. Szabo. *J. Chem. Phys.*, 150:054106, 2019.

[82] Q. Li, B. Lin, and W. Ren. *J. Chem. Phys.*, 151:054112, 2019.

[83] K. L. Chung. *J. Res. Natl. Bur. Stand.*, 50:302–208, 1953.

[84] K. L. Chung. *Illinois J. Math.*, 5:431–435, 1961.

[85] G. L. Sriwastav and S. N. N. Pandit. *Nav. Res. Logist. Q.*, 25:653–658, 1978.

[86] D. J. Sharpe. https://github.com/danieljsharpe/DISCOTRESS, 2020.

[87] P. Coolen-Schrijner and E. A. van Doorn. *Probab. Eng. Inf. Sci.*, 16:351–366, 2002.

[88] S. A. Serebrinsky. *Phys. Rev. E*, 83:037701, 2011.

[89] J. Medhi. In J. Medhi, editor, *Stochastic models in queueing theory*, pages 1–46. Academic Press, San Diego, 2003.

[90] D. P. Heyman. *J. Appl. Probab.*, 32:893–901, 1995.

[91] T. Dayar and W. J. Stewart. *SIAM J. Sci. Comput.*, 17:287–303, 1996.

[92] W. J. Stewart. In W. Grassmann, editor, *Computational Probability*, pages 81–111. Springer, New York, 2000.

[93] J.-H. Prinz, M. Held, J. C. Smith, and F. Noé. *Multiscale Model. Simul.*, 9:545–567, 2011.

[94] J. J. Hunter. *Linear Algebra Appl.*, 549:100–122, 2018.

[95] Z. Zheng, G. Xiao, G. Wang, G. Zhang, and K. Jiang. *Math. Probl. Eng.*, 2017:8217361, 2017.

[96] N. Singhal, C. D. Snow, and V. S. Pande. *J. Chem. Phys.*, 121:415–425, 2004.

[97] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, PA, second edition, 2003.

[98] Y. Saad. *Linear Algebra Appl.*, 34:269–295, 1980.

[99] H. D. Simon. *Linear Algebra Appl.*, 61:101–131, 1984.

[100] C. D. Meyer Jr. *SIAM Rev.*, 31:240–272, 1989.

[101] E. Seneta. *SIAM J. Matrix Anal. Appl.*, 19:556–563, 1998.

[102] E. Meerbach, C. Schütte, and A. Fischer. *Linear Algebra Appl.*, 398:141–160, 2005.

[103] M. Haviv. *SIAM J. Numer. Anal.*, 22:952–966, 1987.

[104] Y. Q. Zhao and D. Liu. *J. Appl. Probab.*, 33:623–629, 1996.

[105] T. Dayar, H. Hermanns, D. Spieler, and V. Wolf. *Numer. Linear Algebra Appl.*, 18:931–946, 2011.

[106] A. Milias-Argeitis and J. Lygeros. *J. Chem. Phys.*, 138:184109, 2013.

[107] T. D. Swinburne and D. J. Wales. *J. Chem. Theory Comput.*, 16:2661–2679, 2020.

[108] M. Athènes and V. V. Bulatov. *Phys. Rev. Lett.*, 113:230601, 2014.

[109] E. Vanden-Eijnden. In M. Ferrario, G. Ciccotti, and K. Binder, editors, *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, pages 453–493. Springer Berlin, Heidelberg, 2006.

[110] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *J. Chem. Phys.*, 125:084110, 2006.

[111] W. E and E. Vanden-Eijnden. *J. Stat. Phys.*, 123:503–523, 2006.

[112] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *Multiscale Model. Simul.*, 7:1192–1219, 2009.

[113] W. E and E. Vanden-Eijnden. *Annu. Rev. Phys. Chem.*, 61:391–420, 2010.

[114] J. J. Hunter. *Spec. Matrices*, 4:151–175, 2016.

[115] J. G. Kemeny and J. L. Snell. *Theory Prob. Its Appl.*, 6:101–105, 1961.

[116] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. *J. Chem. Phys.*, 153:244108, 2020.

[117] M. K. Cameron and E. Vanden-Eijnden. *J. Stat. Phys.*, 156:427–454, 2014.

[118] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 153:024121, 2020.

[119] G. Hummer. *J. Chem. Phys.*, 120:516–523, 2004.

[120] R. B. Best and G. Hummer. *Proc. Natl. Acad. Sci. USA*, 102:6732–6737, 2005.

[121] P. G. Bolhuis. *J. Stat. Phys.*, 145:841–859, 2011.

[122] A. Berezhkovski, G. Hummer, and A. Szabo. *J. Chem. Phys.*, 130:205102, 2009.

[123] J. J. Hunter. *Linear Algebra Appl.*, 410:217–243, 2005.

[124] R. E. Funderlic and C. D. Meyer. *Linear Algebra Appl.*, 76:1–17, 1986.

[125] G. D. Zhang. *SIAM J. Matrix Anal. Appl.*, 14:1112–1123, 1993.

[126] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 151:124101, 2019.

[127] D. Shukla, C. X. Hernández, J. K. Weber, and V. S. Pande. *Acc. Chem. Res.*, 48:414–422, 2015.

[128] E. Suárez, J. L. Adelman, and D. M. Zuckerman. *J. Chem. Theory Comput.*, 12:3473–3481, 2016.

[129] J. P. K. Doye, M. A. Miller, and D. J. Wales. *J. Chem. Phys.*, 110:6896–6906, 1999.

[130] J. P. K. Doye, M. A. Miller, and D. J. Wales. *J. Chem. Phys.*, 111:8417–8428, 1999.

[131] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. *J. Comput. Phys.*, 17:10–18, 1975.

[132] O. M. Becker and M. Karplus. *J. Chem. Phys.*, 106:1495–1517, 1997.

[133] M. Athènes, S. Kaur, G. Adjanor, T. Vanacker, and T. Jourdan. *Phys. Rev. Materials*, 3:103802, 2019.

[134] V. Betz and S. Le Roux. *Stoch. Process. Their Appl.*, 126:3499–3526, 2016.

[135] P. Buchholz. *J. Appl. Probab.*, 31:59–75, 1994.

[136] W. E, T. Li, and E. Vanden-Eijnden. *Proc. Natl. Acad. Sci. USA*, 105:7907–7912, 2008.

[137] J. A. Ward and M. López-García. *Appl. Netw. Sci.*, 4:108, 2019.

[138] G. Hummer and A. Szabo. *J. Phys. Chem. B*, 119:9029–9037, 2015.

[139] A. Kells, Z. E. Mihálka, A. Annibale, and E. Rosta. *J. Chem. Phys.*, 150:134107, 2019.

Chapter 4

Efficient and exact sampling of transition path ensembles on Markovian networks

The problem of flickering trajectories in standard kinetic Monte Carlo (kMC) simulations prohibits sampling of the transition path ensembles (TPEs) on Markovian networks representing many slow dynamical processes of interest. In this chapter, we overcome this problem using knowledge of the metastable macrostates, determined by an unsupervised community detection algorithm, to perform enhanced sampling kMC simulations. We implement two accelerated kMC methods to simulate the nonequilibrium stochastic dynamics on arbitrary Markovian networks, namely weighted ensemble (WE) sampling and kinetic path sampling (kPS). WE-kMC utilizes resampling in pathway space to maintain an ensemble of representative trajectories covering the state space, and kPS utilizes graph transformation to simplify the description of an escape trajectory from a trapping energy basin. Both methods sample individual trajectories governed by the linear master equation with the correct statistical frequency. We demonstrate that they allow for efficient estimation of the time-dependent occupation probability distributions for the metastable macrostates, and of TPE statistics, such as committor probabilities and first passage time distributions. kPS is particularly attractive, since its efficiency is essentially independent of the degree of metastability, and we suggest how the algorithm could be coupled with other enhanced sampling methodologies. We illustrate our approach with results for a network representing the folding transition of a tryptophan zipper peptide, which exhibits a separation of characteristic timescales. We highlight some salient features of the dynamics, most notably, strong deviations from two-state behaviour, and the existence of multiple competing mechanisms.

4.1 Introduction

The stochastic dynamics of many complex systems can be formulated as a continuous-time Markov chain (CTMC),[1] with dynamics governed by the master equation (Sec. 4.2.1).[2–5] In molecular and condensed matter systems, the construction of kinetic networks circumvents the timescale problem,[6,7] which precludes the use of unbiased molecular dynamics (MD) simulations for systems exhibiting rare event dynamics.[8–10] One approach to constructing a master equation representation is to map the stationary points of the potential energy landscape onto the nodes and edges of a network,[11–13] as in discrete path sampling (DPS).[14,15] This strategy avoids explicit simulation of the dynamics, and is therefore especially useful for modeling systems featuring broken ergodicity.[16] Moreover, the resulting network representation of the underlying energy landscape preserves the full dimensionality of the configuration space. Both of these considerations are often important for biomolecular systems,[17] since a suitable low-dimensional projection of the energy landscape does not necessarily exist.[18] Projection of the energy landscape onto inappropriate reaction coordinates is liable to misrepresent the transition state ensemble (TSE) region[19] that is crucial to the description of the rare events.[20]

Many alternative methods for constructing kinetic networks representing the dynamics of continuous-state systems have been developed,[21–41] and we do not aim to review them all here. Kinetic networks described by a linear master equation also appear in social and economic models.[42] Kinetic networks for which the dynamics are governed by a nonlinear master equation, where higher-order terms arise due to interactions of species with discrete populations, are common in systems biology,[43–46] for example in the modeling of gene regulatory networks,[47–51] as well as in epidemiology[52] and ecology,[53] and can be mapped to linear kinetic networks.[51,54,55]

In this chapter, we describe efficient and exact methods for explicit simulation of the nonequilibrium stochastic dynamics for arbitrary Markov chains. In particular, we are interested in performing a detailed analysis of the transition path ensemble (TPE),[47] the set of $\mathcal{A} \leftarrow \mathcal{B}$ transition paths from initial to absorbing macrostates, denoted $\mathcal{B}$ and $\mathcal{A}$, respectively. Key properties characterizing the global $\mathcal{A} \leftarrow \mathcal{B}$ dynamics, such as mean first passage times[1] (MFPTs) and committor probabilities,[19] can be calculated robustly by state reduction methods, as described in the previous chapters.[56–61] Here, our aim is to describe methods that can elucidate how these quantities are encoded in the ensembles of pathways from which they are computed. The MFPT alone is not particularly informative, and the full FPT distribution may have a complex form. The moments of the probability distributions for path properties in the first passage path ensemble can in principle be calculated from the

fundamental matrix of an irreducible (Chapter 1) or a reducible (Chapter 3) Markov chain. However, this approach requires inverting a square matrix of order $|\mathcal{S}|$ or $|\mathcal{Q}|$, respectively. To obtain information on the sequence of events for a typical $\mathcal{A} \leftarrow \mathcal{B}$ transition, the dominant first passage paths can in principle be determined by k shortest paths algorithms with appropriate edge weights (Chapter 2).[62,63] In the metastable regime, however, the dominant first passage paths account for only a small fraction of the $\mathcal{A} \leftarrow \mathcal{B}$ flux,[64] and therefore analysis of the shortest paths alone may be misleading. For these reasons, to gain rigorous, detailed, and quantitative insight into an $\mathcal{A} \leftarrow \mathcal{B}$ dynamical process, it is often necessary to sample the TPE explicitly.

There are various numerical methods for obtaining detailed trajectory information. Direct solution of the master equation by linear algebra methods[54,55] (Chapter 1) rapidly becomes intractable with increasing system size.[65] In strongly metastable stochastic networks, the global dynamical behaviour is dominated by the rare fluctuations across the boundaries between the long-lived states. Separation of characteristic timescales is a ubiquitous feature of realistic models for dynamical processes, including biophysical[8–10,26,66] and biochemical[47–51] systems. In this regime, linear algebra methods are numerically unstable,[60,65] as discussed in the previous chapters, and mean-field methods based on deterministic ordinary differential equations (ODEs) may severely misrepresent the dynamics.[43,52,67–70] Therefore, the most generally applicable approach to analyze the time-dependent occupation probability distribution on kinetic networks is to use kinetic Monte Carlo (kMC) simulation[71–76] to sample the solution to the master equation, either exactly or approximately, by the generation of individual realizations of trajectories.

The problem of 'flickering' trajectories within metastable macrostates[77–80] seriously limits the efficiency of standard rejection-free[75,81,82] exact kMC algorithms (Sec. 4.2.2), such as the Bortz-Kalos-Lebowitz (BKL) algorithm[83] extended herein, and the equivalent Gillespie algorithm[68,84–86] for stochastic reaction networks. Hence there is a need to employ some enhanced sampling methodology[6,7] to accelerate the observation of rare events in kMC simulations. Many solutions have been proposed to the timescale problem associated with standard kMC simulations.[87] Strategies to ensure that the entire state space is representatively sampled include biasing the simulations and reweighting trajectories,[88,89] perturbing existing transition paths,[19] and repeatedly simulating portions of transition paths in parallel based on a division of the state space.[8,90–94]

As discussed in Chapter 1, one class of accelerated kMC methods is based on the formulation of the escape of a trajectory from a metastable trapping basin as an absorbing Markov chain, as in the Monte Carlo with absorbing Markov chains (MCAMC) algorithm.[95,96] The master equation of the absorbing Markov chain can be solved exactly by first passage

time analysis (FPTA).[70,97–99] Alternatively, the master equation can be solved approximately by the mean rate method[70,77,99,100] or by assuming a local equilibrium within the active basin. The metastable macrostates may be specified in advance or can be determined on-the-fly.[70,99] A basin that is being actively sampled may be built up as a Markovian web[101,102] of explored nodes. Methods based on absorbing Markov chains, especially those utilizing FPTA, incur a significant computational overhead that severely limits the feasible size of the trapping basins, and hence the potential computational gains achievable by such methods.[95,99] Employing the approximate alternatives to FPTA, or solving the dynamics within metastable sets of nodes using ODEs,[70] forfeits a statistically exact description of the trajectories within the metastable basins. The graph transformation method[56–61] can be leveraged to keep the number of nodes in a trapping basin small, by iteratively eliminating nodes from the kinetic network, and renormalizing the transition probabilities and waiting times in the reduced network to preserve the mean of the escape time distribution.[78]

An efficient approximate approach to facilitate the escape of trajectories from metastable macrostates is provided by accelerated superbasin kMC (AS-kMC).[103] In AS-kMC, the repeated observation of a transition between a pair of nodes triggers a search to determine the complete metastable basin to which the pair belongs. This neighbour search determines a subnetwork of nodes that are internally connected by fast transition rates, according to a given threshold. Then, the rates of all internode transitions within the basin are raised by a scale factor, in a way that maintains the accuracy of the kMC trajectory within a specified error tolerance. This biasing of the individual transition rates eventually encourages escape from a metastable basin. Many other approaches to accelerating kMC simulations exist, including strategies based on an importance function,[104,105] 'leapfrog' moves,[56] tau-leaping,[106,107] multi-level algorithms,[108–111] sliding windows,[112] uniformization,[113] stochastic complements,[114] waste recycling,[115–117] and more.[118–131]

Here, we analyze two complementary enhanced sampling methods, which facilitate efficient kMC simulations for arbitrary discrete- and continuous-time Markov chains of varying dimensionality and metastability. The first method that we consider is weighted ensemble (WE) sampling, originally proposed in Ref. 132 and pioneered by Zuckerman and co-workers.[93,133–137] The WE methodology has been discussed primarily in the context of stochastic MD simulations, including applications to biomolecular conformational transitions,[138–144] but has also been applied to kMC simulations of stochastic network models, specifically to the solution of nonlinear master equations in systems biology.[48–51] WE sampling belongs to a family of enhanced sampling methods where the state space is partitioned into non-overlapping bins and an ensemble of trajectories are simulated in parallel.[137] Each trajectory is associated with a statistical weight, and the ensemble of trajectories is maintained by resampling in

the pathway space. The simulation of trajectory segments for transitions between the bins facilitates the simulation of complete reactive trajectories between two defined endpoint macrostates of interest $\mathcal{A}$ and $\mathcal{B}$.[145] We describe WE-kMC in more detail in Section 4.2.4. The WE method is exact in sampling the path probability distribution for Markovian dynamics, and can be highly efficient in accelerating the observation of rare events.[93] Since the WE method utilizes multiple trajectories, it provides a natural approach for identifying multiple pathway ensembles.[146] This capability is desirable in the present context, since Markovian networks representing realistic dynamical processes, such as biomolecular conformational transitions,[9,10] frequently contain competing sets of pathways.[64]

The second approach to enhanced sampling of the dynamics on arbitrary finite Markov chains that we consider is kinetic path sampling (kPS),[147,148] which provides a powerful alternative to methods based on the explicit kMC simulation of trajectories. kPS leverages graph transformation[56–61] to generate a stochastic escape path from a defined trapping basin to an absorbing boundary, along with an associated trajectory time, that is exactly consistent with the master equation. The order of internode transitions in the escape path is not computed, but nodes at the absorbing boundary are sampled with the correct probability distribution. The kPS method, which is close in spirit to the MCAMC algorithm,[95,96] is described in more detail in Section 4.2.5.

The WE-kMC and kPS methods have some complementary desirable features. The time complexity of the kPS algorithm is essentially independent of the metastability of the kinetic network, and therefore the method is effectively immune to kinetic trapping. However, the cost of generating an escape path from a trapping basin $\mathbb{B}$, comprising $|\mathbb{E}|$ nodes that are eliminated in the graph transformation stage of the algorithm, scales roughly as $\mathcal{O}(|\mathbb{E}|^3)$. Hence, there is a significant computational overhead associated with the method if the trapping basins are large.[147] The memory requirements of a single iteration of the kPS algorithm likewise scales strongly with the size of the community. In its simplest form, the kPS algorithm does not simulate ordered trajectories on the kinetic network, but instead computes a non-Markovian trajectory on the coarse-grained network defined by the partitioning of the state space. Although kPS can be extended to compute the order of transition events along a detailed escape trajectory on the network, the time complexity is then adversely affected by metastability.[147] In WE-kMC, there is no such restriction on the size of communities, but it is essential that the communities reflect all uncoupled slow dynamical modes of the system, since dynamical modes orthogonal to the bin coordinates must be sampled by standard rejection-free kMC.[48] For this reason, WE-kMC is less efficient for more strongly metastable Markov chains.[149] However, the method is highly parallelizable,[93] and in the context of simulating the dynamics on kinetic networks,

where the bins to which nodes belong are identified simply by labels, the computational overhead associated with the trajectory resampling procedure is negligible. Moreover, by utilizing a protocol to accelerate the establishment of a steady state, WE-kMC can also be used to conduct equilibrium simulations.[134]

Both WE-kMC and kPS are based on a partitioning of the state space into disjoint sets of nodes, although this division need not be known *a priori*. Here we propose the use of an unsupervised stochastic community detection algorithm, namely multi-level regularized[150–152] Markov clustering[153–155] (MLR-MCL), to identify the metastable sets of nodes in a kinetic network,[156] which are then used as the fixed bins for the WE-kMC and kPS simulations. Strategies for the choice of metastable macrostates, which can also be performed adaptively, are discussed in more detail in Section 4.2.3. We present results for a kinetic network representing the folding of the trytophan zipper peptide TZ1,[157] constructed by the DPS methodology,[14,15] which is high-dimensional and exhibits a separation of characteristic timescales (Sec. 4.3.2). We show that both WE-kMC and kPS provide efficient and exact methods for sampling nonequilibrium TPEs in discrete-state stochastic systems. Estimation of the time-dependent occupation probability distributions, committor probabilities,[19] and reactive visitation probabilities for the states of interest, and of the $\mathcal{A} \leftarrow \mathcal{B}$ first passage time distribution, is tractable even for systems that exhibit strong metastability (Sec. 4.3.3). We highlight some salient features of the dynamics for the TZ1 kinetic network. In particular, we note deviations from simple two-state behaviour that arise from the presence of metastable intermediate states, the existence of multiple competing kinetically relevant pathway ensembles, and the increased localization of the TPE in the state space with decreasing temperature.

4.2 Methodology

4.2.1 Master equation dynamics

The simulation methodology and theory that we present in the current work are applicable to an arbitrary discrete- or continuous-time finite Markov chain. We illustrate our approach with a kinetic network constructed using geometry optimization methods[158] to locate the stationary points on a potential energy landscape. Here, local minima and transition states are mapped to the nodes and weighted, bidirectional edges, respectively, of a CTMC.[11–13] The set of nodes $\mathcal{S}$ constitutes the (finite) state space. The edge weights are minimum-to-minimum rate constants, usually estimated by harmonic transition state theory,[159] although any unimolecular rate theory, including methods based on explicit dynamics, may be used. Formally, the partition function can be written as a weighted sum of contributions from local

minima,[160,161] and so all thermodynamic properties can be extracted from the discretized representation of the energy landscape if the local densities of states are known. In practice, the equilibrium occupation probabilities associated with the nodes are usually approximated assuming a locally harmonic density of states.[16,161] Kinetic networks constructed by geometry optimization typically contain tens to hundreds of thousands of nodes, and are sparse.[64] Further details of the discrete path sampling[14,15] (DPS) methodology for the construction and analysis of kinetic networks can be found in recent reviews.[160,162,163]

Markovian dynamics on a kinetic network are described by the linear master equation,[2–5]

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} \Big(K_{ij} p_j(t) - K_{ji} p_i(t) \Big), \tag{4.1}$$

which can be written in matrix notation,

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{K}\mathbf{p}(t). \tag{4.2}$$

Here, $\mathbf{p}(t) = (p_1(t), p_2(t), \ldots, p_{|\mathcal{S}|}(t))^\top$ is the time-dependent occupation probability vector for the nodes of the kinetic network, and $\mathbf{K}$ is the transition rate matrix, as in previous chapters. The off-diagonal elements K_{ij} of $\mathbf{K}$ are the rates for the $i \leftarrow j$ internode transitions. The diagonal elements are set so that the columns of the matrix sum to zero, $K_{jj} = -\sum_{\gamma \neq j} K_{\gamma j}$. At equilibrium, the occupation probability vector is equal to the stationary probability vector $\boldsymbol{\pi}$. In kinetic networks derived from stationary point databases, the detailed balance condition, $K_{ij}\pi_j = K_{ji}\pi_i \;\forall\, i \neq j$, is necessarily satisfied if the densities of states for minima and transition states are assumed to be locally harmonic.[58]

Typical numerical methods for the linear algebra solution of the master equation, $\mathbf{p}(t) = \exp(\mathbf{K}t)\mathbf{p}(0)$, have time complexity $\mathcal{O}(|\mathcal{S}|^3)$, and this direct approach is therefore intractable for kinetic networks of relatively high dimensionality.[54,55] Moreover, for systems exhibiting rare event dynamics, numerical instability is a pervasive problem in methods to calculate the matrix exponential, and propagation of numerical error similarly affects dense linear algebra methods to calculate the MFPT between two endpoint macrostates.[59,60,65] It is the aim of the present chapter to sample the exact solution to the linear master equation (Eqs. 4.1 and 4.2), for transitions between two endpoint macrostates of interest $\mathcal{A}$ and $\mathcal{B}$, by the explicit simulation of trajectories using accelerated kMC.

Let an $a \leftarrow b$ first passage trajectory on a kinetic network (*i.e.* a discrete path[14,15]) connecting nodes $a \in \mathcal{A}$ and $b \in \mathcal{B}$, where $\mathcal{B}$ is the initial and $\mathcal{A}$ the absorbing macrostate, respectively, be denoted by the ordered sequence of visited nodes $\xi^{(a \leftarrow b)} = (a \leftarrow i_n \leftarrow i_{n-1} \leftarrow \ldots \leftarrow i_1 \leftarrow b)$. Here, i is used to denote nonabsorbing nodes, $i \in \mathcal{A}^\mathsf{c}$. The probability

$\mathcal{P}[\xi^{(a\leftarrow b)}]$ of this $\mathcal{A} \leftarrow \mathcal{B}$ first passage path is simply a product of branching probabilities $P_{ij} = K_{ij}/\sum_{\gamma\neq j} K_{\gamma j}$ for the internode transitions along the path, weighted by the probability of a trajectory starting from the initial node b, $\pi_b/\sum_{b'\in\mathcal{B}} \pi_{b'}$. The contribution of a discrete path to the $\mathcal{A} \leftarrow \mathcal{B}$ steady state rate constant[59,164] is the path probability $\mathcal{P}[\xi^{(a\leftarrow b)}]$ weighted by the inverse of the mean waiting time $\tau_b = 1/\sum_{\gamma\neq b} K_{\gamma b}$ for the initial node b. Thus, if the weights associated with $i \leftarrow j$ edges are chosen to be $-\ln P_{ij}$, the set of first passage paths that make the dominant contributions to the steady state rate constant for a kinetic network with a single source node can be extracted using k shortest path algorithms.[62–64] The contribution of an individual first passage path to the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT is given by the product of the path probability $\mathcal{P}[\xi^{(a\leftarrow b)}]$ and the sum of mean waiting times for nodes along the path, excluding the absorbing node a. The $\mathcal{A} \leftarrow \mathcal{B}$ MFPT, and hence the steady state and non-steady state phenomenological rate constants,[59,164] can be calculated robustly by the graph transformation method.[56–61] The kMC methods described below sample paths of the $\mathcal{A} \leftarrow \mathcal{B}$ TPE in proportion to their probabilities $\mathcal{P}[\xi^{(a\leftarrow b)}]$, and therefore yield an unbiased estimate for the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT.

4.2.2 Rejection-free kinetic Monte Carlo

The BKL (or *n-fold way*) algorithm,[83] which we introduced in Chapter 1, is a rejection-free[75,82,165] formulation of the kMC simulation method. In this section, we briefly outline the algorithm in the context of simulating the solution to the master equation for a kinetic network described by the rate matrix $\mathbf{K}$ (Eq. 4.2). Internode transitions $i \leftarrow j$, associated with rates K_{ij}, are assumed to be independent Poisson processes associated with average transition times equal to the waiting time for the j-th node, τ_j. For each node j, a list of possible transitions is constructed, and at each iteration of the algorithm a transition event is selected randomly with probability equal to the branching probability $P_{ij} = K_{ij}/\sum_{\gamma\neq j} K_{\gamma j}$. The move is always accepted, and the simulation clock is incremented by drawing a random value from the exponential distribution of waiting times between transitions, with rate parameter τ_j^{-1}, $p(\Delta t) = \tau_j^{-1}\exp(-\tau_j^{-1}\Delta t)$.[82] The exponential form for the distribution of waiting times follows from the fact that the competing independent Poisson processes for the $i \leftarrow j$ transitions together generate a new Poisson distribution. Advancing the system clock by Δt is achieved in practice by drawing a uniform random number $r \in (0, 1]$ and setting $\Delta t = -\tau_j \ln r$; a formal derivation of this algorithm is presented in Refs. 71 and 84.

Since the BKL algorithm uses the branching probabilities for transition events, the algorithm produces realizations of trajectories governed by the linear master equation (Eq. 4.2) with the exactly correct statistical frequency. Many independent trajectories are required

to achieve proper sampling of $\mathbf{p}(t)$ and hence produce a converged solution of the master equation. Although rejection-free kMC methods eliminate the possibility of self-transitions, and therefore of becoming trapped in any single node associated with a small escape rate, standard simulations can be very inefficient for kinetic networks featuring a separation of characteristic timescales.[77–80] In the following sections, we discuss how the standard BKL algorithm can be extended and modified to incorporate enhanced sampling methods based on a specification of disjoint sets of nodes, which enables this timescale problem to be overcome.

4.2.3 Identifying metastable states of a kinetic network

Both WE-kMC (Sec. 4.2.4) and kPS (Sec. 4.2.5) require a specified partitioning of the state space, or a criterion for defining the metastable macrostates on-the-fly. This partitioning is an essential consideration that strongly affects the efficiency of the algorithms.[156] The community structure must faithfully represent the coordinates corresponding to the independent slow dynamical modes of the system, which describe the rare transition events between metastable states. Hence, the partitioning should appropriately reflect the progress of the overall reactive transition between the two specified endpoint macrostates $\mathcal{A}$ and $\mathcal{B}$.

Here we propose the use of multi-level regularized[150–152] Markov clustering[153–155] (MLR-MCL) to efficiently characterize the metastable sets of nodes in high-dimensional and ill-conditioned kinetic networks, at an adjustable level of timescale resolution. MLR-MCL is a stochastic unsupervised community detection algorithm that uses heuristic operations on a coarse-grained transition probability matrix to obtain a clustering that characterizes the average behaviour of random walks on the network. The partitioning determined by MLR-MCL should therefore characterize the metastable sets of nodes, and hence provides appropriate predefined and fixed bins for use in WE-kMC and kPS simulations.[167] We have found that partitionings of the network obtained using MLR-MCL typically provide an accurate representation of the metastable macrostates,[156] as indicated by widely used graph-theoretic metrics, such as the weighted normalized cut and the conductance.[168] Furthermore, the partitioning can be subsequently refined by variational optimization of the dominant nonzero eigenvalue of the transition rate matrix.[169] Additional detail concerning MLR-MCL and the variational optimization procedure is provided in Appendices 4.A and 4.B, respectively.

The resolution of the community detection can be directly controlled via the choice of input parameters to MLR-MCL, namely the granularity parameter, the number of iterations of the multi-level graph coarsening algortihm, and the lag time at which the initial transition probability matrix is estimated.[150,151] The tuning of these parameters allows flexibility in

the timescale at which the determined macrostates appear metastable. This feature is highly desirable, as the efficiency of WE-kMC is strongly affected by the average escape time from the macrostates, since coordinates that are not correlated with the bins must effectively be sampled by brute force.[48] Similarly, the efficiency of kPS is strongly affected by the size of the macrostates. For this reason, it is favourable that MLR-MCL penalizes overly large communities, thereby addressing a deficiency of the original MCL algorithm.[150,151] Because MLR-MCL is an unsupervised learning algorithm, the overall workflow for the WE-kMC and kPS simulations does not require any prior knowledge of the system. This feature addresses one of the key problems with enhanced sampling methods based on a division of the state space, namely that determination of an appropriate partitioning is a highly non-trivial problem.[17–19,93,137]

Neither WE-kMC nor kPS require that the metastable macrostates are pre-defined or fixed throughout the simulation. We therefore also investigate defining the macrostates adaptively, using a search protocol similar to that employed in AS-kMC.[103] Beginning with the currently occupied node, a breadth-first search procedure is used to build up a group of nodes that are mutually interconnected by transition rates that exceed a specified threshold, thereby ensuring that the resulting subnetwork is 'well-knit'. The search is terminated when the size of the macrostate exceeds a specified limit, or when all transitions to neighbouring nodes of the subnetwork are associated with small transition rates.

4.2.4 Weighted ensemble kinetic Monte Carlo (WE-kMC)

The weighted ensemble (WE) algorithm[93] is a method for resampling[6,170,171] the path probability distribution in pathway space. The procedure can simulate exact nonequilibrium[132] or equilibrium[134] dynamics for a variety of stochastic processes, including Langevin dynamics[136] and dynamics of stochastic reaction networks.[48–51] The WE method employs a partitioning of the state space into bins, which can be performed adaptively,[133,172] and a set of independent trajectories ('walkers'), each associated with a statistical weight. A stochastic splitting and culling procedure, carried out at regular time intervals τ_{R}, maintains a target number of trajectories in each bin throughout the simulation, thus ensuring representative sampling of the entire state space. This resampling procedure is exact for Markovian dynamics, and generates an unbiased sample of the path ensemble for a $\mathcal{A} \leftarrow \mathcal{B}$ transition.[132–134] The independence of the simulated trajectories leads to linear parallel scaling.

To employ the WE methodology for nonequilibrium $\mathcal{A} \leftarrow \mathcal{B}$ stochastic dynamics, we define a set of non-overlapping bins, and specify target numbers of walkers, M_ξ, for each bin, which are not necessarily equal and which can be updated on-the-fly. We also specify a time

interval τ_{R} for conducting the walker resampling procedure. The WE simulation begins by spawning a specified number of weighted trajectories, where the sum of the weights is unity. The starting trajectories can be set up according to any desired initial condition, including spawning many trajectories in multiple different bins and with non-uniform weights. The stochastic dynamics are propagated independently for each of the trajectories, and when all trajectories have exceeded the time interval for resampling, all populated bins are checked for the total numbers of trajectories that they contain. If the number of trajectories in a given bin is less than the target number, then trajectories occupying the bin are chosen randomly, in proportion to their relative weights, to be split. The newly-spawned trajectories each inherit an equal share of the weight of the parent trajectory, and they all share the history of the parent. This procedure is repeated until the number of trajectories in the bin exceeds the target value. If the number of trajectories in a bin is greater than the target number, then trajectories are culled by randomly selecting one of the two trajectories of lowest weight in the bin to survive, in proportion to their relative weights, and breaking ties arbitrarily. The surviving trajectory inherits the weight of the culled trajectory, and this procedure is repeated until the target number of trajectories is met. Then the walkers of the new set of trajectories are again propagated independently until the time for the next resampling operation is reached. The times and weights of walkers hitting the absorbing macrostate $\mathcal{A}$ are recorded, yielding the FPT distribution and an estimate for the MFPT straightforwardly. An overview of the WE-kMC algorithm is illustrated in Fig. 4.1. Because the WE simulation distributes resources to sampling the entire state space, including the crossing of dynamical bottlenecks,[173] the simulation can yield the pathway ensemble even for transitions associated with timescales that are far too long to be accessible by brute force.[93] The WE simulation is repeated many times to achieve sufficient sampling.

The legitimacy of resampling the trajectory ensemble is justified by a simple factorization of the path probability.[133] Let $\xi^{(n\leftarrow m)} = (n \leftarrow n-1 \leftarrow \ldots \leftarrow m+1 \leftarrow m)$ be a trajectory initially at node m and terminating at node n. The path probability $\mathcal{P}[\xi^{(n\leftarrow m)}]$ can be factorized as

$$\begin{aligned}\mathcal{P}[\xi^{(n\leftarrow m)}] &= p_m(0) \prod_{k=m}^{n-1} P_{k+1,k} \\ &= p_m(0) \prod_{k=m}^{m'-1} P_{k+1,k} \prod_{k=m'}^{n-1} P_{k+1,k} \\ &= \mathcal{P}[\xi^{(m'\leftarrow m)}] \prod_{k=m'}^{n-1} P_{k+1,k}. \end{aligned} \tag{4.3}$$

Here, $\mathbf{p}(0)$ specifies the initial probability distribution over nodes, and P_{ij} is the branching

probability for the $i \leftarrow j$ transition. Consider resampling the path probability distribution at a timestep when a particular trajectory $\xi^{(m' \leftarrow m)}$ currently occupies the arbitrary node m'. From Eq. 4.3, any resampling procedure that exactly preserves the probability distribution of trajectories at that time, such as the splitting and culling procedure outlined above, exactly yields the correct probability distribution in pathway space at all future times, provided that members of the new set of trajectories inherit the histories of the trajectories from which they were derived. That is, if the path probabilities associated with any daughter trajectories $\xi^{(n \leftarrow m)}$ spawned from $\xi^{(m' \leftarrow m)}$ are weighted by an equal share of the weight of the parent trajectory, then the correct path probability distribution is preserved. This factorization argument holds for both Markovian and non-Markovian dynamics.

Although the WE method generates an unbiased sample of the pathway ensemble, the correlated histories of the trajectories is problematic when attempting to make accurate statistical estimates of dynamical properties from WE simulation data.[134,135,174] For instance, although the committor probabilities[19] can in principle be computed by tracing the $\mathcal{A} \leftarrow \mathcal{B}$ paths,[47] in practice, reliable estimation requires averaging over the results of many independent WE simulations, to mitigate the effect of the correlated histories of the trajectories within a given WE run. This feature means that WE sampling, and other methods that simulate complete $\mathcal{A} \leftarrow \mathcal{B}$ trajectories by piecing together trajectory segments, such as forward flux sampling,[47,91,116,175–179] are not ideal for the estimation of committor probabilities.

For completeness, we note that WE-kMC can also be used to sample the equilibrium TPE. A steady state must eventually be reached if, when a walker reaches the endpoint macrostate $\mathcal{A}$ in the course of the WE simulation, it is placed back in the initial macrostate $\mathcal{B}$ with its current weight.[180] Let the total weight of trajectories in bin I at the steady state be denoted by w_I^{SS}. The steady state bin weights $\{w_I^{\mathrm{SS}}\}$ satisfy[48]

$$\frac{dw_I^{\mathrm{ss}}}{dt} = \sum_{J \neq I} \left(K_{JI} w_I^{\mathrm{SS}} - K_{IJ} w_J^{\mathrm{SS}} \right) = 0, \tag{4.4}$$

where K_{IJ} denotes the rate for the $I \leftarrow J$ inter-community transition (note the use of capital letter indices to denote macrostates, as opposed to nodes). Eq. 4.4 suggests an iterative scheme, where, by comparing the measured inter-bin fluxes with the expression $F_{IJ} = K_{IJ} w_J$, the transition rates K_{IJ} can be inferred, and then the weights of individual trajectories within the bins can be rescaled so that the $\{w_I\}$ are consistent with Eq. 4.4. It has been demonstrated empirically that this protocol can greatly accelerate the convergence of the system to the true steady state.[134] When the ensemble of walkers has equilibriated,

the $\mathcal{A} \leftarrow \mathcal{B}$ steady state rate constant is given by[135,174]

$$k_{\mathcal{AB}}^{\mathrm{SS}} = \sum_{J \neq \mathcal{A}} F_{\mathcal{A}J}, \tag{4.5}$$

where we have used the well-known Hill relation[181] for the MFPT. We only consider nonequilibrium simulations in the present chapter, and hence we do not obtain the MFPT via Eqs. 4.4 and 4.5. Instead, the MFPT is computed directly as an average over FPTs for the weighted first passage paths.

4.2.5 Kinetic path sampling (kPS)

Kinetic path sampling[147,148] (kPS) is a method for sampling the solution to the linear master equation (Eq. 4.2) without requiring explicit kMC simulation of trajectories. We described kPS in detail in Chapter 1, and we give an overview of the key features of the procedure in the following. The kPS algorithm uses graph transformation[56–61] to reduce the representation of an escape trajectory from the active metastable macrostate. An overview of the stages of the kPS algorithm is illustrated in Fig. 4.2. To generate an escape path from the currently occupied trapping basin, we first define the sets of nodes that constitute the basin $\mathbb{B}$ and the absorbing macrostate $\mathbb{A} \equiv \mathbb{B}^{\mathsf{c}}$. A subset of $|\mathbb{E}|$ nodes of the trapping basin $\mathbb{E} \subseteq \mathbb{B}$ are marked for elimination and queued. The remaining nodes of the trapping basin, which constitute the set $\mathbb{T} \subset \mathbb{B}$, where $\mathbb{B} \equiv \mathbb{E} \cup \mathbb{T}$, are the retained transient nodes. The nodes $\partial\mathbb{A} \subseteq \mathbb{A}$ at the boundary of the absorbing state $\mathbb{A}$, *i.e.* directly connected to at least one node of the set $\mathbb{B}$, are identified. The graph transformation algorithm[56–60] is then used to construct the set of transition probability matrices $\{\mathbf{T}^{(n)}(\tau)\}$, $0 \leq n \leq |\mathbb{E}|$. That is, the set of matrices $\{\mathbf{T}^{(n)}\}$ are formed by the iterative elimination of the $|\mathbb{E}|$ nodes in the set $\mathbb{E} \subseteq \mathbb{B}$ from the subnetwork $\mathbb{B} \cup \partial\mathbb{A}$, where renormalization of the transition probabilities preserves the individual path probabilities and the MFPT for the set of escape trajectories from the current node to the absorbing boundary $\partial\mathbb{A}$.[61] The initial stochastic matrix, $\mathbf{T}^{(0)}$, may be the linearized transition matrix[82] estimated at a lag time τ, or the branching probability matrix. In the latter case, the mean waiting times associated with individual nodes are, in general, non-uniform.

A stochastic path from the currently occupied node $\epsilon \in \mathbb{B}$ to an absorbing node $\alpha \in \partial\mathbb{A}$ at the boundary of the metastable basin is randomly generated by repeatedly drawing new nodes according to a probability distribution defined from $\mathbf{T}^{(0)}$ and $\mathbf{T}^{(|\mathbb{E}|)}$. Concomitantly, a count matrix $\mathbf{H}^{(n)}(\tau)$, containing the number of internode transitions observed for dynamics based on $\mathbf{T}^{(n)}(\tau)$, is recorded for $n = |\mathbb{E}|$. The elements of the matrices $\mathbf{T}^{(n)}$ and $\mathbf{H}^{(n)}$,

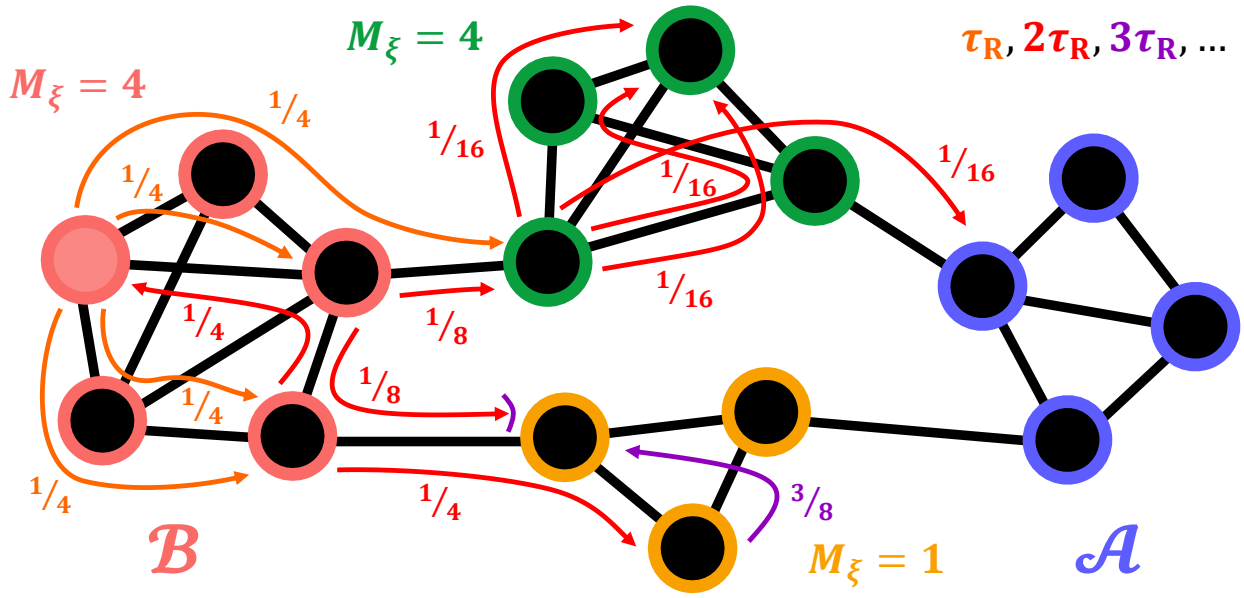


Figure 4.1: An overview of the WE-kMC algorithm. To facilitate sampling the $\mathcal{A} \leftarrow \mathcal{B}$ path ensemble for the transition between two endpoint macrostates of interest, the network is divided into communities characterizing the metastable sets of nodes. Each community is associated with a target number of trajectories, M_ξ. A number of walkers, associated with statistical weights that sum to unity, are spawned according to a specified initial distribution. In the above figure, four walkers with uniform weights are spawned at a particular node (highlighted) of the initial macrostate $\mathcal{B}$. The walkers are propagated independently, and a resampling procedure, carried out at time intervals of τ_R, maintains the weighted set. In the example above, after a time τ_R, one of the walkers has transitioned to another community. The target number of walkers for this community is $M_\xi = 4$, and so the single walker that currently occupies the community is split into four. Each of the daughter trajectories inherits the history of the parent trajectory and an equal share of the weight. To maintain the target number of trajectories in the initial community $\mathcal{B}$, for which $M_\xi = 4$, one of the walkers currently occupying $\mathcal{B}$ is selected, with probability proportional to its weight, and split into two. After time $2\tau_R$, one walker reaches the absorbing macrostate $\mathcal{A}$, and its (weighted) contribution to the $\mathcal{A} \leftarrow \mathcal{B}$ MFPT is recorded. Also at this time, two walkers transition from $\mathcal{B}$ to reach a new community. The target number of walkers for this community is $M_\xi = 1$. Therefore one walker currently occupying this community is chosen to survive, with probability proportional to its statistical weight. The other walker is culled, and the surviving walker, which retains its history, inherits the weight of the culled walker.

for which the n-th node has been eliminated compared to $\mathbf{T}^{(n-1)}$ and $\mathbf{H}^{(n-1)}$, subsume all indirect internode transitions that proceed with n as an intermediate node, where n is visited an arbitrary number of times. Modeling the dynamics using one of the reduced transition matrices therefore greatly reduces the complexity of a sampled escape path. An escape trajectory from a node $\epsilon \in \mathbb{B}$ based on $\mathbf{T}^{(|\mathbb{E}|)}$ contains only a single step, notwithstanding any transitions involving nodes of the set $\mathbb{T}$, if $\mathbb{T} \neq \emptyset$. The kPS algorithm then exploits the fact that $\mathbf{H}^{(n-1)}$ can be generated stochastically from $\mathbf{H}^{(n)}$ given $\mathbf{T}^{(n)}$ and $\mathbf{T}^{(n-1)}$, without explicit simulation of the dynamics using $\mathbf{T}^{(n-1)}$. Note that the sampling rules do not allow self-loop transitions for nodes of the set $\mathbb{T}$, and the kPS algorithm reduces to standard rejection-free kMC[83] in the case where $\mathbb{E} = \emptyset$ and therefore $\mathbb{B} \equiv \mathbb{T}$.

The result of the repeated application of the iterative reverse randomization procedure is the hopping matrix $\mathbf{H}^{(0)}$, for which the elements are the numbers of internode kMC moves along a detailed stochastic path within the trapping basin. This matrix can therefore be used to generate a time associated with the trajectory escaping to the sampled absorbing node $\alpha \in \partial\mathbb{A}$, by sampling from a Gamma distribution. kPS produces escape paths to absorbing nodes that are exactly consistent with the linear master equation (Eq. 4.1), and does not necessarily require *a priori* knowledge of the metastable basins. Since the number of kMC moves along the sampled escape path is calculated within the kPS algorithm, the simulation can always revert to the standard BKL algorithm on-the-fly when it is favourable to do so.

4.3 Results

4.3.1 Simulation setup and performance

We illustrate the sampling methods described above with results for a kinetic network representing the folding of the tryptophan zipper peptide TZ1,[157] constructed by discrete path sampling (DPS).[14,15] The system was modeled using an atomistic potential and implicit solvent. Further details of the force field and the DPS procedures employed, and some preliminary analysis of the dynamics, for instance using Dijkstra's algorithm with appropriate edge weights[62–64] to determine the transition path that makes the dominant contribution to the steady state rate constant (Sec. 4.2.1), can be found in Ref. 157. The stationary point database for TZ1 contains 68780 minima and 99935 transition states, and the corresponding network constitutes a single fully connected component.

In the present chapter, we simulate the nonequilibrium dynamics when the probability density is initially localized at an unfolded node for which the peptide chain is extended (with no native or non-native contacts), and the native fold is treated as an absorbing

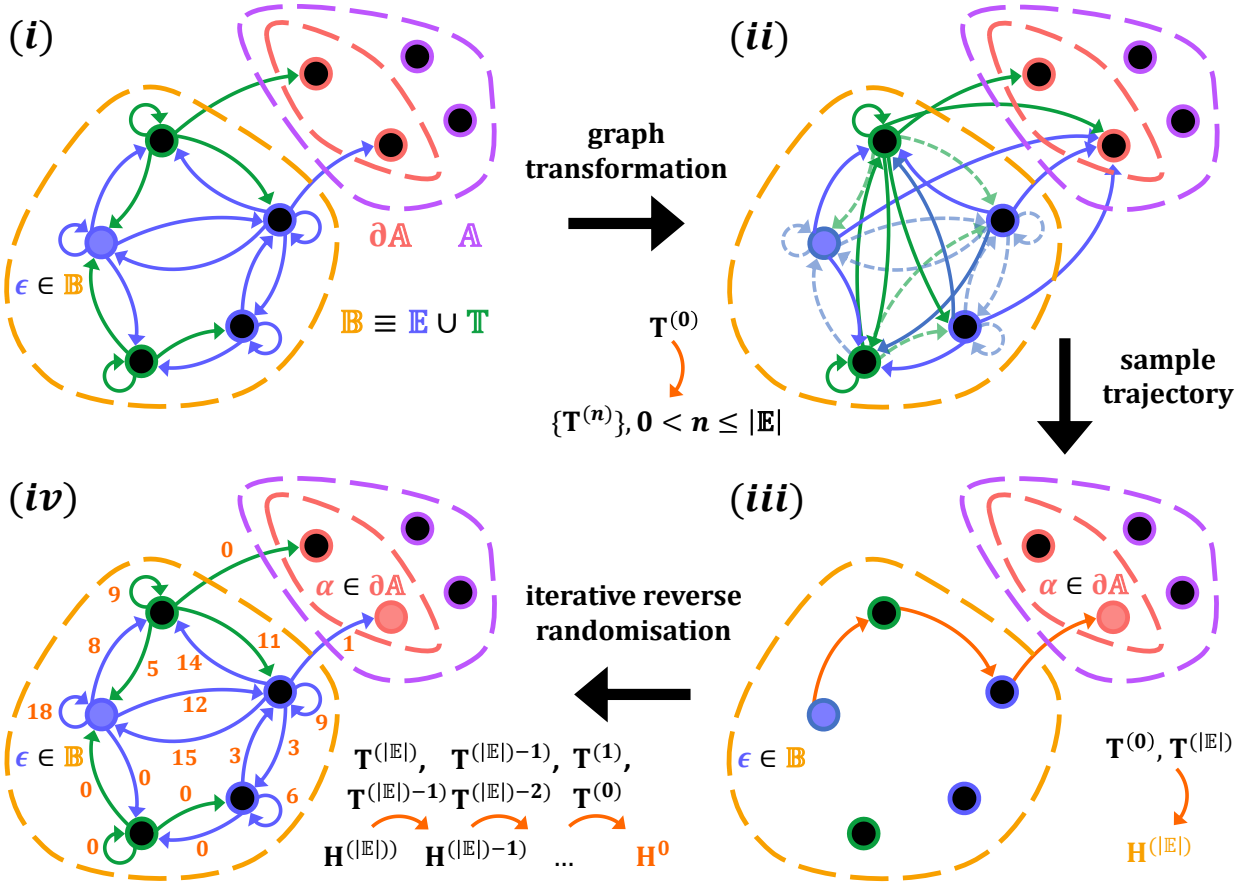


Figure 4.2: An overview of the stages of the kPS algorithm. (i) The network is divided into two sets, the trapping basin $\mathbb{B}$, containing the currently occupied node $\epsilon \in \mathbb{B}$, and the absorbing state $\mathbb{A} \equiv \mathbb{B}^{\mathsf{c}}$. A subset of the trapping basin, $\mathbb{E} \subseteq \mathbb{B}$, comprising nodes to be eliminated by graph transformation, is identified, and the set of remaining nodes is denoted $\mathbb{T} \equiv \mathbb{B} \setminus \mathbb{E}$. The subset of nodes of the absorbing macrostate that are directly connected to the trapping basin constitute the absorbing boundary, $\partial\mathbb{A} \subseteq \mathbb{A}$. The initial transition probability matrix for the subnetwork of interest, $\mathbb{B} \cup \partial\mathbb{A}$, is denoted $\mathbf{T}^{(0)}$. (ii) The $|\mathbb{E}|$ nodes of the set $\mathbb{E}$ are eliminated iteratively by graph transformation, which involves removing transitions to eliminated nodes and renormalization of the transition probabilities to preserve path probabilities for trajectories from the trapping basin to the absorbing boundary. The resulting transition matrices, $\{\mathbf{T}^{(n)}\}$, $0 < n \leq |\mathbb{E}|$, are stored. (iii) A path from the currently occupied node $\epsilon \in \mathbb{B}$ to a node of the absorbing boundary $\alpha \in \partial\mathbb{A}$ is sampled according to a probability distribution based on $\mathbf{T}^{(0)}$ and $\mathbf{T}^{(|\mathbb{E}|)}$. The number of internode $i \leftarrow j$ transitions along this reduced representation of the stochastic escape trajectory thus generated are the elements of the hopping matrix $\mathbf{H}^{(|\mathbb{E}|)}$. (iv) An iterative reverse randomization procedure, exploiting the fact that $\mathbf{H}^{(n-1)}$ can be sampled from $\mathbf{H}^{(n)}$ using a probability distribution based on $\mathbf{T}^{(n)}$ and $\mathbf{T}^{(n-1)}$, is used to generate the elements of the hopping matrix $\mathbf{H}^{(0)}$, the elements of which are the numbers of $i \leftarrow j$ kMC moves on the original subnetwork $\mathbf{T}^{(0)}$. From this information, a time associated with the $\alpha \in \partial\mathbb{A} \leftarrow \epsilon \in \mathbb{B}$ trajectory can be sampled.

macrostate. Besides these unfolded (U) and folded (F) states, we identify two intermediate states of interest, I1 and I2. The I1 macrostate comprises partially folded conformations, which may include on- or off-pathway intermediate states with non-native contacts. The I2 macrostate comprises low-energy conformations for which the backbone is ordered similarly to the native state, but rearrangements of the side chains are required to transition to the native fold. Note also that the macrostate of unfolded structures (U) includes not only high energy extended structures such as the initial node, but also disordered structures that are collapsed (U′). The U, I1, and I2+F macrostates used in the analysis of the time-dependent occupation probability distributions are determined by MLR-MCL at a low resolution, and the folded macrostate F comprises only a small number of the lowest potential energy nodes that are interconnected by fast transition rates, identified manually. The WE-kMC and kPS simulations are based on 390 communities determined by MLR-MCL at a higher resolution, with the manually-chosen nodes of the set F designated as the absorbing macrostate. Further information on the community structure detection is included in Appendix 4.C. We compare the results from low (300 K) and high (330 K) temperature simulations. The kinetic network models in each case are obtained by calculating the transition rates and stationary probabilities at the chosen temperature, assuming locally harmonic densities of states for the minima and transition states on the potential energy landscape (Sec. 4.2.1).

The results from the kPS and WE-kMC simulations, each obtained from 20000 first passage paths, are nearly identical, and the MFPTs estimated from the simulation data are consistent with the exact values calculated using graph transformation[56–61] (Table 4.1). Note that the FPTs reported here are not directly comparable with experiment; to analyze folding rates would require extended definitions of the endpoint sets of nodes to reflect the experimental states. Our principal interest here is in the diagnosis of alternative pathways and convergence of sampling algorithms. We do not provide a detailed performance comparison of the two methods, since their efficiency is strongly affected by the simulation setup, which is highly flexible, and the optimal parameter choices are system-dependent and can only be discovered empirically through extensive testing. We simply note that it is essential to incorporate appropriate considerations into the design of the simulation protocol, and that both accelerated kMC methods considered here are many orders of magnitude faster than brute-force kMC. In kPS simulations, it is essential that the communities are not too large, since the graph transformation procedure then incurs a significant computational overhead, and that the simulation reverts to the standard rejection-free kMC algorithm when it is favourable to do so. In the present chapter, a fixed number of rejection-free kMC steps are taken after each kPS basin escape, to ensure that the trajectory moves away

from the boundaries between communities, thereby avoiding the computation of expensive kPS iterations for trivial recrossings between communities. We use the same communities, determined by MLR-MCL, for the kPS and WE-kMC simulations, but the optimal choices for the communities in each case could be rather different,[156] owing to the various factors affecting the efficiency of the methods. The simulation parameters are described in Appendix 4.C.

The kPS simulation data presented below were obtained in approximately 200 CPU hours using Intel Core i7-5820K 3.30GHz processors. The CPU time was the same for simulations at temperatures of $T = 300\,\mathrm{K}$ and $T = 330\,\mathrm{K}$, since the time complexity of kPS is largely independent of the metastability of the kinetic network.[60,147] The WE-kMC simulation data were obtained in approximately 1000 CPU hours for a temperature of 300 K, and in around 840 CPU hours for a temperature of 330 K. In contrast, simulation of a single folding trajectory at 330 K by the BKL algorithm (Sec. 4.2.2) requires, on average, around 10 hours of CPU time, and low-probability paths at the tail of the FPT distribution require much more CPU time. Brute-force kMC simulation is therefore unfeasible. Both kPS and WE-kMC simulations employing an adaptive definition of the communities were slower than simulations based on predefined communities determined by MLR-MCL, which suggests that the MLR-MCL communities accurately characterize the metastable macrostates and are appropriately balanced in size. Therefore the additional computational time associated with the breadth-first search procedure to identify communities on-the-fly is an unnecessary computational expense for this system.

The superior performance of kPS compared to WE-kMC in this particular instance can be ascribed to the presence of strong kinetic traps in the kinetic network, for which a very large number of kMC steps (sometimes more than 10^{12}) are required to escape the corresponding community. The computational time for a kPS iteration is essentially agnostic to the number of kMC steps for internode transitions, which are the elements of the hopping matrices $\{\mathbf{H}^{(n)}\}$, and are not explicitly simulated. Instead, these values only enter the calculation as parameters in the binomial and negative binomial distributions from which the elements of the next hopping matrix $\mathbf{H}^{(n-1)}$ are drawn, and in the Gamma distribution from which the time associated with the basin escape trajectory is sampled.[147,148] In WE-kMC, these kMC steps must be explicitly taken in order to escape from the community, and therefore the flickering problem,[77–80] while not as serious as in standard kMC, may still hinder the WE-kMC calculation. Hence, the required CPU time for the WE-kMC simulation is adversely affected by decreasing temperature. We anticipate that with alternative computational resources or refinement of the WE-kMC simulation protocol, for instance by dividing the state space into more communities, and increasing the target numbers of walkers for particular communities, significant gains in the efficiency of the WE-kMC calculation could be achieved.

Table 4.1: MFPTs for the folding transition of TZ1 calculated by various methods. The graph transformation result is exact.[56–61] The values from WE-kMC and kPS explicit simulation data were calculated from 20000 first passage paths, and are associated with a standard error. The simulations were performed using a predefined and fixed partitioning of the network into communities determined by MLR-MCL.

	MFPT $\times 10^{11}$ / ns	
Method	$T = 300\,\mathrm{K}$	$T = 330\,\mathrm{K}$
Graph transformation	9.1275	3.3386
WE-kMC	8.8 $\pm$0.4	3.1 $\pm$0.3
kPS	8.9 $\pm$0.3	3.2 $\pm$0.2

4.3.2 Folding mechanism for the TZ1 peptide

To characterize the mechanistic features of the folding transition for TZ1, we calculate the vector $\mathbf{p}(t)$ containing the time-dependent occupation probabilities for the four states of interest, U, F, I1, and I2, described above (Fig. 4.3). Representative trajectories from the explicit simulations are shown in Fig. 4.4, alongside the transition path that makes the single largest contribution to the F $\leftarrow$ U steady state rate constant,[63,64,164] where the transition times are chosen to be the mean waiting times associated with the individual nodes along this shortest path. It is immediately apparent from these calculations that the F $\leftarrow$ U transition of TZ1 does not conform to a simple two-state model of the dynamics, since the macrostate I1 may persist on appreciable timescales (Fig. 4.3).

There is a rapid collapse of the initially occupied node of the state U, the extended conformation, to a more compact state (denoted U$'$), which similarly contains no native or non-native contacts. These conformations are therefore grouped into the same macrostate U in the calculation of the occupation probabilities for the key states, and this unfolded macrostate has a lifetime of around 10^6 ns (nanoseconds). Following escape from the unfolded macrostate U, around 70% of the paths avoid becoming trapped in the I1 state, and subsequently the I2 state, and therefore there is a relatively steep increase in the occupation probability of the native folded state F, on a timescale of around 10^8 ns. The remaining $\sim 30\%$ of the first passage paths do not follow this simple fast-folding mechanism, but instead become trapped in the I1 state, with a lifetime of around 10^{12} ns, before rapidly proceeding to the native state F via the I2 state. The latter folding mechanism becomes slightly more favoured at the lower temperature (Fig. 4.3). The kinetic traps in the TZ1 kinetic network that have a significant effect on the folding dynamics therefore correspond to partially folded conformations where the backbone is not properly arranged (I1). Kinetic traps corresponding to low-energy states with an ordered backbone but improperly positioned side chains (I2) are not as strongly

metastable, and have a low probability to appear along a transition path. They therefore have only a small effect on the folding dynamics (Fig. 4.3). Hence, once the peptide backbone adopts a conformation similar to the native state, the peptide almost always proceeds rapidly to the native fold.

The single path that makes the dominant contribution to the steady state rate constant,[164] determined using Dijkstra's algorithm with appropriate edge weights,[62–64] is clearly not representative of explicitly simulated trajectories (Fig. 4.4). We made the same observation for a model system in Chapter 2. While the sequence of conformational change events along this shortest path is consistent with the family of simulated first passage paths corresponding to fast downhill folding, the shortest path contains no useful temporal information for this system. That is, it is not possible to identify the states that, in practice, are associated with long lifetimes. For kinetic networks featuring metastability, realistic transition paths feature a large number of flickers.[77–80] For the shortest path at 300 K, the first passage time is $t_{\mathrm{FPT}} \approx 10^{3.5}$ ns, but the path probability of folding trajectories with $t_{\mathrm{FPT}} < 10^5$ ns is negligible (Fig. 4.5). Therefore the set of shortest paths, which can be determined by an appropriate k shortest paths algorithm,[63,64] make a negligible contribution to the folding flux for TZ1. Moreover, because the number of kMC steps along the second family of folding trajectories, which become trapped in the I1 state, is even larger than for the first family, the path probability for any one member of this family of longer-timescale trajectories is exceedingly small. Therefore these transition paths cannot feasibly be identified using a k shortest paths algorithm,[64] even though the paths collectively make an important contribution to the MFPT (Fig. 4.5). These observations demonstrate the value of explicitly simulating trajectories to obtain dynamical information.

Evidently, the folding energy landscape for TZ1 clearly does not satisfy the criteria outlined by Zwanzig in Ref. 182 that ought to be satisfied for the folding transition to exhibit simple two-state kinetics. It is true that there is effectively a single well-defined native folded node (the global potential energy minimum) and a large number of unfolded nodes, such that any one individual unfolded node makes only a very small contribution to the partition function for the unfolded macrostate. However, sets of unfolded nodes can be grouped into metastable clusters. The folding landscape therefore violates a vital condition required for the observation of two-state kinetics: relatively high energy barriers separate metastable unfolded, misfolded, and partially folded states, and so the full ensemble of non-native structures is not in overall local equilibrium. The existence of metastable on-pathway partially folded and off-pathway misfolded states complicates the dynamical behaviour of the peptide by acting as strong kinetic traps, and their effect is clear in the complex form of the FPT distribution (Fig. 4.5). The effect of some of the features of the

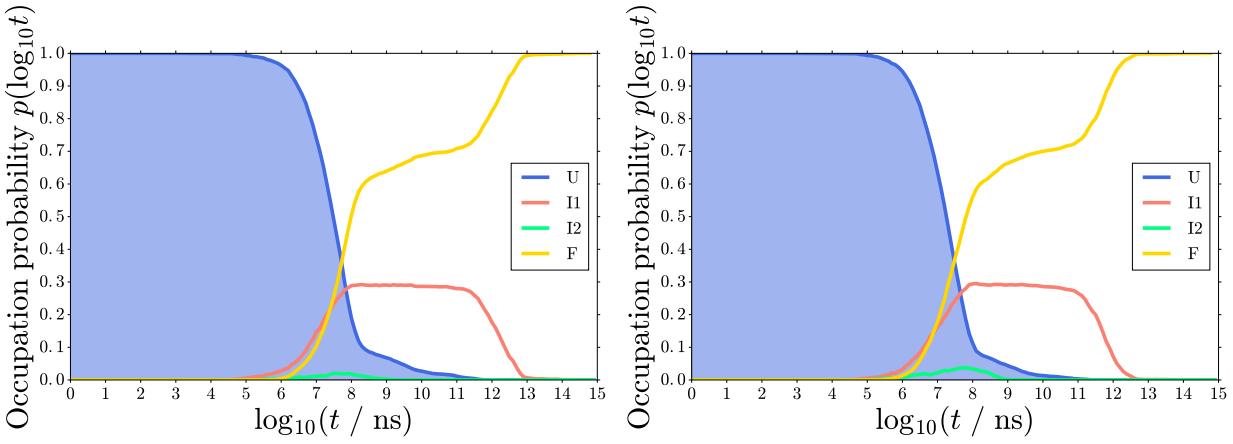


Figure 4.3: Time-dependent occupation probability distribution for the four key states in the course of the F ← U transition at (a) $T = 300\,\mathrm{K}$ and (b) $T = 330\,\mathrm{K}$, obtained from 20000 transition paths simulated using the kPS algorithm with *a priori* communities determined by MLR-MCL. The areas under the curves corresponding to the occupation probabilities of initial (U) and absorbing (F) states are shaded, to aid visualization of the progress of the folding transition.

folding landscape on the observed dynamics may be exacerbated by finite sampling of the database of stationary points on the potential energy landscape. However, the existence of deviations from a single-funnel energy landscape, and the consequent appearance of complex features in TPE statistics contrasting with simple fast-folding behaviour expected for single-funnel folding landscapes (Sec. 4.3.3), are not a result of sampling error. A two-state kinetic model oversimplifies the folding dynamics of TZ1, even though it is a small peptide that folds rapidly in experiments.[157]

4.3.3 Transition path ensemble statistics

The complete FPT distributions for the F ← U transition at temperatures of 300 K and 330 K are shown in Fig. 4.5. Notably, the FPT distribution for the folding transition of TZ1 does not follow a simple Poissonian form, but is instead double-peaked. The complex form of the FPT distribution reflects the fact that there exist multiple competing mechanisms that are kinetically relevant. In particular, the existence of two peaks in the FPT distribution, one corresponding to a much longer timescale, suggests that the paths can be broadly classified into two families, as noted in Sec. 4.3.2. The first family of paths, which constitute around 70% of the simulated transition paths, correspond to fast 'downhill' folding to the native state F. In contrast, members of the second family of paths become trapped in one or more metastable partially folded intermediate states, collectively represented by the macrostate I1. The separation of the first passage path ensemble into two competing sets of paths

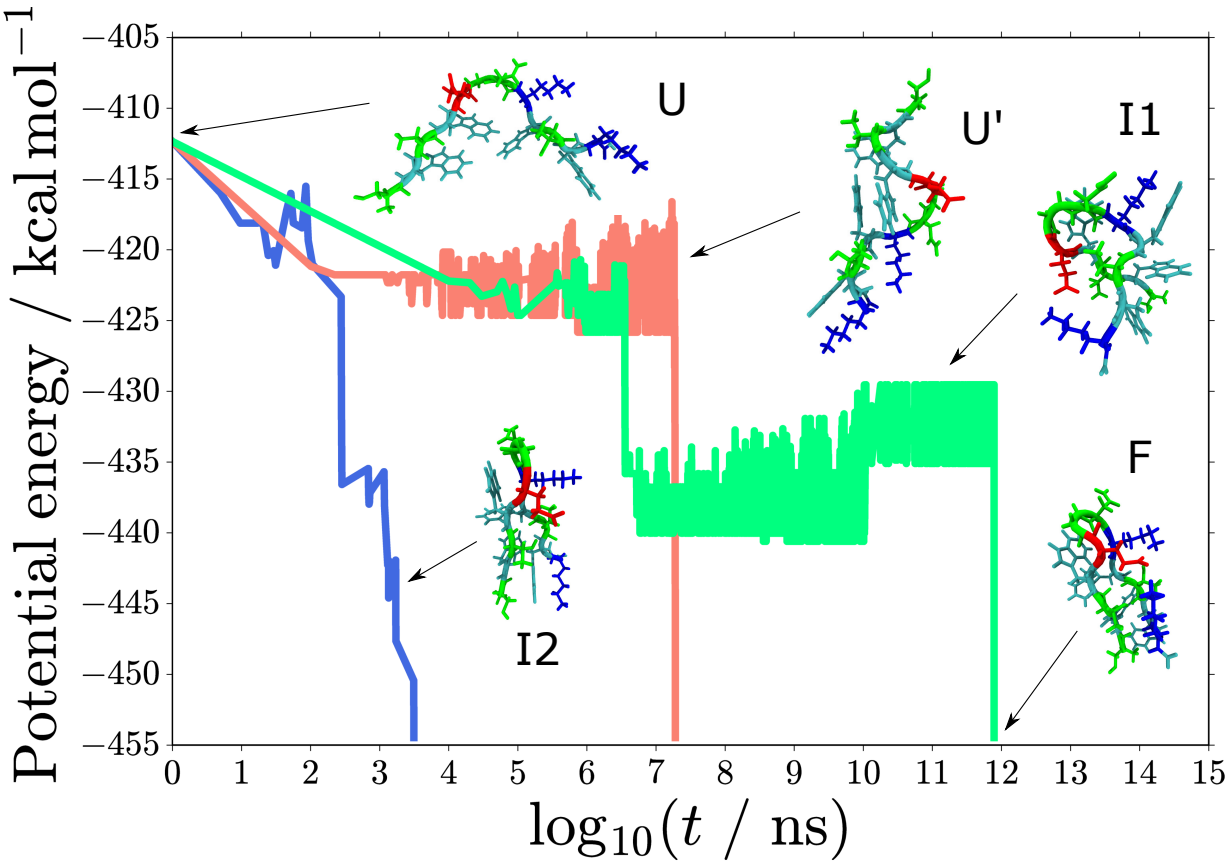


Figure 4.4: Representative trajectories for the F ← U transition. The path that makes the single largest contribution to the steady state rate constant[62–64] at a temperature of 300 K is shown in blue. Two representative trajectories, corresponding to each of the two major mechanisms for the folding transition, obtained from WE-kMC simulations at 300 K, are also shown. The trajectory marked in red corresponds to a straightforward folding mechanism in which the peptide is trapped in a collapsed unfolded state (U′) on a timescale of around 10^6 ns, and then rapidly folds to the native state (F). The trajectory shown in green corresponds to the second, more complex, mechanism in which the peptide becomes trapped in an intermediate partially folded state I1, with a lifetime of around 10^{12} ns. The existence of two separate kinetically relevant mechanisms, where one mechanism is associated with a significantly longer timescale, is evident from the FPT distribution in Fig. 4.5. The tryptophan residues, which arrange to form an interlocking zipper-like motif of aromatic rings in the native state, are coloured in cyan in the TZ1 structures.

that each make a substantial contribution to the MFPT, one corresponding to fast downhill transitions and the other corresponding to longer-timescale pathways that become trapped in a metastable intermediate state, has been observed in other simulation studies on the folding of simple peptides[183–186] and nucleic acid oligomers.[187,188] Both local maxima of the FPT distribution are shifted to longer timescales with decreasing temperature, and the difference in the MFPTs at the two temperatures is around half an order of magnitude (Table 4.1). The shorter-timescale peak in the FPT distribution is much sharper at the lower temperature, suggesting that a small number of transition paths in the subensemble of pathways corresponding to the fast-folding mechanism become increasingly dominant with decreasing temperature. Curiously, the longer-timescale peak of the FPT distribution actually becomes slightly broader at the lower temperature, owing to the increased influence of the kinetic traps.

There are a very small number of paths giving rise to a tail in the FPT distribution, with $t_{\mathrm{FPT}} \approx 10^{14}\,\mathrm{ns}$, collectively accounting for around 0.01% of the transition path probability. These paths become trapped in a metastable cluster of nodes corresponding to low-energy misfolded structures. From this kinetic trap, the peptide must largely unfold before transitioning to the native state is possible. It is relatively common for the FPT distributions in realistic kinetic networks to be fat-tailed, so that extremal values for the FPT make a non-negligible contribution to the MFPT.[48,189] For such systems, brute-force simulations are inefficient, since inadequate computational resources are used for representative sampling of the tail region of the FPT distribution, and it may be desirable to employ a trajectory reweighting scheme.[89]

To characterize the features of the TPE at a microscopic level of detail, we calculate the $\mathcal{A} \leftarrow \mathcal{B}$ committor probability q_j^+ and the $\mathcal{A} \leftarrow \mathcal{B}$ (reactive) visitation probability r_j^+ for the nodes j of the network. The $\mathcal{A} \leftarrow \mathcal{B}$ committor probability for the j-th node is defined as the probability that a trajectory initially at node j will reach the absorbing macrostate $\mathcal{A}$ before returning to the initial set $\mathcal{B}$.[19,47,59,60] By definition, $q^+_{b\in\mathcal{B}} = 0$ and $q^+_{a\in\mathcal{A}} = 1$. The $\mathcal{A} \leftarrow \mathcal{B}$ (reactive) visitation probability for the j-th node is defined as the conditional probability that a trajectory visits node j, given that the trajectory is a direct $\mathcal{A} \leftarrow \mathcal{B}$ transition path.[190] We introduced the reactive visitation probability in Chapter 3, and derived an expression for the $\{r_j^+\}$ that can be evaluated straightforwardly from the committor probabilities and from the fundamental matrix of the reducible Markov chain corresponding to the reactive process. The committor probability is an 'ideal' one-dimensional reaction coordinate characterizing the progress of the $\mathcal{A} \leftarrow \mathcal{B}$ transition, and is especially useful for identifying the transition state ensemble (TSE) region, defined by nodes associated with values for the committor probability close to 0.5.[47] The TSE essentially defines the boundary between the effective

basins of attraction associated with the endpoint macrostates $\mathcal{A}$ and $\mathcal{B}$.[19] The reactive visitation probability is a measure of the extent to which the TPE is localized in the state space, and provides a convenient metric for identifying sets of kinetically relevant pathways that are separated in the state space.

The committor and reactive visitation probabilities for nodes j, with $r_j^+ \geq 0.01$, for the F $\leftarrow$ U transition at $T = 330\,\mathrm{K}$ are shown in the form of potential energy disconnectivity graphs[191,192] in Fig. 4.6. At a temperature of 330 K, there are 7818 nodes with $r_j^+ \geq 0.01$ and 2008 nodes with $r_j^+ \geq 0.1$, compared to 68780 nodes in total. Thus the reactive visitation probability is quite localized in the state space. This is especially true at the lower temperature of 300 K, for which there are 4046 nodes with $r_j^+ \geq 0.01$ and 1541 nodes with $r_j^+ \geq 0.1$. The increased localization of the TPE in pathway space with decreasing temperature has also been observed in kinetic networks for peptide folding transitions constructed from replica exchange MD simulation data.[183] There are a very small number of nodes of the I2 state for which the values of the reactive visitation probability are close to unity, $r_j^+ \approx 1$ (Fig. 4.6b), and hence there exists a well-defined region of the state space through which the vast majority of folding transition paths are channeled. However, these nodes do *not* correspond to a dynamical bottleneck (*i.e.* to the TSE), since they are associated with committor probabilities close to unity, $q_j^+ \approx 1$. That is, the transition from these nodes to the native folded state F is largely irreversible. In fact, the folding transition almost always proceeds very rapidly once the I2 state is reached (Fig. 4.3). Another notable feature of the reactive visitation probability in the state space is the cluster of nodes for which $r_j^+ \approx 0.3$, which comprise a subset of the I1 macrostate. This observation is consistent with the simulated time-dependent occupation probabilities (Fig. 4.3) and the FPT distribution (Fig. 4.5), which show that around 30% of transition paths become trapped in the I1 macrostate and hence are of a comparatively long timescale. Although these nodes correspond to relatively high-energy partially folded structures, their associated committor probabilities are $q_j^+ \approx 1$, and therefore the peptide is strongly committed to folding at this point.

Inspection of the distribution of committor probabilities demonstrates that the folding transition of TZ1 exhibits multi-state kinetics (Fig. 4.6a). For an ideal two-state system, the vast majority of nodes are associated with committor probabilities $q_j^+ \approx 0$ or $q_j^+ \approx 1$, and can therefore be divided into two well-defined sets. The small number of nodes for which $q_j^+ \approx 0.5$ constitute the TSE, and have a dominant effect on the global dynamical properties of the $\mathcal{A} \leftarrow \mathcal{B}$ transition, including the MFPT.[19] Conversely, for a system exhibiting diffusive dynamics, there is a continuous spread of committor probabilities for nodes. Clearly, the distribution of the committor probabilities for nodes for the F $\leftarrow$ U transition of TZ1 does

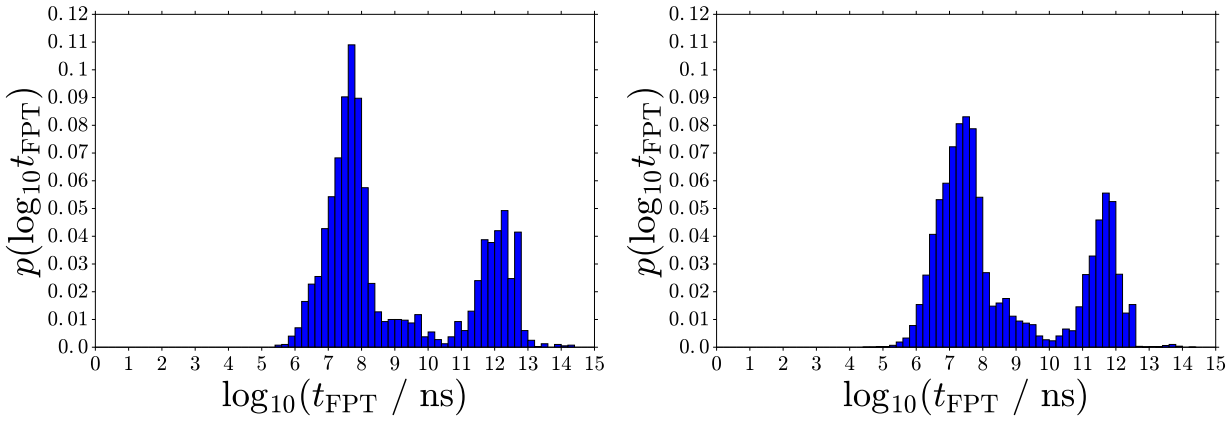


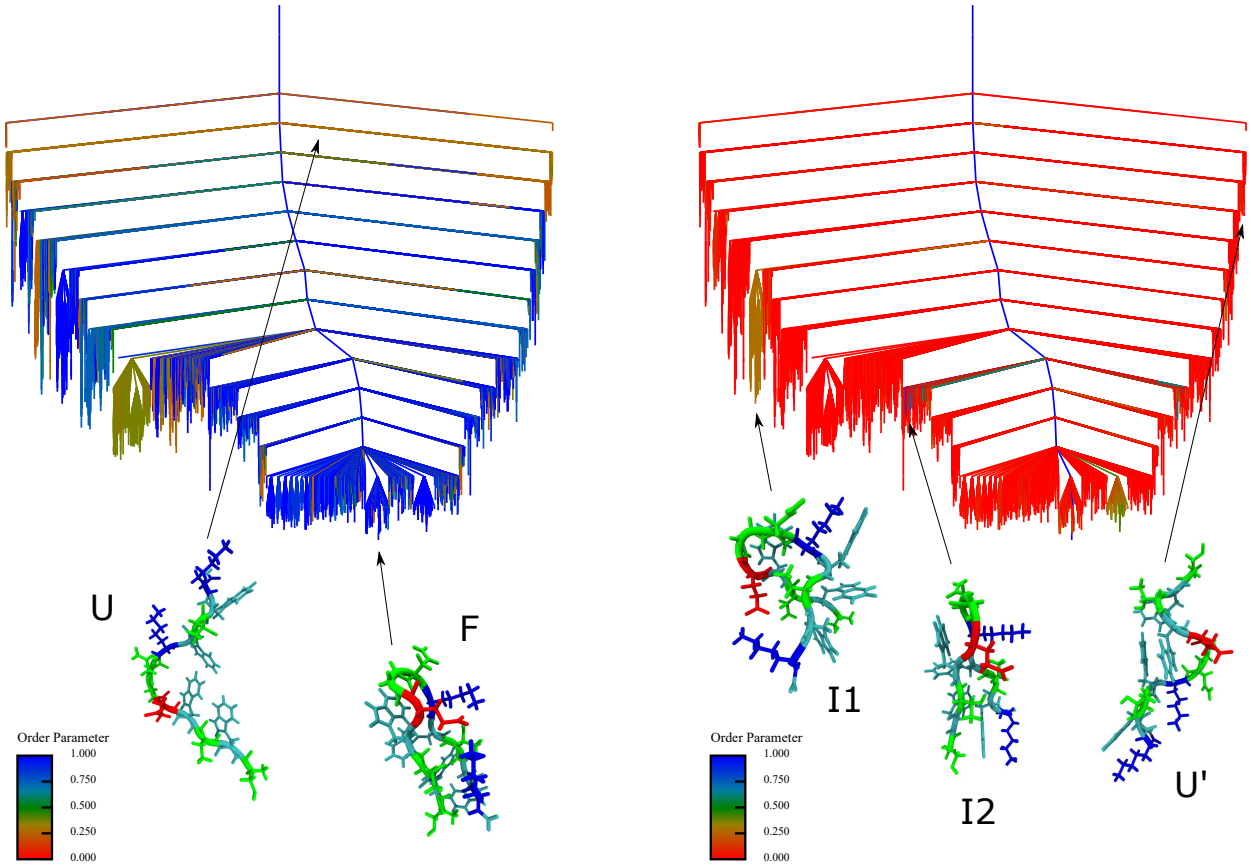
Figure 4.5: Histogram of the FPT distribution for the F ← U transition at (a) $T = 300\,\mathrm{K}$ and (b) $T = 330\,\mathrm{K}$, obtained from 20000 transition paths simulated using the kPS algorithm with *a priori* communities determined by MLR-MCL.

not correspond to either of these dynamical regimes. Instead, it is possible to identify clusters of nodes with similar intermediate values for the committor probability, and the nodes comprising the TSE are not localized in the state space. Furthermore, and perhaps counterintuitively, the committor probabilities do not correlate strongly with the potential energy. There are some nodes associated with potential energy values similar to that of the native state, but have committor probabilities $q_j^+ < 0.5$, and, conversely, there are many high-energy nodes with committor probabilities $q_j^+ \approx 1$.

4.4 Discussion

4.4.1 Features of the methodology

We have discussed two accelerated kinetic Monte Carlo (kMC) algorithms, weighted ensemble[93,132–137] kMC (WE-kMC) and kinetic path sampling[147,148] (kPS), which sample trajectories in exact accordance with the linear master equation (Eq. 4.1) that governs the Markovian dynamics on arbitrary kinetic networks. In particular, we have considered the problem of sampling the nonequilibrium $\mathcal{A} \leftarrow \mathcal{B}$ transition path ensemble (TPE) between two endpoint macrostates of interest, $\mathcal{A}$ and $\mathcal{B}$. The choice of enhanced sampling kMC methods employed herein is motivated by their desirable complementary features. WE-kMC is highly parallelizable and can be adapted to sample the equilibrium TPE, while the time complexity of kPS is essentially independent of the metastability of the kinetic network. Both methods overcome the 'flickering' problem that precludes the application of standard kMC to metastable kinetic networks.[77–80] They also both require a division of the state space, and their efficiency



(a) (b)

Figure 4.6: Disconnectivity graph[191,192] with leaves coloured according to (a) $\mathcal{A} \leftarrow \mathcal{B}$ committor probabilities q_j^+ and (b) $\mathcal{A} \leftarrow \mathcal{B}$ visitation probabilities r_j^+ for nodes j of the kinetic network, for the F $\leftarrow$ U transition at a temperature of 330 K. Only nodes of the network for which $r_j^+ \geq 0.01$ are included in the tree. The data were obtained from 20000 transition paths simulated using the kPS algorithm with *a priori* communities determined by MLR-MCL. The vertical axis corresponds to potential energy, with an incremental value of 3 kcal mol^{-1} for the superbasin analysis.[191]

is affected by the extent to which this partitioning faithfully characterizes the metastable macrostates. The choice of the disjoint sets is therefore a crucial consideration, and indeed is often the most challenging aspect in the implementation of enhanced sampling methods.[93,137] Here, we address this problem by employing a fast and numerically stable stochastic community detection algorithm, namely multi-level regularized Markov clustering (MLR-MCL),[150–155] to identify metastable sets of nodes on the kinetic network (Sec. 4.A). The MLR-MCL algorithm is unsupervised, and therefore our simulation strategy is fully automated. An initial partitioning can also be refined by an unsupervised variational optimization scheme (Sec. 4.B). Construction of a kinetic network by DPS (Sec. 4.2.1) does not require a low-dimensional projection of the underlying energy landscape, which might obscure features that have a dominant effect on the dynamics,[17–20] and so sampling paths on such networks is a powerful method to understand transition mechanisms in complex and high-dimensional systems.

The $\mathcal{O}(|\mathbb{E}|^3)$ time complexity for a single kPS iteration to simulate an escape trajectory from a trapping basin $\mathbb{B}$, where $|\mathbb{E}|$ nodes of the basin are eliminated in the graph transformation stage of the algorithm, leads to problems when a metastable macrostate is naturally large. This issue may be alleviated by allowing for a number of retained transient nodes, so that $\mathbb{T} \neq \emptyset$, thereby keeping the number of eliminated nodes $|\mathbb{E}|$ manageable, and effectively transforming the kPS iteration into a hybrid BKL-kPS scheme.[147] However, depending on the choice of nodes belonging to the set $\mathbb{T}$, this approach may reintroduce the problem of flickering trajectories, requiring the explicit simulation of a large number of standard rejection-free kMC steps. Alternatively, the network can be preprocessed by one of a number of methods, although we then forfeit exact sampling of the original kinetic network. Use of a recursive regrouping scheme[16] to subsume nodes interconnected by fast transition rates, according to a specified threshold, can be highly effective in removing the effects of the groups of nodes that are primarily responsible for the flickering.[189] Preprocessing of the network by graph transformation,[56–61] which preserves the path probabilities in their reduced representation, and which introduces renormalized waiting times for nodes to preserve the MFPT from any given node to a set of absorbing nodes, can also be used to avoid any one trapping basin becoming too large. This idea has been used to limit the size of trapping basins when using the FPTA method[70,95–99] to solve the master equation for an absorbing Markov chain in Ref. 78, and is an interesting possible direction for further work.

4.4.2 Comparison to alternative enhanced sampling methods

There are several other exact enhanced sampling methods that are closely related to WE-kMC, in the sense that they employ a division of the state space to simulate complete trajectories between two endpoint macrostates of interest in a piecewize fashion, and maintain a set of 'walkers' on the state space that are simulated in parallel.[137,193] We expect them to perform similarly, but there are some factors that may make a particular enhanced sampling method more favourable for a given system. We conclude this chapter by giving a brief overview of popular exact enhanced sampling methods alternative to those implemented in the present chapter. We highlight relative advantages and disadvantages of the methods, and draw attention to how the methods could be adapted and optimized for the problem of sampling the TPE on arbitrary Markovian kinetic networks, as opposed to the more common problem of performing simulations on a continuous state space. In particular, we suggest how some of these methods may be coupled with the kPS[147,148] (or, in the same way, the MCAMC[95,96]) algorithm.

Milestoning[31,38,92,194] utilizes a partitioning of the state space into disjoint sets, each characterized by an 'anchor' node. Short trajectories are initialized at so-called milestones, which are hypersurfaces at the interfaces between the states, with probabilities reflecting the equilibrium distribution. The flux across a milestone is measured from the first passage time distributions of incident trajectories initialized from neighbouring milestones. The MFPTs between all pairs of milestones can be computed from this information. In a kinetic network, the analogue of a hypersurface is a set of boundary nodes separating a pair of communities. The communities could be determined by any appropriate community detection algorithm, such as MLR-MCL.[168] Since the complete trajectory time distribution for inter-milestone transitions is an ingredient in the estimation of the coarse-grained MFPT matrix, milestoning provides a natural method for the estimation of reduced Markov chains.[194] Furthermore, milestoning does not depend on a separation of timescales, and therefore the choice of partitioning of the state space is less crucial, and the method remains effective when applied to highly diffusive dynamical processes.[193]

Nonequilibrium umbrella sampling[90,195–197] (NEUS) and the related tilting algorithm[94] likewise employ a partitioning of the state space into arbitrary nonoverlapping sets to achieve distributed sampling, and aim to calculate the flux across interfaces. Again, for discrete-state systems, this partitioning can be obtained using MLR-MCL, as in the present work. In these methods, each state is assigned a number of walkers and an initial weight. Every time a walker reaches an interface, an incremental amount of weight is transferred from the state to the neighbouring state associated with the interface. The walker is then moved to

a new interface of the original state, with probability in proportion to the associated flux. Eventually the bin weights converge to steady state values, and the $\mathcal{A} \leftarrow \mathcal{B}$ flux can be inferred from the number of trajectories crossing individual interfaces per unit time.[197] The tilting algorithm[94] variant of NEUS allows for more rapid convergence to the steady state. Despite its relatively high computational cost,[197] the flexibility in defining the states, and in the distribution of computational resources via specification of the numbers of walkers for each state, makes NEUS a powerful method for sampling TPEs at a steady state.

If it is more natural to divide the state space by nonintersecting interfaces, which for kinetic networks could be achieved automatically by the repeated application of minimum-cut algorithms,[64] then it may be preferable to employ forward-flux sampling (FFS),[47,91,116,175–179] which can be thought of as a particular case of NEUS. In FFS, trajectory segments are simulated starting from a distribution at the current interface. The trajectory pieces either reach the next interface, in which case the incident points are stored for use in the initial distribution of trajectories starting from this succeeding interface, or return to the initial macrostate $\mathcal{B}$. Note that, unlike milestoning, FFS does not employ the equilibrium probability distribution at the interfaces, and therefore simulates nonequilibrium, instead of equilibrium, TPEs. Coordinates orthogonal to the reaction coordinate that defines the nesting of interfaces must effectively be sampled by brute force. Therefore FFS is most useful for simulating rare event systems that can be projected onto a single dimension without significant loss of information, in which case the comparably small computational overhead of FFS makes the method attractive.[197] The treatment of the successive trajectory pieces in a serial fashion leads to a propagation of errors,[91] and significant computational effort is expended simulating trajectories that do not reach the next interface but instead return to the initial macrostate, especially if there are intermediate states acting as strong kinetic traps.[116] Recent advances are focussed on addressing these issues.[198] Enhanced sampling methods based on the parallel simulation of multiple trajectory segments share many of the same shortcomings. In particular, the correlated histories of trajectories necessitate that rigorous statistical tests be employed to evaluate the quality of the simulation data.[137]

In milestoning and FFS, trajectory segments are simulated from one hypersurface (in discrete state space, a set of boundary nodes) to another, and unlike NEUS (or WE-kMC), there is not continual 'feedback' between adjacent hypersurfaces, although FFS must be carried out in a serial fashion. This feature means that milestoning and FFS are well-suited for use in conjunction with kPS (or MCAMC) using only modest computational resources, since the calculation can be ran by focussing on individual communities of nodes in turn. Therefore, the graph transformation stage of the kPS simulation,[147,148] and the spectral decomposition of the transition matrix in the MCAMC algorithm,[70,95,96,99] which are the

computational bottlenecks of the respective methods, need only be carried out once for each community. After storing the relevant information to undo the graph transformation in kPS, the iterative reverse randomization procedure[147,148] can be repeated to generate the desired number of sample trajectory segments within the community. Similarly, the eigenspectrum of a community can be used to repeatedly generate sample trajectory segments within a MCAMC simulation. Since kPS and the FPTA[70,99] variant of the MCAMC algorithm correctly preserve the FPT distribution, and sample nodes at the absorbing boundary of the currently occupied community with the exactly correct probabilities, the milestoning and FFS methods used in conjunction with kPS or FPTA will yield unbiased estimates for MFPTs. These hybrid methods will also yield an unbiased sample of the TPE, albeit with reduced resolution of the pathways, since information on the dynamics within communities is lost. This loss is fairly inconsequential, since the communities ought to reflect the metastable states within which the trajectories flicker unproductively.

4.5 Conclusions

The advanced kMC methods employed in this chapter, weighted ensemble[93,132–135,137] kMC (WE-kMC) (Sec. 4.2.4) and kinetic path sampling[147,148] (kPS) (Sec. 4.2.5), allow for a detailed quantitative analysis of the $\mathcal{A} \leftarrow \mathcal{B}$ path ensembles on arbitrary finite discrete- and continuous-time Markov chains, and remain efficient even for networks that are strongly metastable and of high dimensionality. We have demonstrated our simulation workflow, in which the MLR-MCL[150–155] (Sec. 4.A) unsupervised community detection algorithm is used to define *a priori* fixed bins in the accelerated kMC simulations, with optional refinement of the partitioning by a variational optimization procedure (Sec. 4.B), with results for a kinetic network representing the folding of the TZ1 peptide[157] (Sec. 4.3.2) constructed by discrete path sampling (Sec. 4.2.1).[14,15] The folding transition for TZ1 exhibits complex multi-state and multi-pathway kinetics (Sec. 4.3.3), and simulation of the folding transition by brute-force kMC is unfeasible. The choice of partitioning of the state space is a crucial consideration that strongly affects the efficiency of enhanced sampling algorithms,[17–19,93,137] and any appropriate community detection[168] algorithm could be used to obtain this partitioning in a kinetic network. Our proposed variational optimization scheme (Sec. 4.B) to refine the initially determined communities of nodes ensures that the metastable macrostates are accurately characterized, and therefore makes our automated simulation workflow robust. Future work could discuss the choice of community detection algorithm for the purpose of determining an initial approximation to the metastable sets of nodes, which can subsequently be used to guide accelerated kMC simulations, as in the present work, or to determine a

reduced Markov chain.[199]

Another avenue for further investigation is to implement and benchmark alternative enhanced sampling methods, such as those described in Sec. 4.4.2, for accelerating kMC simulations on arbitrary kinetic networks. In particular, it is desirable to have access to methods that are well-suited to sampling the equilibrium TPE, a problem to which the protocol for establishing a steady state in WE sampling[134] does not provide an ideal solution. Hybrid algorithms combining kPS or FPTA[70,95,96,99] with milestoning[38,92,194] and with forward flux sampling (FFS)[47,91,116,175–179] are particularly promising approaches for efficient and exact sampling of equilibrium and nonequilibrium TPEs, respectively. We have developed the DISCOTRESS (DIscrete State COntinuous Time Rare Event Simulation Suite) software to perform enhanced sampling simulations on arbitrary Markovian networks. Application of these advanced kMC methods to a variety of systems will yield insight into how features of the TPE, such as the existence of multiple competing mechanisms, arise from the topology of the kinetic network. The identification of archetypal classes[200,201] of stochastic dynamics and corresponding network topologies will provide fundamental understanding concerning how dynamical observables such as the MFPT arise from microscopic features of the TPE, and therefore of how these macroscopic dynamical properties are encoded in the underlying energy landscape.

4.A Multi-level regularized Markov clustering

The basic Markov clustering (MCL) algorithm[153–155] is a deterministic method for the unsupervised detection of community structure in weighted and directed networks, by performing operations on a stochastic matrix to artificially simulate the properties of an average random walk on the network. The main loop of the MCL algorithm iterates the following three operations until convergence is achieved. Firstly, the expansion operation, where the product of the transition matrix with itself is computed. The expansion operation effectively lengthens the timescale of the average random walk characterized by the updated transition matrix, and thus allows for probability flow between different regions of the network. Secondly, the inflation operation, where the Hadamard power of the transition matrix is computed, given a granularity parameter $r > 1$. The inflation operation effectively augments the probability current for transitions between regions of the network where the flow is strong, and diminishes the probability current where it is already weak. Lastly, elements of the matrix with values below a threshold are pruned, and the columns of the matrix are renormalized. With the repeated application of this sequence of operations, the transition matrix becomes increasingly sparse, since the probability distribution of flows becomes progressively more

localized. At convergence, each node appears in only a single column of the transition matrix. Thus the output of the MCL algorithm is a matrix that is doubly idempotent with respect to the expansion and inflation operations. The converged matrix can be interpreted as a clustering, where nodes corresponding to nonzero elements in particular rows of the final stochastic matrix belong to the same community. Each community is characterized by an attractor node in the output matrix, namely the nodes associated with rows containing one or more nonzero transition probabilities, with the interpretation that there is a net flow from all nodes of a given community to the corresponding attractor. The resolution of the community detection can be increased by increasing the value of the granularity parameter r. The capability to tune the resolution of the clustering, i.e. to adjust the timescales characterized by the community structure, by a direct choice of input parameter is a desirable feature for the present purpose.

For generic networks, the input matrix in the MCL procedure is usually obtained naïvely by simple renormalization of the columns of the network adjacency matrix (after adding self-loops), and possibly employing weight transformation heuristics to improve the quality of the output clustering.[150,151] In the present context, where we have a CTMC parameterized by a transition rate matrix, we require a stochastic matrix that properly represents the dynamics on the kinetic network. To ensure that the resulting stochastic matrix includes nonzero probabilities for self-loop transitions, we use the linearized transition probability matrix,[147] $\mathbf{T}_{\text{lin}}(\tau) = \mathbf{I} + \tau\mathbf{K}$, where $\mathbf{I}$ is the identity matrix, instead of the branching probability matrix. Provided that $\tau \leq \min\{-(K_{jj})^{-1} : \forall\, j \in \mathcal{S}\}$, then the linearized transition matrix is column-stochastic, as required (with uniform mean waiting times, τ_j, for all nodes j). The lag time provides another input parameter that governs the resolution of the community detection. In fact, it may be preferable to control the resolution of the clustering via the lag time τ rather than through the granularity parameter r, since the MCL algorithm is highly sensitive to r.[150,151]

There are a number of issues with the MCL algorithm in its simplest form. One problem is scalability: the expansion operation has time complexity $\mathcal{O}(|\mathcal{S}|^2)$ for dense matrices, and is therefore prohibitively expensive for large networks. A solution to this problem is to pre-process the network by using heavy edge matching (HEM) to iteratively determine coarsened networks that retain the topological features of the original network. In the HEM procedure, nodes are randomly selected in turn, and matched to a currently unmatched neighbour, if such a node exists, according to the shared edge of greatest weight. Each pair of matched nodes is contracted into a 'supernode', for which the set of edges is the union of the edges involving the corresponding nodes of the refined graph. The HEM procedure is repeated until the most coarse network comprises a number of nodes below a given threshold. Of course,

the number of iterations of the HEM procedure affects the resolution of the clustering, and so this threshold must be chosen carefully. Within this multi-level framework, a small number of iterations of MCL are run for a coarsened transition matrix, after which the flow is projected onto the next refined transition matrix, according to the mapping of nodes required to undo the HEM procedure.[150,151] Since the coarse transition matrices are of low dimensionality, the expansion operations in the early stages of the MLR-MCL procedure are fast. It is desirable that the coarse network retains the attractor nodes associated with the communities of the original network. Hence, in our implentation of the HEM procedure for Markov chains, which have bidirectional edges, the unmatched node corresponding to the shared edge of greatest *incoming* weight is mapped to each randomly selected node. In the final stage of the multi-level MCL algorithm, a large number of MCL iterations are performed on the projected transition matrix representing the complete network. Since the transition matrices become more sparse as the algorithm progresses, these later iterations can be achieved efficiently by exploiting a compressed sparse row data structure and parallelizble sparse matrix-matrix multiplication algorithms. Thus within the multi-level framework, the expansion operation should at no point become prohibitively slow.

A second issue concerns the quality of the output community structure. In particular, the standard MCL algorithm has a tendency to produce a large number of clusters. This overfitting effect arises since there is no penalty associated with divergence of columns of the transition matrix that correspond to neighbouring nodes. The proposed solution to this problem is to replace the expansion step with a regularization step, in which the product of the transition matrix with a regularisation matrix is computed.[150,151] In the simplest case, the regularization matrix is the initial transition matrix at each step of the multi-level refinement. It can be shown that this is the optimal choice of regularization matrix for smoothing the distribution of probability flow out of nodes,[150] thereby preventing overfitting. That is, columns of the transition matrix corresponding to neighbouring nodes tend to remain similar, and hence the corresponding nodes tend to be associated with the same attractor, provided that transitions between the nodes have high probability. The regularization effect can be tuned by constructing a regularization matrix at every iteration according to a specified balance parameter, b.[151] A balance parameter of $b = 1$ corresponds to the simplest form of regularization, described above. Multi-level regularised MCL (MLR-MCL), which incorporates the aforementioned modifications, scales more favourably than, and yields fewer and more balanced communities compared to, the basic MCL algorithm.[150,151]

4.B Variational optimization procedure to refine metastable macrostates

The methodology described herein relies on a partitioning of the nodes of a finite Markov chain into metastable macrostates. Consider the stochastic matrix $\mathbf{T}$, for which the state space $\mathcal{S}$ is partitioned into the set of macrostates $\mathcal{C} \equiv \{\mathcal{X}, \mathcal{Y}, \ldots\}$. In practice, if the community detection algorithm used to determine $\mathcal{C}$ was chosen appropriately, it is likely that the obtained clustering characterizes the metastable sets of nodes in the Markovian network with relatively high accuracy. However, there are liable to be misclassifications of nodes at the intercommunity boundaries. These errors may arise since, for instance, many community detection algorithms for generic networks are based on heuristics or objective functions that do not precisely correspond to the aim of identifying metastable macrostates in Markov chains. In addition, many state-of-the-art community detection algorithms, including MLR-MCL (Sec. 4.A), are stochastic.[168] Although only a small fraction of nodes may be misclassified, the states at the intercommunity boundaries typically have a dominant effect on the slow dynamics. Hence, any misclassifications are likely to have a profound effect on the resulting properties of a lumped (reduced) Markov chain obtained by the estimation of intercommunity transition probabilities or rates, and are also detrimental to the efficiency of advanced simulation algorithms such as WE-kMC (Sec. 4.2.4), kPS (Sec. 4.2.5) and MCAMC.[95]

We now propose a procedure to refine an initial partitioning of a Markov chain into metastable communities. Our strategy leverages the existence of a variational principle for the second dominant eigenvalue, λ_2, (or similarly, the average mixing time, ζ_{K}, introduced in Chapter 1) of the reduced transition matrix obtained from the local equilibrium approximation (LEA).[202–204] That is, for a given community structure $\mathcal{C}$ associated with the stochastic matrix $\mathbf{T}$, there is an upper bound on the second dominant eigenvalue $\lambda_2^{\mathcal{C}}$ of the lumped Markov chain $\mathbf{T}^{\mathcal{C}}$ with intercommunity transition probabilities $T^{\mathcal{C}}_{\mathcal{X}\mathcal{Y}} = \mathbf{1}_{\mathcal{X}}^{\top}\mathbf{T}_{\mathcal{X}\mathcal{Y}}\boldsymbol{\pi}_{\mathcal{Y}}\ \forall\, \mathcal{X}, \mathcal{Y} \in \mathcal{C}$; $\lambda_2^{\mathcal{C}} < \lambda_2$. Likewise, for an irreducible Markov chain, there is an upper bound on the average mixing time (the expected time to reach the stationary distribution, which is independent of the initial condition[199]) of the lumped Markov chain given by the LEA: $\zeta_{\mathrm{K}}^{\mathcal{C}} < \zeta_{\mathrm{K}}$.[205] Hence, if we employ a procedure to perturb the community structure $\mathcal{C}$ and thus obtain an updated reduced transition matrix $\mathbf{T}^{\mathcal{C}}$, then we can recompute either $\lambda_2^{\mathcal{C}}$ or $\zeta_{\mathrm{K}}^{\mathcal{C}}$ and use this quantity as a metric to test if the proposed perturbation improved the quality of the clustering. The second dominant eigenvalue corresponds to the timescale of the slowest relaxation process, and the average mixing time is effectively a sum of timescales for all relaxation processes.

Hence, these properties provide rigorous and interpretable objective functions to assess the extent to which a community structure $\mathcal{C}$ characterizes the metastable macrostates of a finite Markov chain. The lumped Markov chain $\mathbf{T}^{\mathcal{C}}$ given by the LEA can be recomputed trivially if the stationary distribution of $\mathbf{T}$ is known. Moreover, the second dominant eigenvalue (or the average mixing time) only needs to be calculated for the *reduced* Markov chain at each iteration. Since the number of macrostates is typically small, $|\mathcal{C}| \ll |\mathcal{S}|$, and the reduced transition matrix should be well-conditioned if $\mathcal{C}$ approximates the true metastable communities of nodes, the variational optimization procedure is efficient and stable.

There are many ways that this perturbation framework could be implemented in practice. In the simplest possible refinement scheme, the assigned community is switched for only a single boundary node at each iteration, and a queue of intercommunity edges is maintained, with connections corresponding to fast transitions prioritized for perturbation. The proposed switching moves are accepted greedily; that is, perturbations leading to an increase (decrease) in $\lambda_2^{\mathcal{C}}$ or $\zeta_{\mathrm{K}}^{\mathcal{C}}$ are always accepted (rejected), respectively. This basic version of the procedure, which may be sufficient to determine the optimal partitioning when the initial community structure $\mathcal{C}$ is a close approximation to the true metastable sets of nodes, is illustrated in Fig. 4.7. More sophisticated implementations of this procedure may use stochastic criteria to select intercommunity edges and associated nodes for reassignment, and apply a Metropolis condition to accept proposed perturbations to the community structure. In practice, we have found empirically that simulated annealing provides an effective means to refine the intercommunity boundaries. Simulated annealing uses a Metropolis acceptance criterion with an artificial temperature that decreases throughout the simulation according to a specified protocol, so that the probability of accepting moves that decrease the objective function becomes smaller as the simulation progresses, and the heuristic eventually reduces to a greedy heuristic. A further possibility is to allow for the reassignment of multiple nodes simultaneously, which may be especially useful for accelerating the convergence of a global optimization algorithm when the initial clustering $\mathcal{C}$ is a poor representation of the metastable communities. These advanced moves could be achieved using a breadth first search procedure initialized from a chosen boundary node, incorporating only nodes associated with an intercommunity transition probability or rate that exceeds a specified threshold.

4.C Simulation parameters

The following simulation parameters were used to obtain the results for the kinetic network representing the folding of the tryptophan zipper peptide described in this chapter. The

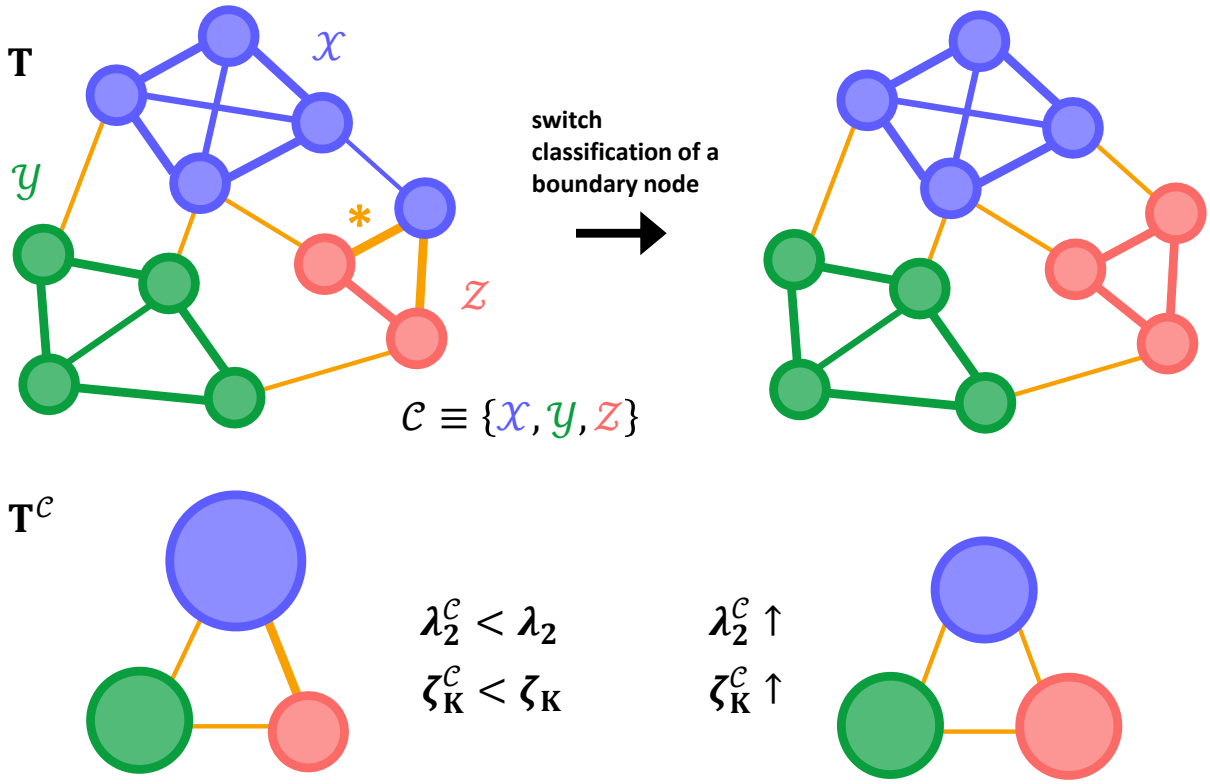


Figure 4.7: Schematic illustration of a single iteration of the proposed variational optimization procedure to refine a community structure $\mathcal{C} \equiv \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ associated with a stochastic matrix $\mathbf{T}$. The edges indicate bidirectional transitions, with the intercommunity transitions highlighted in yellow, and transition probabilities (or rates) are correlated with the edge thickness. The initial clustering clearly does not precisely characterize the metastable sets of nodes: there is a node of the set $\mathcal{X}$ for which there is a slow $\mathcal{X} \leftarrow \mathcal{X}$ transition and fast $\mathcal{Z} \leftrightarrow \mathcal{X}$ transitions. Hence, the average mixing time $\zeta_{\mathrm{K}}^{\mathcal{C}}$ for the lumped Markov chain estimated by the local equilibrium approximation (LEA), with transition probabilities $T_{\mathcal{X}\mathcal{Y}}^{\mathcal{C}} = \mathbf{1}_{\mathcal{X}}^{\top} \mathbf{T}_{\mathcal{X}\mathcal{Y}} \boldsymbol{\pi}_{\mathcal{Y}}$ $\forall\, \mathcal{X}, \mathcal{Y} \in \mathcal{C}$, is erroneously fast. The timescale of the slowest relaxation process for this reduced chain, which directly relates to the second dominant eigenvalue $\lambda_2^{\mathcal{C}}$, is likewise spuriously fast. In the first step of an iteration of the variational refinement procedure, an intercommunity edge (indicated by $*$) is selected for perturbation. This selection can be made either stochastically or deterministically, with prioritization based on the values of the transition probabilities or rates. Next, one of the two nodes associated with the chosen edge is selected. Again, this selection can be informed by relevant criteria - here, the blue node, of the set $\mathcal{X}$, that is associated with the chosen intercommunity edge also features a connection corresponding to a slow intracommunity transition, which suggests that this node is misclassified. The community assigned to this node is switched to that of a neighbouring node (here, $\mathcal{Z}$), and an updated reduced matrix is computed from the LEA. The new lumped Markov chain $\mathbf{T}^{\mathcal{C}}$ has increased timescales for the slowest relaxation process and for the average mixing time. That is, the lumped Markov chain better represents the slow dynamics of the original system, and the proposed perturbation to the community structure $\mathcal{C}$ is thus accepted.

simulations were performed using the DISCOTRESS software (available under the GNU General Public License at github.com/danieljsharpe/DISCOTRESS).

4.C.1 Determination of communities

To obtain fixed *a priori* communities of the TZ1 kinetic network in the kPS and WE-kMC simulations, the multi-level regularized Markov clustering (MLR-MCL) algorithm[150,151] (Sec. 4.A) was used with granularity parameter $r = 1.05$, balance parameter $b = 1$, pruning threshold $\varepsilon = 10^{-6}$, maximum number of residual nodes following the heavy edge matching procedure equal to $n_{\mathrm{HEM}} = 800$, number of curtailed MCL iterations at each stage of the multi-level procedure $N_{\mathrm{cur}} = 3$, and an initial transition probability matrix estimated at a lag time 10^{-16} s. There were 390 resulting communities, the smallest of which was the manually chosen absorbing state described below, comprised of 17 nodes. The largest community comprised 5762 nodes. The majority of communities contained around 100 nodes.

To obtain communities on-the-fly, a breadth-first search procedure was used, where neighbouring nodes with an associated transition rate corresponding to an energy barrier height $< 4\,\mathrm{kcal\,mol^{-1}}$ are incorporated into the current community, with a maximum size of 3000 nodes.

4.C.2 Definition of endpoint states

The three key metastable macrostates described in the main text, namely the U, I1, and I2+F states, were identified with the MLR-MCL algorithm at a low resolution, using input parameters $r = 1.025$, $b = 1$, $\varepsilon = 10^{-6}$, $n_{\mathrm{HEM}} = 2000$, $N_{\mathrm{cur}} = 3$, and $\tau = 10^{-13}$ s. The macrostate representing the native fold, denoted F in the main text, comprised 17 nodes chosen manually, including the global potential energy minimum, and the nodes connected to this native node by transition rates corresponding to energy barrier heights $< 1\,\mathrm{kcal\,mol^{-1}}$. In both the WE-kMC and kPS simulations, the initial probability density was localized at a particular high-energy node of the unfolded macrostate, corresponding to an extended conformation with no native or non-native contacts. The set of folded nodes F was treated as an absorbing macrostate in the nonequilibrium stochastic dynamics simulations.

4.C.3 Kinetic path sampling simulation parameters

The kinetic path sampling[147,148] (kPS) simulations used the branching probability matrix as the initial transition probability matrix in the graph transformation[56–61] stage of the algorithm. All nodes of the current trapping basin were always eliminated ($\mathbb{B} \equiv \mathbb{E}$ and $\mathbb{T} \equiv \emptyset$ for all kPS iterations). 50000 rejection-free kMC[82,83] moves were conducted following

each kPS iteration, to avoid trivial recrossings at the boundaries between communities. 20000 independent first passage paths were simulated.

4.C.4 Weighted ensemble kMC simulation parameters

In the weighted ensemble[48,93] kMC (WE-kMC) simulations, the target numbers of walkers for each community were chosen to be uniform, equal to $M_\xi = 100$. The trajectory resampling procedure was conducted at time intervals of $\tau_{\mathrm{R}} = 10^6\,\mathrm{s}$. The 20000 simulated first passage paths were obtained from 40 independent WE runs.

Bibliography

[1] J. R. Norris. *Markov Chains*. Cambridge University Press, New York, USA, 1997.

[2] J. Goutsias and G. Jenkinson. *Phys. Rep.*, 529:199–264, 2013.

[3] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, Amsterdam, Netherlands, 1992.

[4] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists*. Academic Press, New York, USA, 1992.

[5] R. Zwanzig. *J. Stat. Phys.*, 30:255–262, 1983.

[6] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, San Diego, California, second edition, 2001.

[7] C. Chipot and A. Pohorille. *Free Energy Calculations*. Springer-Verlag, Berlin, Germany, 2007.

[8] J. D. Chodera and F. Noé. *Curr. Op. Struct. Biol.*, 25:135–144, 2014.

[9] J. A. Joseph, K. Röder, D. Chakraborty, R. G. Mantell, and D. J. Wales. *Chem. Commun.*, 53:6974–6988, 2017.

[10] K. Röder, J. A. Joseph, B. E. Husic, and D. J. Wales. *Adv. Theory Simul.*, 2:1800175, 2019.

[11] D. J. Wales. *Energy Landscapes*. Cambridge University Press, Cambridge, UK, 2003.

[12] F. Noé and J. C. Smith. In A. Deutsch, L. Brusch, J. Byrne, G. de Vries, and H.-P. Herzel, editors, *Mathematical Modeling of Biological Systems, Volume I.*, pages 125–144. Birkhäuser, Boston, 2007.

[13] F. Noé and S. Fischer. *Curr. Op. Struct. Biol.*, 18:154–162, 2008.

[14] D. J. Wales. *Mol. Phys.*, 100:3285–3305, 2002.

[15] D. J. Wales. *Mol. Phys.*, 102:891–908, 2004.

[16] D. J. Wales and P. Salamon. *Proc. Natl. Acad. Sci. USA*, 111:617–622, 2014.

[17] A. Mardt, L. Pasquali, J. Wu, and F. Noé. *Nat. Commun.*, 9:5, 2018.

[18] A. Ma and A. R. Dinner. *J. Phys. Chem. B*, 109:6769–6779, 2005.

[19] C. Dellago, P. G. Bolhuis, and P. L. Geissler. *Adv. Chem. Phys.*, 123:1–78, 2002.

[20] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. *J. Chem. Phys.*, 134:174105, 2011.

[21] N. M. Amato, K. A. Dill, and G. Song. *J. Comput. Biol.*, 10:239–255, 2003.

[22] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma. *J. Comput. Biol.*, 10:257–281, 2003.

[23] S. V. Krivov and M. Karplus. *J. Phys. Chem. B*, 110:12689–12698, 2006.

[24] W. C. Swope, J. W. Pitera, and F. Suits. *J. Phys. Chem. B*, 108:6571–6581, 2004.

[25] N. Singhal, C. D. Snow, and V. S. Pande. *J. Chem. Phys.*, 121:415–425, 2004.

[26] F. Noé, I. Horenko, C. Schütte, and J. C. Smith. *J. Chem. Phys.*, 126:155102, 2007.

[27] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. *Proc. Natl. Acad. Sci. USA*, 106:19011–19016, 2009.

[28] J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, and W. C. Swope. *J. Chem. Phys.*, 126:155101, 2007.

[29] N.-V. Buchete and G. Hummer. *J. Phys. Chem. B*, 112:6057–6069, 2008.

[30] G. R. Bowman, V. S. Pande, and F. Noé (Eds.). *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Springer, Netherlands, first edition, 2014.

[31] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden. *J. Chem. Phys.*, 134:204105, 2011.

[32] W. Zheng, M. Andrec, E. Gallicchio, and R. M. Levy. *J. Phys. Chem. B*, 113:11702–11709, 2009.

[33] F. Rao and A. Caflisch. *J. Mol. Biol.*, 342:299–306, 2004.

[34] F. Rao and M. Karplus. *Proc. Natl. Acad. Sci. USA*, 107:9152–9157, 2010.

[35] L. Gong and X. Zhou. *J. Phys. Chem. B*, 114:10266–10276, 2010.

[36] B. Fačkovec, E. Vanden-Eijnden, and D. J. Wales. *J. Chem. Phys.*, 143:044119, 2015.

[37] G. C. Boulougouris and D. N. Theodorou. *J. Chem. Phys.*, 130:044905, 2009.

[38] S. Viswanath, S. M. Kreuzer, A. E. Cardenas, and R. Elber. *J. Chem. Phys.*, 139:174105, 2013.

[39] P. D. Dixit, A. Jain, G. Stock, and K. A. Dill. *J. Chem. Theory Comput.*, 11:5464–5472, 2015.

[40] P. D. Dixit, J. Wagoner, C. Weistuch, S. Pressé, K. Ghosh, and K. A. Dill. *J. Chem. Phys.*, 148:010901, 2018.

[41] P. D. Dixit and K. A. Dill. *J. Chem. Phys.*, 150:054105, 2019.

[42] D. Helbing. *Quantitative Sociodynamics*. Springer-Verlag, Berlin, second edition, 2010.

[43] T. Székely Jr. and K. Burrage. *Comput. Struct. Biotechnol. J.*, 12:14–25, 2014.

[44] D. Schnoerr, G. Sanguinetti, and R. Grima. *J. Phys. A: Math. Theor.*, 50:093001, 2017.

[45] D. J. Warne, R. E. Baker, and M. J. Simpson. *J. R. Soc. Interface*, 16:20180943, 2019.

[46] G. Simoni, F. Reali, C. Priami, and L. Marchetti. *Wiley Interdiscip. Rev. Syst. Biol. Med.*, 11:e1459, 2019.

[47] R. J. Allen, P. B. Warren, and P. R. ten Wolde. *Phys. Rev. Lett.*, 94:018104, 2005.

[48] R. M. Donovan, A. J. Sedgewick, J. R. Faeder, and D. M. Zuckerman. *J. Chem. Phys.*, 139:115105, 2013.

[49] R. M. Donovan, J. J. Tapia, D. P. Sullivan, J. R. Faeder, R. F. Murphy, M. Dittrich, and D. M. Zuckerman. *PLoS Comput. Biol.*, 12:e1004611, 2016.

[50] B. K. Chu, M. J. Tse, R. R. Sato, and E. L. Read. *BMC Syst. Biol.*, 11:14, 2017.

[51] M. J. Tse, B. K. Chu, C. P. Gallivan, and E. L. Read. *PLoS Comput. Biol.*, 14:e1006336, 2018.

[52] L. J. S. Allen. In F. Brauer, P. van den Driessche, and J. Wu, editors, *Mathematical Epidemiology*, pages 81–130. Springer-Verlag, Berlin, 2008.

[53] L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology*. Prentice Hall, Upper Saddle River, New Jersey, 2003.

[54] B. Munsky and M. Khammash. *J. Chem. Phys.*, 124:044104, 2006.

[55] K. N. Dinh and R. B. Sidje. *Phys. Biol.*, 13:035003, 2016.

[56] S. A. Trygubenko and D. J. Wales. *Mol. Phys.*, 104:1497–1507, 2006.

[57] S. A. Trygubenko and D. J. Wales. *J. Chem. Phys.*, 124:234110, 2006.

[58] D. J. Wales. *Int. Rev. Phys. Chem.*, 25:237–282, 2006.

[59] D. J. Wales. *J. Chem. Phys.*, 130:204111, 2009.

[60] J. D. Stevenson and D. J. Wales. *J. Chem. Phys.*, 141:041104, 2014.

[61] R. S. MacKay and J. D. Robinson. *Phil. Trans. Roy. Soc. A*, 376:20170232, 2018.

[62] D. A. Evans and D. J. Wales. *J. Chem. Phys.*, 121:1080–1090, 2004.

[63] J. M. Carr and D. J. Wales. In A. Solov'yov and J.-P. Connerade, editors, *Latest Advances in Atomic Cluster Collisions: Structure and Dynamics from the Nuclear to the Biological Scale*, pages 321–330. Imperial College Press, London, 2008.

[64] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 151:124101, 2019.

[65] T. J. Frankcombe and S. C. Smith. *Theor. Chem. Acc.*, 124:303–317, 2009.

[66] B. E. Husic and V. S. Pande. *J. Am. Chem. Soc.*, 140:2386–2896, 2018.

[67] D. T. Gillespie. *J. Chem. Phys.*, 113:297–306, 2000.

[68] D. T. Gillespie, A. Hellander, and L. R. Petzold. *J. Chem. Phys.*, 138:170901, 2013.

[69] D. Schultz, A. M. Walczak, J. N. Onuchic, and P. G. Wolynes. *Proc. Natl. Acad. Sci. USA*, 105:19165–19170, 2008.

[70] K. A. Fichthorn and Y. Lin. *J. Chem. Phys.*, 138:164104, 2013.

[71] K. A. Fichthorn and W. H. Weinberg. *J. Chem. Phys.*, 95:1090–1096, 1991.

[72] A. F. Voter. In K. E. Sickafus and E. A. Kotomin, editors, *Radiation Effects in Solids*, pages 1–23. Springer, Dordrecht, Netherlands, 2005.

[73] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, Cambridge, UK, third edition, 2009.

[74] A. P. J. Jansen. *An Introduction to Kinetic Monte Carlo Simulations of Surface Reactions*. Springer Berlin, Heidelberg, Germany, 2012.

[75] H. M. Cuppen, L. J. Karssemeijer, and T. Lamberts. *Chem. Rev.*, 113:8840–8871, 2013.

[76] M. Andersen, C. Panosetti, and K. Reuter. *Front. Chem.*, 7:00202, 2019.

[77] D. R. Mason, R. E. Rudd, and A. P. Sutton. *Comp. Phys. Comm.*, 160:140–157, 2004.

[78] V. V. Bulatov, T. Oppelstrup, and M. Athènes. Technical report, Lawrence Livermore National Laboratory, 2011.

[79] L. Xu and G. Henkelman. *J. Chem. Phys.*, 129:114104, 2008.

[80] F. El-Mellouhi, N. Mousseau, and L. J. Lewis. *Phys. Rev. B*, 78:153202, 2008.

[81] J. D. Muñoz, M. A. Novotny, and S. J. Mitchell. *Phys. Rev. E*, 67:026101, 2003.

[82] S. A. Serebrinsky. *Phys. Rev. E*, 83:037701, 2011.

[83] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. *J. Comp. Phys.*, 17:10–18, 1975.

[84] D. T. Gillespie. *J. Comp. Phys.*, 22:403–434, 1976.

[85] D. T. Gillespie. *J. Phys. Chem.*, 81:2340–2361, 1977.

[86] D. T. Gillespie. *Annu. Rev. Phys. Chem.*, 58:35–55, 2007.

[87] J. T. Berryman and T. Schilling. *J. Chem. Phys.*, 133:244101, 2010.

[88] D. M. Zuckerman and T. B. Woolf. *J. Chem. Phys.*, 111:9475–9484, 1999.

[89] P. B. Warren and R. J. Allen. *Mol. Phys.*, 116:3104–3113, 2018.

[90] A. Warmflash, P. Bhimalapuram, and A. R. Dinner. *J. Chem. Phys.*, 127:154112, 2007.

[91] R. J. Allen, C. Valerani, and P. R. ten Wolde. *J. Phys.: Condens. Matter*, 21:463102, 2009.

[92] A. K. Faradjian and R. Elber. *J. Chem. Phys.*, 120:10880–10889, 2004.

[93] D. M. Zuckerman and L. T. Chong. *Annu. Rev. Biophys.*, 46:43–57, 2017.

[94] E. Vanden-Eijnden and M. Venturoli. *J. Chem. Phys.*, 131:044120, 2009.

[95] M. A. Novotny. *Phys. Rev. Lett.*, 74:1–5, 1995.

[96] M. A. Novotny. *Comput. Phys. Commun.*, 147:659–664, 2002.

[97] M. A. Novotny. In D. Stauffer, editor, *Annual Reviews of Computational Physics: Vol. 9*, pages 153–210. World Scientific, Singapore, 2001.

[98] C. S. Deo and D. J. Srolovitz. *Modell. Simul. Mater. Sci. Eng.*, 10:581–596, 2002.

[99] B. Puchala, M. L. Falk, and K. Garikipati. *J. Chem. Phys.*, 132:134104, 2010.

[100] M. Athènes, P. Bellon, and G. Martin. *Phil. Mag. A*, 76:565–585, 1997.

[101] G. C. Boulougouris and D. Frenkel. *J. Chem. Theory Comput.*, 1:389–393, 2005.

[102] G. C. Boulougouris and D. N. Theodorou. *J. Chem. Phys.*, 127:084903, 2007.

[103] A. Chatterjee and A. F. Voter. *J. Chem. Phys.*, 132:194101, 2010.

[104] W. Cai, M. H. Kalos, M. de Koning, and V. V. Bulatov. *Phys. Rev. E*, 66:046703, 2002.

[105] M. de Koning, W. Cai, B. Sadigh, T. Oppelstrup, M. H. Kalos, and V. V. Bulatov. *J. Chem. Phys.*, 122:074103, 2005.

[106] D. T. Gillespie. *J. Chem. Phys.*, 115:1716–1733, 2001.

[107] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis. *J. Chem. Phys.*, 122:024112, 2005.

[108] D. F. Anderson and D. J. Higham. *Multiscale Model. Simul.*, 10:146–179, 2012.

[109] C. Lester, C. A. Yates, M. B. Giles, and R. E. Baker. *J. Chem. Phys.*, 142:024113, 2015.

[110] M. B. Giles. In J. Dick, F. Y. Kuo, G. W. Peters, and I. H. Sloan, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 83–103. Springer Berlin, Heidelberg, Germany, 2013.

[111] D. F. Anderson, D. J. Higham, and Y. Sun. *SIAM J. Numer. Anal.*, 52:3106–3127, 2014.

[112] V. Wolf, R. Goel, M. Mateescu, and T. A. Henzinger. *BMC Syst. Biol.*, 4:42, 2010.

[113] C. H. L. Beentjes and R. E. Baker. *J. Chem. Phys.*, 150:154107, 2019.

[114] A. Milias-Argeitis and J. Lygeros. *J. Chem. Phys.*, 138:184109, 2013.

[115] D. Frenkel. *Proc. Natl. Acad. Sci. USA*, 101:17571–17575, 2004.

[116] R. J. Allen, D. Frenkel, and P. R. ten Wolde. *J. Chem. Phys.*, 124:024102, 2006.

[117] M. Athènes. Technical report, Université Paris Saclay; Université Paris Sud, 2018.

[118] G. Korniss, M. A. Novotny, and P. A. Rikvold. *J. Comput. Phys.*, 153:488–508, 1999.

[119] A. Chatterjee and D. G. Vlachos. *J. Chem. Phys.*, 124:064110, 2006.

[120] R. E. Rudd, D. R. Mason, and A. P. Sutton. *Prog. Mater. Sci.*, 52:319–332, 2007.

[121] A. Slepoy, A. P. Thompson, and S. J. Plimpton. *J. Chem. Phys.*, 128:205101, 2008.

[122] P. Terrier, M. Athènes, T. Jourdan, G. Adjanor, and G. Stoltz. *J. Comput. Phys.*, 350:280–295, 2017.

[123] D. Perez, B. P. Uberuaga, and A. F. Voter. *Comput. Mater. Sci.*, 100:90–103, 2015.

[124] H. Resat, H. S. Wiley, and D. A. Dixon. *J. Phys. Chem. B*, 105:11026–11034, 2001.

[125] M. A. Snyder, A. Chatterjee, and D. G. Vlachos. *Comput. Chem. Eng.*, 29:701–712, 2005.

[126] J. Goutsias. *J. Chem. Phys.*, 122:184102, 2005.

[127] A. Samant and D. G. Vlachos. *J. Chem. Phys.*, 123:144114, 2005.

[128] W. E, D. Liu, and E. Vanden-Eijnden. *J. Chem. Phys.*, 123:194107, 2005.

[129] W. E, D. Liu, and E. Vanden-Eijnden. *J. Comput. Phys.*, 221:158–180, 2007.

[130] A. La Magna and S. Coffa. *Comput. Mater. Sci.*, 17:21–33, 2000.

[131] C. D. Van Siclen. *J. Phys.: Condens. Matter*, 19:072201, 2007.

[132] G. A. Huber and S. Kim. *Biophys. J.*, 70:97–110, 1996.

[133] B. W. Zhang, D. Jasnow, and D. M. Zuckerman. *J. Chem. Phys.*, 132:054107, 2010.

[134] D. Bhatt, B. W. Zhang, and D. M. Zuckerman. *J. Chem. Phys.*, 133:014110, 2010.

[135] E. Suárez, S. Lettieri, M. C. Zwier, C. A. Stringer, S. R. Subramanian, L. T. Chong, and D. M. Zuckerman. *J. Chem. Theory Comput.*, 10:2658–2667, 2014.

[136] H. Feng, R. Costaouec, E. Darve, and J. A. Izaguirre. *J. Chem. Phys.*, 142:214113, 2015.

[137] L. T. Chong, A. S. Saglam, and D. M. Zuckerman. *Curr. Opin. Struct. Biol.*, 43:88–94, 2017.

[138] A. Rojnuckarin, S. Kim, and S. Subramaniam. *Proc. Natl. Acad. Sci. USA*, 95:4288–4292, 1998.

[139] B. W. Zhang, D. Jasnow, and D. M. Zuckerman. *Proc. Natl. Acad. Sci. USA*, 104:18043–18048, 2007.

[140] J. L. Adelman, A. L. Dale, M. C. Zwier, D. Bhatt, L. T. Chong, D. M. Zuckerman, and M. Grabe. *Biophys. J.*, 101:2399–2407, 2011.

[141] A. Dickson, A. M. Mustoe, L. Salmon, and C. L. Brooks. *Nucleic Acids Res.*, 42:12126–12137, 2014.

[142] A. S. Saglam and L. T. Chong. *J. Phys. Chem. B*, 120:117–122, 2016.

[143] M. C. Zwier, A. J. Pratt, J. L. Adelman, J. W. Kaus, D. M. Zuckerman, and L. T. Chong. *J. Phys. Chem. Lett.*, 7:3440–3445, 2016.

[144] A. S. Saglam and L. T. Chong. *Chem. Sci.*, 10:2360–2372, 2019.

[145] M. C. Zwier and L. T. Chong. *Curr. Opin. Pharmacol.*, 10:745–752, 2010.

[146] B. W. Zhang, D. Jasnow, and D. M. Zuckerman. 2009. arXiv:0902.2772.

[147] M. Athènes and V. V. Bulatov. *Phys. Rev. Lett.*, 113:230601, 2014.

[148] M. Athènes, S. Kaur, G. Adjanor, T. Vanacker, and T. Jourdan. *Phys. Rev. Materials*, 3:103802, 2019.

[149] B. W. Zhang, D. Jasnow, and D. M. Zuckerman. *J. Chem. Phys.*, 126:074504, 2007.

[150] V. Satuluri and S. Parthasarathy. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 737–746. ACM New York, June 2009.

[151] V. Satuluri and S. Parthasarathy. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 247–256. ACM New York, Aug. 2010.

[152] Y.-K. Shih and S. Parthasarathy. *Bioinformatics*, 28:i473–i479, 2012.

[153] S. van Dongen. *Graph Clustering by Flow Simulation.* PhD thesis, University of Utrecht, 2000.

[154] A. J. Enright, S. van Dongen, and C. A. Ouzounis. *Nucleic Acids Res.*, 30:1575–1584, 2002.

[155] S. van Dongen. *SIAM J. Matrix Anal. Appl.*, 30:121–141, 2008.

[156] D. J. Sharpe and D. J. Wales. (in preparation).

[157] J. A. Joseph, C. S. Whittleston, and D. J. Wales. *J. Chem. Theory Comput.*, 12:6109–6117, 2016.

[158] R. G. Mantell, C. E. Pitt, and D. J. Wales. *J. Chem. Theory Comput.*, 12:6182–6191, 2016.

[159] B. Peters. *Reaction Rate Theory and Rare Events*. Elsevier, Oxford, UK, 2017.

[160] D. J. Wales. *Annu. Rev. Phys. Chem.*, 69:401–425, 2018.

[161] M. Griffiths and D. J. Wales. *J. Chem. Theory Comput.*, 15:6865–6881, 2019.

[162] D. J. Wales. *Curr. Op. Struct. Biol.*, 20:3–10, 2010.

[163] D. J. Wales. *Phil. Trans. Roy. Soc. A*, 370:2877–2899, 2012.

[164] T. D. Swinburne and D. J. Wales. *J. Chem. Theory Comput.*, 16:2661–2679, 2020.

[165] K. Reuter. In O. Deutschmann, editor, *Modeling Heterogeneous Catalytic Reactions: From the Molecular Process to the Technical System*, pages 71–111. Wiley-VCH, Weinheim, Germany, 2011.

[166] T. D. Swinburne and D. Perez. *Phys. Rev. Materials*, 2:053802, 2018.

[167] D. Gfeller, P. De Los Rios, A. Caflisch, and F. Rao. *Proc. Natl. Acad. Sci. USA*, 104:1817–1822, 2007.

[168] S. Fortunato. *Phys. Rep.*, 486:75–174, 2010.

[169] D. J. Sharpe and D. J. Wales. (in preparation).

[170] B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM, Philadelphia, PA, 1982.

[171] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, NY, USA, first edition, 2001.

[172] D. Aristoff. *ESAIM: M2AN*, 52:1219–1238, 2018.

[173] J. L. Adelman and M. Grabe. *J. Chem. Phys.*, 138:044105, 2013.

[174] E. Suárez, A. J. Pratt, L. T. Chong, and D. M. Zuckerman. *Protein Sci.*, 25:67–78, 2016.

[175] R. J. Allen, D. Frenkel, and P. R. ten Wolde. *J. Chem. Phys.*, 124:194111, 2006.

[176] C. Valeriani, R. J. Allen, M. J. Morelli, D. Frenkel, and P. R. ten Wolde. *J. Chem. Phys.*, 127:114109, 2007.

[177] E. E. Borrero and F. A. Escobedo. *J. Chem. Phys.*, 127:164101, 2007.

[178] N. B. Becker, R. J. Allen, and P. R. ten Wolde. *J. Chem. Phys.*, 136:174118, 2012.

[179] N. B. Becker and P. R. ten Wolde. *J. Chem. Phys.*, 136:174119, 2012.

[180] D. Bhatt and I. Bahar. *J. Chem. Phys.*, 137:104101, 2012.

[181] T. L. Hill. *Free Energy Transduction and Biochemical Cycle Kinetics*. Springer-Verlag, New York, NY, USA, 1989.

[182] R. Zwanzig. *Proc. Natl. Acad. Sci. USA*, 94:148–150, 1997.

[183] W. Zheng, E. Gallicchio, N. Deng, M. Andrec, and R. M. Levy. *J. Phys. Chem. B*, 115:1512–1523, 2011.

[184] F. Marinelli, F. Pietrucci, A. Laio, and S. Piana. *PLoS Comput. Biol.*, 5:e1000452, 2009.

[185] J. Jurazsek and P. G. Bolhuis. *Proc. Natl. Acad. Sci. USA*, 103:15859–15864, 2006.

[186] W. Du and P. G. Bolhuis. *J. Chem. Phys.*, 140:195102, 2014.

[187] G. Portella and M. Orozco. *Angew. Chem. Int. Ed.*, 49:7673–7676, 2010.

[188] G. Pinamonti, J. Zhao, D. E. Condon, F. Paul, F. Noé, D. H. Turner, and G. Bussi. *J. Chem. Theory Comput.*, 13:926–934, 2017.

[189] T. D. Swinburne, D. Kannan, D. J. Sharpe, and D. J. Wales. (submitted).

[190] H. Jung, K. Okazaki, and G. Hummer. *J. Chem. Phys.*, 147:152716, 2017.

[191] O. M. Becker and M. Karplus. *J. Chem. Phys.*, 106:1495–1517, 1997.

[192] D. J. Wales, M. A. Miller, and T. R. Walsh. *Nature*, 394:758–760, 1998.

[193] P. G. Bolhuis and C. Dellago. In L. Lipkowitz, editor, *Reviews in Computational Chemistry, Vol. 27*, pages 111–210. Wiley, Hoboken, New Jersey, 2010.

[194] J. M. Bello-Rivas and R. Elber. *J. Chem. Phys.*, 142:094102, 2015.

[195] A. Dickson, A. Warmflash, and A. R. Dinner. *J. Chem. Phys.*, 130:074104, 2009.

[196] A. Dickson, A. Warmflash, and A. R. Dinner. *J. Chem. Phys.*, 131:154104, 2009.

[197] A. Dickson and A. R. Dinner. *Annu. Rev. Phys. Chem.*, 61:441–459, 2010.

[198] S. Hussain and A. Haji-Akbari. *J. Chem. Phys.*, 152:060901, 2020.

[199] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. *J. Chem. Phys.*, 153:244108, 2020.

[200] B. Barzel and A.-L. Barabási. *Nat. Phys.*, 9:673–681, 2013.

[201] U. Harush and B. Barzel. *Nature Communications*, 8:2181, 2017.

[202] G. Hummer and A. Szabo. *J. Phys. Chem. B*, 119:9029–9037, 2015.

[203] L. Martini, A. Kells, R. Covino, G. Hummer, N.-V. Buchete, and E. Rosta. *Phys. Rev. X*, 7:031060, 2017.

[204] A. Kells, Z. E. Mihálka, A. Annibale, and E. Rosta. *J. Chem. Phys.*, 150:134107, 2019.

[205] A. Kells, V. Koskin, E. Rosta, and A. Annibale. *J. Chem. Phys.*, 152:104108, 2020.

Chapter 5

Conclusions and Outlook

We conclude the thesis by giving a brief overview of the theoretical and methodological advances reported herein. We discuss some recent related work that is beyond the scope of the current thesis, and suggest possible further extensions and variations of our proposed frameworks. Finally, we highlight the significance of the novel theory and algorithms described here by suggesting potential applications to realistic models, where our approach may lead to new insights into the dynamical behaviour of pertinent systems.

In this thesis we have reported various novel computational methods to analyze the dynamics of finite Markov chains. Crucially, our strategies scale favourably and are numerically stable, ensuring that the procedures remain applicable to the high-dimensional and ill-conditioned Markov chains that are typically encountered in the modeling of realistic dynamical systems.[1–5] The numerical analysis of *nearly reducible* Markov chains, comprising a mixture of fast and slow processes, is often intractable using conventional algorithms owing to the propagation of error in the finite precision arithmetic.[6–16] We have demonstrated the utility of our proposed approaches with applications to numerically challenging models that are relevant to current problems in physical science.

Before the developments of the current work, there existed robust procedures to compute quantities characterizing the global dynamics, such as the stationary distribution (by the Grassmann-Taksar-Heyman algorithm[17,18] or uncoupling-coupling procedures[19–24]), mean first passage times (MFPTs) for a transition to an absorbing state (by the graph transformation algorithm[25–28]), and the average mixing time (by the FUND[29,30] and REFUND[31] algorithms) (Chapter 1). We have extended the family of state reduction algorithms to compute microscopic quantities, allowing for detailed analysis to probe the relationship between local regions of a Markovian network and the slow, global dynamics. Specifically, we have devised efficient algorithms to compute committor probabilities[32,33] for nodes (the probability that a path initialized at a node is a transition path, i.e. hits the absorbing state before visiting the initial state), and the expected numbers of times that nodes are visited on paths prior to absorption (Chapter 3). We then derived an expression for the probability that a node is visited along a transition path, which is readily evaluated using the above information. Thus, all of the quantities required to identify the individual nodes of a Markov chain that have a dominant effect in modulating the overall transition can be obtained in a numerically stable manner. In addition to this nodewise analysis, we proposed a pathwise analysis to quantitatively assess the dynamical relevance of alternative competing mechanisms for a transition of interest (Chapter 2). This framework uses knowledge of the committor probabilities and stationary distribution to exactly decompose the overall reactive flux for a transition into additive contributions from a finite set of transition flux-paths. Further extensions of the graph transformation algorithm generalize the procedure to any path property that is a sum of contributions from individual transitions (Chapter 2), and incorporate a backward pass phase to compute the MFPTs for transitions from all nonabsorbing nodes in a single computation (Chapter 3).

In Chapter 4 we proposed a framework for the efficient and exact sampling of trajectories on a Markovian network, which avoids the problem of devoting excessive computational resources to simulate the unproductive flickering of trajectories that are trapped in metastable

states.[34,35] This flickering issue renders standard kinetic Monte Carlo methods prohibitively inefficient for simulating trajectories on nearly reducible Markov chains.[36] Our strategy uses the multi-level regularized[37,38] Markov clustering[39–41] (MLR-MCL) unsupervised community detection algorithm to efficiently obtain an initial clustering of the network into metastable states, which is subsequently refined by a variational optimization procedure. The resulting communities are employed in kinetic path sampling[42,43] (kPS) simulations, which uses a state reduction algorithm to sample the numbers of internode transitions in a trajectory segment that escapes from the currently occupied community.[44]

The separation of characteristic timescales that defines a nearly reducible Markov chain[3,45,46] can be exploited to obtain a reduced Markovian network where the metastable macrostates are represented by individual nodes.[47–49] In related work that is beyond the scope of the thesis, we have contributed to devising strategies for accurately determining the transition probabilities or rates of a reduced Markov chain, for a given partitioning of the original network. One approach to this problem uses local eigendecompositions of the respective communities to estimate appropriate coarse-grained transition rates.[50] The reduced discrete- or continuous-time Markov chain, which is optimal in the sense of most accurately preserving the occupation number correlation functions of the communities, can be obtained via inversion of the matrix of pairwise MFPTs for all transitions between nodes of the original Markov chain.[51–53] To minimize numerical error in this process, the MFPT matrix can be computed efficiently and robustly using state reduction methods.[54] Another possible route to determine a coarse-grained Markov chain is to use the simulation strategy of Chapter 4 to sample many short-timescale trajectories efficiently.[55] The probabilities or rates for transitions between the predefined communities can then be inferred from the trajectory data using maximum-likelihood[56–59] or Gibbs sampling[60–63] methods. Finally, a somewhat unorthodox approach to the dimensionality reduction problem is to eliminate a subset of nodes from each community by renormalization.[64] This latter scheme requires that the choice of eliminated nodes does not lead to the significant loss of information on the slow dynamics for intercommunity transitions. Conceiving heuristics to prioritize nodes for elimination in this framework therefore raises interesting theoretical questions on the relationship between network topology and global dynamics.[65]

These advanced approaches to assigning coarse-grained transition probabilities or rates, which accurately preserve the global dynamics of the original Markovian network, are less sensitive to the choice of communities than simply invoking the local equilibrium approximation.[54,66] Nonetheless, since nodes at the boundaries of metastable states play a critical role in facilitating the intercommunity transitions, the choice of clustering algorithm is a pivotal consideration in dimensionality reduction workflows. The numerical stability of uncoupling-

coupling procedures, and of the block formulation of the graph transformation algorithm, likewise relies on a partitioning of the network into communities that appropriately reflect the metastable sets of nodes.[1,24] Similarly, the kPS algorithm requires a suitable choice of metastable communities in order to simulate paths efficiently.[42,44] Our variational optimization procedure to refine the boundaries of an initial clustering, described in Chapter 4, attenuates the influence of the initial clustering method. This property is useful for improving the reproducibility of a dimensionality reduction workflow, since state-of-the-art scalable community detection algorithms, such as MLR-MCL, are typically stochastic.[67–69] Moreover, many popular community detection algorithms are based on heuristics or objectives that are not necessarily appropriate in the context of Markov chains and are therefore liable to misclassify nodes,[70] especially those at the boundaries of metastable macrostates. The proposed variational optimization procedure is based on a rigorous objective function, namely the second dominant eigenvalue of the transition probability or rate matrix, or the Kemeny constant. Hence, this process improves the quality of an initial clustering in a readily interpretable way.

While the variational optimization procedure can partially correct for misclassified nodes at the boundaries of the communities, the choice of initial clustering procedure remains important. A suitable community detection algorithm should be scalable, numerically stable, require no prior knowledge of the Markov chain besides the edge weights, and be based on a heuristic or objective that explicitly considers the edge weights to be Markovian transition probabilities or rates. Clustering algorithms that are based on the eigenvectors of the Markov chain, such as the original[71,72] and robust[73,74] Perron cluster-cluster analysis algorithms (PCCA and PCCA+, respectively), are not numerically stable,[10] but could be applied to weakly metastable Markov chains. Clustering algorithms that are based on the modularity objective function,[75] such as the Louvain algorithm,[76] are efficient, but often fail to characterize long-lived macrostates in Markov chains.[77]

In Chapter 4, we identified MLR-MCL as a community detection procedure having a desirable balance of properties for application to Markov chains. A second promising candidate procedure is the InfoMap algorithm,[78] which has favourable time complexity, yields a hierarchical clustering, and incorporates regularization via a single free parameter. The map equation that serves as the objective function in the InfoMap algorithm,[79] which is optimized by a stochastic node reassignment procedure,[76] has a rigorous interpretation relating to the theory of random walks. An alternative approach is the BACE (Bayesian agglomerative clustering engine) algorithm,[80] where the two nodes that are associated with the smallest pairwise Bayes factor, which indicates that the nodes share similar transition probabilities to neighbouring states, are merged at each iteration. However, the BACE

algorithm has cubic time complexity in the number of nodes, and is therefore not feasible for application to Markov chains with a large number of nodes. The BACE algorithm has the advantage that an explicit stopping criterion can be specified; either once the desired number of communities has been reached, or when the lowest Bayes factor exceeds a threshold. It is reasonable to suggest that the "best" community detection procedure depends on both the system and the intended application. For instance, kPS requires that the metastable communities are not too large,[42,44] but this consideration is irrelevant for determining a reduced Markov chain by the local equilibrium approximation or the Hummer-Szabo relation.[51,53] Extensive benchmarking of alternative clustering approaches for different purposes would serve as an important reference to guide practitioners.

Another potential avenue to extend the methodological advances of the present work is to use the kPS algorithm in conjunction with enhanced sampling methods for handling a set of independent trajectories.[81,82] A particularly attractive possibility is to utilize the exact milestoning approach,[83,84] wherein short trajectory segments are initialized at the boundaries between macrostates, termed *milestones*, and terminate at neighbouring milestones.[85] These trajectory data can then be used to calculate MFPTs for the transitions between all pairs of milestones.[33,86] Hence, milestoning simulations provide a numerical strategy for the dimensionality reduction of Markov chains that is scalable and offers a complementary perspective to approaches based on estimating transition rates between communities of nodes. Milestoning simulations are readily parallelizable by focussing on trajectory segments between appropriately defined milestones separately. Moreover, the objects required to estimate the first passage time distributions between milestones, i.e. the graph-transformed networks in kPS, can be stored and subsequently recycled. Hence, combined milestoning and kPS provides a powerful method to simulate the *equilibrium* path ensemble, which is otherwise challenging to access efficiently.[44] Practical aspects of milestoning, such as suitable milestone placement, and considerations for other enhanced sampling methods, differ significantly for the discrete- compared to the more usual continuous-state case, and represent an interesting direction for further query.

The algorithms described in the thesis are implemented in a new software package, `DISCOTRESS` (github.com/danieljsharpe/DISCOTRESS), a `C++` program for the efficient simulation and numerically stable analysis of nearly reducible discrete- and continuous-time finite Markov chains. `DISCOTRESS` is freely available to download under the GNU General Public License, and is provided with extensive documentation and tutorials. The software is highly flexible, allowing for a variety of calculations to be performed conveniently. In addition to the default operating mode of the program, namely to simulate the nonequilibrium first passage path ensemble,[87] there are further options to simulate: the equilibrium (i.e. steady

state) path ensemble,[88] fixed-timescale trajectories that are not necessarily conditioned on endpoint sets of states, and many short-timescale trajectories to harvest data for estimating a reduced Markov chain.[55] Each of these methods is compatible with any chosen kinetic Monte Carlo method. Through the specification of bins, which are distinct from the communities leveraged in a kPS simulation, nodewise statistics, such as committor and reactive visitation probabilities, can be estimated from trajectory data for arbitrary groups of nodes. The software also includes special methods to conduct exact analyses using various state reduction algorithms, and a class implementing the recursive enumeration algorithm[89] to determine the dominant transition paths and their relative importance. The code is object-oriented, and therefore it is possible to extend the software with a custom class to handle a set of trajectories that are propagated independently by an unspecified kinetic Monte Carlo algorithm,[44] for example to perform a milestoning simulation. The availability of advanced algorithms to analyze ill-conditioned Markovian networks will aid researchers in diverse disciplines to extract macroscopic and microscopic dynamical information on computationally challenging models.

We have illustrated our proposed methodologies with applications to Markov chains representing dynamical processes of current interest in the physical sciences. In particular, we have analyzed the dominant pathways and key influential states for two configurational transitions, namely, the solid-solid transition of an atomic cluster (Chapter 3) and the folding transition of a peptide (Chapter 4). The properties of these systems, and of the benchmark eight-state system of Chapter 2, are typical of Markov chains representing the energy landscape of a continuous-state physical system.[90] For such models, the exponential sensitivity of transition rates to the heights of energy barriers leads to metastability and consequent ill-conditioning.[26] The numerically stable and efficient methods of the current work, and the dimensionality reduction methods based on the extension of the concepts presented herein, will therefore lend themselves to varied applications in analyzing the dynamics of complex physical systems such as glassy materials[91] and biomolecules.[92,93]

Nearly reducible Markov chains also arise naturally in many other disciplines.[5] For instance, extinction of a species is a rare event in a discrete-state population dynamics model of an ecosystem.[94–96] An extreme weather event is a low-probability transition in a climate dynamics simulation.[97] In economic models, large-scale market changes take place on a much slower timescale than small fluctuations arising from individual trades.[98] The capability to perform computational analyses that were previously intractable could yield new insights into the dynamics of realistic systems, including those mentioned above.

Our methodology could also lead to improved fundamental understanding of Markovian networks, such as the complex interplay between network topology and dynamics, and how

local features of a network are manifested in the global dynamics. In the current work, we have found common features shared between the transition path ensembles[88] for three different systems characterized by an underlying energy landscape: a benchmark model, an atomic cluster, and a peptide. Namely, the set of transition paths that are associated with a non-negligible proportion of the productive flux is highly localized in the state space, and the kinetically relevant states, which have a dominant influence on the macroscopic dynamical properties of the Markov chain, are highly localized also. Systematic investigation of archetypal Markovian networks could lead to the identification of universality classes for dynamical behaviour, following similar frameworks for networks where the dynamics are governed by deterministic ordinary differential equations.[99,100]

Bibliography

[1] P. J. Courtouis and S. P. *Linear Algebra Appl.*, 76:59–70, 1986.

[2] E. Vanden-Eijnden and J. Weare. *Commun. Pure Appl. Math*, 65:1770–1803, 2012.

[3] M. K. Cameron. *J. Chem. Phys.*, 141:184113, 2014.

[4] T. Gan and M. Cameron. *J. Nonlinear Sci.*, 27:927–972, 2017.

[5] C. Pérez-Espigares and P. I. Hurtado. *Chaos*, 29:083106, 2019.

[6] W. Grassmann and D. A. Stanford. In W. Grassmann, editor, *Computational Probability*, pages 153–203. Springer, New York, 2000.

[7] J. R. Koury, D. F. McAllister, and W. J. Stewart. *SIAM J. Alg. Discr. Meth.*, 5:164–186, 1984.

[8] D. P. Heyman. *SIAM J. Alg. Discr. Meth.*, 8:226–232, 1987.

[9] D. P. Heyman and A. Reeves. *ORSA J. Comp.*, 1:52–60, 1989.

[10] B. Philippe, Y. Saad, and W. J. Stewart. *Oper. Res.*, 40:1156–1179, 1992.

[11] C. A. O'Cinneide. *Numer. Math.*, 65:109–120, 1993.

[12] C. D. Meyer Jr. *SIAM J. Matrix Anal. Appl.*, 15:715–728, 1994.

[13] C. A. O'Cinneide. *Numer. Math.*, 73:507–519, 1996.

[14] D. P. O'Leary and Y.-J. J. Wu. *SIAM J. Matrix Anal. Appl.*, 17:470–488, 1996.

[15] D. J. Hartfiel and C. D. Meyer Jr. *Linear Algebra Appl.*, 272:193–203, 1998.

[16] J. L. Barlow. *SIAM J. Matrix Anal. Appl.*, 22:230–241, 2000.

[17] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. *Oper. Res.*, 33:1107–1116, 1985.

[18] T. J. Sheskin. *Oper. Res.*, 33:228–235, 1985.

[19] C. D. Meyer Jr. *SIAM Rev.*, 31:240–272, 1989.

[20] C. D. Meyer. *Linear Algebra Appl.*, 114-115:69–94, 1989.

[21] P. J. Schweitzer. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 63–87. Marcel Dekker, New York, 1991.

[22] M. Haviv. *SIAM J. Numer. Anal.*, 22:952–966, 1987.

[23] W. J. Stewart and W. Wu. *ORSA J. Comp.*, 4:336–350, 1992.

[24] T. Dayar and W. J. Stewart. *SIAM J. Sci. Comput.*, 17:287–303, 1996.

[25] D. J. Wales. *J. Chem. Phys.*, 130:204111, 2009.

[26] J. D. Stevenson and D. J. Wales. *J. Chem. Phys.*, 141:041104, 2014.

[27] R. S. MacKay and J. D. Robinson. *Phil. Trans. Roy. Soc. A*, 376:20170232, 2018.

[28] T. D. Swinburne and D. J. Wales. *J. Chem. Theory Comput.*, 16:2661–2679, 2020.

[29] D. P. Heyman. *SIAM J. Matrix Anal. Appl.*, 16:954–963, 1995.

[30] D. P. Heyman and D. P. O'Leary. *SIAM J. Matrix Anal. Appl.*, 19:534–540, 1998.

[31] I. Sonin and J. Thornton. *SIAM J. Matrix Anal. Appl.*, 23:209–224, 2001.

[32] J.-H. Prinz, M. Held, J. C. Smith, and F. Noé. *Multiscale Model. Simul.*, 9:545–567, 2011.

[33] A. M. Berezhkovskii and A. Szabo. *J. Chem. Phys.*, 150:054106, 2019.

[34] M. A. Novotny. *Phys. Rev. Lett.*, 74:1–5, 1995.

[35] D. R. Mason, R. E. Rudd, and A. P. Sutton. *Comput. Phys. Comm.*, 160:140–157, 2004.

[36] B. Puchala, M. L. Falk, and K. Garikipati. *J. Chem. Phys.*, 132:134104, 2010.

[37] V. Satuluri and S. Parthasarathy. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 737–746. ACM New York, June 2009.

[38] V. Satuluri and S. Parthasarathy. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 247–256. ACM New York, Aug. 2010.

[39] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[40] A. J. Enright, S. van Dongen, and C. A. Ouzounis. *Nucleic Acids Res.*, 30:1575–1584, 2002.

[41] S. van Dongen. *SIAM J. Matrix Anal. Appl.*, 30:121–141, 2008.

[42] M. Athènes and V. V. Bulatov. *Phys. Rev. Lett.*, 113:230601, 2014.

[43] M. Athènes, S. Kaur, G. Adjanor, T. Vanacker, and T. Jourdan. *Phys. Rev. Materials*, 3:103802, 2019.

[44] D. J. Sharpe and D. J. Wales. *J. Chem. Phys.*, 153:024121, 2020.

[45] E. Meerbach, C. Schütte, and A. Fischer. *Linear Algebra Appl.*, 398:141–160, 2005.

[46] M. K. Cameron and E. Vanden-Eijnden. *J. Stat. Phys.*, 156:427–454, 2014.

[47] P. Buchholz. *J. Appl. Probab.*, 31:59–75, 1994.

[48] W. E, T. Li, and E. Vanden-Eijnden. *Proc. Natl. Acad. Sci. USA*, 105:7907–7912, 2008.

[49] J. A. Ward and M. López-García. *Appl. Netw. Sci.*, 4:108, 2019.

[50] T. D. Swinburne, D. Kannan, D. J. Sharpe, and D. J. Wales. *J. Chem. Phys.*, 153:134115, 2020.

[51] G. Hummer and A. Szabo. *J. Phys. Chem. B*, 119:9029–9037, 2015.

[52] A. Kells, Z. E. Mihálka, A. Annibale, and E. Rosta. *J. Chem. Phys.*, 150:134107, 2019.

[53] A. Kells, V. Koskin, E. Rosta, and A. Annibale. *J. Chem. Phys.*, 152:104108, 2020.

[54] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. *J. Chem. Phys.*, 153:244108, 2020.

[55] G. R. Bowman, V. S. Pande, and F. Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Springer, Netherlands, first edition, 2014.

[56] N.-V. Buchete and G. Hummer. *J. Phys. Chem. B*, 112:6057–6069, 2008.

[57] N.-V. Buchete and G. Hummer. *Phys. Rev. E*, 77:030902, 2008.

[58] P. Metzner, E. Dittmer, T. Jahnke, and C. Schütte. *J. Comput. Phys.*, 227:353–375, 2007.

[59] R. T. McGibbon and V. S. Pande. *J. Chem. Phys.*, 143:034109, 2015.

[60] P. Metzner, F. Noé, and C. Schütte. *Phys. Rev. E*, 80:021106, 2009.

[61] B. Trendelkamp-Schroer and F. Noé. *J. Chem. Phys.*, 138:164113, 2013.

[62] B. Trendelkamp-Schroer, H. Wu, F. Paul, and F. Noé. *J. Chem. Phys.*, 143:174101, 2015.

[63] B. Trendelkamp-Schroer and F. Noé. *Phys. Rev. X*, 6:011009, 2016.

[64] D. Kannan, D. J. Sharpe, T. D. Swinburne, and D. J. Wales. (unpublished), 2020.

[65] D. Kannan. *Dimensionality Reduction of Complex Networks with Rare Events*. PhD thesis, University of Cambridge, 2020.

[66] L. Martini, A. Kells, R. Covino, G. Hummer, N.-V. Buchete, and E. Rosta. *Phys. Rev. X*, 7:031060, 2017.

[67] F. D. Malliaros and M. Vazirgiannis. *Phys. Rep.*, 533:95–142, 2013.

[68] Z. Yang, R. Algesheimer, and C. J. Tessone. *Sci. Rep.*, 6:30750, 2016.

[69] S. Fortunato and D. Hric. *Phys. Rep.*, 659:1–44, 2016.

[70] S. Fortunato. *Phys. Rep.*, 486:75–174, 2010.

[71] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. *Linear Algebra Appl.*, 315:39–59, 2000.

[72] F. Noé, I. Horenko, C. Schütte, and J. C. Smith. *J. Chem. Phys.*, 126:155102, 2007.

[73] P. Deuflhard and M. Weber. *Linear Algebra Appl.*, 398:161–184, 2005.

[74] S. Kube and M. Weber. *J. Chem. Phys.*, 126:024103, 2007.

[75] M. E. J. Newman and M. Girvan. *Phys. Rev. E*, 69:026113, 2004.

[76] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. *J. Stat. Mech.*, 10:P10008, 2008.

[77] D. Gfeller, P. De Los Rios, A. Caflisch, and F. Rao. *Proc. Natl. Acad. Sci. USA*, 104:1817–1822, 2007.

[78] M. Rosvall and C. T. Bergstrom. *Proc. Natl. Acad. Sci. USA*, 105:1118–1123, 2008.

[79] M. Rosvall, D. Axelsson, and C. T. Bergstrom. *Eur. Phys. J. Special Topics*, 178:13–23, 2009.

[80] G. R. Bowman. *J. Chem. Phys.*, 137:134111, 2012.

[81] J. T. Berryman and T. Schilling. *J. Chem. Phys.*, 133:244101, 2010.

[82] R. J. Allen, D. Frenkel, and P. R. ten Wolde. *J. Chem. Phys.*, 124:024102, 2006.

[83] J. M. Bello-Rivas and R. Elber. *J. Chem. Phys.*, 142:094102, 2015.

[84] D. Aristoff, J. M. Bello-Rivas, and R. Elber. *Multiscale Model. Simul.*, 14:301–322, 2016.

[85] R. Elber. *Q. Rev. Biophys.*, 50:e8, 2017.

[86] S. Viswanath, S. M. Kreuzer, A. E. Cardenas, and R. Elber. *J. Chem. Phys.*, 139:174105, 2013.

[87] M. von Kleist, C. Schütte, and W. Zhang. *J. Stat. Phys.*, 170:809–843, 2018.

[88] P. Metzner, C. Schütte, and E. Vanden-Eijnden. *Multiscale Model. Simul.*, 7:1192–1219, 2009.

[89] V. M. Jiménez and A. Marzal. In J. S. Vitter and C. D. Zaroliagis, editors, *Algorithm Engineering: 3rd International Workshop, WAE'99, London, UK*, pages 15–29. Springer Berlin, Heidelberg, 1999.

[90] D. J. Wales. *Annu. Rev. Phys. Chem.*, 69:401–425, 2018.

[91] S. P. Niblett, M. Biedermann, D. J. Wales, and V. K. de Souza. *J. Chem. Phys.*, 147:152726, 2017.

[92] J. A. Joseph, K. Röder, D. Chakraborty, R. G. Mantell, and D. J. Wales. *Chem. Commun.*, 53:6974–6988, 2017.

[93] K. Röder, J. A. Joseph, B. E. Husic, and D. J. Wales. *Adv. Theory Simul.*, 2:1800175, 2019.

[94] L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology*. Prentice Hall, Upper Saddle River, New Jersey, 2003.

[95] L. J. S. Allen. *Stochastic Population and Epidemic Models*. Springer, Cham, 2015.

[96] B. S. Lindley, L. B. Shaw, and I. B. Schwartz. *EPL*, 108:58008, 2014.

[97] N. Malik and U. Ozturk. *Chaos*, 30:090401, 2020.

[98] H. A. Simon and A. Ando. *Econometrica*, 29:111–138, 1961.

[99] B. Barzel and A.-L. Barabási. *Nature Physics*, 9:673–681, 2013.

[100] U. Harush and B. Barzel. *Nature Communications*, 8:2181, 2017.