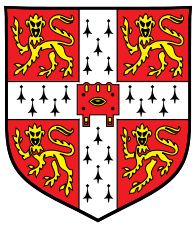


University of Cambridge
Department of Chemistry

Pathways and Energy Landscapes

A thesis submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Semen A. Trygubenko

Darwin College

January 2006

Declaration

The work described in this dissertation was carried out in the Department of Theoretical Chemistry at the University of Cambridge between October 2002 and January 2006. The contents are the work of the author and contain nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements, and have not previously been submitted for any degree or qualification at another institution. The number of words does not exceed 60 000.

Semen A. Trygubenko

Cambridge, January 2006

Acknowledgements

I am glad I have the opportunity to express my sincere gratitude to all the people that accompanied and supported me during my study.

I had the pleasure of being supervised by Dr. David Wales and would like to thank him for the high-quality guidance I received in the duration of the Ph.D. course.

The second person I gratefully acknowledge is Dr. Catherine Pitt who introduced me to the Debian GNU/Linux operating system and various scientific applications that were heavily used throughout the development of this work.

My appreciation also goes to Dr. Thomas Middleton for providing transition state databases for the binary Lennard-Jones system, Dr. David Evans for his Dijkstra-based utilities that locate shortest paths in connected databases of minima and transition states, Dr. Mark Miller for many KMC-related discussions as well as his programs for creating and manipulating disconnectivity graphs, and Dr. Dmytro Hovorun and Dr. Pavel Hobza who greatly influenced my decision to do a Ph.D. in this area.

The chain of my gratitude would definitely be incomplete without special thanks to Tetyana Bogdan, Dr. Joanne Carr and Tim James, for many stimulating discussions we had on various aspects of my research, the Life, the Universe and everything.

Lastly, I would like to thank Cambridge Commonwealth Trust, Cambridge Overseas Trust and Darwin College for financial support. Most of the calculations were performed using computational facilities funded by the Isaac Newton Trust.

To My
Teachers

Abstract

A modification of the nudged elastic band (NEB) method is presented that enables stable optimisations to be run using both the limited-memory quasi-Newton (L-BFGS) and slow-response quenched velocity Verlet minimisers. The performance of this new 'doubly nudged' DNEB method is analysed in conjunction with both minimisers and compared with previous NEB formulations. We find that the fastest DNEB approach (DNEB/L-BFGS) can be quicker by up to two orders of magnitude. Applications to permutational rearrangements of the seven-atom Lennard-Jones cluster (LJ_7) and highly cooperative rearrangements of LJ_{38} and LJ_{75} are presented.

Secondly, we propose new measures of localisation and cooperativity for the analysis of atomic rearrangements. We show that for both clusters and bulk material cooperative rearrangements usually have significantly lower barriers than uncooperative ones, irrespective of the degree of localisation. We also find that previous methods used to sample stationary points are biased towards rearrangements of particular types. Linear interpolation between local minima in double-ended transition state searches tends to produce cooperative rearrangements, while random perturbations of all the coordinates, as sometimes used in single-ended searches, has the opposite effect.

Thirdly, we report a new algorithm for constructing pathways between local minima that involve a large number of intervening transition states on the PES. A significant improvement in efficiency has been achieved by changing the strategy for choosing successive pairs of local minima that serve as endpoints for the next search. We employ Dijkstra's algorithm to identify the 'shortest' path corresponding to missing connections within an evolving database of local minima and the transition states that connect them.

Finally, we describe an exact approach for calculating the total transition probabilities in finite-state discrete-time Markov processes. All the states and the rules for transitions between them must be known in advance. We can then calculate averages over a given ensemble of paths for both additive and multiplicative properties in a non-stochastic and non-iterative fashion. In particular, we can calculate the mean first passage time between arbitrary groups of stationary points for discrete path sampling databases, and hence extract phenomenological rate constants. We present a number of examples to demonstrate the efficiency and robustness of this approach.

Contents

List of Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno minimisation algorithm [1–3]
BFM	Bellman-Ford-Moore algorithm
BKL	Bortz-Kalos-Lebowitz algorithm
BLJ	Binary Lennard-Jones potential
CG	Conjugate Gradient
CHARMM	Chemistry at HARvard Molecular Modelling program [4]
CHARMM19	United-atom CHARMM force field [5]
CHARMM22	All-atom CHARMM force field [6]
CPU	Central Processing Unit
CS	Chain-of-States approach
DFP	Davidon-Fletcher-Powell optimisation method
DGT	Dense-optimised Graph Transformation method
DNEB	Doubly Nudged Elastic Band method [7]
DPS	Discrete Path Sampling method [8–10]
EEF1	Effective Energy Function 1 solvation potential [11]
EF	Eigenvector-Following method [12–24]
GT	Graph Transformation method
KMC	Kinetic Monte Carlo method
L-BFGS	Limited memory version of the BFGS algorithm [2]
LJ	Lennard-Jones potential
LST	Linear Synchronous Transit method
MB	Müller-Brown surface [25]
MCAMC	Monte Carlo with Absorbing Markov Chains [26]
MD	Molecular Dynamics method

MM	Matrix Multiplication method
NEB	Nudged Elastic Band method [27–30]
NMR	Nuclear Magnetic Resonance
NR	Newton-Raphson approach [31]
ORT	Overall Rotation and Translation
PDB	Protein Data Bank
PES	Potential Energy Surface
QST	Quadratic Synchronous Transit method
QVV	Quenched Velocity Verlet minimisation algorithm [27]
RMS	Root-Mean-Square
SDGT	A version of the GT method that is a combination of SGT and DGT
SGT	Sparse-optimised Graph Transformation method
SMM	Sparse-optimised Matrix Multiplication method
SQVV	Slow-response Quenched Velocity Verlet minimisation algorithm [7]
SRW	Simple Random Walk
VV	Velocity Verlet integrator [32]

Glossary of Symbols

Consistency is the last refuge of the unimaginative.

Oscar Wilde

$\langle x \rangle$	The mean value of variable x	81
$\emptyset$	Empty set	105
$\hat{}$	Denotes a unit vector; $\hat{\mathbf{x}} = \mathbf{x}/\lvert\mathbf{x}\rvert$	12
$\lvert\,\rvert$	A vector norm; $\lvert\mathbf{x}\rvert = \sqrt{\sum_{i=1}^{dim(\mathbf{x})} x_i^2}$; also, cardinality of a set	10
$\ominus$	Denotes symmetric difference of two sets	100
$\otimes$	Denotes vector direct product, a.k.a. dyadic; $\mathbf{a} \otimes \mathbf{b}^T = \mathbf{c}$, $c_{i,j} = a_i b_j$	18
$(\,,\,)$	A scalar product of two vectors, a.k.a. dot product; $(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \cdot \mathbf{b}$	12
$\{\,\}$	A set of objects; $\{\mathbf{x}_i\}_1^n = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n\}$	10
$\mathbf{0}$	A matrix or a vector filled with zeros	15
A	One of the two superstates in two-state kinetic model	79
$\mathbf{A}$	Approximation to the inverse Hessian matrix, $\mathbf{H}^{-1}$	18
$Adj[i]$	Set of all the nodes adjacent to node i	94
$AdjIn[i]$	Set of all the nodes connected to node i via incoming edges	75
$AdjOut[i]$	Set of all the nodes connected to node i via outgoing edges	75
B	One of the two superstates in two-state kinetic model	79
BLJ_n	Binary Lennard-Jones liquid with n atoms in a periodic cell	65
$\mathbf{C}$	Square matrix, columns of which are eigenvectors	20
C_N	A chain graph with N nodes	85
D	Endpoint separation	57
$\Delta_i(j)$	ith displacement in magnitude of an atom between structures $j-1$ and j	61
E	A set of edges	85
$\mathcal{E}_\alpha^G$	Probability of escape from G starting from α in a single step	80
E_-	Energy barrier corresponding to the reverse reaction	43

E_+	Energy barrier corresponding to the forward reaction	43
F	Set of all minima connected to the final endpoint	40
$\mathcal{F}$	Frequency distribution function	55
G_N	An arbitrary graph with N nodes	89
$\mathbf{H}(\mathbf{x})$	Hessian matrix evaluated at point $\mathbf{x}$	15
I	A set containing all the minima that do not belong to $A \cup B$	79
K_N	A complete graph with N nodes	89
L	Lagrangian function	22
LJ_n	n-atom Lennard-Jones cluster	30
$\mho(\mathbf{x})$	The set of all possible values of the control variable $\mathbf{x}$	15
N	Number of atoms; number of nodes in a graph	55
N_c	Cooperativity index	55
N_f	Number of frames or points sampled along a path	54
N_i	Number of images in a band	10
N_p	Participation index	55
$\widetilde{N}$	Participation index evaluated using the endpoints alone	42
O_k	Displacement overlap evaluated for k atoms using displacements $d_i(j)$	58
$\mathcal{O}(\,)$	$f(n) = \mathcal{O}\Big(g(n)\Big)$ means $0 \leqslant f(n) \leqslant cg(n)$ holds for some constants $c > 0$	87
$\mathbf{P}$	Transition probability matrix	74
P_i^{eq}	Equilibrium occupation probability of state i	79
$P_i(t)$	Occupation probability of state i at time t	78
$P_{j,i}$	Probability of transition from state i to state j	75
$\mathcal{P}_\xi$	Pathway probability	81
R_N	A random graph with N nodes	102
$\mathbf{R}_\alpha$	$3N$-dimensional rotation matrix about axis α, $\alpha \in \{x, y, z\}$	21
$\mathbb{R}$	The set of all real numbers	20
S	Set of all minima connected to the starting endpoint	40
Σ_α^G	Total probability of escape from G if started at node α	80
$\mathcal{S}_{\alpha,\beta}^G$	Sum of weights of all pathways connecting α and β and confined to G	81
T	Temperature	74
Θ_k	Displacement overlap evaluated for k atoms using displacements $\Delta_i(j)$	61

$\mathcal{T}_i^G$	Mean escape time from graph G if started at node i	83
U	Set of all minima that do not belong to $S \cup F$	40
$\Upsilon(\mathbf{x}, \varepsilon)$	A set of feasible points contained in the neighbourhood ε of $\mathbf{x}$	15
V	Potential energy functional; also, a set of graph nodes	2
$\boldsymbol{\mathcal{V}}$	$3N$-dimensional vector of velocities*	17
$\widetilde{V}$	Spring potential	10
$W(a, b)$	Weight of the shortest path $\xi = a \leftarrow b$; $W(a, b) = -\ln(\mathcal{W}_\xi)$	134
$\mathbb{W}$	The set of whole numbers; $\mathbb{W} = \{0, 1, 2, \dots\}$	76
$\mathcal{W}_\xi$	Product of branching probabilities associated with path ξ	81
$\mathbf{X}$	$3N$-dimensional vector representing a point in configuration space	10
Ξ	Pathway ensemble	80
a	A state that belongs to a superstate A	79
α	Pathway nonlinearity index	57
b	A state that belongs to a superstate B	79
β	Energy barrier asymmetry index	43
$\mathbf{c}$	Eigenvector	20
$\det \mathbf{M}$	A determinant [33] (a scalar-valued function) of matrix $\mathbf{M}$	21
d_i	Integrated path length for atom i; also, degree of node i	55
$d_i(j)$	Displacement of atom i between structures $j-1$ and j	55
$e_{j,i}$	Directed edge that describes a transition from node i to node j	75
ϵ	A parameter in LJ potential (the depth of the potential energy well)	2
ε	Small positive parameter	15
η	Number of atomic degrees of freedom	12
$f(\mathbf{x})$	Objective function of a vector argument $\mathbf{x}$	15
$\mathbf{g}$	$3N$-dimensional gradient vector of the true potential	10
γ	Kurtosis of a distribution evaluated using moments about the mean	56
γ'	Kurtosis of a distribution evaluated using moments about the origin	57
$\widetilde{\mathbf{g}}^{\parallel}$	Spring gradient vector component parallel to the path	13
$\widetilde{\mathbf{g}}^{\perp}$	Spring gradient vector component perpendicular to the path	12
$\widetilde{\mathbf{g}}$	$3N$-dimensional gradient vector of the spring potential	10

*This vector and the other vectors defined here are column vectors.

$\mathbf{g}^{\parallel}$	True gradient vector component parallel to the path	10
$\mathbf{g}^{\perp}$	True gradient vector component perpendicular to the path	10
i, j, k	Indices; range and meaning may vary depending on the context	16
k_B	Boltzmann's constant	74
$k_{j,i}$	Rate constant for transitions from state i to state j	78
k_{spr}	Spring force constant	12
$l(\xi)$	Length of path ξ	82
λ	Eigenvalue	20
m	Atomic mass	17
m_n	nth moment of a distribution function about the mean	56
m'_n	nth moment of a distribution function about the origin	57
n	Time parameter of a discrete-time stochastic process	77
$o(\,)$	$f(n) = o\Big(g(n)\Big)$ means $0 \leqslant f(n) \leqslant cg(n)$ holds for all constants $c > 0$*	135
$\mathbf{p}$	Search direction vector	16
π	Path length asymmetry index	42
$\mathbf{r}_i(j)$	Three-dimensional Cartesian coordinates vector of atom i for structure j	54
s	Integrated path length	37
σ	A parameter in the LJ potential ($2^{1/6}\sigma$ is the pair equilibrium separation)	2
$\boldsymbol{\tau}$	$3N$-dimensional tangent vector	12
τ_i	Mean waiting time in state i before escape	78
t	Time	17
δt	Time integration step	17
^{T}	A matrix or vector transpose	15
ϖ	Step size	16
v_i	ith graph node	85
$w(u, v)$	Weight of the undirected edge connecting nodes u and v	44
x_i	ith component of vector $\mathbf{x}$	15
ξ	A pathway	81
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Vectors; dimensionality and meaning may vary depending on the context	15

*Otherwise known as an upper bound that is not asymptotically tight.

Chapter 1

Introduction

> The Dark Side of the Force is a pathway to many abilities some consider to be unnatural.
>
> George Lucas, Revenge of the Sith

Knowledge of the potential energy surface* (PES) and the ability to use this knowledge grant extraordinary powers of prediction about the structure, dynamics and thermodynamics of any molecular system [9]. A potential, also known as a force field, is used to formally specify the PES in theoretical studies.

1.1 The Force Field

A general force field can be written as a series of terms representing the interactions between increasingly large sets of atoms [34, 35]:

$$V = \epsilon^{(0)} + \sum_{\alpha}^{N} \epsilon_{\alpha}^{(1)} + \sum_{\alpha}^{N} \sum_{\beta<\alpha}^{N} \epsilon_{\alpha,\beta}^{(2)} + \sum_{\alpha}^{N} \sum_{\beta<\alpha}^{N} \sum_{\gamma<\beta}^{N} \epsilon_{\alpha,\beta,\gamma}^{(3)} + \cdots , \tag{1.1}$$

where N is the total number of atoms, and the two-body term $\epsilon_{\alpha,\beta}^{(2)}$, for instance, describes the interaction of two atoms α and β.

Three-body and higher order terms in Equation 1.1 are often neglected, such as, for

*In this thesis terms 'potential energy landscape' and PES are used synonymously. At the time of writing searching for "potential energy surface" and "potential energy landscape" in Google® yielded 259,000 and 17,800 hits, respectively.

example, in the Lennard-Jones (LJ) pair potential [9, 36], which takes the form

$$V = 4\epsilon \sum_{\beta<\alpha}^{N} \left[\left(\frac{\sigma}{r_{\alpha,\beta}} \right)^{12} - \left(\frac{\sigma}{r_{\alpha,\beta}} \right)^{6} \right], \tag{1.2}$$

where $r_{\alpha,\beta}$ is the distance between atoms α and β, ϵ is the depth of the potential energy well, and $2^{1/6}\sigma$ is the pair equilibrium separation. This is an approximate potential as its form is a trade-off between the accurate reproduction of the interaction between closed-shell atoms and mathematical and computational simplicity. In this thesis we will use it to describe atomic clusters of various sizes.

1.2 Creating a Coarse-grained Model

It is often possible to gain new insight into the properties of a molecular system by expressing them in terms of stationary points of the PES, i.e. points where the gradient of the potential vanishes [9, 37]. Such a coarse-grained picture may be appropriate if the system spends most of its time in the vicinity of these points and the properties of interest can be expressed in terms of the properties of these points only. In realistic applications it may also be the only way forward, as the corresponding PES's are usually complex.

The most important stationary points are minima and the transition states that connect them. Here we define a minimum as a stationary point where the Hessian, the second derivative matrix, has no negative eigenvalues, while a transition state is a stationary point with precisely one such eigenvalue [38].

The number of stationary points on the PES generally scales exponentially with system size [39–43], which necessitates an appropriate sampling strategy of some sort for larger systems. In particular, to analyse kinetic properties a representative sample is usually obtained, which generally involves extensive use of single-ended and double-ended transition state searching techniques [7, 9, 30].

Locating transition states on a PES also provides an important tool in the study of dynamics using statistical rate theories [44–48]. Unfortunately, it is significantly harder to locate transition states than local minima, since the system must effectively 'balance on a knife-edge' in one degree of freedom. Many algorithms have been suggested for this purpose, and the most efficient method may depend upon the nature of the system. For

example, different considerations probably apply if second derivatives can be calculated relatively quickly, as for many empirical potentials [9]. Transformation to an alternative coordinate system may also be beneficial for systems bound by strongly directional forces [49–57].

Single-ended transition state searches [12–23, 54, 58–86] only require an initial starting geometry. The result of a single-ended search may be a transition state that is not connected to the starting point by a steepest-descent path, and such methods can be useful for building up databases of stationary points to provide a non-local picture of the potential energy surface, including thermodynamic and dynamic properties [9, 86]. However, double-ended searches [27–29, 53, 87–107] require two endpoint geometries, a mechanism to generate a set of configurations between them, and a suitable functional (or gradient) to be evaluated and minimised. The most successful single- and double-ended methods currently appear to be based upon hybrid eigenvector-following [12–24] and the nudged elastic band approach [7, 27, 29, 30, 108–110], respectively. The two search types are often used together, since double-ended transition state searches do not produce a tightly converged transition state and further refinement may be needed [7, 9].

1.3 Working with a Coarse-grained Model

In Chapter 2 and Chapter 4 of this thesis coarse-grained models of a PES are discussed in graph-theoretical terms. Nowadays a flourishing branch of mathematics and computer science, graph theory arguably started in the year of 1736 with Leonhard Euler's paper on the seven bridges of Königsberg, where he abstracted from landmasses and bridges to highlight the connectivity, and proved that it is impossible to cross every bridge exactly once in a single walk that starts and ends at the same point.

A graph* is defined as a set of nodes with connections between them called edges [111]. A coarse-grained picture of a PES therefore naturally fits into this definition, with nodes representing minima and edges representing the transition states that connect them. This approach was adopted in a number of previous energy landscapes studies, examples being the characterisation of dynamics in a region of a PES [8, 112–114] and detailed topological analyses of semi-complete PES samples [115].

*In modern literature the term 'network' is often used synonymously with the term 'graph'.

A directed edge is defined by its origin and destination nodes and can be travelled in one direction only. A graph composed of directed edges is termed a directed graph (digraph). Although every transition state facilitates both forward and reverse reactions, these are usually inequivalent, which leads to a symmetric digraph representation of a PES, i.e. to a digraph that is composed of pairs of complementary edges that share the same endpoints. Studies aimed at elucidating the global topology of a PES usually do not make such a distinction and deal with undirected graphs.

Interesting properties of the PES's of small Lennard-Jones clusters were recently discovered by Doye and Massen in a study of the corresponding undirected graph representations [115]. It appears that such graphs have features similar to both small-world and scale-free graphs. (A graph is said to have a small-world property if most pairs of nodes are connected by relatively short paths [111, 116, 117]. Such a graph can be easily obtained via a random rewiring of regular grid or lattice. The degree of a node is defined as the total number of its neighbours, and a graph is termed scale-free if its distribution of degrees follows a power law [111, 116].) While scale-free graphs are usually obtained via preferential attachment during growth [118], the origin of scale-free topology in graphs corresponding to PES's may lie in an Apollonian-like [119] packing of the basins of attraction [115]. Following the 'inherent structure' PES partitioning due to Stillinger and Weber [41, 42], a basin of attraction of a minimum can be defined as a set of points in configuration space connected to that minimum via steepest-descent paths. Topological studies of PES's are important because they can provide further insights into the PES connectivity and, consequently, increase our understanding of the relationship between the structural organisation of the PES and the observed physical and chemical properties.

Describing physical phenomena such as, for example, Brownian motion [120] and diffusion [121], requires more sophisticated graph models that allow for different types of edges. In edge-weighted graphs a label (weight or cost) is associated with every edge. Such graphs will be used in this thesis for various purposes. For example, an important part of the path-finding method described in Chapter 2 is the undirected edge-weighted graph representation, where every node is connected to every other node via an edge with a weight that is a function of Euclidean distance. In Appendix E we explain how

a weighted graph approach can be used to identify the fastest reaction pathways.

Both Brownian motion and diffusion were extensively studied in the past with the help of stochastic processes such as the random walk [121]. In mathematics and physics, a random walk is a formalisation of the intuitive idea of taking successive steps, each in a random direction [33]. If the number of possible directions is predefined and finite, a random walk is very easy to realize on an edge-weighted graph. A walk on a graph is defined as a sequence of edges such that the second node of each edge (except for the last edge) is the first node of the next edge. If the weights of the outgoing edges for every graph node add up to unity the graph is called probabilistic. A single step of a random walk confined to a probabilistic graph constitutes choosing the next graph node from the neighbours of the current node with a probability that equals the weight of the corresponding edge.

In Chapter 4 we will use random walks to model unimolecular chemical reactions [122]. An elementary reaction pathway is described as a transition from one state to a neighbouring state via a single transition state. Valid predictions of the sequence of such steps, known as the reaction mechanism, and a time scale associated with it are the holy grails of modern theoretical chemistry. Node-weighted probabilistic graphs are needed to address the time scale issue, because the time spent by a system before the transition occurs is likely to vary from state to state. An alternative description based on a probabilistic graph where each node is allowed to have a connection to itself is a more general approach to achieve this objective [123]. Both these models are employed in Chapter 4 in calculations of the average time for the reaction known as the mean first passage time.

The aforementioned graph model with self-connections is known by scientists in the fields of probability and stochastic processes as a discrete-time Markov chain — a stochastic process with a discrete state space [123]. The question of the first moment of the distribution of first passage times in stochastic processes is one of the most basic ones and is over two hundred years old. The first mean first passage time calculation was probably done by Jacob Bernoulli in the beginning of eighteenth century. In his groundbreaking work on probability titled ‘The art of conjecture’, which was posthumously published by his nephew Nicholas Bernoulli in 1713, he describes the techniques for

calculating the duration of various games of chance [124]. A number of methods for efficient calculation of the mean first passage times was developed since then, some of which are yet to be fully utilised in chemical applications.

In this thesis I will attempt to address two important stages in an energy-landscapes-based approach to the analysis of chemical reactivity, namely, finding reaction (or rearrangement) pathways and extracting the kinetic information from the obtained pathway ensemble. As the meaning of the term pathway (or path) varies from chapter to chapter I will spend some time in the introductory sections clarifying the terminology.

1.4 Thesis Overview

The results of the work that I have carried out are presented in three chapters.

The main focus of Chapter 2 is on double-ended methods for finding transition states. A detailed review of one of the leading methods from that class is followed by the discussion of our modifications and improvements that allowed us to extend its applicability. Results for a model two-dimensional surface and Lennard-Jones clusters of several sizes are presented. The chapter culminates with an application to finding folding paths for a family of small peptides known as tryptophan zippers.

Chapter 3 is devoted to discussion of two exciting properties of rearrangement pathways — cooperativity and localisation. A new measure of cooperativity suitable for applications to atomic rearrangements is introduced and subsequently used to establish the links between cooperativity of a single-step rearrangement, the energy barrier height and the difficulty of locating the corresponding transition state with both single-ended and double-ended methods.

In Chapter 4 we deal with compact representations of large pathway ensembles borrowing ideas from graph theory and the theory of random processes. The main theme is the development of faster methods for calculation of mean escape times for graphs of increasing complexity. We devise a number of approaches for extracting this kinetic information and compare them to well-established techniques such as kinetic Monte Carlo and discrete path sampling.

Chapter 5 summarises the achievements of the work described in this thesis and suggests the directions for future research.

Chapter 2

Finding Rearrangement Pathways

> Remember when life's path is steep
> to keep your mind even.
>
> Horace (65 BC - 8 BC)

2.1 Introduction

Our principal concern in this chapter is the development of the double-ended NEB approach [27–30]. The earliest double-ended methods were probably the linear and quadratic synchronous transit algorithms (LST and QST) [125], which are entirely based on interpolation between the two endpoints. In LST the highest energy structure is located along the straight line that links the two endpoints. QST is similar in spirit, but approximates the reaction path using a parabola instead of a straight line. Neither interpolation is likely to provide a good estimate of the path except for very simple reactions, but they may nevertheless be useful to generate initial guesses for more sophisticated double-ended methods.

Another approach is to reduce the distance between reactant and product by some arbitrary value to generate an 'intermediate', and seek the minimum energy of this intermediate structure subject to certain constraints, such as fixed distance to an endpoint. This is the basis of the 'Saddle' optimisation method [126] and the 'Line Then

Plane' [127] algorithm, which differ only in the definition of the subspace in which the intermediate is allowed to move. The latter method optimises the intermediate in the hyperplane perpendicular to the interpolation line, while 'Saddle' uses hyperspheres. The minimised intermediate then replaces one of the endpoints and the process is repeated.

There are also a number of methods that are based on a 'chain-of-states' (CS) approach, where several images of the system are somehow coupled together to create an approximation to the required path. The CS methods mainly differ in the way in which the initial guess to the path is refined. In the 'Chain' method [128] the geometry of the highest energy image is relaxed first using only the component of the gradient perpendicular to the line connecting its two neighbours. The process is then repeated for the next-highest energy neighbours. The optimisation is terminated when the gradient becomes tangential to the path. The 'Locally Updated Planes' method [129] is similar, but the images are relaxed in the hyperplane perpendicular to the reaction coordinate, rather than along the line defined by the gradient, and all the images are moved simultaneously.

The NEB approach introduced some further refinements to these CS methods [30]. It is based on a discretised representation of the path originally proposed by Elber and Karplus [88], with modifications to eliminate corner-cutting and sliding-down problems [27], and to improve the stability and convergence properties [29]. Maragakis *et al.* applied the NEB method to various physical systems ranging from semiconductor materials to biologically relevant molecules. They report that use of powerful minimisation methods in conjunction with the NEB approach was unsuccessful [107]. These problems were attributed to instabilities with respect to the extra parameters introduced by the springs.

The main result of the present contribution is a modified 'doubly nudged' elastic band (DNEB) method, which is stable when combined with the L-BFGS minimiser. In comparing the DNEB approach with other methods we have also analysed quenched velocity Verlet minimisation, and determined the best point at which to remove the kinetic energy. Extensive tests show that the DNEB/L-BFGS combination provides a significant performance improvement over previous implementations. We therefore

outline a new strategy to connect distant minima, which is based on successive DNEB searches to provide transition state candidates for refinement by eigenvector-following.

2.2 A Double-ended Method: Nudged Elastic Band

In the present work we used the nudged elastic band [27, 28] (NEB) and eigenvector-following [12–19, 22–24] (EF) methods for locating and refining transition states. In the NEB approach the path is represented as a set of images $\{\mathbf{X}_1, \mathbf{X}_2 ... \mathbf{X}_{N_i}\}$ that connect the endpoints $\mathbf{X}_0$ and $\mathbf{X}_{N_i+1}$, where $\mathbf{X}_i$ is a vector containing the coordinates of image i (Figure 2.1) [29]. In the usual framework of double-ended methodologies [130] the endpoints are stationary points on the PES (usually minima), which are known in advance. In addition to the true potential, V_i, which binds the atoms within each image, equivalent atoms in N_i adjacent images are interconnected by $N_i + 1$ springs according to a parabolic potential,

$$\widetilde{V} = \tfrac{1}{2} k_{spr} \sum_{i=1}^{N_i+1} |\mathbf{X}_i - \mathbf{X}_{i-1}|^2. \tag{2.1}$$

Subsequently these potentials will be referred to as the 'true potential' and the 'spring potential', respectively.

The springs are intended to hold images on the path during optimisation — otherwise they would slide down to the endpoints, or to other intermediate minima [88]. Occasionally, depending on the quality of the initial guess, we have found that some images may converge to higher index stationary points. One could imagine the whole construction as a band or rope that is stretched across the PES, which, if optimised, is capable of closely following a curve defined in terms of successive minima, transition states, and the intervening steepest-descent paths.

In practice, the above formulation encounters difficulties connected with the coupling between the 'true' and 'spring' components of the potential. The magnitude of the springs' interference with the true potential is system dependent and generally gives rise to corner-cutting and sliding-down problems [27]. It is convenient to discuss these difficulties in terms of the components of the true gradient, $\mathbf{g}$, and spring gradient, $\widetilde{\mathbf{g}}$, parallel and perpendicular to the path. The parallel component of the gradient $\mathbf{g}^{\parallel}$ at image i on the path is obtained by projecting out the perpendicular component $\mathbf{g}^{\perp}$ us-

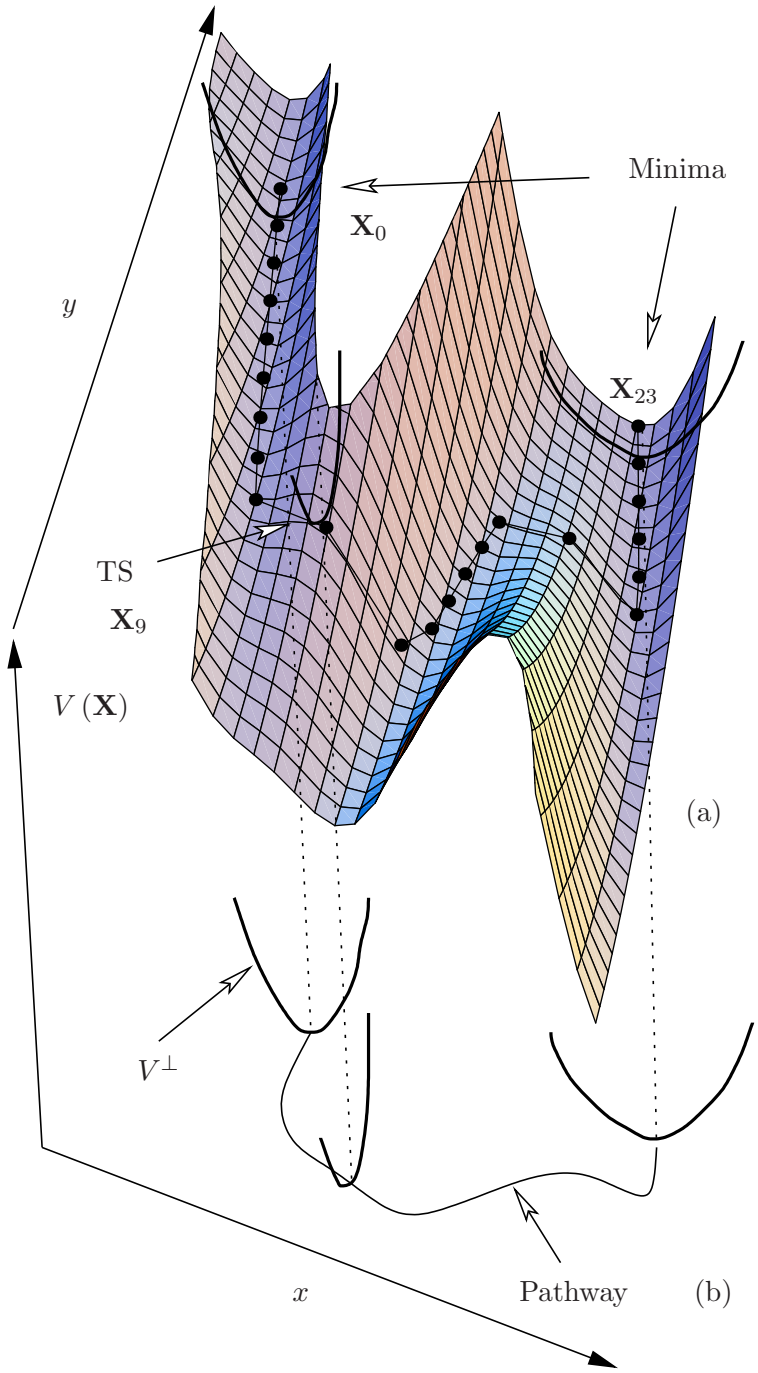


Figure 2.1: Graphical representation of the nudged elastic band approach. (a) The optimised nudged elastic band for a two-dimensional model surface. The band contains 21 images and connects two minima $\mathbf{X}_0$ and $\mathbf{X}_{23}$. Image $\mathbf{X}_9$ has the highest energy and might therefore be used to estimate transition state properties or as a starting guess for further refinement. (b) 'Nudging': the NEB depicted in (a) is projected onto the xy plane and feels only the perpendicular component of the true gradient from the effective potential $V^{\perp}$.

ing an estimate of the tangent to the path. The parallel and perpendicular components for image i are:

$$\mathbf{g}_i^{\parallel} = (\bigtriangledown_i V_i, \hat{\boldsymbol{\tau}}_i)\,\hat{\boldsymbol{\tau}}_i, \qquad \mathbf{g}_i^{\perp} = \bigtriangledown_i V_i - \mathbf{g}_i^{\parallel}, \tag{2.2}$$

where $V_i = V(\mathbf{X}_i)$, the unit vector $\hat{\boldsymbol{\tau}}_i$ is the tangent, and $(\bigtriangledown_i V_i, \hat{\boldsymbol{\tau}}_i)$ denotes the scalar product of vectors $\bigtriangledown_i V_i$ and $\hat{\boldsymbol{\tau}}_i$. Here and throughout this work we denote unit vectors by a hat. The complete gradient, $\mathbf{g}$, has $N_i \times \eta$ components for a band of N_i images with η atomic degrees of freedom each.

Corner-cutting has a significant effect when a path experiences high curvature. Here $\widetilde{\mathbf{g}}^{\perp}$ is large, which prevents the images from following the path closely because the spring force necessarily has a significant component perpendicular to the tangent. The sliding-down problem occurs due to the presence of $\mathbf{g}^{\parallel}$, which perturbs the distribution of images along the path, creating high-resolution regions (around the local minima) and low-resolution regions (near the transition states) [27]. Both problems significantly affect the ability of the NEB method to produce good transition state candidates. We have found that sliding-down and corner-cutting are interdependent and cannot both be remedied by adjusting the spring force constant k_{spr}; increasing k_{spr} may prevent sliding-down but it will make corner-cutting worse.

The aforementioned problems can sometimes be eliminated by constructing the NEB gradient from the potential in the following way: $\mathbf{g}^{\parallel}$ and $\widetilde{\mathbf{g}}^{\perp}$ are projected out, which gives the elastic band its 'nudged' property [28]. Removal of $\mathbf{g}^{\parallel}$ can be thought of as bringing the path into a plane or flattening the PES [Figure 2.1 (b)], while removal of $\widetilde{\mathbf{g}}^{\perp}$ is analogous to making the images heavier so that they favour the bottom of the valley at all times.

The choice of a method to estimate the tangent to the path is important for it affects the convergence of the NEB calculation. Originally, the tangent vector, $\hat{\boldsymbol{\tau}}_i$, for image i was obtained by normalising the line segment between the two adjacent images, $i+1$ and $i-1$ [27]:

$$\hat{\boldsymbol{\tau}}_i = \frac{\mathbf{X}_{i+1} - \mathbf{X}_{i-1}}{|\mathbf{X}_{i+1} - \mathbf{X}_{i-1}|}. \tag{2.3}$$

However, kinks can develop during optimisation of the image chain using this definition of $\hat{\boldsymbol{\tau}}_i$. It has been shown [29] that kinks are likely to appear in the regions where the ratio $g_i^{\parallel}/g_j^{\perp}$ is larger than the length of the line segment, $|\boldsymbol{\tau}|$, used in estimating the

tangent [Figure 2.2 (a)].

Both the above ratio, the image density and $|\hat{\boldsymbol{\tau}}|$ can vary depending on the system of interest, the particular pathway and other parameters of the NEB calculation. From Equation 2.3 it can be seen that $\hat{\boldsymbol{\tau}}_i$, and, hence, the next step in the optimisation of image i, is determined by its neighbours, which are not necessarily closer to the path than image i. Therefore, a better approach in estimating the $\hat{\boldsymbol{\tau}}_i$ would be to use only one neighbour, since then we only need this neighbour to be better converged than image i.

There are two neighbours to select from, and it is natural to use the higher-energy one for this purpose, since steepest-descent paths are easier to follow downhill than uphill:

$$\hat{\boldsymbol{\tau}}_i = \frac{(j-i)\,(\mathbf{X}_j - \mathbf{X}_i)}{|\mathbf{X}_j - \mathbf{X}_i|}, \tag{2.4}$$

where i and j are two adjacent images with energies V_i and V_j, and $V_i < V_j$. In this way, an image i that has one higher-energy neighbour j behaves as if it is 'hanging' on to it [Figure 2.2(b)].

The above tangent formulation requires special handling of extrema along the path, and a mechanism for switching $\hat{\boldsymbol{\tau}}$ at such points was proposed [29]. It also fails to produce an even distribution of images in regions with high curvature [Figure 2.2 (c)]. We presume that Henkelman and Jónsson substitute $(\widetilde{\mathbf{g}}, \hat{\boldsymbol{\tau}})\,\hat{\boldsymbol{\tau}}$ by $|\widetilde{\mathbf{g}}|\hat{\boldsymbol{\tau}}$ in Equation 2.2 to obtain a spring gradient formulation that will keep the images equispaced when the tangent from Equation 2.4 is used in the projections [28]:

$$\widetilde{\mathbf{g}}_i^{\parallel} = k_{spr}\Big(|\mathbf{X}_i - \mathbf{X}_{i-1}| - |\mathbf{X}_{i+1} - \mathbf{X}_i|\Big)\hat{\boldsymbol{\tau}}_i. \tag{2.5}$$

2.3 Optimisation of the Nudged Elastic Band

In the present work the NEB approach has been used in combination with two minimisers, namely the quenched velocity Verlet (QVV) and the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithms. It is noteworthy that the objective function corresponding to the projected NEB gradient is unknown, but it is not actually required in either of the minimisation algorithms that we consider.

Optimisation is a general term that refers to finding stationary points of a function. Many problems in computational chemistry can be formulated as optimisation of a

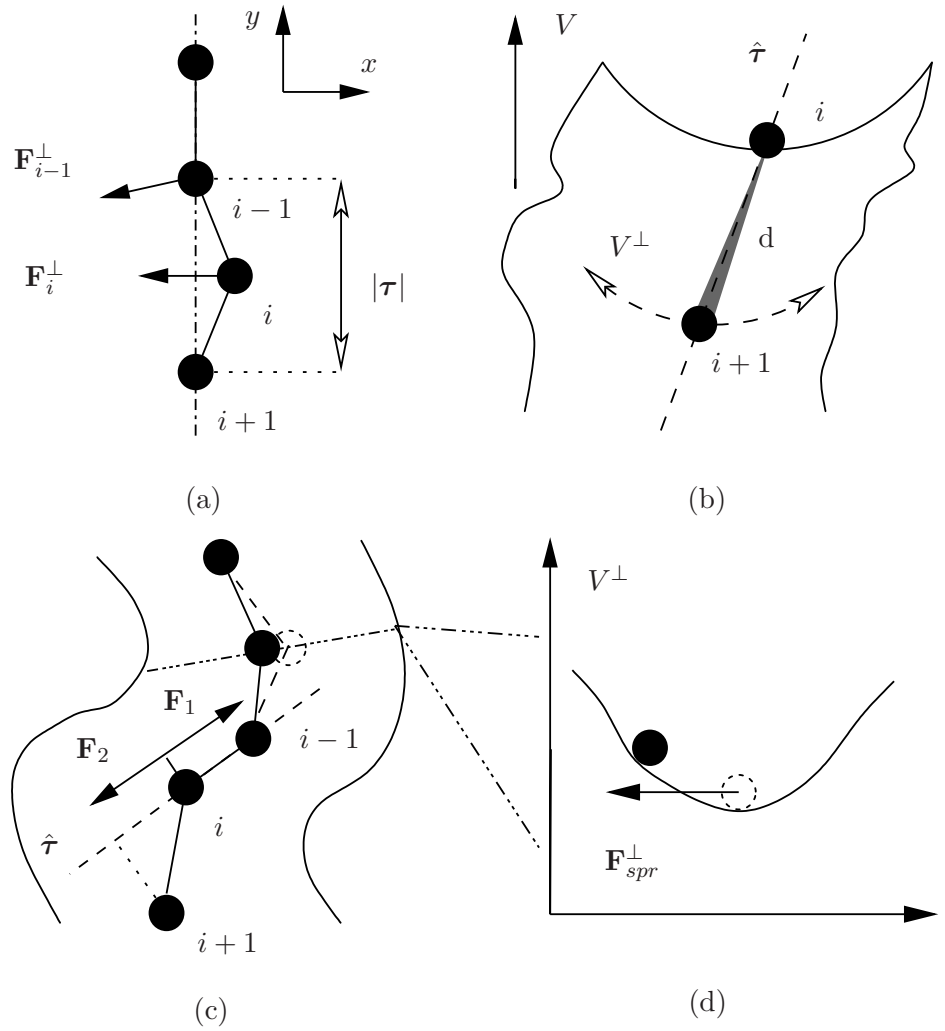


Figure 2.2: Details of recent NEB implementations. (a) Conditions under which kinks appear during optimisation of the NEB using the tangent estimated from the line segment $\hat{\boldsymbol{\tau}}$ connecting images $i+1$ and $i-1$. Displacement of image i from the path (dash-dotted line) creates forces $\mathbf{F}^{\perp}_{i-1} = -\mathbf{g}^{\perp}_{i-1}$ and $\mathbf{F}^{\perp}_{i} = -\mathbf{g}^{\perp}_{i}$. While $\mathbf{F}^{\perp}_{i}$ is a restoring force that originates from $V^{\perp}$, $\mathbf{F}^{\perp}_{i-1}$ is destabilising and originates from $V^{\parallel}$ (and is non-zero due to the fact that the tangent at image $i-1$ has changed after displacement of image i). For the case of small displacements the potential may be resolved into two contributions, $V^{\perp} = k^{\perp}x^2/2$ and $V^{\parallel} = -k^{\parallel}y$, and kinks will not appear if $k^{\parallel}/k^{\perp} < |\hat{\boldsymbol{\tau}}|$. (b) Tangent estimate using the higher energy neighbour: image $i+1$ is 'hanging' on to image i. The separation d is controlled by the lower-lying images ($> i+1$) but not V. (c) An NEB that follows the curved region of the path: since the spring force $\mathbf{F}_1$ acting on image i is compensated by projection $\mathbf{F}_2$, the distribution of images becomes uneven. (d) Corner-cutting displayed on a cross-section of the curved part of the path depicted in (c): the image is displaced from the path due to the presence of $\mathbf{F}^{\perp}_{spr}$.

multidimensional function, NEB optimisation being one of the many. The goal of an optimisation problem can be formulated as follows: find a combination of parameters (independent variables) that optimise a given quantity (the objective function), possibly with some restrictions on the allowed parameter ranges [131]. To know exactly what we are looking for optimality conditions need to be defined. The condition

$$f(\mathbf{x}^*) < f(\mathbf{y}) \qquad \forall \mathbf{y} \in \mho(\mathbf{x}), \mathbf{y} \neq \mathbf{x}^*, \tag{2.6}$$

where $\mho(\mathbf{x})$ is the set of all possible values of the control variable $\mathbf{x} = (x_1, x_2, ...x_n)^T$, defines the global optimum $\mathbf{x}^*$ of the function $f(\mathbf{x})$. For an unconstrained problem $\mho(\mathbf{x})$ is infinitely large, and finding the corresponding global minimum may be a difficult task. Similarly, a point $\mathbf{x}^*$ is a strong local minimum of $f(\mathbf{x})$ if

$$f(\mathbf{x}^*) < f(\mathbf{y}) \qquad \forall \mathbf{y} \in \Upsilon(\mathbf{x}^*, \varepsilon), \mathbf{y} \neq \mathbf{x}^*, \tag{2.7}$$

where $\Upsilon(\mathbf{x}^*, \varepsilon)$ is a set of feasible points contained in the neighbourhood ε of $\mathbf{x}^*$. For a weak local minimum only an inequality $f(\mathbf{x}^*) \leqslant f(\mathbf{y})$ must be satisfied in Equation 2.7.

More easily identified optimality conditions could be used instead of Equation 2.6 and Equation 2.7 if $f(\mathbf{x})$ is a function with continuous first and second derivatives, namely, the stationary point and the strong local minimum conditions. A stationary point is a point where the gradient vanishes:

$$g_i(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_i} = 0, \tag{2.8}$$

and a strong local minimum is a stationary point where the Hessian matrix

$$[\mathbf{H}(\mathbf{x})]_{ij} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \tag{2.9}$$

is positive-definite:

$$\mathbf{z}^T \mathbf{H}(\mathbf{x}) \mathbf{z} > 0 \qquad \forall \mathbf{z} \neq \mathbf{0}. \tag{2.10}$$

Formally, optimisation of an NEB qualifies as a nonlinear unconstrained continuous multivariable optimisation problem. There are many algorithms available for solving problems of this type that differ in their computational requirements, convergence and other properties. However, NEB optimisation is augmented with an additional difficulty: due to the projections involved the objective function that is being minimised

is unknown. Nevertheless, several optimisation techniques were found to be stable in working with NEB. Most of these are based on steepest-descent methods and, consequently, are not very efficient. The L-BFGS and QVV minimisers discussed in this section are based on the deterministic approach for finding local minima.

The basic structure of any local minimiser can be summarised as follows:

- guess initial geometry $\mathbf{x}_0$;
- for i = 1,2,... until convergence criterion is satisfied:
    1. test if convergence is achieved for current geometry $\mathbf{x}_i$
    2. construct the search direction $\hat{\mathbf{p}}_i$
    3. determine the step length ϖ_i
    4. update the geometry to $\mathbf{x}_i + \varpi_i \hat{\mathbf{p}}_i$

The descent direction $\hat{\mathbf{p}}$ is defined as the one along which the directional derivative is negative:

$$\left(\mathbf{g}\left(\mathbf{x}\right), \hat{\mathbf{p}}\right) < 0. \tag{2.11}$$

Negativity of the left hand side guarantees that a lower function value will be found along $\hat{\mathbf{p}}$ provided that the step is sufficiently small. According to the way algorithms choose the search direction they are classified as non-derivative, gradient and second-derivative. Steepest-descent is an example of a gradient-based method, while BFGS belongs to the class of quasi-Newton methods (or quasi-second-derivative methods). The steepest-descent search direction is defined as the direction antiparallel to the gradient at the current point

$$\hat{\mathbf{p}} = -\hat{\mathbf{g}}, \tag{2.12}$$

whereas second-derivative-based methods determine the search direction using the Newton-Raphson equation [31]:

$$\mathbf{p} = -\mathbf{H}^{-1}\mathbf{g}. \tag{2.13}$$

The step length ϖ can be determined using either 'line search' or 'trust radius' approaches. Line search is essentially a one-dimensional minimisation of the function $\widetilde{f}(\varpi) = f(\mathbf{x}_i + \varpi \mathbf{p}_i)$, which is performed at every step [131]. In the trust-radius-based approach the step length is adjusted dynamically depending on how well the

minimisation algorithm predicts the change in the object function value [9]. There is no clear evidence for the superiority of one method over the other [131]; however, in combination with the limited-memory version of BFGS algorithm the trust ratio approach was found to be more efficient for treating problems with discontinuities such as problems involving periodic boundary conditions with cutoffs [9].

2.3.1 Quenched Velocity Verlet Minimiser

The QVV method is based on the velocity Verlet algorithm [32] (VV) as modified by Jónsson *et al.* [27], and was originally used for NEB optimisation. VV is a symplectic integrator that enjoys widespread popularity, primarily in molecular dynamics (MD) simulations where it is used for numerical integration of Newton's equations of motion. Its main advantage over the standard Verlet method is the minimisation of the round-off errors. At each time step δt the coordinates and the velocities $\boldsymbol{\mathcal{V}}$ are updated from the coupled first-order differential equations in the following manner [32]:

$$\mathbf{X}\left(t+\delta t\right)=\mathbf{X}\left(t\right)+\delta t\boldsymbol{\mathcal{V}}\left(t\right)-\frac{\delta t^{2}}{2m}\mathbf{g}\left(t\right), \tag{2.14}$$

$$\boldsymbol{\mathcal{V}}\left(t+\frac{1}{2}\delta t\right)=\boldsymbol{\mathcal{V}}\left(t\right)-\frac{\delta t}{2m}\mathbf{g}\left(t\right), \tag{2.15}$$

$$\boldsymbol{\mathcal{V}}\left(t+\delta t\right)=\boldsymbol{\mathcal{V}}\left(t+\frac{1}{2}\delta t\right)-\frac{\delta t}{2m}\mathbf{g}\left(t+\delta t\right), \tag{2.16}$$

where m is the atomic mass. The algorithm involves two stages, with a force evaluation in between. First the positions are updated according to Equation 2.14, and the velocities at midstep $t+\delta t/2$ are then computed using Equation 2.15. After the evaluation of the gradient at time $t+\delta t$ the velocity is updated again [Equation 2.16] to complete the move. To obtain minimisation it is necessary to remove kinetic energy, and this can be done in several ways. If the kinetic energy is removed completely every step the algorithm is equivalent to a steepest-descent minimisation, which is rather inefficient. Instead, it was proposed by Jónsson *et al.* [27] to keep only the velocity component that is antiparallel to the gradient at the current step. If the force is consistently pointing in the same direction the system accelerates, which is equivalent to increasing the time step [27]. However, a straightforward variable time step version of the above algorithm was reported to be unsuccessful [110].

2.3.2 L-BFGS Minimiser

The BFGS algorithm belongs to the class of variable metric methods and is slightly different from the Davidon-Fletcher-Powell (DFP) method in the way in which the correction term is constructed [132] (see below). It has become generally recognised that the BFGS algorithm is superior to DFP in convergence tolerances, roundoff error and other empirical issues [31]. The idea of the variable metric, or quasi-Newton, method is to build up a good approximation to the inverse Hessian matrix $\mathbf{H}^{-1}$ by constructing a sequence of matrices $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \dots\}$ with the following property:

$$\lim_{i\to\infty} \mathbf{A}_i = \mathbf{H}^{-1}. \tag{2.17}$$

If we are successful in doing so, combining two Newton-Raphson equations (see Equation 2.13) for consecutive iterations i and $i+1$, we find that $\mathbf{A}_{i+1}$ should satisfy

$$\mathbf{x}_{i+1} - \mathbf{x}_i = \varpi_i \hat{\mathbf{p}}_i = -\mathbf{A}_{i+1}\left(\mathbf{g}_{i+1} - \mathbf{g}_i\right), \tag{2.18}$$

where we have used the property $\mathbf{A}_{i+1} = \mathbf{A}_i$, which must hold if both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are in the neighbourhood of minimum $\mathbf{x}^*$.

At every iteration i the new approximation to the inverse Hessian $\mathbf{A}_{i+1}$ should be constructed in order to calculate the next step. The update formula must be consistent with Equation 2.18 and could take the form

$$\mathbf{A}_{i+1} = \mathbf{A}_i + \widetilde{\mathbf{A}}, \tag{2.19}$$

where the correction term $\widetilde{\mathbf{A}}$ is constructed using the gradient difference $\delta\mathbf{g} = \mathbf{g}_{i+1} - \mathbf{g}_i$, the step $\delta\mathbf{x} = \mathbf{x}_{i+1} - \mathbf{x}_i$, and the inverse Hessian matrix from the previous iteration $\mathbf{A}_i$. The BFGS update correction is [31]

$$\widetilde{\mathbf{A}} = \frac{\delta\mathbf{x}\otimes\delta\mathbf{x}}{(\delta\mathbf{x},\delta\mathbf{g})} - \frac{\mathbf{u}\otimes\mathbf{u}}{(\delta\mathbf{g},\mathbf{u})} + (\delta\mathbf{g},\mathbf{u})\left[\left(\frac{\delta\mathbf{x}}{(\delta\mathbf{x},\delta\mathbf{g})} - \frac{\mathbf{u}}{(\delta\mathbf{g},\mathbf{u})}\right)\otimes\left(\frac{\delta\mathbf{x}}{(\delta\mathbf{x},\delta\mathbf{g})} - \frac{\mathbf{u}}{(\delta\mathbf{g},\mathbf{u})}\right)\right], \tag{2.20}$$

where $\otimes$ denotes the direct product of two vectors, and $\mathbf{u} = \mathbf{H}_i\delta\mathbf{g}$.

Since at every iteration the old Hessian is overwritten with a new one $n^2/2 + n/2$ storage locations are needed. This is an entirely trivial disadvantage over the conjugate gradient methods for any modest value of n [31]. However, for large-scale problems it is advantageous to be able to specify the amount of storage BFGS is allowed to use.

There is a version of the BFGS algorithm that allows us to vary the amount of storage available to BFGS and is particularly suitable for large-scale problems [2]. The difference between this limited memory BFGS (L-BFGS) algorithm and the standard BFGS is in the Hessian matrix update. L-BFGS stores each correction separately and the Hessian matrix is never formed explicitly. To store each correction to the Hessian $2n$ storage locations are needed [1]. The user can specify the number of corrections L-BFGS is allowed to store. Every iteration $\mathbf{Ag}$ is computed according to a recursive formula described by Nocedal [1]. For the first m iterations L-BFGS is identical to the BFGS method. After the first m iterations the oldest correction is discarded. Thus, by varying m the iteration cost can also be controlled, which makes the method very efficient and flexible. In general, because only m recent corrections are stored L-BFGS is better able to use additional storage to accelerate convergence. Here we employed a modified version of Nocedal's L-BFGS implementation [3] in which the line search was removed and the maximum step size was limited for each image separately.

2.4 A Single-ended Method: Eigenvector-following

Single-ended methods use only the function and its derivatives to search for a transition state from the starting point (the initial guess). Since a transition state is a local maximum in one direction but a minimum in all the others it is not generally possible to use standard minimisation methods for this purpose.

Newton-type optimisation methods are based on approximating the objective function locally by a quadratic model and then minimising that function approximately using, for example, the Newton-Raphson (NR) approach [31]. However, the NR algorithm can converge to a stationary point of any index [130]. To converge to a transition state the algorithm may need to start from a geometry at which the Hessian has exactly one negative eigenvalue and the corresponding eigenvector is at least roughly parallel to the reaction coordinate. Locating such a point can be a difficult task and several methods have been developed that help to increase the basins of attraction of transition states, which gives more freedom in choosing the starting geometry [9, 19, 22, 24].

The most widely used single-ended transition state search method is eigenvector-following (EF). In it simplest form it requires diagonalisation of the Hessian matrix [12–

15]. For larger systems it is advantageous to use hybrid EF methods that avoid either calculating the Hessian or diagonalising it [8, 22, 23, 133].

We start by reviewing the theory behind the Newton-Raphson methods. Solving the eigenvalue problem

$$\mathbf{HC} = \mathbf{C\Lambda} \tag{2.21}$$

yields the matrix $\mathbf{C}$, the columns of which are the eigenvectors $\mathbf{c}_i$, and the diagonal matrix $\mathbf{\Lambda}$, with eigenvalues λ_i on the diagonal. The matrix $\mathbf{C}$ defines a unitary transformation to a new orthogonal basis $\{\mathbf{c}_i\}$, in which the original Hessian matrix $\mathbf{H}$ is diagonal

$$\mathbf{C}^{-1}\mathbf{HC} = \mathbf{\Lambda}, \tag{2.22}$$

so that Equation 2.13 is simplified to

$$p'_i = -\frac{g'_i}{\lambda_i}, \tag{2.23}$$

where $\mathbf{g}'$ is the gradient vector in the new basis, which is related to the original gradient vector $\mathbf{g}$ as

$$\mathbf{g}' = \mathbf{C}^T\mathbf{g}, \tag{2.24}$$

because of the unitarity of $\mathbf{C}$

$$\mathbf{C}^{-1} = \mathbf{C}^T. \tag{2.25}$$

The unnormalised search direction $\mathbf{p}'$ is defined similarly.

After taking a step of length $\varpi = |\mathbf{p}|$ in the direction $\mathbf{p}$, as prescribed by Equation 2.23, the energy changes by

$$\Delta V' = -\frac{1}{2}\mathbf{g}'\mathbf{\Lambda}^{-1}\mathbf{g}' \tag{2.26}$$

provided our quadratic approximation to $V(\mathbf{X})$ is perfect. In general $\lambda_i \in \mathbb{R}$ and terms in Equation 2.26 with $\lambda_i < 0$ and $\lambda_i > 0$ increase and decrease the energy, respectively. For isolated molecules and bulk models with periodic boundary conditions the Hessian matrix is singular: there are three zero eigenvalues that correspond to overall translations. Non-linear isolated molecules have three additional zero eigenvalues due to rotations. Luckily, in all cases for which $\lambda_i = 0$ the analytic form of the corresponding eigenvectors $\mathbf{c}_i$ is known so the eigenvalue shifting procedure could be applied:

$$\mathbf{H}' = \mathbf{H} + \sum_i \widetilde{\lambda}_i \mathbf{c}_i \otimes \mathbf{c}_i^T, \tag{2.27}$$

where $\widetilde{\lambda}_i$ are the new eigenvalues. In this work all the zero eigenvalues were shifted by the same amount (10^6). As a result, $\det \mathbf{H}' \neq 0$ and problems with applying Equation 2.23 do not arise.

The analytic forms of the $\mathbf{c}_i$'s corresponding to translations along the x, y and z directions are

$$\hat{\mathbf{c}}_1 = \sqrt{\frac{1}{N}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \ \hat{\mathbf{c}}_2 = \sqrt{\frac{1}{N}} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}, \text{ and } \hat{\mathbf{c}}_3 = \sqrt{\frac{1}{N}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \end{pmatrix}, \tag{2.28}$$

respectively. The eigenvector that corresponds to rotation about the x axis by a small angle ϕ can be obtained as

$$\hat{\mathbf{c}}_4 = \widetilde{\mathbf{R}}_x \hat{\mathbf{x}} - \hat{\mathbf{x}}, \tag{2.29}$$

where $\hat{\mathbf{x}} = (x_1, y_1, z_1, x_2, \dots)^T$ is a normalised $3N$-dimensional vector describing the position before the rotation, and $\widetilde{\mathbf{R}}_x$ is Maclaurin series expansion of the $3N$-dimensional rotation matrix [134] $\mathbf{R}_x$ with respect to a small angle ϕ truncated to the second order. The displacement vectors due to infinitesimal rotation about the x, y and z axes are therefore

$$\hat{\mathbf{c}}_4 = \begin{pmatrix} 0 \\ -\frac{\phi^2}{2} y_1 + \phi z_1 \\ -\phi y_1 - \frac{\phi^2}{2} z_1 \\ 0 \\ -\frac{\phi^2}{2} y_2 + \phi z_2 \\ -\phi y_2 - \frac{\phi^2}{2} z_2 \\ \vdots \end{pmatrix}, \ \hat{\mathbf{c}}_5 = \begin{pmatrix} -\frac{\phi^2}{2} x_1 - \phi z_1 \\ 0 \\ \phi x_1 - \frac{\phi^2}{2} z_1 \\ -\frac{\phi^2}{2} x_2 - \phi z_2 \\ 0 \\ \phi x_2 - \frac{\phi^2}{2} z_2 \\ \vdots \end{pmatrix}, \text{ and } \hat{\mathbf{c}}_6 = \begin{pmatrix} -\frac{\phi^2}{2} x_1 + \phi y_1 \\ -\phi x_1 - \frac{\phi^2}{2} y_1 \\ 0 \\ -\frac{\phi^2}{2} x_2 + \phi y_2 \\ -\phi x_2 - \frac{\phi^2}{2} y_2 \\ 0 \\ \vdots \end{pmatrix}, \tag{2.30}$$

respectively.

If started from a point at which the Hessian matrix has n negative eigenvalues, the

NR method is expected to converge to a stationary point of index n. However, the step taking procedure may change the index as optimisation proceeds.

Introducing Lagrange multipliers allows a stationary point of specified index to be located. Consider a Lagrangian function

$$L = -V'(\mathbf{x}'_0) - \sum_{\alpha=1}^{3N} \left[g'_\alpha(\mathbf{x}'_0)\, p'_\alpha + \frac{1}{2}\lambda_\alpha\, p'^2_\alpha - \frac{1}{2}\mu_\alpha(p'^2_\alpha - c^2_\alpha) \right], \tag{2.31}$$

where a separate Lagrange multiplier μ_α is used for every eigendirection α, c_α is a constraint on the step p'_α, λ_α is the αth eigenvalue of $\mathbf{H}'$, which is defined in Equation 2.27, and $\mathbf{x}'_0$ is the point about which the potential energy function $V'(\mathbf{x})$ was expanded. Primes, as before, denote that components are expressed in the orthogonal basis. Differentiating Equation 2.31 with respect to $\mathbf{p}'$ and using the condition for a stationary point the optimal step can be obtained:

$$p'_\alpha = \frac{g'_\alpha(\mathbf{x}'_0)}{\mu_\alpha - \lambda_\alpha}. \tag{2.32}$$

The predicted energy change corresponding to this step is

$$\Delta V' = \sum_{\alpha=1}^{3N} \frac{(\mu_\alpha - \lambda_\alpha/2)}{(\mu_\alpha - \lambda_\alpha)^2} g'_\alpha(\mathbf{x}'_0)^2. \tag{2.33}$$

We have some freedom in choosing the Lagrange multipliers μ_α as long as for the eigendirections for which an uphill (downhill) step is to be taken $\mu_\alpha > \lambda_\alpha/2$ ($\mu_\alpha < \lambda_\alpha/2$). The following choice of Lagrange multipliers allows the recovery of the Newton-Raphson step in the vicinity of a stationary point [9]:

$$\mu_\alpha = \lambda_\alpha \pm \frac{1}{2}|\lambda_\alpha| \left(1 + \sqrt{1 + 4g'_\alpha(\mathbf{x}'_0)^2/\lambda^2_\alpha} \right), \tag{2.34}$$

with the plus sign for an uphill step, and the minus sign for a downhill step.

In cases when Hessian diagonalisation is impossible or undesirable the smallest eigenvalue and a corresponding eigenvector can be found using an iterative method [135].

2.5 RESULTS

The springs should distribute the images evenly along the NEB path during the optimisation, and the choice of k_{spr} must be made at the beginning of each run. It has been suggested by Jónsson and coworkers that since the action of the springs is only

felt along the path the value of the spring constant is not critical as long as it is not zero [27]. If k_{spr} is set to zero then sensible behaviour occurs for the first several tens of iterations only; even though $\mathbf{g}^{\parallel}$ is projected out, $\hat{\boldsymbol{\tau}}$ fluctuates and further optimisation will eventually result in the majority of images gradually sliding down to local minima [27].

2.5.1 Slow-response Quenched Velocity Verlet

In practice we find that the value of k_{spr} affects the convergence properties and the stability of the optimisation process. This result depends on the type of minimiser employed and may also depend on minimiser-specific settings. Here we analyse the convergence properties of NEB minimisations using the QVV minimiser (NEB/QVV) and their dependence on the type of velocity quenching. From previous work it is not clear when is the best time to perform quenching during the MD minimisation of the NEB [27–29]. Since the VV algorithm calculates velocities based on the gradients at both current and previous steps quenching could be applied using either of these gradients.

Specifically, it is possible to quench velocities right after advancing the system using Equation 2.14, at the half-step in the velocity evaluation (quenching intermediate velocities at time $t+\delta t/2$) using either the old or new gradient [Equation 2.15], or after completion of the velocity update. In Figure 2.4 we present results for the stability of NEB/QVV as a function of the force constant parameter for three of these quenching approaches. We will refer to an NEB optimisation as stable for a certain combination of parameters (e.g. time integration step, number of images) if the NEB steadily converges to a well-defined path and/or stays in its proximity until the maximal number of iterations is reached or the convergence criterion is satisfied.

We have performed some of the tests on the Müller-Brown (MB) two-dimensional surface [25]. This widely used surface does not present a very challenging or realistic test case, but if an algorithm does not behave well for this system it is unlikely to be useful. MB potential is defined as a sum of four terms, each of which takes the form

$$V_i(x,y) = \alpha \exp\left[a\,(x-x_0)^2 + b\,(x-x_0)\,(y-y_0) + c\,(y-y_0)^2\right], \tag{2.35}$$

where x and y are variables and α, a, b, c, x_0 and y_0 are parameters. In the form

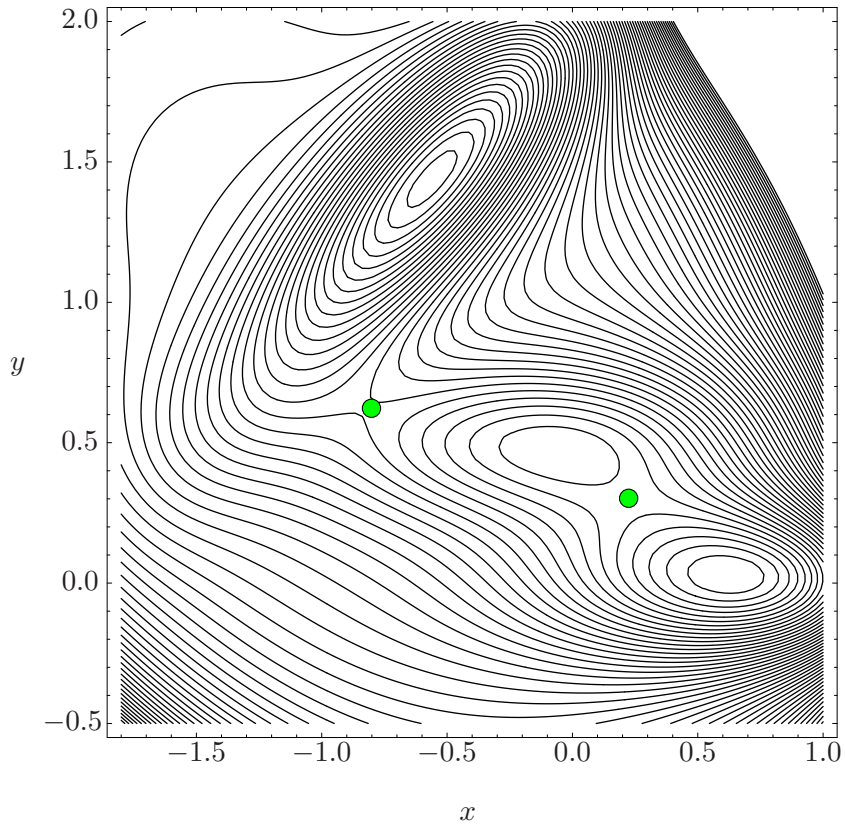


Figure 2.3: A contour plot of Müller-Brown surface [25]. Transition states are designated with green circles.

$V_i(\alpha, a, b, c, x_0, y_0)$ these parameters can be written down as $V_1(-200, -1, 0, -10, 1, 0)$, $V_2(-100, -1, 0, -10, 0, 0.5)$, $V_3(-170, -6.5, 11, -6.5, -0.5, 1.5)$ and $V_4(15, 0.7, 0.6, 0.7, -1, 1)$. A contour plot of Müller-Brown surface is depicted in Figure 2.3.

Figure 2.4 shows the results of several thousand optimisations for a 17-image band with the MB two-dimensional potential [25] using QVV minimisation and a time step of 0.01 (consistent units) for different values of k_{spr}. Each run was started from the initial guess obtained using linear interpolation and terminated when the root-mean-square (RMS) gradient became less than 0.01. We define the RMS gradient for the NEB as

$$g^{\perp}_{\mathrm{RMS}} = \sqrt{\frac{\sum_{i=1}^{N_i} |\mathbf{g}_i^{\perp}|}{N_i \eta}} \tag{2.36}$$

where N_i is the number of images in the band and η is the number of atomic degrees of freedom available to each image.

It seems natural to remove the velocity component perpendicular to the gradient

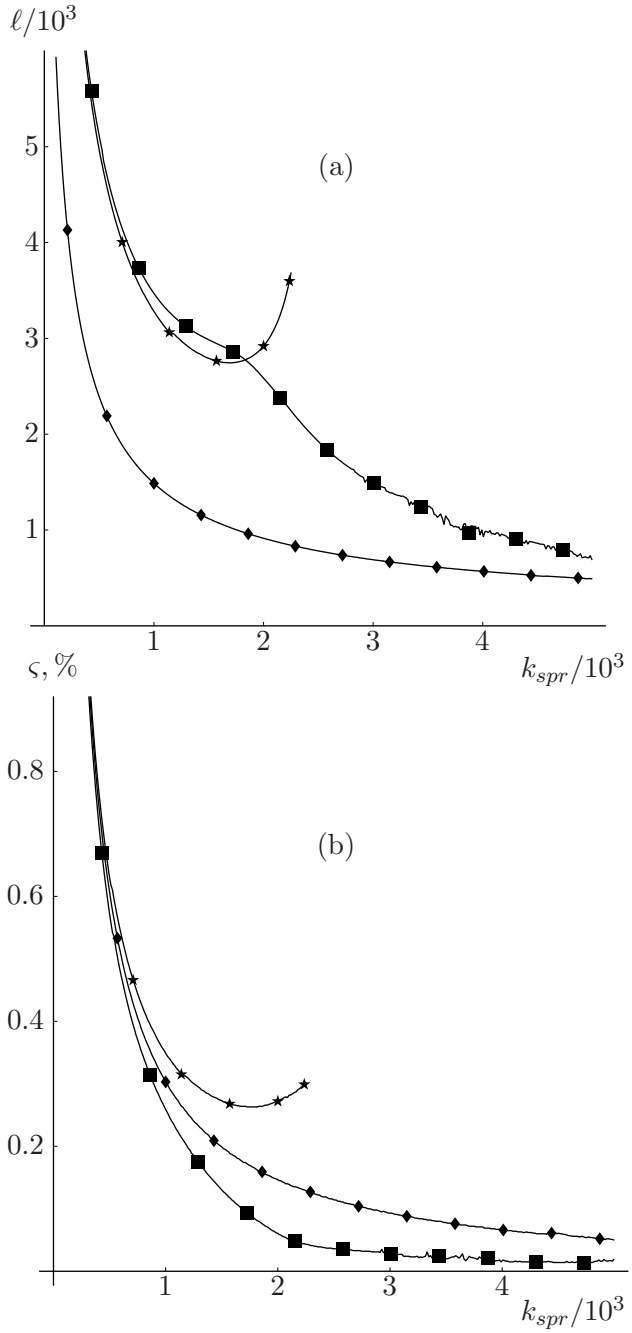


Figure 2.4: (a) Number of iterations, ℓ, and (b) average deviation from the average image separation, ς, as a function of the spring force constant, k_{spr}, obtained using a 17-image NEB on the Müller-Brown surface [25]. Minimisation was performed using QVV with time step 0.01 and RMS force termination criterion 0.01. The number of iterations is shown for velocity quenching after the coordinate update (diamonds), after the gradient evaluation (squares) and at the half-step through the velocities update (stars).

at the current point when the geometry $\mathbf{X}(t)$, gradient $\mathbf{g}(t)$ and velocity $\boldsymbol{\mathcal{V}}(t)$ are available, i.e.

$$\boldsymbol{\mathcal{V}}_{\mathrm{Q}}(t) = \Big(\boldsymbol{\mathcal{V}}(t), \hat{\mathbf{g}}(t)\Big)\hat{\mathbf{g}}(t), \tag{2.37}$$

where $\boldsymbol{\mathcal{V}}_{\mathrm{Q}}(t)$ is the velocity vector after quenching. However, we found this approach to be the least stable of all — the optimisation was slow and convergence was very sensitive to the magnitude of the time step. Hence we do not show any results for this type of quenching.

From Figure 2.4 (a) we see that the best approach is to quench the velocity after the coordinate update. The optimisation is then stable for a wide range of force constant values, and the images on the resulting pathway are evenly distributed. In this quenching formulation the velocity response to a new gradient direction is retarded by one step in coordinate space: the step is still taken in the direction $\boldsymbol{\mathcal{V}}(t)$ but the corresponding velocity component is removed. To implement this slow-response QVV (SQVV) it is necessary to modify the VV algorithm described in Section 2.5.1 by inserting Equation 2.37 in between the two stages described by Equation 2.14 and Equation 2.15.

The second-best approach after SQVV is to quench the velocity at midstep $t + \delta t/2$ using the new gradient. On average, this algorithm takes twice as long to converge the NEB to a given RMS gradient tolerance compared to SQVV. However, the method is stable for the same range of spring force constant values and produces a pathway in which the images are equispaced more accurately than the other formulations [see Figure 2.4 (b)].

The least successful of the three QVV schemes considered involves quenching velocities at mid-step using the gradient from the previous iteration (stars in Figure 2.4). Even though the number of iterations required is roughly comparable to that obtained by quenching using the new gradient, it has the smallest range of values for the force constant where it is stable. Some current implementations of NEB [136, 137] (intended for use in combination with electronic structure codes) use this type of quenching in their QVV implementation.

We have also conducted analogous calculations for more complicated systems such as permutational rearrangements of Lennard-Jones clusters. The results are omitted for brevity, but agree with the conclusions drawn from the simpler 2D model described

above. The same is true for the choice of force constant investigated in the following section.

2.5.2 Choice of the Force Constant

We find that if the force constant is too small many more iterations are needed to converge the images to the required RMS tolerance, regardless of the type of quenching. In addition, the path exhibits a more uneven image distribution. This result occurs because at the initial stage the images may have very different gradients from the true potential along the band, because they lie far from the required path, and the gradient of the true potential governs the optimisation. When the true RMS force is reduced the springs start to play a more important role. But at this stage the forces are small and so is the QVV step size. The influence of the springs is actually most important during the initial optimisation stage, for it can determine the placement of images in appropriate regions. It is less computationally expensive to guide an image into the right region at the beginning of an optimisation than to restore the distribution afterwards by dragging it between two minima through a transition state region.

If k_{spr} is too big the NEB never converges to the required RMS gradient tolerance value. Instead, it stays in proximity to the path but develops oscillations: adjacent images start to move in opposite directions. For all types of quenching we observed similar behaviour when large values of the force constant were used. This problem is related to the step in coordinate space that the optimiser is taking: for the SQVV case simply decreasing the time step remedies this problem.

2.5.3 Comparison of SQVV and L-BFGS Minimisers for the MB Surface

We tested the NEB/L-BFGS method by minimising a 17-image NEB for the two-dimensional Müller-Brown surface [25]. Our calculations were carried out using the `OPTIM` program [138]. The NEB method in its previous formulation [28] and a modified L-BFGS minimiser [9] were implemented in `OPTIM` in a previous discrete path sampling study [8]. We used the same number of images, initial guess and termination criteria as described in Section 2.5.1 to make the results directly comparable.

Figure 2.5 shows the performance of the L-BFGS minimiser as a function of k_{spr}.

We used the following additional L-BFGS specific settings. The number of corrections in the BFGS update was set to $m = 4$ (Nocedal's recommendation for the number of corrections is $3 \leqslant m \leqslant 7$, see Reference [3]), the maximum step size was 0.1, and we limited the step size for each image separately, i.e.

$$|\mathbf{p}_j| \leqslant 0.1, \tag{2.38}$$

where $\mathbf{p}_j$ is the step for image j. The diagonal elements of the inverse Hessian were initially set to 0.1.

From Figure 2.5 it can be seen that the performance of L-BFGS minimisation is relatively independent of the choice of force constant. All the optimisations with $30 \leqslant k_{spr} \leqslant 10,000$ converged to the steepest-descent path, and, for most of this range, in less than 100 iterations. This method therefore gives roughly an order of magnitude improvement in speed over SQVV minimisation [see Figure 2.4 (a)].

We found it helpful to limit the step size while optimising the NEB with the L-BFGS minimiser. The magnitude and direction of the gradient on adjacent images can vary significantly. Taking bigger steps can cause the appearance of temporary discontinuities and kinks in the NEB. The NEB still converges to the correct path, but it takes a while for these features to disappear and the algorithm does not converge any faster.

2.5.4 Doubly Nudged Elastic Bands

The NEB/QVV approach has previously been systematically tested on systems with around $\eta = 100$ degrees of freedom [107]. However, in the majority of cases these test systems could be divided into a 'core' and a smaller part that actually changes significantly. The number of active degrees of freedom is therefore significantly smaller than the total number in these tests. For example, prototropic tautomerisation of cytosine nucleic acid base ($\eta = 33$) involves motion of one hydrogen atom along a quasi-rectilinear trajectory accompanied by a much smaller distortion of the core.

We have therefore tested the performance of the NEB/SQVV and NEB/L-BFGS schemes for more complicated rearrangements of Lennard-Jones (LJ) clusters to validate the results of Section 2.5.2, and to investigate the stability and performance of both approaches when there are more active degrees of freedom. Most of our test cases involve permutational isomerisation of the LJ_7, LJ_{38} and LJ_{75} clusters. These examples

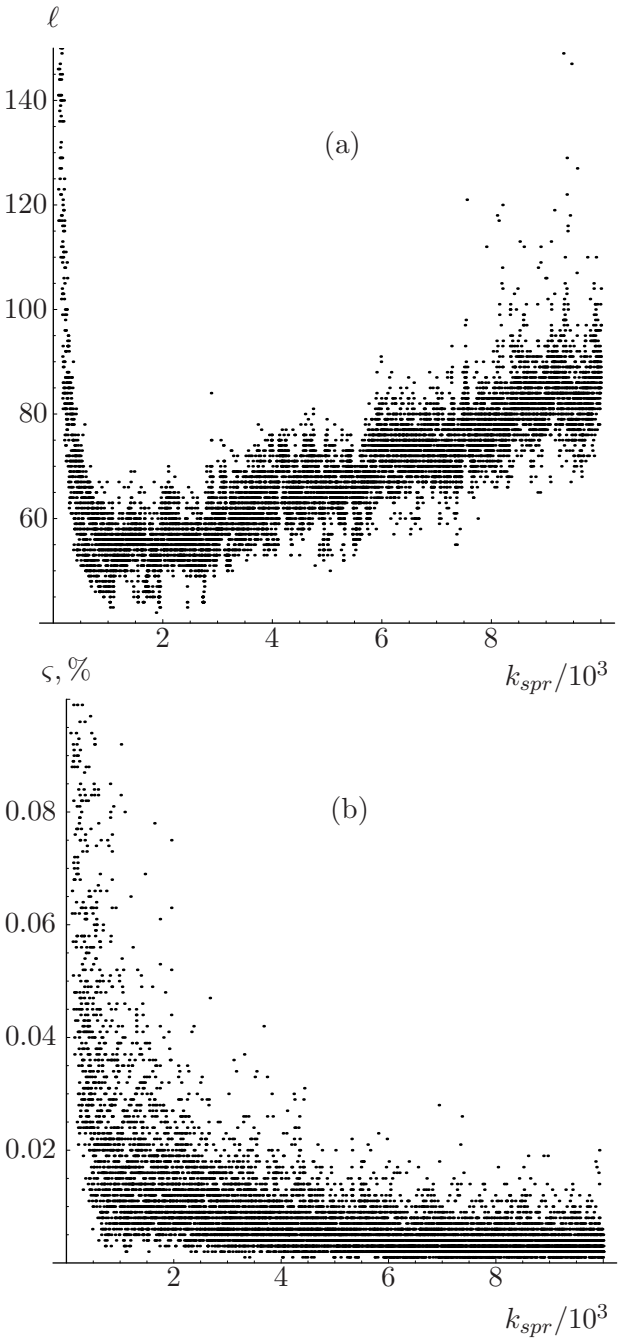


Figure 2.5: (a) Number of iterations, ℓ, and (b) average deviation from average image separation, ς, as a function of the spring force constant, k_{spr}, obtained using a 17-image NEB for the Müller-Brown surface [25]. Minimisation was performed using L-BFGS with number of corrections $m = 4$, maximum step size 0.1 and RMS force termination criterion 0.01.

include cases with widely varying separation between the endpoints, integrated path length, number of active degrees of freedom and cooperativity.

Permutational rearrangements are particularly interesting because it is relatively difficult to produce an initial guess for the NEB run. In contrast, linear interpolation between the endpoints was found to provide a useful initial guess for a number of simpler cases [27]. For example, it was successfully used to construct the NEB for rearrangements that involve one or two atoms following approximately rectilinear trajectories, and for migration of a single atom on a surface [107]. For more complex processes an alternative approach adopted in previous work is simply to supply a better initial guess 'by hand', e.g. construct it from the images with unrelaxed geometries containing no atom overlaps [107]. The 'detour' algorithm described in previous calculations that employ the ridge method could also be used to avoid 'atom-crashing' in the initial interpolation [93].

It has previously been suggested that it is important to eliminate overall rotation and translation (ORT) of each image during the optimisation of an NEB [27]. We have implemented this constraint in the same way as Jónsson *et al.*, by freezing one atom, restricting the motion of a second atom to a plane, and constraining the motion of a third atom to a line by zeroing the appropriate components of the NEB gradient.

We were able to obtain stable convergence in NEB/L-BFGS calculations only for simple rearrangements, which confirms that straightforward L-BFGS optimisation of the NEB is unstable [107]. Figure 2.6 shows the performance of the NEB/SQVV [Figure 2.6 (a)] and NEB/L-BFGS [Figure 2.6 (b)] approaches for one such rearrangement. These calculations were carried out using a 7-image NEB both with (diamonds) and without (stars) removing ORT for isomerisation of an LJ_7 cluster (global minimum $\rightarrow$ second-lowest minimum). The number of iterations, ℓ, is proportional to the number of gradient evaluations regardless of the type of minimiser. Hence, from Figure 2.6 we conclude that for this system NEB/L-BFGS is faster than NEB/SQVV by approximately two orders of magnitude. However, removal of ORT leads to instability in the NEB/L-BFGS optimisation: the images do not stay in proximity to the required path for long and instead diverge from it [see inset in Figure 2.6 (b)].

By experimentation we have found that the main source of the instabilities is the

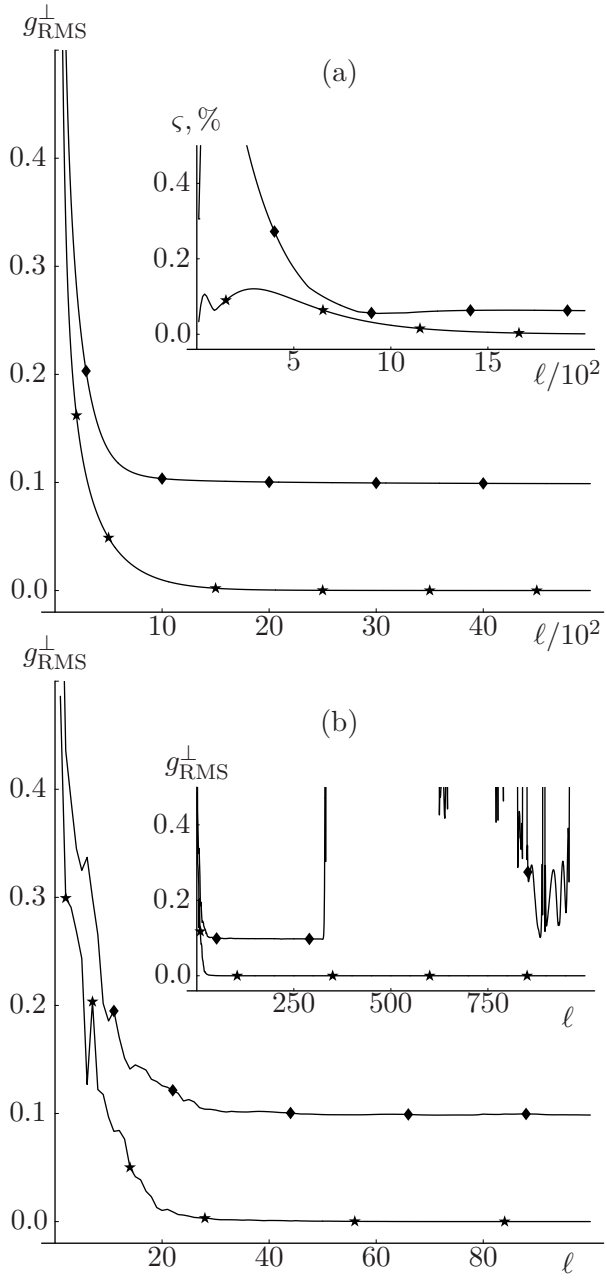


FIGURE 2.6: RMS gradient $g^{\perp}_{\mathrm{RMS}}$ as a function of iteration number ℓ. A 7-image NEB was used to model an isomerisation path in the LJ_7 cluster (global minimum → second-lowest minimum). Minimisation was performed using the SQVV (a) and L-BFGS (b) methods. Results are shown for minimisations with and without removing overall rotation and translation (diamonds and stars, respectively). The inset in (a) depicts the average deviation from the average image separation, ς, as a function of iteration number for minimisations using SQVV, while the inset in (b) shows $g^{\perp}_{\mathrm{RMS}}$ recorded for 1000 iterations of L-BFGS minimisations. These calculations were all continued for a fixed number of iterations, regardless of convergence.

complete removal of $\widetilde{\mathbf{g}}^{\perp}$. Instead, the inclusion of some portion of $\widetilde{\mathbf{g}}^{\perp}$ in the NEB gradient, i.e.

$$\mathbf{g}_{\mathrm{NEB}} = \mathbf{g}^{\perp} + \widetilde{\mathbf{g}}^{\parallel} + \widetilde{\mathbf{g}}^{*}, \tag{2.39}$$

where $\widetilde{\mathbf{g}}^{*} = \xi\widetilde{\mathbf{g}}^{\perp}$, makes the NEB/L-BFGS calculations stable but introduces some additional corner-cutting, as well as an extra parameter, ξ. Since we use the transition state candidates from NEB as starting points for further EF calculations the corner-cutting is not a drawback as long as the transition state candidates are good enough. By adjusting ξ in the range of $(0.01, 0.1)$ we were able to achieve satisfactory performance for the NEB/L-BFGS method in a number of cases. However, an alternative modification, described below, proved to be even more successful.

The drawback of the NEB gradient described by Equation 2.39 stems from the interference of $\mathbf{g}^{\perp}$ and $\xi\widetilde{\mathbf{g}}^{\perp}$, and becomes particularly noticeable when the projection of $\xi\widetilde{\mathbf{g}}^{\perp}$ on $\mathbf{g}^{\perp}$ and $\mathbf{g}^{\perp}$ itself are of comparable magnitude. This problem is analogous to the interference of $\mathbf{g}$ and $\widetilde{\mathbf{g}}$ in the original elastic band method, which was previously solved by 'nudging' [28]. We have therefore constructed the gradient of a new 'doubly' nudged elastic band (DNEB) using

$$\widetilde{\mathbf{g}}^{*} = \widetilde{\mathbf{g}}^{\perp} - (\widetilde{\mathbf{g}}^{\perp}, \hat{\mathbf{g}}^{\perp})\hat{\mathbf{g}}^{\perp}. \tag{2.40}$$

In this formulation some corner-cutting may still occur because the images tend to move cooperatively during optimisation; the spring gradient $\widetilde{\mathbf{g}}^{\perp}_{\mathrm{DNEB}}$ acting on one image can still indirectly interfere with the true gradients of its neighbours. In our calculations this drawback was not an issue, since we are not interested in estimating properties of the path directly from its discrete representation. Instead we construct it from steepest-descent paths calculated after converging the transition states tightly using the EF approach. We have found DNEB perfectly adequate for this purpose.

We have implemented DNEB method in our `OPTIM` program [138]. Equation 2.2 was used to obtain components of both spring gradient and true gradient, Equation 2.4 to calculate pathway tangent, and Equations 2.39-2.40 to evaluate the final DNEB gradient. We note that although we employed the improved tangent from Equation 2.4, we did not use Equation 2.5 to obtain $\widetilde{\mathbf{g}}^{\parallel}$. This implementation detail was not clarified in Reference [7], and we thank Dr. Dominic R. Alfonso for pointing that out.

We have also tested a number of approaches that might be useful if one wants to produce a full pathway involving a number of transition states for a complicated rearrangement in just one NEB run. One of these, for instance, is a gradual removal of the $\widetilde{\mathbf{g}}^*$ component from the NEB gradient once some convergence criterion is achieved. This removal works remarkably well, particularly in situations with high energy initial guesses, which occur frequently if the guessing is fully automated. This adjustment can be thought of as making the band less elastic in the beginning in order to resolve the highest-energy transition state regions first.

2.5.5 Comparison of the DNEB/L-BFGS and DNEB/SQVV Methods for Permutational Isomerisations of LJ_7

It is sometimes hard to make a direct comparison of different double-ended methods for a particular rearrangement because the calculations may converge to different paths. Another problem concerns the choice of a consistent termination criterion: the RMS force usually converges to some finite system-dependent value, which in turn may depend on the number of images and other parameters. A low-energy chain of NEB images does not necessarily mean that a good pathway has been obtained, since it may arise because more images are associated with regions around local minima, rather than the higher energy transition state regions. Here we present the results of DNEB/L-BFGS, DNEB/SQVV and, where possible, NEB/SQVV calculations for all the distinct permutational rearrangements of the global minimum for the LJ_7 cluster (see Figure 2.7 for the endpoints and nomenclature).

It is possible to draw a firm conclusion as to how well the NEB represents the pathway when the corresponding stationary points and steepest-descent paths are already known. We therefore base our criterion for the effectiveness of an NEB calculation on whether we obtain good estimates of all the transition states. By considering several systems of increasing complexity we hope to obtain comparisons that are not specific to a particular pathway.

Connections between two minima are defined by calculating an approximation to the two steepest-descent paths that lead downhill from each transition state, and two transition states are considered connected if they are linked to the same minimum via

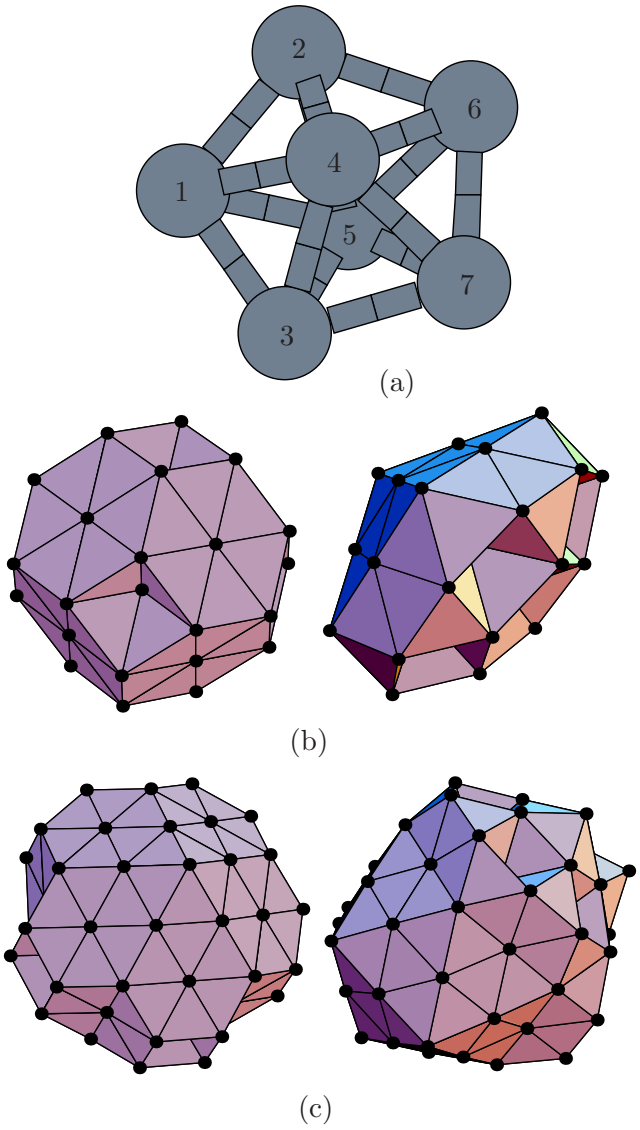


Figure 2.7: Structures of the most stable isomers for (a) LJ_7, (b) LJ_{38} and (c) LJ_{75} clusters, which were used as endpoints in the NEB calculations. The first endpoint was the global minimum in each case. For LJ_{38} and LJ_{75} the second endpoint was chosen to be second-lowest minimum shown on the right in parts (b) and (c), respectively, while a permutational isomer of the global minimum was used as the second endpoint in all the LJ_7 calculations. The notation 1–2 denotes an LJ_7 rearrangement where the second endpoint is structure (a) with atoms 1 and 2 swapped. The structures and numbering employed for LJ_{38} and LJ_{75} are defined at http://www-wales.ch.cam.ac.uk/∼sat39/DNEBtests/. Picture of LJ_7 was generated using `XMakemol` program written by Dr. Matthew Hodges [139]. Structures of LJ_{38} and LJ_{75} clusters were visualised using a `Mathematica` [140] notebook for making and manipulating triangulated polyhedra written by Dr. David Wales [141].

a steepest-descent path. We will say that minima are 'connected' if there exists a path consisting of one or more transition states and intermediate minima linking them. Permutational isomers of the same minimum are distinguished in these calculations. We refer to the chain of images produced by the NEB calculation as 'connected' if going downhill from each transition state using steepest-descent minimisation yields a set of minima that contains the endpoints linked together.

For NEB/SQVV calculations we used the NEB formulation defined in Reference [28]. DNEB is different from the above method because it includes an additional component in the NEB gradient, as described by Equation 2.39 and Equation 2.40. In addition, for the following DNEB calculations we did not remove overall rotation and translation (ORT), because we believe it is unnecessary when our gradient modification is used. To converge transition state candidates tightly we employed EF optimisation, limiting the maximum number of EF iterations to five with an RMS force convergence tolerance of 10^{-5}. (Standard reduced units for the Lennard-Jones potential are used throughout this work.) Initial guesses for all the following calculations were obtained by linear interpolation between the endpoints. To prevent 'atom-crashing' from causing overflow in the initial guess we simply perturbed such images slightly using random atomic displacements of order 10^{-2} reduced units.

In each case we first minimised the Euclidean distance between the endpoints with respect to overall rotation and translation using the method described in Reference [142].* SQVV minimisation was performed with a time step of 0.01 and a maximum step size per degree of freedom of 0.01. This limit on the step size prevents the band from becoming 'discontinuous' initially and plays an important role only during the first 100 or so iterations. The limit was necessary because for the cases when the endpoints are permutational isomers linear interpolation usually yields bands with large gradients, and it is better to refrain from taking excessive steps at this stage. We did not try to select low energy initial guesses for each rearrangement individually, since one of our primary concerns was to automate this process. For the same reason, all the L-BFGS optimisations were started from guesses preoptimised using SQVV until the RMS force dropped below 2.0.

*Structure alignment methods that are based on finding a pseudorotation matrix that satisfies Eckart axis conditions might equally well be used for this purpose [143–147].

Table 2.1: The minimal number of images and total number of gradient calls (in parentheses) are shown for degenerate rearrangements of LJ_7. The image range was $2 \leqslant N_i \leqslant 20$ and the iteration range was $1 < \ell \leqslant 3,000N_i$. Each SQVV calculation was started from the guess produced using linear interpolation, while guesses for L-BFGS runs were preoptimised using DNEB/SQVV until the RMS force dropped below 2.0. Every iteration the images that satisfy $V_i > V_{i\pm1}$ were optimised further using eigenvector-following. The transition state candidates that converged to a true transition state within five iterations were used to generate the connected minima using energy minimisation. If this procedure yielded a connected pathway the calculation was terminated and the rest of the parameter range was not explored. Otherwise, the number of images was incremented and the procedure repeated. The number of gradient calls is a product of the number of images and the total number of iterations. For the L-BFGS calculations the number of iterations includes the SQVV preoptimisation steps (100 on average) and the actual number of L-BFGS steps. Dashes signify cases where we were unable to obtain a connected pathway.

Method	1–2	2–3	3–4	4–5
DNEB/L-BFGS	5(1720)	18(30276)	11(2486)	18(8010)
DNEB/SQVV	16(21648)	–	10(14310)	–

Table 2.1 shows the minimum number of images and gradient calls required to produce a connected pathway using the DNEB/L-BFGS and DNEB/SQVV methods. These calculations were run assuming no prior knowledge of the path. Normally there is no initial information available on the integrated path length or the number of intermediate minima between the endpoints, and it takes some experimentation to select an appropriate number of images. Our strategy is therefore to gradually increase the number of images to make the problem as computationally inexpensive as possible. Hence we increment the number of images and maximum number of NEB iterations in each calculation until a connected path is produced, in the sense defined above. The permitted image range was $2 \leqslant N_i \leqslant 20$ and the maximum number of NEB iterations ranged from $1 \leqslant l \leqslant 3,000N_i$. We were unable to obtain connected pathways for any of the four LJ_7 rearrangements using the NEB/SQVV approach.

Table 2.2 presents the results of analogous calculations where we keep the number of

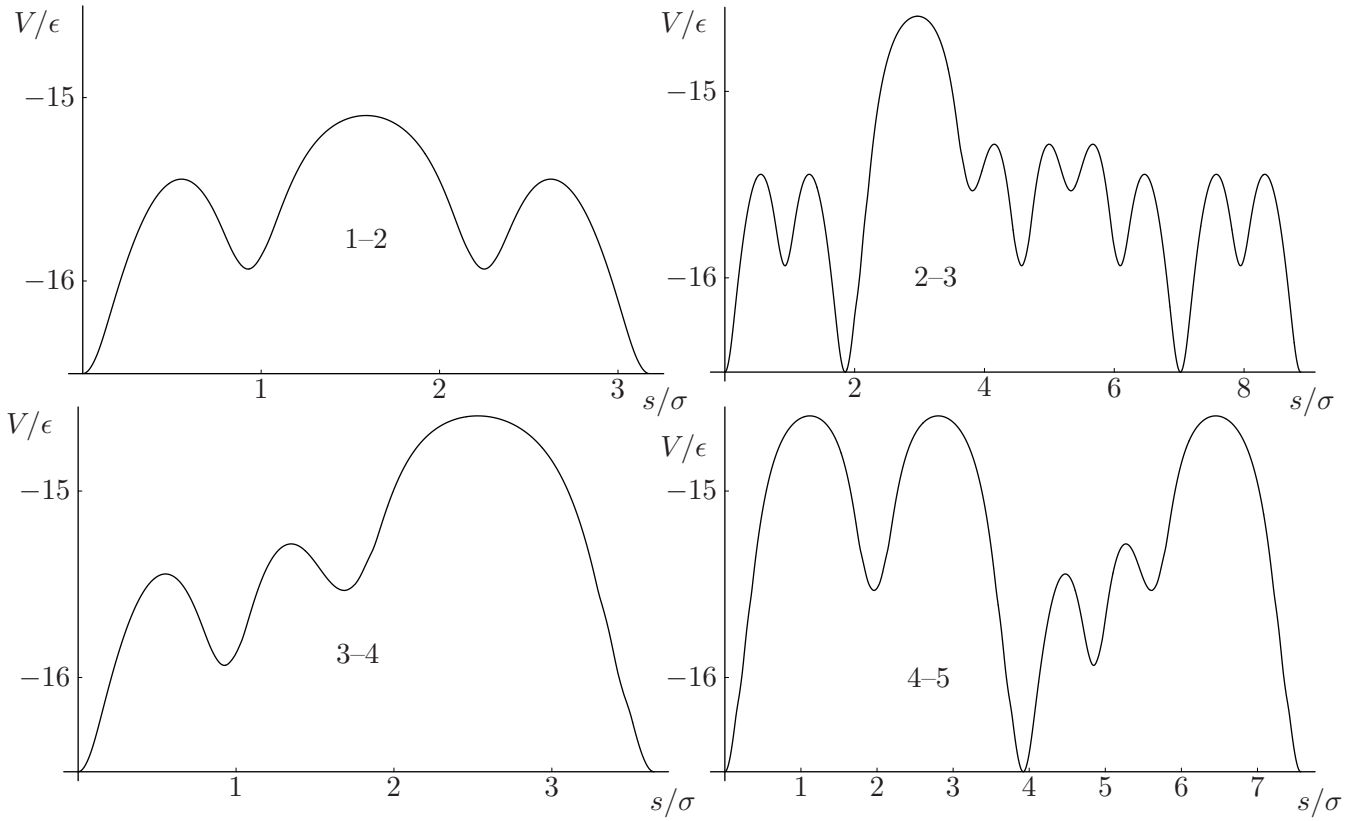


Figure 2.8: The potential energy, V, as a function of the integrated path length, s, for four degenerate rearrangements of LJ_7. These profiles were constructed using energy minimisation to characterise the paths connected to transition states obtained by EF refinement of candidate structures obtained from DNEB calculations [19].

images fixed to 50. Unlike the performance comparison where the number of images is kept to a minimum (Table 2.1), these results should provide insight into the performance of the DNEB approach when there are sufficient images to resolve all the transition states. All the optimisations for a particular rearrangement converged to the same or an enantiomeric pathway unless stated otherwise. The energy profiles that correspond to these rearrangements are shown in Figure 2.8.

From Table 2.1 and Table 2.2 we conclude that in all cases the DNEB/L-BFGS approach is more than an order of magnitude faster than DNEB/SQVV. It is also noteworthy that the DNEB/SQVV approach is faster than NEB/SQVV because overall rotation and translation are not removed. Allowing the images to rotate or translate freely can lead to numerical problems, namely a vanishing norm for the tangent vector,

Table 2.2: The minimal number of iterations needed to produce connected pathways for four degenerate rearrangements of LJ_7 using a 50-image NEB. The strategy of this calculation is identical to the one described in the caption to Table 2.1, except that the number of images was fixed.

Method	1–2	2–3	3–4	4–5
DNEB/L-BFGS	131[a]	493	171	326
DNEB/SQVV	1130	15178	2777	23405[b]
NEB/SQVV	11088[b]	–	30627	–

[a] The number of iterations is the sum of the SQVV preoptimisation steps (100 on average) and the actual number of iterations needed by L-BFGS minimiser. [b] This value is not directly comparable since DNEB converged to a different path that contains more intermediate minima. Dashes signify cases where we were unable to obtain a connected pathway.

when the image density is very large or the spring force constant is too small. However, when overall rotation and translation are not allowed there is less scope for improving a bad initial guess, because the images are more constrained. This constraint usually means that more images are needed or a better initial guess is required. Our experience is that such constraints usually slow down convergence, depending on which degrees of freedom are frozen: if these are active degrees of freedom (see above) the whole cluster must move instead, which is usually a slow, concerted multi-step process.

2.5.6 A Revised Connection Algorithm

In previous work we have used the NEB approach to supply transition state guesses for further EF refinement [8, 133]. Double-ended searches are needed in these discrete path sampling runs to produce alternative minimum–transition state–minimum $\cdots$ sequences from an initial path. The end minima that must be linked in such calculations may be separated by relatively large distances, and a detailed algorithm was described for building up a connected path using successive transition state searches. The performance of the DNEB/L-BFGS approach is sufficiently good that we have changed this connection strategy in our `OPTIM` program. In particular, the DNEB/L-BFGS method can often provide good candidates for more than one transition state at a time, and

may even produce all the necessary transition states on a long path. However, it is still generally necessary to consider multiple searches between different minima in order to connect a pair of endpoints. In particular, we would like to use the minimum number of NEB images possible for reasons of efficiency, but automate the procedure so that it eventually succeeds or gives up after an appropriate effort for any pair of minima that may arise in a discrete path sampling run. These calculations may involve the construction of many thousands of discrete paths. As in previous work we converge the NEB transition state candidates using eigenvector-following techniques and then use L-BFGS energy minimisation to calculate approximate steepest-descent paths. These paths usually lead to local minima, which we also converge tightly. The combination of NEB and hybrid eigenvector-following techniques [22, 23] is similar to using NEB with a 'climbing image' as described in Reference [29].

The initial parameters assigned to each DNEB run are the number of images and the number of iterations, which we specify by image and iteration densities. The iteration density is the maximum number of iterations per image, while the image density is the maximum number of images per unit distance. The distance in question is the Euclidean separation of the endpoints, which provides a crude estimation of the integrated path length. This approach is based on the idea that knowing the integrated path length, which means knowing the answer before we start, we could have initiated each DNEB run with the same number of images per unit of distance along the path. In general it is also impossible to provide a lower bound on the number of images necessary to fully resolve the path, since this would require prior knowledge of the number of intervening stationary points. Our experience suggests that a good strategy is to employ as small an image and iteration density as possible at the start of a run, and only increase these parameters for connections that fail.

All NEB images, i, for which $V_i > V_{i\pm1}$ are considered for further EF refinement. The resulting distinct transition states are stored in a database and the corresponding energy minimised paths were used to identify the minima that they connect. New minima are also stored in a database, while for known minima new connections are recorded. Consecutive DNEB runs aim to build up a connected path by progressively filling in connections between the endpoints or intermediate minima to which they are

connected. This is an advantageous strategy because the linear interpolation guesses usually become better as the separation decreases, and therefore fewer optimisation steps are required. Working with sections of a long path one at a time is beneficial because it allows the algorithm to increase the resolution only where it is needed. Our experience is that this approach is generally significantly faster than trying to characterise the whole of a complex path with a single chain of images.

When an overall path is built up using successive DNEB searches we must select the two endpoints for each new search from the database of known minima. It is possible to base this choice on the order in which the transition states were found, which is basically the strategy used in our previous work [8, 133]. We have found that this approach is not flexible or general enough to overcome difficulties that arise in situations when irrelevant transition states are present in the database. A better strategy is to connect minima based upon their Euclidean separation. For this purpose it is convenient to classify all the minima into those already connected to the starting endpoint (the S set), the final endpoint (the F set), and the remaining minima, which are not connected to either endpoint (the U set). The endpoints for the next DNEB search are then chosen as the two that are separated by the shortest distance, where one belongs to S or F, and the other belongs to a different set. The distance between these endpoints is then minimised with respect to overall rotation and translation, and an initial guess for the image positions is obtained using linear interpolation. Further details of the implementation of this algorithm and the `OPTIM` program are available online [138].

2.5.7 Applications to Isomerisation of LJ_{38} and LJ_{75}

As test cases for this algorithm we have considered various degenerate rearrangements of LJ_7, LJ_{13}, LJ_{38} and LJ_{75}. (A degenerate rearrangement is one that links permutational isomers of the same structure [9, 148].) In addition, we have considered rearrangements that link second lowest-energy structure with a global minimum for LJ_{38} and LJ_{75} clusters. The PES's of LJ_{38} and LJ_{75} have been analysed in a number of previous studies [113, 149–151], and are known to exhibit a double-funnel morphology: for both clusters the two lowest-energy minima are structurally distinct and well separated in

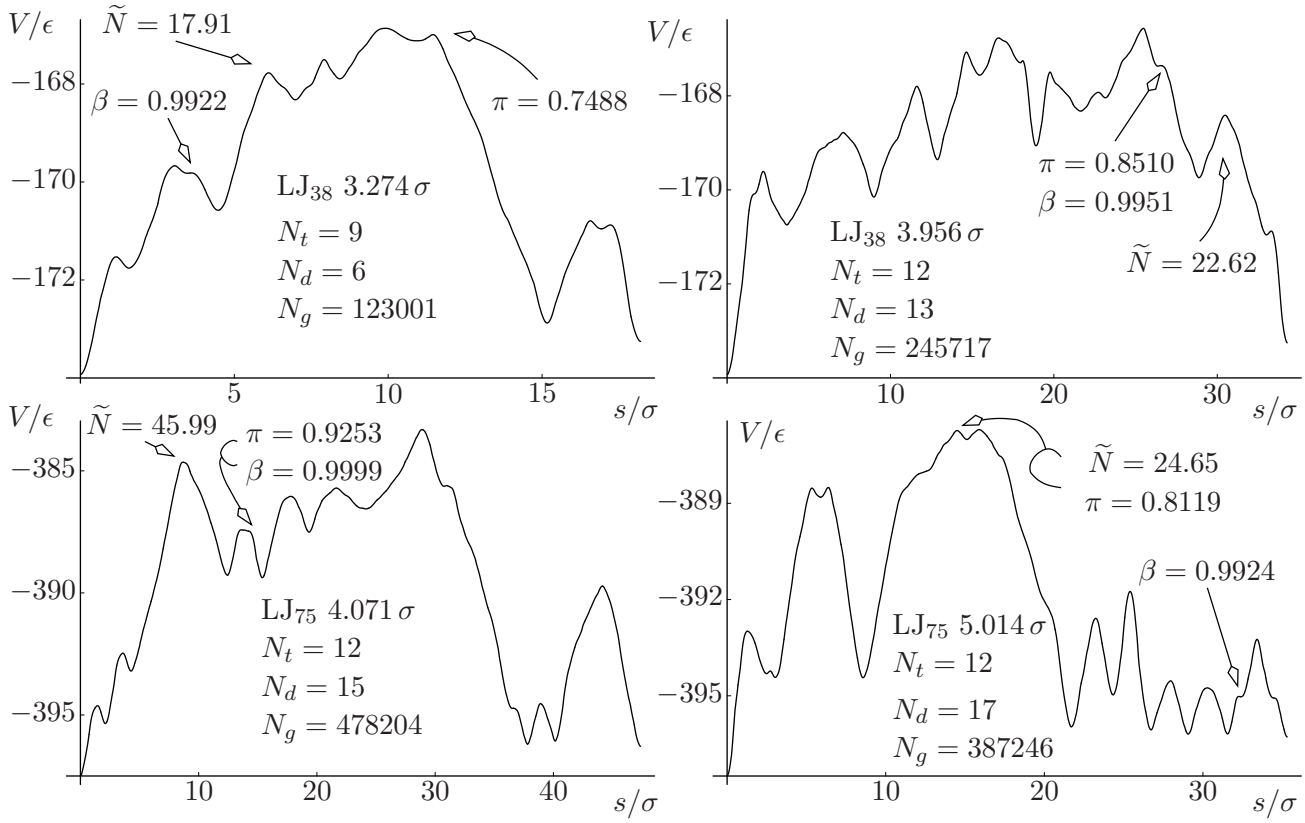


FIGURE 2.9: The potential energy, V, as a function of integrated path length, s, for pathways linking the two lowest minima of LJ_{38} and LJ_{75}. Calculations were initiated between two different sets of permutational isomers of these minima. For each profile the number of transition states, N_t, number of DNEB runs, N_d, and the total number of gradient calls, N_g, are shown. Maximum values of $\widetilde{N}$, β and π are marked next to the corresponding transition states. The endpoints were illustrated in Figure 2.7.

configuration space. This makes them useful benchmarks for the above connection algorithm. Figure 2.9 depicts the energy profiles obtained using the revised connection algorithm for rearrangements between the two lowest minima of each cluster. In each case we have considered two distinct paths that link different permutational isomers of the minima in question, and these were chosen to be the permutations that give the shortest Euclidean distances. These paths will be identified using the distance between the two endpoints; for example, in the case of LJ_{38} we have paths LJ_{38} $3.274\,\sigma$ and LJ_{38} $3.956\,\sigma$, where $2^{1/6}\,\sigma$ is the pair equilibrium separation for the LJ potential.

For each calculation we used the following settings: the initial image density was

set to 10, the iteration density to 30, and the maximum number of attempts to connect any pair of minima was limited to three. If a connection failed for a particular pair of minima then up to two more attempts were allowed before moving to the pair with the next smallest separation. For the second and third attempts the number of images was increased by 50% each time. The maximum number of EF optimisation steps was set to 30 with an RMS force convergence criterion of 10^{-5}. In Figure 2.9 every panel is labelled with the separation between the endpoints, the number of transition states in the final pathway, the number of DNEB runs required, and the total number of gradient calls.

Individual pathways involving a single transition state have been characterised using indices [152] such as

$$\widetilde{N} = \frac{\left(\sum_t |\mathbf{X}_t(S) - \mathbf{X}_t(F)|^2\right)^2}{\sum_t |\mathbf{X}_t(S) - \mathbf{X}_t(F)|^4}, \tag{2.41}$$

which is a measure of the number of atoms that participate in the rearrangement.* Here $\mathbf{X}_t(S)$ and $\mathbf{X}_t(F)$ are the position vectors of atom t in the starting and finishing geometries, respectively. The largest values are marked in Figure 2.9 next to the corresponding transition state. It is noteworthy that the pathways LJ_{38} $3.956\,\sigma$ and LJ_{75} $4.071\,\sigma$ both involve some highly cooperative steps, and the average value of $\widetilde{N}$ is more than 12 for both of them.

We have found that it is usually easier to locate good transition state candidates for a multi-step path if the stationary points are separated by roughly equal distances, in terms of the integrated path length. Furthermore, it seems that more effort is needed to characterise a multi-step path when transition states involving very different path lengths are present. In such cases it is particularly beneficial to build up a complete path in stages. To further characterise this effect we introduce a path length asymmetry index π defined as

$$\pi = \frac{|s_+ - s_-|}{s_+ + s_-}, \tag{2.42}$$

where s_+ and s_- are the two integrated path lengths corresponding to the two downhill steepest-descent paths from a given transition state. For example, in rearrangement LJ_{38} $3.956\,\sigma$, five steps out of nine have $\pi > 0.5$.

*A modification of this Stillinger and Weber's participation index as well as a new cooperativity index will be presented in Chapter 3.

Barrier asymmetry also plays a role in the accuracy of the tangent estimate, the image density required to resolve particular regions of the path, and in our selection process for transition state candidates, which is based on the condition $V_i > V_{i\pm1}$.[†] To characterise this property we define a barrier asymmetry index, β, as

$$\beta = \frac{|E_+ - E_-|}{\max(E_+, E_-)}, \tag{2.43}$$

where E_+ and E_- are the barriers corresponding to the forward and reverse reactions, respectively. The test cases in Figure 2.9 include a variety of situations, with barrier asymmetry index β ranging from 0.004 to 1.000. The maximum values of π and β are shown next to the corresponding transition states in this figure.

We note that the total number of gradient evaluations required to produce the above paths could be reduced significantly by optimising the DNEB parameters or the connection strategy in each case. However, our objective was to find parameters that give reasonable results for a range of test cases, without further intervention.

2.5.8 A Dijkstra-based Selector

An essential part of the connection algorithm is a mechanism to incorporate the information obtained in all the previous searches into the next one. For large endpoint separations guessing the initial pathway can be difficult, and there is a large probability of finding many irrelevant stationary points at the beginning of the calculation.

The connection algorithm described earlier uses one double-ended search per cycle. However, we have found that this approach can be overwhelmed by the abundance of stationary points and pathways for complicated rearrangements. We therefore introduce the idea of an unconnected pathway and make the connection algorithm more focused by allowing more than one double-ended search per cycle.

Before each cycle a decision must be made as to which minima to try and connect next. Various strategies can be adopted, for example, selection based on the order in which transition states were found [8], or, selection of minima with the minimal separation in Euclidean distance space [7]. However, when the endpoints are very distant in configuration space, neither of these approaches is particularly efficient. The

[†]Unless there is no maximum in the profile, in which case we consider for transition state searching the highest energy image.

number of possible connections that might be tried simply grows too quickly if the S, F and U sets become large. However, the new algorithm described below seems to be very effective.

The modified connection algorithm we have used in the present work is based on a shortest path method proposed by Dijkstra [153, 154]. We can describe the minima that are known at the beginning of each connection cycle as a complete graph [155], $G = (M, E)$, where M is the set of all minima and E is the set of all the edges between them. Edges are considered to exist between every pair of minima u and v, even if they are in different S, F or U sets, and the weight of the edge is chosen to be a function of the minimum Euclidean distance between them [142]:

$$w(u,v) = \begin{cases} 0, & \text{if } u \text{ and } v \text{ are connected via a single transition state,} \\ \infty, & \text{if } n(u,v) = n_{max}, \\ f(D(u,v)), & \text{otherwise,} \end{cases} \tag{2.44}$$

where $n(u, v)$ is the number of times a pair (u, v) was selected for a connection attempt, n_{max} is the maximal number of times we may try to connect any pair of minima, and $D(u, v)$ is the minimum Euclidean distance between u and v. f should be a monotonically increasing function, such as $f(D(u, v)) = D(u, v)^2$. We denote the number of minima in the set $M = S \cup U \cup F$, as m, and the number of edges in the set E as $e = m(m - 1)/2$.

Using the Dijkstra algorithm [153, 154] and the weighted graph representation described above, it is possible to determine the shortest paths between any minima in the database. The source is selected to be one of the endpoints. Upon termination of the Dijkstra algorithm, a shortest path from one endpoint to the other is extracted. If the weight of this pathway is non-zero, it contains one or more 'gaps'. Connection attempts are then made for every pair (u, v) of adjacent minima in the pathway with non-zero $w(u, v)$ using the DNEB approach [7].

The computational complexity of the Dijkstra algorithm is at worst $O(m^2)$, and the memory requirements scale in a similar fashion. The most appropriate data structure is a weighted adjacency matrix. For the calculations presented here, the single source shortest paths problem was solved at the beginning of each cycle, which took less than

10% of the total execution time for the largest database encountered. We emphasise here that once an initial path has been found, the perturbations considered in typical discrete path sampling (DPS) calculations will generally involve attempts to connect minima that are separated by far fewer elementary rearrangements than the endpoints. It is also noteworthy that the initial path is unlikely to contribute significantly to the overall rate constant. Nevertheless, it is essential to construct such a path to begin the DPS procedure.

The nature of the definition of the weight function allows the Dijkstra algorithm to terminate whenever a second endpoint, or any minimum connected to that endpoint via a series of elementary rearrangements, is reached. This observation reduces the computational requirements by an amount that depends on the distribution of the minima in the database among the S, U and F sets. One of the endpoints is always a member of the S set, while the other is a member of F set. Either one can be chosen as the source, and we have found it most efficient to select the one from the set with fewest members. However, this choice does not improve the asymptotic bounds of the algorithm.

2.5.9 Applications to Tryptophan Zippers

Tryptophan zippers are stable fast-folding β-hairpins designed by Cochran *et al.* [156], which have recently generated considerable interest [157, 158]. In the present work we have obtained native to denatured state rearrangement pathways for five tryptophan zippers: trpzip 1, trpzip 2, trpzip 3, trpzip 3-I and trpzip 4. The notation is adopted from the work of Du *et al.* [158]. All these peptides contain twelve residues, except for trpzip 4, which has sixteen. Tryptophan zippers 1, 2, 3 and 3-I differ only in the sequence of the turn. Experimental measurements of characteristic folding times for these peptides have shed some light on the significance of the turn sequence in determining the stability and folding kinetics of peptides with the β-hairpin structural motif [158].

To model these molecules we used a modified CHARMM19 force field [4], with symmetrised Asn, Gln and Tyr dihedral angle and Cter improper dihedral angle terms, to ensure that rotamers of these residues have the same energies and geometries. These

changes to the standard CHARMM19 force field are described in detail in Appendix A. Another small modification concerned the addition of a non-standard amino acid, D-proline, which was needed to model trpzip 3. The implicit solvent model EEF1 was used to account for solvation [11], with a small change to the original implementation to eliminate discontinuities [159].

We have used the Dijkstra-based connection algorithm to obtain folding pathways for all five trpzip peptides. In each case the first endpoint was chosen to be the native state structure, which, for 1, 2 and 4 trpzips, was taken from the Protein Data Bank (PDB) [160]. There are no NMR structures available for 3 and 3-I, so for these peptides the first endpoint was chosen to be the putative global minimum obtained using the basin-hopping method [161–163]. The folded state for trpzip 2 is depicted in Figure 2.10. The total charge of this molecule is $2e\,(\text{two Lys}) - e\,(\text{Glu}) = e$. However, in our calculations the total charge was zero because in the CHARMM19 force field ionic sidechains and termini are neutral when the EEF1 solvation model is used [11]. The second endpoint was chosen to be an extended structure, which was obtained by simply minimising the energy of a conformation with all the backbone dihedral angles set to 180 degrees. All the stationary points (including these obtained during the connection procedure) were tightly converged to reduce the root-mean-squared force below $10^{-10}\,\text{kcal}\,\text{mol}^{-1}\,\text{Å}^{-1}$. The unfolded state for trpzip 2 is depicted in Figure 2.11 for example.

Each of the five trpzip pathway searches was conducted on a single Pentium 4 3.0 GHz 512 Mb cache CPU and required less than 24 hours of CPU time. The timings could certainly be improved by optimising the various parameters employed throughout the searches. However, it is more important that the connections actually succeed in a reasonably short time. It only requires one complete path to seed a DPS run, and we expect the DPS procedure to reduce the length of the initial path by a least a factor of two in sampling the largest contributions to the effective two-state rate constants. The results of all the trpzip calculations are shown in Figure 2.12.

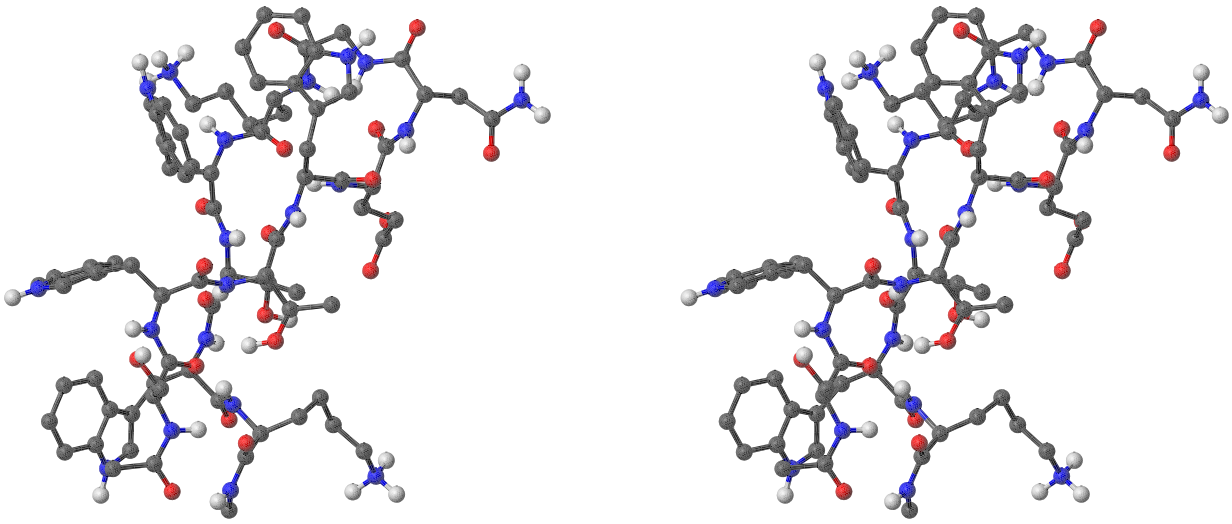


FIGURE 2.10: The folded state of tryptophan zipper 2 (shown in stereo). This structure contains 12 residues and two terminal capping groups, containing a total of 148 CHARMM19 atoms. The sequence of residues is Ace–Ser–Trp–Thr–Trp–Glu–Asn–Gly–Lys–Trp–Thr–Trp–Lys–Cbx. The naming convention for amino acid residues is the same as in Reference [164]. N- and C-termini are capped with the standard CHARMM19 blocking groups, 'Ace' (–CO–CH_3) and 'Cbx' (–NH–CH_3), respectively. The structure was obtained by minimising the first out of 20 structures deposited in the Protein Data Bank (PDB ID: 1LE1) by Cochran, Skelton, and Starovasnik [156]. The RMS force and modified [165] CHARMM19 energy are $< 10^{-10}\,\mathrm{kcal\,mol^{-1}\,\AA^{-1}}$ and $-358.3130612\,\mathrm{kcal\,mol^{-1}}$, respectively. The structural motif of this *de novo* peptide is a β-hairpin. It can be seen that the backbone is stabilised by a U-turn and six hydrogen bonds while four hydrophobic tryptophan sidechains are packed nicely on the side. Red, blue and light grey balls denote oxygen, nitrogen and hydrogen atoms, respectively. Dark grey balls denote carbon atoms as well as CHARMM19 united atoms of types CH, CH2 and CH3. Ace and Cbx each contain a single united atom of type CH3. There is one CH2 'atom' in each of the four tryptophans (connecting the 5-membered ring of the sidechain with the backbone). This figure was prepared using MolMol [166].

2.6 SUMMARY

One of the two most important results of this chapter is probably the doubly nudged elastic band formulation, in which a portion of the spring gradient perpendicular to the path is retained. With this modification we found that L-BFGS minimisation of the images is stable, thus providing a significant improvement in efficiency. Con-

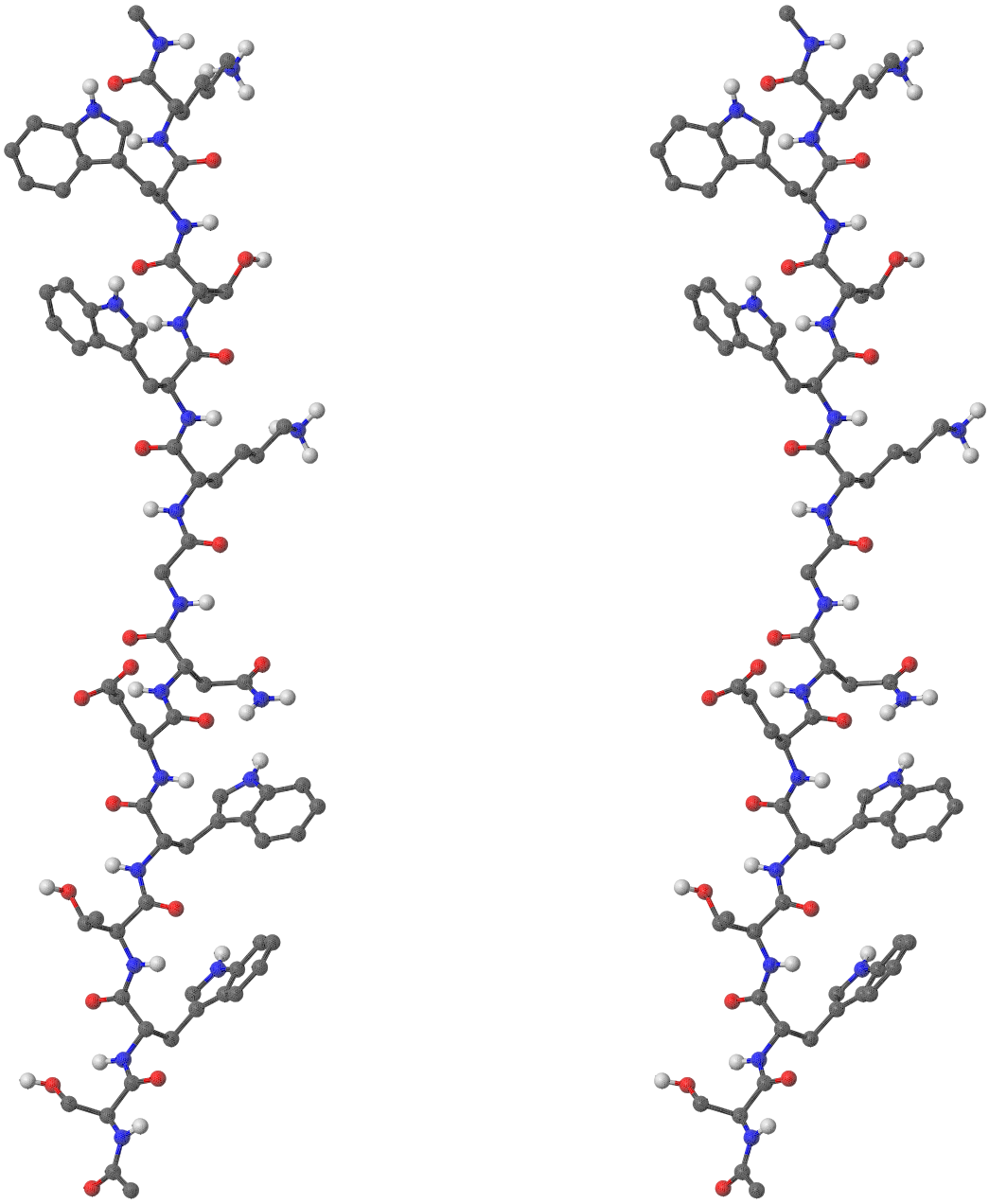
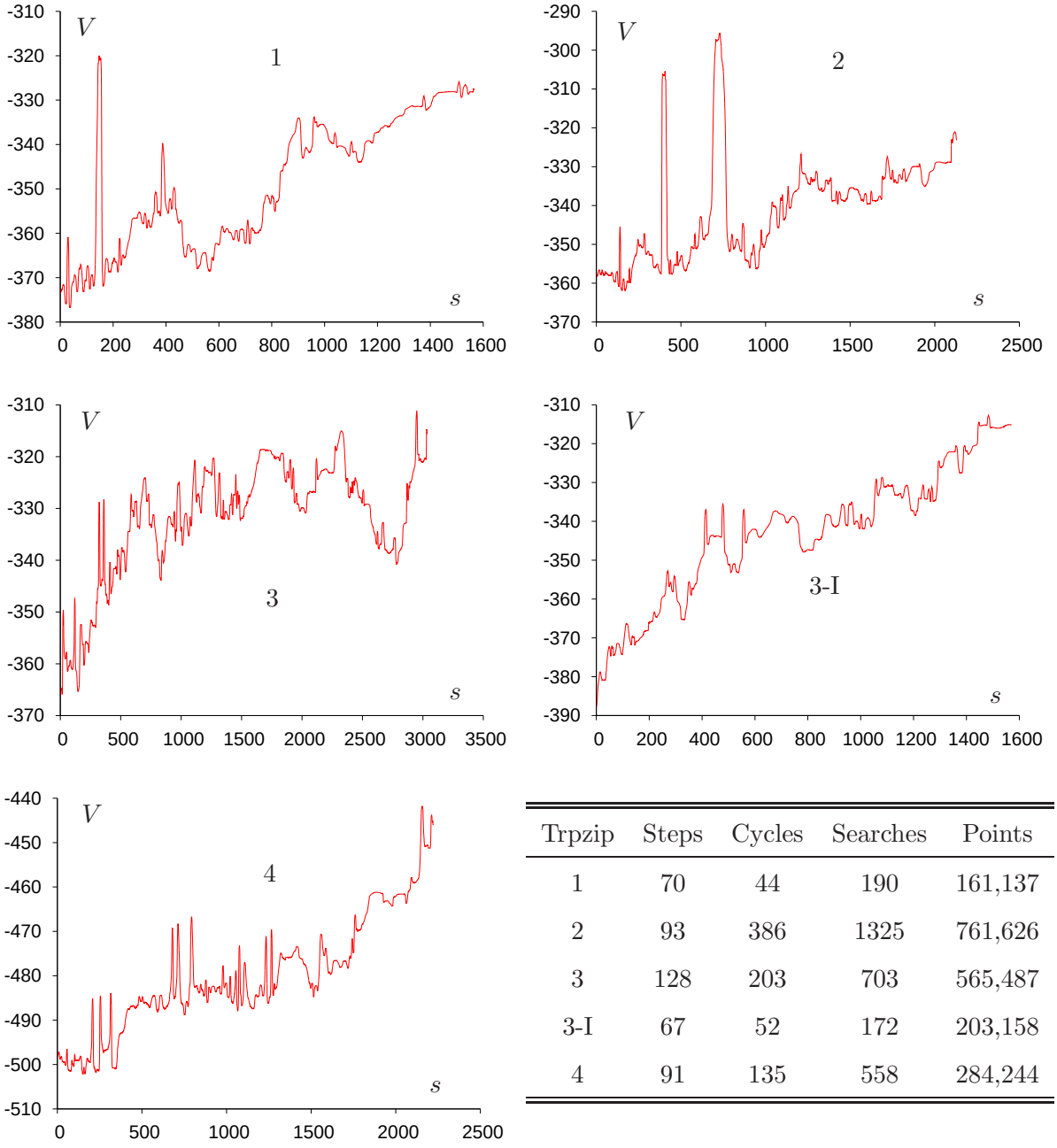


FIGURE 2.11: The unfolded state of tryptophan zipper 2 (shown in stereo). This structure was obtained by minimising a conformation with all the backbone dihedral angles set to 180 degrees. The energy of this structure is $-323.0716439\,\mathrm{kcal\,mol^{-1}}$. The RMS deviation from the minimised PDB structure depicted in Figure 2.10 is 122.70 Å. See the caption of Figure 2.10 for notation and other details.



Trpzip	Steps	Cycles	Searches	Points
1	70	44	190	161,137
2	93	386	1325	761,626
3	128	203	703	565,487
3-I	67	52	172	203,158
4	91	135	558	284,244

FIGURE 2.12: Energy profiles for native to denatured state rearrangements of tryptophan zippers found by the Dijkstra-based connection algorithm. For each profile the number of steps in the pathway, the number of connection algorithm cycles, the total number of DNEB searches and the total number of stationary points in the database (recorded upon termination of the algorithm) are shown. The total number of stationary points is presented in the form n, m, where n is the number of minima and m is the number of transition states. The potential energy, V, is given in the units of kcal/mol, and the integrated path length, s, is given in the units of Å.

straints such as elimination of overall rotation and translation are not required, and the DNEB/L-BFGS method has proved to be reliable for relatively complicated cooperative rearrangements in a number of clusters.

In comparing the performance of the L-BFGS and quenched velocity Verlet (QVV) methods for optimising chains of images we have also investigated a number of alternative QVV schemes. We found that the best approach is to quench the velocity after the coordinate update, so that the velocity response to the new gradient lags one step behind the coordinate updates. However, this slow-response QVV (SQVV) method does not appear to be competitive with L-BFGS.

We have revised our previous scheme [8] for constructing connections between distant minima using multiple transition state searches. Previously we have used an NEB/L-BFGS framework for this purpose, with eigenvector-following refinement of transition state candidates and characterisation of the connected minima using energy minimised approximations to the steepest-descent paths [8, 133]. When the DNEB/L-BFGS approach is used we have found that it is better to spend more effort in the DNEB phase of the calculation, since a number of good transition state guesses can often be obtained even when the number of images is relatively small. In favourable cases a complete path linking the required endpoints may be obtained in one cycle. Of course, this was always the objective of the NEB approach [27–30], but we have not been able to achieve such results reliably for complex paths without the current modifications.

When a number of transition states are involved we still find it more efficient to build up the overall path in stages, choosing endpoints that become progressively closer in space. This procedure has been entirely automated within the `OPTIM` program, which can routinely locate complete paths for highly cooperative multi-step rearrangements, such as those connecting different morphologies of the LJ_{38} and LJ_{75} clusters.

For complex rearrangements the number of elementary steps involved may be rather large, and new methods for constructing an initial path are needed. This path does not need to be the shortest, or the fastest, but it does need to be fully connected. The second most important result of this chapter is probably a connection procedure based upon Dijkstra's shortest path algorithm, which enables us to select the most promising

paths that include missing connections for subsequent double-ended searches. We have found that this approach enables initial paths containing more than a hundred steps to be calculated automatically for a variety of systems. Results have been presented for trpzip peptides here and for buckminsterfullerene, the GB1 hairpin and the villin headpiece subdomain elsewhere [167]. These paths will be employed to seed future discrete path sampling calculations. This approach greatly reduces the computational demands of the method, and has allowed us to tackle more complicated problems. The new algorithm has also been implemented within our `OPTIM` program, and a public domain version is available for download from the Internet [138].

Chapter 3

Properties of Rearrangement Pathways

> There is a difference between knowing the path and walking the path.
>
> Andy and Larry Wachowski, The Matrix

3.1 Introduction

The number of elementary rearrangements increases exponentially with system size as for the number of transition states. For instance, there are approximately 30,000 such pathways on the PES of the 13-atom cluster bound by the Lennard-Jones potential. When permutation-inversion isomers are included, this number increases by a factor of order $2 \times N!$ [9]. For PES's that support such a large number of stationary points a whole range of properties is spanned by the corresponding rearrangement pathways. Understanding these properties can be helpful in answering questions such as why some rearrangement pathways are harder to find than others and whether there is any correlation between these properties that we could potentially utilise in studying PES's.

Two activation barriers can be defined for each pathway in terms of the energy difference between the transition state and each of the minima. For non-degenerate rearrangements [9, 148] the two sides of the path are termed uphill and downhill, where the uphill barrier is the larger one, which leads to the higher minimum. The barriers

and the normal modes of the minima and transition states can be used to calculate rate constants using harmonic transition state theory [45, 46, 48]. More sophisticated treatments based on anharmonic densities of states are possible but can be hard to reconcile with the coarse-grained view of the landscape adopted here, as a detailed knowledge of the basins of attraction is required [168, 169].

For each local minimum a catchment basin can be defined in terms of all the configurations from which steepest-descent paths lead to that minimum [170]. Some of these paths originate from transition states on the boundary of the catchment basin, which connect a given minimum to adjacent minima. The integrated path length for such rearrangements provides a measure of the separation between local minima, and may be related to the density of stationary points in configuration space. The integrated path length is usually approximated as the sum of Euclidean distances between configurations sampled along appropriate steepest-descent paths [9]. It provides a convenient coordinate for monitoring the progress of the reaction.

Calculated pathways can always be further classified mechanistically. For example, some rearrangements preserve the nearest-neighbour coordination shell for all the atoms. In previous studies of bulk models these cage-preserving pathways were generally found to outnumber the more localised cage-breaking processes, which are necessary for atomic transport [171]. It was found that the barriers for cage-breaking and cage-preserving processes were similar for bulk LJ systems, while the cage-breaking mechanisms have significantly higher barriers for bulk silicon modelled by the Stillinger-Weber potential [171].

For minima separated by increasing distances in configuration space, the pathways that connect them are likely to involve more and more elementary steps, and are not unique. Finding such paths in high-dimensional systems can become a challenging task [7, 8]. Some difficulties have been attributed to instabilities and inefficiencies in transition state searching algorithms [7, 107], as well as the existence of very different barrier height and path length scales [7]. A new algorithm for locating multi-step pathways in such cases was presented in the previous chapter.

In the present work we have used the doubly nudged elastic band (DNEB) method [7] in conjunction with eigenvector-following (EF) algorithms [12–24] to locate rearrange-

ment pathways in various systems. The LJ potential was used to describe the 13- and 75-atom Lennard-Jones clusters, LJ_{13} and LJ_{75}. We have also considered a binary LJ (BLJ) system with parameters $\sigma_{AA} = 1.0$, $\sigma_{BB} = 0.88$, $\sigma_{AB} = 0.8$, $\epsilon_{AA} = 1.0$, $\epsilon_{BB} = 0.5$, $\epsilon_{AB} = 1.5$, where A and B are atom types. The mixture with $A : B$ ratio 80 : 20 provides a popular model bulk glass-former [171, 172]. We employed a periodically repeated cubic cell containing 205 A atoms and 51 B atoms. The density was fixed at $1.2\sigma_{AA}^{-3}$ and the Stoddard-Ford scheme was used to prevent discontinuities [173].

The motivation for this work was our observation that construction of some multi-step pathways using the connection algorithm described in Chapter 2 is particularly difficult. We were unable to relate these difficulties to simple properties of the pathways such as the integrated path length, the uphill and downhill barriers, or the barrier and path length asymmetries. Instead, more precise measures of localisation and cooperativity are required, as shown in the following sections. It also seems likely that such tools may prove useful in analysing the dynamics of supercooled liquids, where processes such as intrabasin oscillations and interbasin hopping have been associated with different rearrangement mechanisms [174]. In particular, cooperativity is believed to play an important role at low temperatures in glass-forming systems [175], and dynamic heterogeneity may result in decoupling between structural relaxation and transport properties for supercooled liquids [176].

3.2 LOCALISATION

The outcome of a pathway calculation for an atomic system will generally be a set of intermediate geometries, and the corresponding energies, for points along the two unique steepest-descent paths that link a transition state to two local minima. This discrete representation is a convenient starting point for our analysis of localisation and cooperativity. We number the structures along the path $j = 1,\ 2, \ldots,\ N_f$ starting from one of the two minima and reversing the other steepest-descent path, so that structure N_f corresponds to the other minimum. The transition state then lies somewhere between frames 1 and N_f. We define the three-dimensional vector $\mathbf{r}_i(j)$ to contain the Cartesian coordinates of atom i for structure j, i.e. $\mathbf{r}_i(j) = \Big(X_i(j), Y_i(j), Z_i(j)\Big)$, where $X_i(j)$ is the X coordinate of atom i in structure j, etc.

For each atom i we also define the displacement between structures $j-1$ and j as

$$d_i(j) = \Big|\mathbf{r}_i(j) - \mathbf{r}_i(j-1)\Big|. \tag{3.1}$$

Hence the sum of displacements

$$d_i = \sum_{j=2}^{N_f} d_i(j), \tag{3.2}$$

is an approximation to the integrated path length for atom i, which becomes increasingly accurate for smaller step sizes. The total integrated path length, s, is approximated as

$$s = \sum_{j=2}^{N_f} \sqrt{\sum_{i=1}^{N} d_i\,(j)^2}, \tag{3.3}$$

where N is the total number of atoms. s is a characteristic property of the complete path, and is expected to correlate with parameters such as the curvature and barrier height for short paths [9, 177, 178].

The set $\{d_1, d_2, \ldots, d_N\}$ containing all N values of d_i will be denoted $\{d\}$, and analogous notation will be used for other sets below. We will also refer to the frequency distribution function, which can provide an alternative representation of such data [179]. For example, the frequency distribution function $\mathcal{F}$ for a given continuous variable, x, tells us that x occurs in a certain interval $\mathcal{F}(x)$ times.

Our objective in the present analysis is to provide a more detailed description of the degree of 'localisation' and 'cooperativity' corresponding to a given pathway. The first index we consider is N_p, which is designed to provide an estimate of how many atoms participate in the rearrangement. We will refer to a rearrangement as localised if a small fraction of the atoms participate in the rearrangement, and as delocalised in the opposite limit. The second index we define, N_c, is intended to characterise the number of atoms that move simultaneously, i.e. cooperatively. We will refer to a rearrangement as cooperative if most of the atoms that participate in the rearrangement move simultaneously, and as uncooperative otherwise.

The nth moment about the mean for a data set $\{x_1, x_2, ..., x_M\}$ is the expectation value of $(x_i - \langle x\rangle)^n$, where $\langle x\rangle = \sum_{i=1}^{M} x_i/M$ and M is the number of elements in the

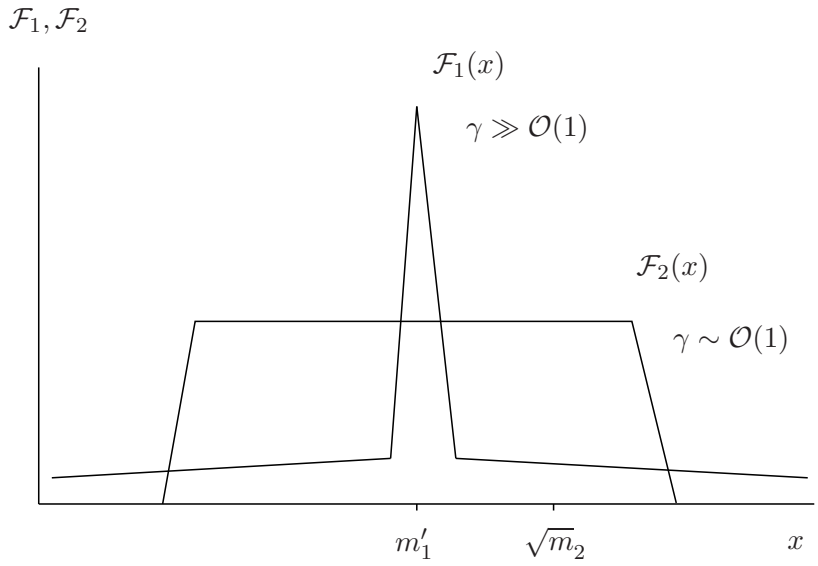


FIGURE 3.1: Two frequency distribution functions $\mathcal{F}_1$ and $\mathcal{F}_2$ of a continuous variable x are contrasted. Both functions have the same average m_1' and standard deviation $\sqrt{m}_2$. However, due to the long tails, $\mathcal{F}_1$ has a significantly larger fourth moment m_4 and, hence, a larger kurtosis, γ. If $\gamma > 3$ the distribution is said to be peaked or leptocurtic. Distributions with $\gamma > 3$ are known as platycurtic or heavy-tailed.

set. Hence for the set $\{d\}$ defined above we define the moments, m_n, as

$$m_n = \frac{1}{N} \sum_{i=1}^{N} (d_i - \langle d \rangle)^n \,. \tag{3.4}$$

The kurtosis of the set $\{d\}$ is then defined as the dimensionless ratio

$$\gamma(d) = \frac{m_4}{(m_2)^2}, \tag{3.5}$$

and provides a measure of the shape of the frequency distribution function corresponding to $\{d\}$. If only one of the atoms moves, or all atoms except one move by the same amount, then $\gamma(d) = N$. Alternatively, if half the atoms move by the same amount whilst the others are stationary, then $\gamma(d) = 1$. Hence, a distribution with a broad peak and rapidly decaying tails will have a small kurtosis, $\gamma \sim \mathcal{O}(1)$, while a distribution with a sharp peak and slowly decaying tails will have a larger value (Figure 3.1). The kurtosis can therefore identify distributions that contain large deviations from the average value [179]. For comparison, a Gaussian distribution has $\gamma = 3$ and a uniform distribution has $\gamma = 1.8$.

The above results show that the kurtosis $\gamma(d)$ can be used to quantify the degree of localisation or delocalisation of a given rearrangement. However, it has the serious disadvantage that highly localised and delocalised mechanisms both have large values of $\gamma(d)$. Since we are interested in estimating the number of atoms that move relative to the number with small or zero displacements, a better approach is to use moments taken about the origin, rather than about the mean, i.e.

$$m'_n = \frac{1}{N}\sum_{i=1}^{N} d_i^n, \tag{3.6}$$

following Stillinger and Weber [152]. Note that while $m_1 = 0$, the first moment m'_1 is the mean value. We therefore estimate the number of atoms that participate in the rearrangement, N_p, as

$$N_p = \frac{N}{\gamma'(d)}, \tag{3.7}$$

where

$$\gamma'(d) = \frac{m'_4}{(m'_2)^2}. \tag{3.8}$$

For the system with N atoms, if only one atom moves $N_p = 1$, while if K atoms move by the same amount, $N_p = K$.

A similar index to N_p has been employed in previous work [9, 19, 180] using only the displacements between the two local minima, which corresponds to taking $N_f = 2$ in Equation 3.2. Using d_i values based upon a sum of displacements that approximates the integrated path length for atom i, rather than the overall displacement between the two minima, better reflects the character of the rearrangement, as it can account for the nonlinearity of the pathway. To describe this property more precisely we introduce a pathway nonlinearity index defined by

$$\alpha = \frac{s - D}{s}, \tag{3.9}$$

where D is the Euclidean distance between the endpoints,

$$D = \sqrt{\sum_{i=1}^{N} (\mathbf{r}_i(N_f) - \mathbf{r}_i(1))^2}. \tag{3.10}$$

We calculated the α values for a database of 31,342 single transition state pathways of LJ_{75} (hereafter referred to as the LJ_{75} database). The average value of α was 0.4

with a standard deviation of 0.2, and, hence, there is a significant loss of information if γ' is calculated only from the endpoints using $N_f = 2$. Comparison of the two indices for the LJ_{75} database revealed many examples where neglect of intermediate structures produces a misleading impression of the number of atoms that move. The definition in Equation 3.7 is therefore suggested as an improvement on previous indices of localisation [9, 19, 152, 180].

3.3 Cooperativity

N_p atoms can participate in a rearrangement according to a continuous range of co-operativity. At one end of the scale there are rearrangements where N_p atoms all move simultaneously [see Figure 3.2 (a)]. Although these paths exhibit the highest degree of correlated atomic motion they do not usually pose a problem for double-ended transition state search algorithms [7, 30]. Linear interpolation between the two minima tends to generate initial guesses that lie close to the true pathway, particularly if $\alpha \sim 0$. At the opposite extreme, atoms can move almost one at a time, following a 'domino' pattern [see Figure 3.2 (b)]. Locating a transition state for such rearrangements may require a better initial guess, since linear interpolation effectively assumes that all the coordinates change at the same rate.

The degree of correlation in the atomic displacements can be quantified by considering the displacement 'overlap'

$$O_k = \sum_{j=2}^{N_f} O_k\left(j\right) = \sum_{j=2}^{N_f} \min\Big[d_{c(1)}(j), d_{c(2)}(j), ..., d_{c(k)}(j)\Big], \tag{3.11}$$

where the index k indicates that O was calculated for k atoms numbered $c(1)$, $c(2)$,...,$c(k)$. $\mathbf{c}$ is a k-dimensional vector that represents a particular choice of k atoms from N, and hence there are $C_N^k = N!/k!(N-k)!$ possible values of O_k. The index O can be thought of as a measure of how the displacements of the atoms $c(1)$, $c(2)$, etc. overlap along the pathway. For example, if two atoms move at different times then O_2 is small for this pair because the minimum displacement in Equation 3.11 is always small. However, if both atoms move in the same region of the path then O_2 is larger.

We now explain how the statistics of the overlaps, O_k, can be used to extract a measure of cooperativity (Figure 3.3). Suppose that m atoms move simultaneously in

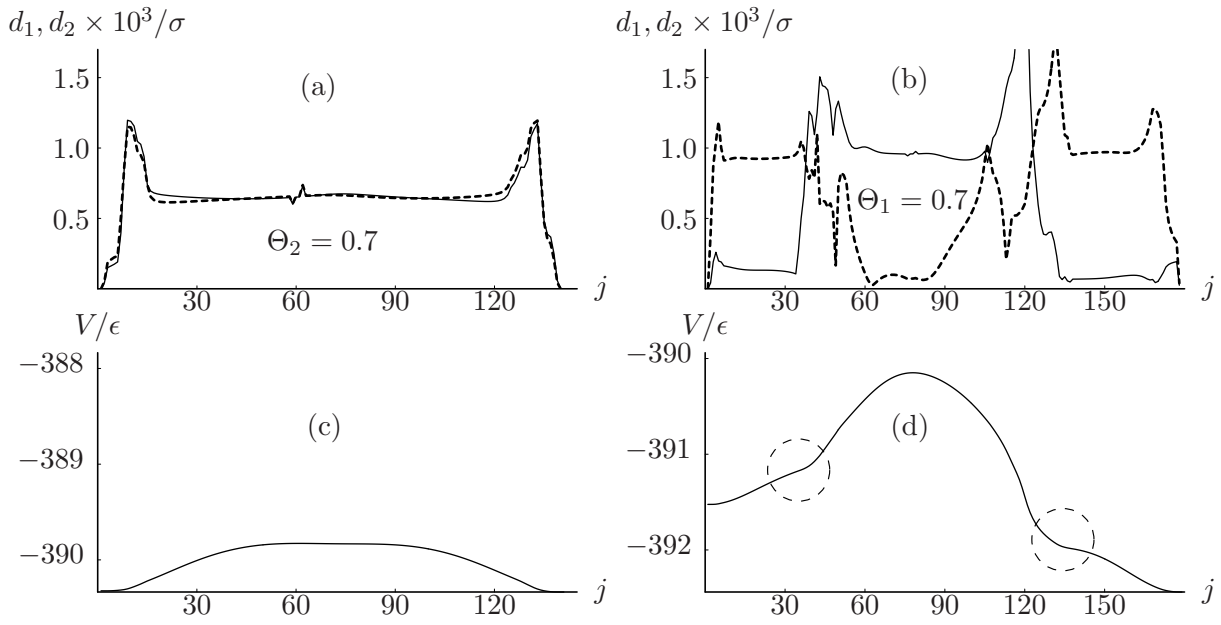


FIGURE 3.2: Comparison of cooperative (a) and uncooperative (b) rearrangements of the LJ_{75} cluster, for mechanisms that are localised mainly on two atoms. The displacement d as a function of the frame number j is shown for the two atoms that move the most. Panels (c) and (d) illustrate the potential energy V as a function of frame number j for the rearrangements in (a) and (b), respectively. Dashed circles indicate flatter parts of the energy profile, which correspond to the most cooperative regions of the pathway.

a hypothetical rearrangement. Then all the overlaps O_k for $k > m$ will be relatively small, because one or more atoms are included in the calculation whose motion is uncorrelated with the others. For overlaps O_k with $k \leqslant m$ the set of O_k for all possible choices of k atoms from N will exhibit some large values and some small. The large values occur when all the chosen atoms are members of the set that move cooperatively, while other choices give small values of O_k. Hence the kurtosis of the set $\{O_k\}$, $\gamma'(O_k)$, calculated from moments taken about the origin, will be large for $k \leqslant m$, and small for $k > m$.

To obtain a measure of how many atoms move cooperatively we could therefore calculate $\gamma'(O_2)$, $\gamma'(O_3)$, etc. and look for the value of k where $\gamma'(O_k)$ falls in magnitude. However, to avoid an arbitrary cut-off, it is better to calculate the kurtosis of the set $\{\gamma'(O_2), \gamma'(O_3), ..., \gamma'(O_k)\}$, or $\gamma'[\gamma'(O)]$ for short. There are $N-2$ members of this set, and by analogy with the definition of $N_p = N/\gamma'(d)$, we could define a cooperativity

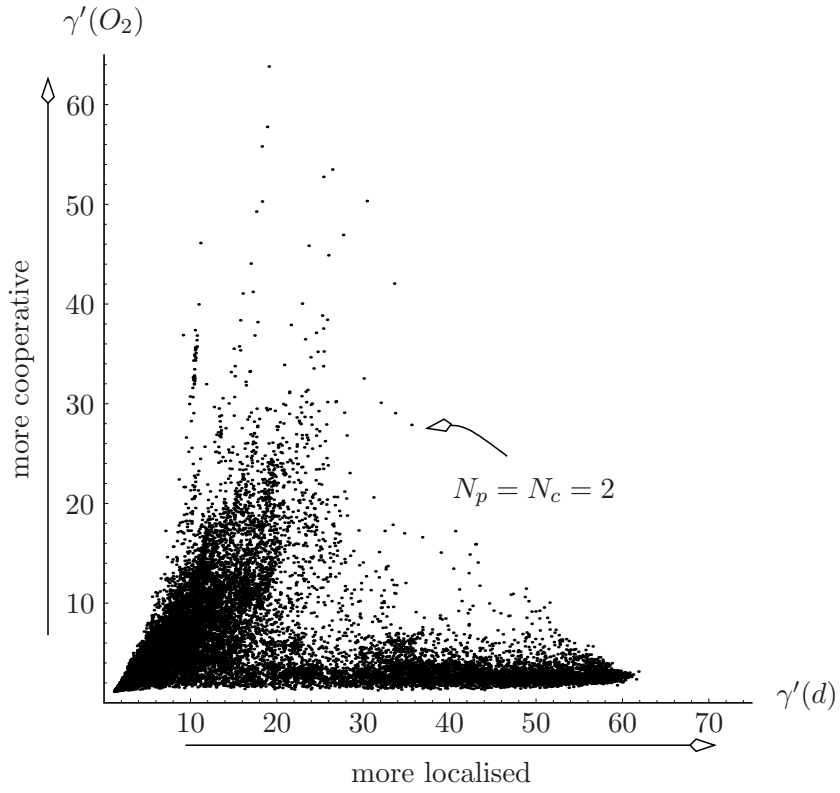


FIGURE 3.3: $\gamma'(O_2)$ plotted against $\gamma'(d)$ for the LJ_{75} pathway database. The figure shows how $\gamma'(O_2)$ can discriminate between rearrangements that have similar values of $\gamma'(d)$ but different cooperativity. The data point for the most cooperative rearrangement localised on two atoms has $N_p = N_c = 2$.

index $N_c = (N-2)/\gamma'[\gamma'(O)] + 1$. Then, if $\gamma'(O_2)$ is large, and all the other $\gamma'(O_k)$ are small, we obtain $\gamma'(\gamma'(O)) \sim N - 2$ and $N_c \sim 2$, correctly reflecting the number of atoms that move together.

In practice, there are several problems with the above definition of N_c. Calculating N_c in this way quickly becomes costly as the number of atoms and/or number of frames in the pathway increases, because the number of elements in the set $\{O_k\}$ varies combinatorially with k. Secondly, as k approaches N the distribution of all the possible values for O_k becomes more and more uniform. Under these circumstances deviations from the mean that are negligible in comparison with the overall displacement can produce large kurtosises. Instead, we suggest a modified (and simpler) definition of N_c, which better satisfies our objectives.

We first define the overlap of atomic displacements in a different manner. It can be

seen from Equation 3.11 that the simultaneous displacement of l atoms is included in each set of overlaps $\{O_k\}$ with $k \leqslant l$. For example, if three atoms move cooperatively then both the $\{O_2\}$ and $\{O_3\}$ sets will include large elements corresponding to these contributions. Another redundancy is present within $\{O_k\}$, since values in this set are calculated for all possible subsets of k atoms and the displacement of each atom is therefore considered more than once. However, we can avoid this redundancy by defining a single $k-$overlap, rather than dealing with C_N^k different values.

Recall that $d_i(j)$ is the displacement of atom i between frames $j-1$ and j. The ordering of the atoms is arbitrary but remains the same for each frame number j. We now define $\Delta_i(j)$ as the displacement of atom i in frame j, where index i numbers the atoms in frame j in descending order, according to the magnitude of $d_i(j)$, e.g. atom 1 in frame 2 is now the atom with the maximum displacement between frames 1 and 2, atom 2 has the second largest displacement etc. As the ordering may vary from frame to frame, the atoms labelled i in different frames can now be different. This relabelling greatly simplifies the notation we are about to introduce. Consider the k-overlap defined as

$$\Theta_k = \frac{1}{\Delta_{tot}} \sum_{j=2}^{N_f} \Big[\Delta_k(j) - \Delta_{k+1}(j) \Big], \tag{3.12}$$

where k ranges from 1 to N, $\Delta_{tot} = \sum_{j=2}^{N_f} \Delta_1(j)$ and $\Delta_{N+1}(j)$ is defined to be zero for all j. A schematic illustration of this construct is presented in Figure 3.4. For example, if only two atoms move in the course of the rearrangement, and both are displaced by the same amount (which may vary from frame to frame), the only non-zero overlap will be Θ_2.

We can now define an index to quantify the number of atoms that move cooperatively as

$$N_c = \sum_{k=1}^{N} k\Theta_k. \tag{3.13}$$

If only one atom moves during the rearrangement then $N_c = 1$, while if K atoms displace cooperatively during the rearrangement then $N_c = K$. This definition is independent of the total displacement, the integrated path length, and the number of atoms, which makes it possible to compare N_c indices calculated for different systems.

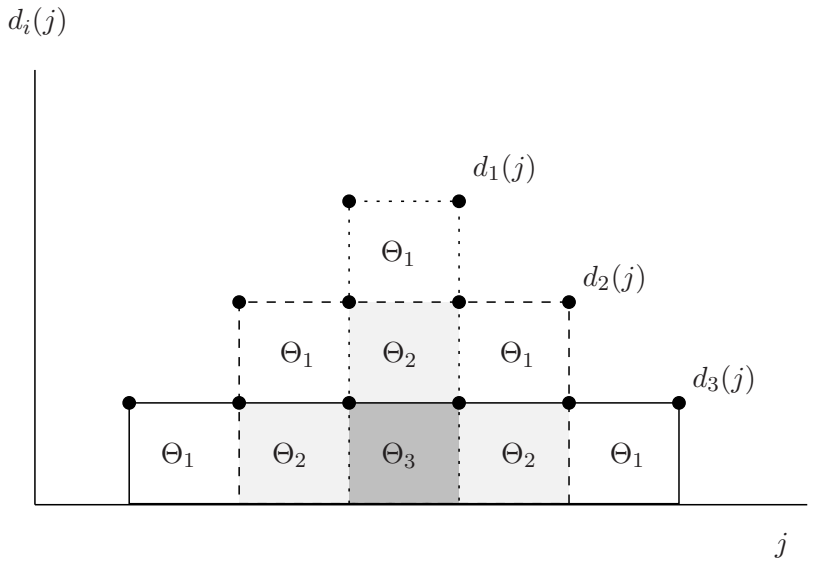


Figure 3.4: The Θ indices for a hypothetical rearrangement localised on three atoms. For each of these atoms the displacement d as a function of frame number j is shown. The d_i values in successive frames are connected with dotted (d_1), dashed (d_2) and solid (d_3) lines. The corresponding contributions to Θ_1, Θ_2, and Θ_3 are shown as shaded squares and are labelled accordingly. If the remaining $N-3$ atoms do not participate, and the area of one square is S, the only non-zero overlaps will be Θ_1, Θ_2, and Θ_3 with values 5/9, 3/9 and 1/9, respectively.

3.4 Applications to LJ_{13} and LJ_{75} Clusters and BLJ_{256} Liquid

Figure 3.2 shows results for the most cooperative and uncooperative processes we have found for the LJ_{75} cluster that are localised mainly on two atoms. In these calculations we have used the database of transition states that was found previously as the result of a discrete path sampling calculation conducted for this system [8, 10]. The cooperative rearrangement [Figure 3.2(a,c)] is the one with the maximum two-overlap Θ_2. For this pathway $\Theta_2 = 0.7$, $N_p = 3.4$, and $N_c = 7.7$. The values of N_p and N_c both reflect the fact that the motion of the two atoms is accompanied by a slight distortion of the cluster core. This example shows that while N_p and N_c allow us to quantify localisation and cooperativity, and correctly reflect the number of atoms that participate and move cooperatively in ideal cases, there will not generally be a simple correspondence between their values and the number of atoms that move. This complication is due to the fact that small displacements of atoms in the core will generally occur, no matter how localised the rearrangement is. In addition, the data reduction

performed in Equation 3.7 and Equation 3.13 means that a range of pathways can give the same value for N_p or N_c. Since the size of the contribution from a large number of small displacements depends on the shape of the displacement distribution function the number of possibilities grows with the size of the index.

The uncooperative rearrangement depicted in the Figure 3.2(b,d) was harder to identify. In principle we could have selected the pathway that maximises Θ_1 from all the rearrangements localised on two atoms. However, this approach picks out rearrangements localised on one atom, where distortion of the core accounts for the value of $N_p > 1$. Instead, we first selected from all the rearrangements with $N_p < 4$ those where two atoms move by approximately the same amount, while the displacement of any other atom is significantly smaller. These are the rearrangements that maximise $1/\gamma(\{d_1, d_2\})$, where d_1 and d_2 are the total displacements of the two atoms that move the most. After this procedure we selected the rearrangement with the maximum value of Θ_1. Figure 3.2 (b) shows that this rearrangement features the displacement of one atom at a time, and the atom that moves first also moves last. For this pathway the values of Θ_1, N_p and N_c are 0.7, 3.8 and 5.3, respectively. Further illustrations and movies of the corresponding rearrangements are available online [181].

Figure 3.2 illustrates several general trends that we have observed for cluster rearrangements. Firstly, we have found that the barrier height is smaller for the cooperative rearrangements [Figure 3.2(c,d)]. Usually atoms that move cooperatively are neighbours. Rearrangements generally involve a change of the environment for the atoms that move. Cooperative motion can reduce this perturbation since for any of the participants the local environment is partly preserved because it moves with the atom in question. Flatter points on the energy profile [circled in Figure 3.2(d)] usually signify a change in the mechanism, i.e. one group of atoms stops moving and another group starts. By comparing (b) and (d) in Figure 3.2 we conclude that flatter points on the energy profile correlate with the most cooperative parts of this rearrangement.

A simple correlation between barrier heights and N_p and N_c does not seem to exist. The barrier height is not a function of cooperativity alone, but also of the energetics of the participating atoms. The way the N_p and N_c indices have been defined can make them insensitive to details of the rearrangements that will affect the energetics. For

instance, neither index depends on the location of the participating atoms or the directionality of their motion. In most cases cooperatively moving particles are adjacent, i.e. localised in space; however, long-distance correlations of atomic displacements also occur. One such case is depicted in Figure 3.5 (a). This path is nearly symmetric with respect to the integrated path length ($\pi = 0.01$), but is very asymmetric with respect to the uphill and downhill barrier heights ($\beta = 0.91$). This rearrangement has $N_p = N_c = 10$. Interestingly, N_p and N_c calculated separately for both sides of the pathway are very similar, i.e. the two steepest-descent pathways cannot be distinguished using these indices. Close inspection of this rearrangement reveals that one side of the pathway involves the rearrangement of two atomic triplets that share a vertex, while the other side involves the drift of all five atoms on the surface of the cluster [see the insets in Figure 3.5 (a)]. Although N_c does not distinguish these cases, the motion in the second side of the path is more cooperative. The participating atoms move together, which results in a significantly lower downhill barrier.

N_p and N_c also describe properties of the whole pathway. A significant number of pathways that we observed were rather non-uniform, i.e. very cooperative phases alternated with uncooperative ones. To distinguish such pathways in the LJ_{75} database we calculated a set $\{N_p\}$ containing N_p's evaluated for each pair of adjacent frames. Then a selection of pathways was made with $m_2/(m_1')^2 < 0.01$, where $m_2/(m_1')^2$ is the moment ratio evaluated for the set $\{N_p\}$. While this procedure ensured that N_p corresponds closely to the number of atoms that moves between any two snapshots of the rearrangement, it did not distinguish cases where different atoms contribute to the value of N_p in different frames [see Figure 3.5 (b)]. The average uphill and downhill barriers for this subset of rearrangements are 100 times smaller than the average barriers for the complete LJ_{75} database (Table 3.1). Figure 3.6 shows that N_p and N_c calculated for these rearrangements are highly correlated and span a range of values, implying that widely different pathways are represented. Finally, all the selected pathways are an order of magnitude shorter than the average path length for the whole database, even though this database contains many short rearrangements localised on one atom.

Figure 3.7 shows the values of the N_p and N_c indices plotted against each other for

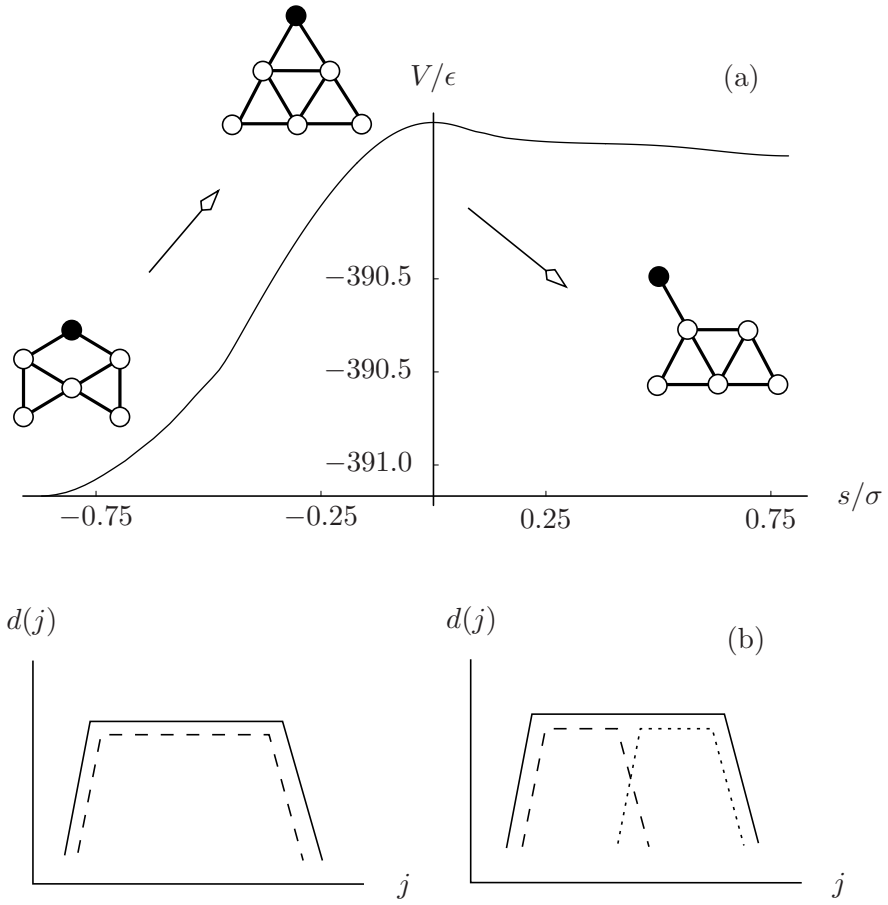


Figure 3.5: Two limitations of the cooperativity index N_c. (a) N_c is not sensitive to the spatial positions of the cooperatively moving atoms, nor to the directionality of their motion. The energy profile is depicted for a rearrangement of the LJ_{75} cluster, which is very asymmetric with respect to barrier height but has similar integrated path lengths on either side of the transition state. The cooperativity index N_c evaluated separately for the two sides is about 10 in both cases. The motion of the five atoms that displace the most is shown schematically relative to a reference atom (black). (b) The displacement of two (left) and three (right) atoms d as a function of the frame number j is shown schematically for a hypothetical pathway. The rearrangement on the left is more cooperative because two atoms move together over a longer region of the path. However, the current definition of N_c does not distinguish between these two cases.

pathway databases calculated for LJ_{13}, LJ_{75} and BLJ_{256}. The two databases for BLJ_{256} labelled as 1 and 12 are taken from Reference [171] and correspond to databases BLJ1 and BLJ12 in that paper. BLJ1 and BLJ12 were obtained using two different sampling schemes intended to provide an overview of a wide range of configuration space and

Table 3.1: Average uphill and downhill barriers, average integrated path length for the LJ_{75} rearrangement pathway database. Values are given for the whole database containing 31,342 paths, and for a subset containing the 57 most cooperative paths. The units of energy and distance are ϵ and σ, respectively.

	All	Cooperative
Uphill barrier	3.03	0.06
Downhill barrier	0.97	0.03
Path length	3.08	0.58

a thorough probe of a smaller region, respectively. These databases were constructed by systematic exploration of the PES, and we refer the reader to the original work for further details [171]. Each BLJ_{256} database contains 10,000 transition states. The LJ_{13} and LJ_{75} databases were obtained in discrete path sampling (DPS) studies [8, 10]. and contain 28,756 and 31,342 transition states, respectively. Figure 3.7 is a density plot where darker shading signifies a higher concentration of data points. The outlying points are connected by a solid line to define the area in which all the points lie. Figure 3.7 shows that as N_p grows the allowed range of N_c increases, especially for LJ_{13}. For the LJ_{75} database rearrangements with $N_c > N/2$ appear to be very rare or poorly sampled. Figure 3.7 also shows that for all these systems rearrangements localised on two or three atoms dominate. This result may be an intrinsic property. However, it may also be due to the geometric perturbation scheme used in producing the starting points for the transition state searches employed in generating these databases. For databases BLJ1 and BLJ12 there are significantly more rearrangements with larger values of N_p and N_c compared to LJ_{75}, which suggests that the abundance of very localised rearrangements for clusters may be a surface effect. The apparent absence of cooperative rearrangements in LJ_{75} database for large values of N_p may be due to the fact that only the pathways between compact phases of this cluster were sampled thoroughly, i.e. rearrangements involving liquid-like structures that are expected to have larger values of N_c are probably underrepresented.

Figure 3.8 depicts the average barrier as a function of the participation and cooper-

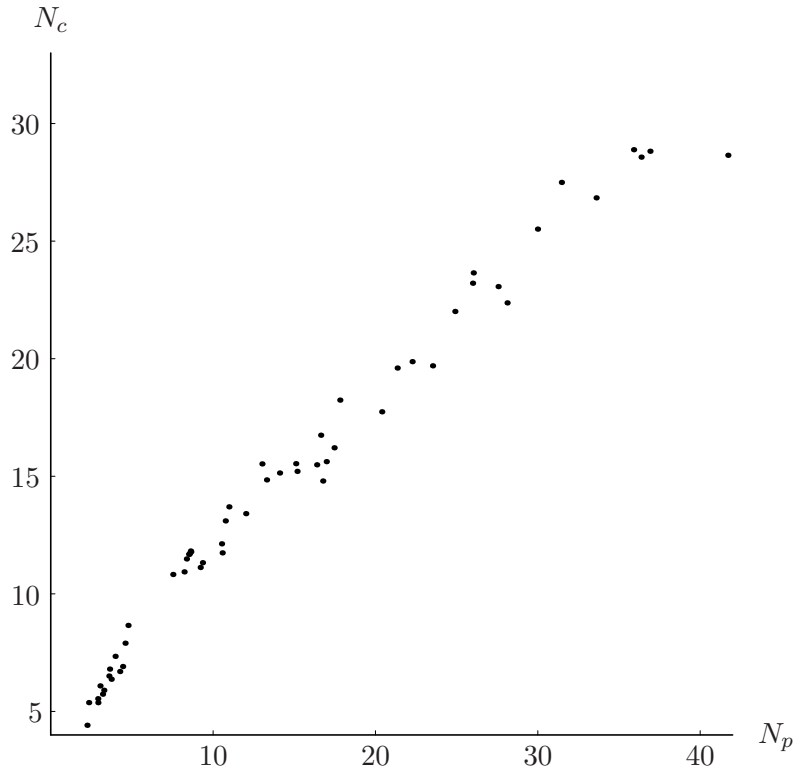


Figure 3.6: N_c as a function of N_p calculated for the 57 most cooperative rearrangements from the LJ_{75} pathway database.

ativity indices. N_p and N_c were calculated separately for both sides of the pathway and the corresponding barriers (uphill or downhill) were averaged to produce a density plot of barrier height. Data boundaries do not coincide with those shown in Figure 3.7 due to numerical imprecision in preparing Figure 3.7. Figure 3.8 illustrates that for each system cooperative rearrangements have the lowest barriers, irrespective of the value of N_p. For clusters, cooperative rearrangements have lower barriers than uncooperative rearrangements with N_p as small as $1-3$, while for bulk barriers corresponding to rearrangements with low N_p become comparable to these for very cooperative rearrangements with high N_p.

In further computational experiments we found that attempts to connect the endpoints of uncooperative pathways using the algorithm described in Chapter 2 either required more images and iterations or converged to an alternative pathway. In some cases additional difficulties arose, such as convergence to a higher index saddle instead of a transition state, which can happen if the linear interpolation guess conserves a

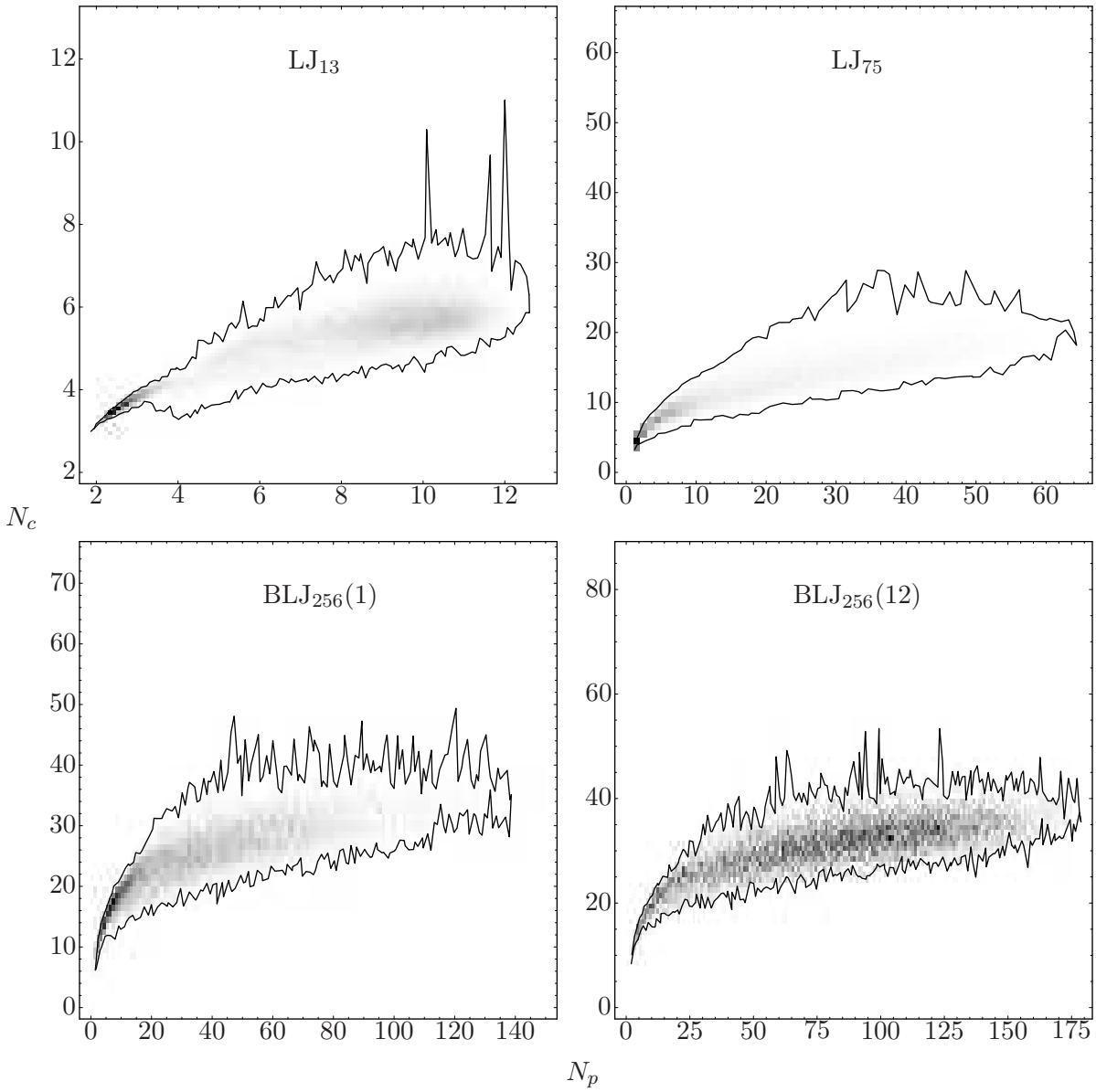


FIGURE 3.7: N_c as a function of N_p calculated from four pathway databases for LJ_{13}, LJ_{75} and BLJ_{256} systems. Due to the large number of data points we employ a density plot, where the darkest shading corresponds to the highest concentration of points. Outlying points are connected to illustrate the boundaries of the data area. The two BLJ databases are taken from Reference [171].

symmetry plane. Figure 3.9 shows N_c calculated from Equation 3.13 plotted against N_p for the LJ_{75} pathway database. Knowing the integrated path length, s, for each pathway we started doubly nudged elastic band calculations with three images per unit of distance and 30 iterations per image. Most of the points that correspond to runs

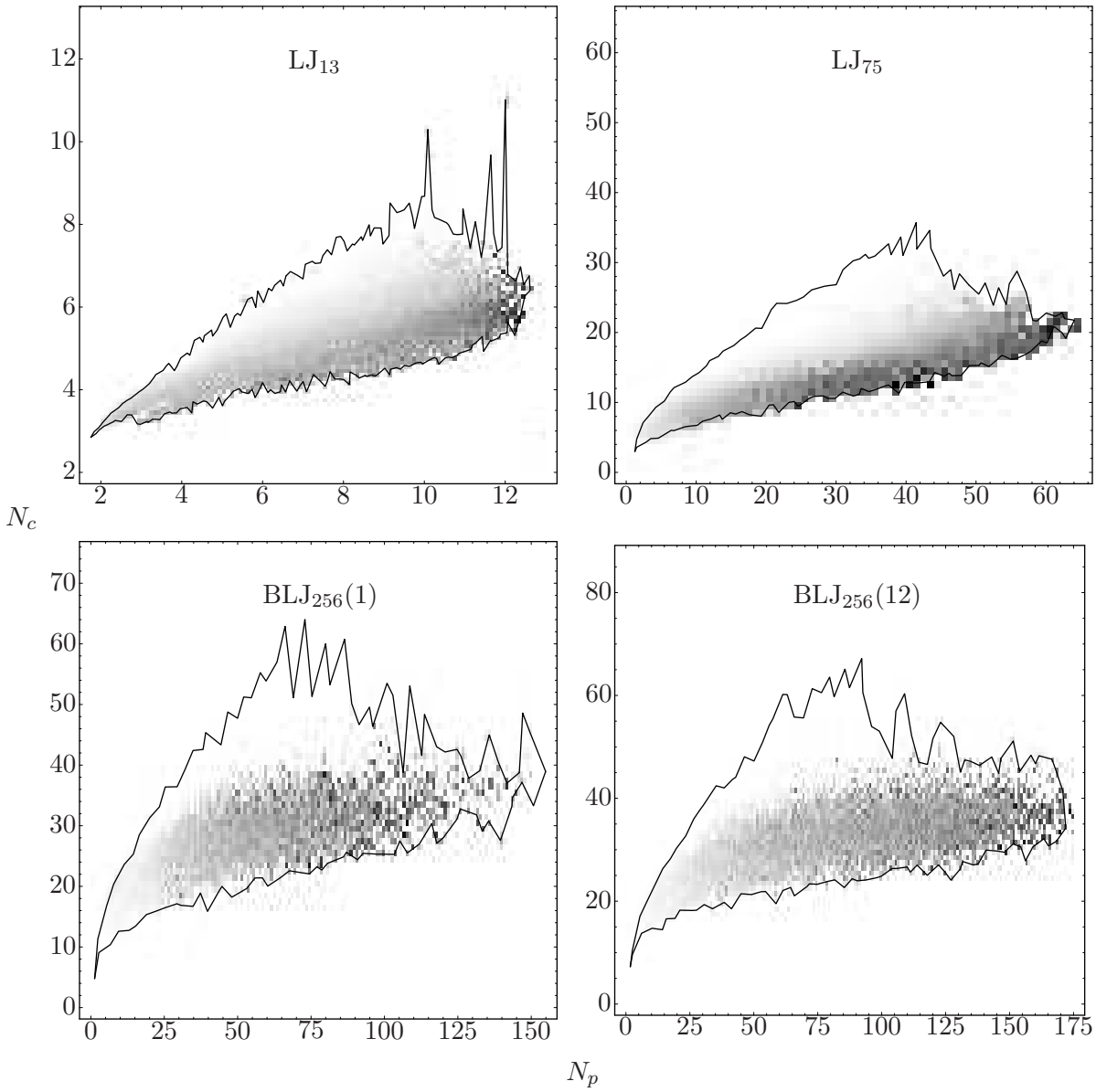


FIGURE 3.8: Average barrier height as a function of N_p and N_c calculated for the same LJ_{13}, LJ_{75} and BLJ_{256} pathway databases used in Figure 3.7. The indices were calculated separately for the two sides of each path. In this case the darkest shading corresponds to the highest barriers. Outlying points are connected to illustrate the boundaries of the data area. The two BLJ databases are taken from Reference [171].

that failed or converged to an alternative pathways are concentrated in the region of small values for N_c/N_p.

As can be seen from Figure 3.7, the LJ_{13} database contains significantly more pathways with large values of N_p and N_c compared to LJ_{75}, where most of the re-

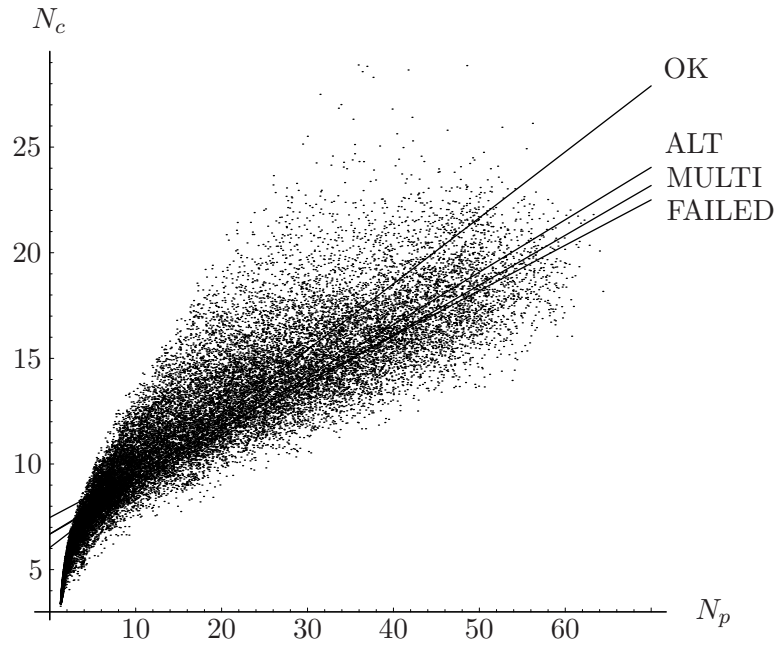


Figure 3.9: N_c as a function of N_p calculated for the LJ_{75} pathway database. For each pathway we conducted DNEB calculations [7] assuming prior knowledge of the path. Every DNEB calculation employed $3s$ images and $90s$ iterations in each case, where s is the integrated path length. Out of $31,342$ DNEB runs $25,158$ yielded a connected pathway while the rest did not (FAILED). Connected pathways are classified further as one-step pathways involving the correct transition state (OK) or an alternative transition state (ALT), or as multi-step pathways (MULTI), which involve more than one transition state. For each set of data points best fit straight lines obtained from linear regression are shown and labelled appropriately.

arrangements are localised on two atoms. The fact that the LJ_{13} database is almost exhaustive then suggests that localised rearrangements either start to dominate as the system size increases or that the sampling scheme used for LJ_{75} was biased towards such mechanisms. Systematic sampling of the configuration space for stationary points often employs perturbations of every degree of freedom followed by minimisation [162]. The LJ_{13} database was obtained in this fashion, while the LJ_{75} database was generated during the DPS approach [8]. In this procedure discrete paths are perturbed by replacing local minima with structures obtained after perturbing all the coordinates and minimising. To investigate whether the perturbation scheme can affect the resulting database of stationary points in more detail we consider the case of LJ_{13}, since nearly all the transition states are known. Figure 3.10 presents the results of two independent

runs aimed at locating most of the transition states for this system. Every cycle a perturbation was applied to a randomly selected transition state from the database and the resulting geometry was used as a starting point for a new transition state search using eigenvector-following [19, 22–24, 72]. Only distinct permutation-inversion isomers were saved. In the first run (bottom curve) every degree of freedom was perturbed by $0.4x$, where x is a random number in the interval $[-1, 1]$ [162]. For the second run (top curve) we introduced a perturbation scheme including correlation. $2 \leqslant K \leqslant N/2$ atoms out of N were displaced by a vector $0.4(x_1, x_2, x_3)$, where the components x_1, x_2 and x_3 are again random numbers drawn from $[-1, 1]$. The K atoms to be displaced were selected based on their relative positions in the cluster. One atom was first selected at random, while the remaining $K - 1$ were chosen to be its closest neighbours. The top curve was generated from a run with $K = 6$. Both runs required approximately the same time to produce two nearly identical databases, each containing about $29,000$ pathways. However, as can be seen from Figure 3.7, random perturbation of all the degrees of freedom results in uncooperative rearrangements being found first, while employing correlated perturbations has the opposite effect.

3.5 Summary

The most important result of this chapter is probably the introduction of an index to quantify the cooperativity of atomic rearrangements. With this new measure it becomes possible to correlate cooperativity and barrier heights, and to show that cooperative rearrangements generally have lower barriers and shorter path lengths. We hope that these results will shed new light on relaxation mechanisms in complex systems, such as glasses and biomolecules, in future applications. For example, in a peptide or protein a large geometrical change can result from a rearrangement that could be described in terms of a single dihedral angle. In glasses and supercooled liquids an important research goal is to understand how observed dynamical properties, such as atomic diffusion and correlation functions [171, 174, 182, 183], are related to features of the underlying PES. The classification of elementary rearrangements as ‘cage-breaking’ or ‘cage-preserving’ [171, 184], and the emergence of structures such as ‘megabasins’ [171, 184–186] can now be investigated more precisely in terms of locali-

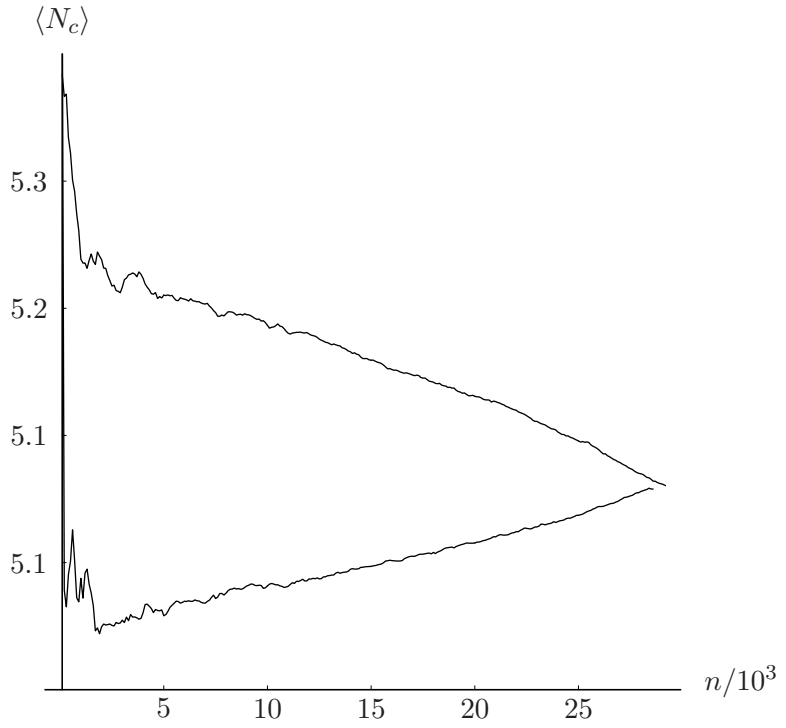


FIGURE 3.10: The average value of N_c for LJ_{13} pathway databases as new paths are added. 6-atom correlated perturbations (top curve) and random perturbations of every degree of freedom (bottom curve) were used to produce starting points for refinement by eigenvector-following [19, 22–24, 72]. Average values were calculated every time 100 new pathways were added.

sation and cooperativity.

We have also demonstrated that cooperative rearrangements are relatively easy to characterise using double-ended transition state searching algorithms, since linear interpolation produces an effective initial guess. Uncooperative rearrangements are usually harder to find using such methods, and alternative initial guesses may be helpful in these cases.

Single-ended transition state searching has been used both in conjunction with double-ended methods, and as a way to sample potential energy surfaces for stationary points. Stationary point databases constructed using random perturbations followed by quenching are likely to be biased towards uncooperative rearrangements. We have therefore outlined a strategy for generating initial guesses appropriate to single-ended transition state searching algorithms, which instead favours cooperative rearrangements. This approach also includes a parameter that is likely to influence the

degree of localisation.

Chapter 4

Ensembles of Rearrangement Pathways

I dreamed a thousand new paths...
I woke and walked my old one.

Chinese proverb

4.1 Introduction

Stochastic processes are widely used to treat phenomena with random factors and noise. Markov processes are an important class of stochastic processes for which future transitions do not depend upon how the current state was reached. Markov processes restricted to a discrete, finite, or countably infinite state space are called Markov chains [123, 187, 188]. The parameter that is used to number all the states in the state space is called the time parameter. Many interesting problems of chemical kinetics concern the analysis of finite-state samples of otherwise infinite state space [9].

When analysing the kinetic databases obtained from discrete path sampling (DPS) studies [8] it can be difficult to extract the phenomenological rate constants for processes that occur over very long time scales [9]. DPS databases are composed of local minima of the potential energy surface (PES) and the transition states that connect them. While minima correspond to mechanically stable structures, the transition states specify how these structures interconvert and the corresponding rates. Whenever the potential

energy barrier for the event of interest is large in comparison with k_BT the event becomes rare, where T is the temperature and k_B is Boltzmann's constant.

The most important tools previously employed to extract kinetic information from a DPS stationary point database are the master equation [189], kinetic Monte Carlo [190, 191] (KMC) and matrix multiplication (MM) methods [8]. The system of linear master equations in its matrix formulation can be solved numerically to yield the time evolution of the occupation probabilities starting from an arbitrary initial distribution. This approach works well only for small problems, as the diagonalisation of the transition matrix, $\mathbf{P}$, scales as the cube of the number of states [9]. In addition, numerical problems arise when the magnitude of the eigenvalues corresponding to the slowest relaxation modes approaches the precision of the zero eigenvalue corresponding to equilibrium [192]. The KMC approach is a stochastic technique that is commonly used to simulate the dynamics of various physical and chemical systems, examples being formation of crystal structures [193], nanoparticle growth [194] and diffusion [195]. The MM approach provides a way to sum contributions to phenomenological two-state rate constants from pathways that contain progressively more steps. It is based upon a steady-state approximation, and provides the corresponding solution to the linear master equation [189, 196]. The MM approach has been used to analyse DPS databases in a number of systems ranging from Lennard-Jones clusters [8, 10] to biomolecules [133, 197].

Both the standard KMC and MM formulations provide rates at a computational cost that generally grows exponentially as the temperature is decreased. In this chapter we describe alternative methods that are deterministic and formally exact, where the computational requirements are independent of the temperature and the time scale on which the process of interest takes place.

4.1.1 Graph Theory Representation of a Finite-state Markov Chain

In general, to fully define a Markov chain it is necessary to specify all the possible states of the system and the rules for transitions between them. Graph theoretical representations of finite-state Markov chains are widely used [187, 198–200]. Here we adopt a digraph [154, 201] representation of a Markov chain, where nodes represent

the states and edges represent the transitions of non-zero probability. The edge $e_{i,j}$ describes a transition from node j to node i and has a probability $P_{i,j}$ associated with it, which is commonly known as a routing or branching probability. A node can be connected to any number of other nodes. Two nodes of a graph are adjacent if there is an edge between them [202].

For digraphs all connections of a node are classified as incoming or outgoing. The total number of incoming connections is the in-degree of a node, while the total number of outgoing connections is the out-degree. In a symmetric digraph the in-degree and out-degree are the same for every node [154]. $AdjIn[i]$ is the set of indices of all nodes that are connected to node i via incoming edges that finish at node i. Similarly, $AdjOut[i]$ is the set of the indices of all the nodes that are connected to node i via outgoing edges from node i. The degree of a graph is the maximum degree of all of its nodes. The expectation value for the degree of an undirected graph is the average number of connections per node.

For any node i the transition probabilities $P_{j,i}$ add up to unity,

$$\sum_j P_{j,i} = 1, \tag{4.1}$$

where the sum is over all $j \in AdjOut[i]$. Unless specified otherwise all sums are taken over the set of indices of adjacent nodes or, since the branching probability is zero for non-adjacent nodes, over the set of all the nodes.

In a computer program dense graphs are usually stored in the form of adjacency matrices [154]. For sparse graphs [201] a more compact but less efficient adjacency-lists-based data structure exists [154]. To store a graph representation of a Markov chain, in addition to connectivity information (available from the adjacency matrix), the branching probabilities must be stored. Hence for dense graphs the most convenient approach is to store a transition probability matrix [187] with transition probabilities for non-existent edges set to zero. For sparse graphs, both the adjacency list and a list of corresponding branching probabilities must be stored.

4.1.2 The Kinetic Monte Carlo Method

The KMC method can be used to generate a memoryless (Markovian) random walk and hence a set of trajectories connecting initial and final states in a DPS database.

Many trajectories are necessary to collect appropriate statistics. Examples of pathway averages that are usually obtained with KMC are the mean path length and the mean first passage time. Here the KMC trajectory length is the number of states (local minima of the PES in the current context) that the walker encounters before reaching the final state. The first passage time is defined as the time that elapses before the walker reaches the final state. For a given KMC trajectory the first passage time is calculated as the sum of the mean waiting times in each of the states encountered.

Within the canonical Metropolis Monte Carlo approach a step is always taken if the proposed move lowers the energy, while steps that raise the energy are allowed with a probability that decreases with the energy difference between the current and proposed states [32]. An efficient way to propagate KMC trajectories was suggested by Bortz, Kalos, and Lebowitz (BKL) [190]. According to the BKL algorithm, a step is chosen in such a way that the ratios between transition probabilities of different events are preserved, but rejections are eliminated. Figure 4.1 explains this approach for a simple discrete-time Markov chain. The evolution of an ordinary KMC trajectory is monitored by the 'time' parameter n, $n \in \mathbb{W}$, which is incremented by one every time a transition from any state is made. The random walker is in state 1 at time $n = 0$. The KMC trajectory is terminated whenever an absorbing state is encountered. As $P_{1,1}$ approaches unity transitions out of state 1 become rare. To ensure that every time a random number is generated (one of the most time consuming steps in a KMC calculation) a move is made to a neighbouring state we average over the transitions from state 1 to itself to obtain the Markov chain depicted in Figure 4.1 (b). Transitions from state 1 to itself can be modelled by a Bernoulli process [33] with the probability of success equal to $P_{1,1}$. The average time for escape from state 1 is obtained as

$$\tau_1 = (1 - P_{1,1}) \sum_{n=0}^{\infty} (n+1)(P_{1,1})^n = \frac{1}{(1 - P_{1,1})}, \tag{4.2}$$

which can be used as a measure of the efficiency of trapping [203]. Transition probabilities out of state 1 are renormalised:

$$\begin{aligned} P_{\alpha,1'} &= \frac{P_{\alpha,1}}{1 - P_{1,1}}, \\ P_{\beta,1'} &= \frac{P_{\beta,1}}{1 - P_{1,1}}. \end{aligned} \tag{4.3}$$

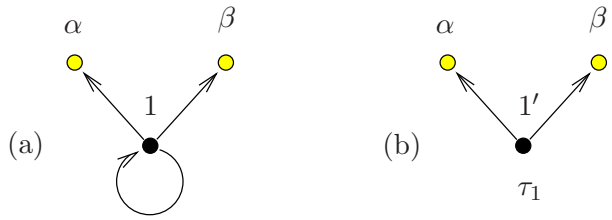


FIGURE 4.1: BKL algorithm for propagating a KMC trajectory applied to a three-state Markov chain. (a) The transition state diagram is shown where states and transitions are represented by circles and directed arrows, respectively. The Markov chain is parametrised by transition probabilities $P_{\alpha,1}$, $P_{\beta,1}$ and $P_{1,1}$. Absorbing states α and β are shaded. If $P_{1,1}$ is close to unity the KMC trajectory is likely to revisit state 1 many times before going to α or β. (b) State 1 is replaced with state $1'$. The new Markov chain is parametrised by transition probabilities $P_{\alpha,1'}$, $P_{\beta,1'}$ and the average time for escape from state 1, τ_1. Transitions from state $1'$ to itself are forbidden. Every time state $1'$ is visited the simulation 'clock' is incremented by τ_1.

Similar ideas underlie the accelerated Monte Carlo algorithm suggested by Novotny [26]. According to this 'Monte Carlo with absorbing Markov chains' (MCAMC) method, at every step a Markov matrix, $\mathbf{P}$, is formed, which describes the transitions in a subspace S that contains the current state α , and a set of adjacent states that the random walker is likely to visit from α. A trajectory length, n, for escape from S is obtained by bracketing a uniformly distributed random variable, r, as

$$\sum_{\beta} \left[\mathbf{P}^n\right]_{\beta,\alpha} < r \leqslant \sum_{\beta} \left[\mathbf{P}^{n-1}\right]_{\beta,\alpha} . \tag{4.4}$$

Then an n-step leapfrog move is performed to one of the states $\gamma \notin S$ and the simulation clock is incremented by n. State γ is chosen at random with probability

$$\left[\mathbf{R}\mathbf{P}^{n-1}\right]_{\gamma,\alpha} / \sum_{\gamma} \left[\mathbf{R}\mathbf{P}^{n-1}\right]_{\gamma,\alpha} , \tag{4.5}$$

where $R_{\gamma,\alpha}$ is the transition probability from state $\alpha \in S$ to state $\gamma \notin S$. Both the BKL and MCAMC methods can be many orders of magnitude faster than the standard KMC method when kinetic traps are present.

In chemical kinetics transitions out of a state are described using a Poisson process, which can be considered a continuous-time analogue of Bernoulli trials. The transition

probabilities are determined from the rates of the underlying transitions as

$$P_{j,i} = \frac{k_{j,i}}{\sum_{\alpha} k_{\alpha,i}}. \tag{4.6}$$

There may be several escape routes from a given state. Transitions from any state to directly connected states are treated as competing independent Poisson processes, which together generate a new Poisson distribution [179]. n independent Poisson processes with rates $k_1,\ k_2,\ k_3, \ldots,\ k_n$ combine to produce a Poisson process with rate $k = \sum_{i=1}^{n} k_i$. The waiting time for a transition to occur to any connected state is then exponentially distributed as $k \exp(-kt)$ [204].

Given the exponential distribution of waiting times the mean waiting time in state i before escape, τ_i, is $1/\sum_j k_{j,i}$ and the variance of the waiting time is simply τ_i^2. Here $k_{j,i}$ is the rate constant for transitions from i to j. When the average of the distribution of times is the property of interest, and not the distribution of times itself, it is sufficient to increment the simulation time by the mean waiting time rather than by a value drawn from the appropriate distribution [9, 205]. This modification to the original KMC formulation [206, 207] reduces the cost of the method and accelerates the convergence of KMC averages without affecting the results.

4.1.3 Discrete Path Sampling

The result of a DPS simulation is a database of local minima and transition states from the PES [8–10]. To extract thermodynamic and kinetic properties from this database we require partition functions for the individual minima and rate constants, $k_{\alpha,\beta}$, for the elementary transitions between adjacent minima β and α. We usually employ harmonic densities of states and statistical rate theory to obtain these quantities, but these details are not important here. To analyse the global kinetics we further assume Markovian transitions between adjacent local minima, which produces a set of linear (master) equations that governs the evolution of the occupation probabilities towards equilibrium [189, 196]

$$\frac{dP_\alpha(t)}{dt} = \sum_{\beta} k_{\alpha,\beta} P_\beta(t) - P_\alpha(t) \sum_{\beta} k_{\beta,\alpha}, \tag{4.7}$$

where $P_\alpha(t)$ is the occupation probability of minimum α at time t.

All the minima are classified into sets A, B and I. When local equilibrium is assumed within the A and B sets we can write

$$P_a(t) = \frac{P_a^{\rm eq} P_A(t)}{P_A^{\rm eq}} \quad \text{and} \quad P_b(t) = \frac{P_b^{\rm eq} P_B(t)}{P_B^{\rm eq}}, \tag{4.8}$$

where $P_A(t) = \sum_{a\in A} P_a(t)$ and $P_B(t) = \sum_{b\in B} P_b(t)$. If the steady-state approximation is applied to all the intervening states $i \in I = \{i_1, i_2, i_3, \ldots, i_{n_i}\}$, so that

$$\frac{dP_i(t)}{dt} = 0, \tag{4.9}$$

then Equation 4.7 can be written as [9]

$$\begin{aligned} \frac{dP_A(t)}{dt} &= k_{A,B} P_B(t) - k_{B,A} P_A(t), \\ \frac{dP_B(t)}{dt} &= k_{B,A} P_A(t) - k_{A,B} P_B(t). \end{aligned} \tag{4.10}$$

The rate constants $k_{A,B}$ and $k_{B,A}$ for forward and backward transitions between states A and B are the sums over all possible paths within the set of intervening minima of the products of the branching probabilities corresponding to the elementary transitions for each path:

$$\begin{aligned} k_{A,B}^{\rm DPS} &= \sum_{a\leftarrow b}{}' \frac{k_{a,i_1}}{\sum_{\alpha_1} k_{\alpha_1,i_1}} \frac{k_{i_1,i_2}}{\sum_{\alpha_2} k_{\alpha_2,i_2}} \cdots \frac{k_{i_{n-1},i_n}}{\sum_{\alpha_n} k_{\alpha_n,i_n}} \frac{k_{i_n,b} P_b^{\rm eq}}{P_B^{\rm eq}} \\ &= \sum_{a\leftarrow b}{}' P_{a,i_1} P_{i_1,i_2} \cdots P_{i_{n-1},i_n} \frac{k_{i_n,b} P_b^{\rm eq}}{P_B^{\rm eq}}, \end{aligned} \tag{4.11}$$

and similarly for $k_{B,A}$ [8]. The sum is over all paths that begin from a state $b \in B$ and end at a state $a \in A$, and the prime indicates that paths are not allowed to revisit states in B. In previous contributions [8, 10, 133, 197] this sum was evaluated using a weighted adjacency matrix multiplication (MM) method, which will be reviewed in Section 4.2.

4.1.4 KMC AND DPS AVERAGES

We now show that the evaluation of the DPS sum in Equation 4.11 and the calculation of KMC averages are two closely related problems.

For KMC simulations we define sources and sinks that coincide with the set of initial states B and final states A, respectively.* Every cycle of KMC simulation involves the

*Terminology taken from graph theory. In probability theory, state i is called absorbing if $P_{i,i} = 1$, which coincides with our definition of a sink.

generation of a single KMC trajectory connecting a node $b \in B$ and a node $a \in A$. A source node b is chosen from set B with probability $P_b^{\mathrm{eq}}/P_B^{\mathrm{eq}}$.

We can formulate the calculation of the mean first passage time from B to A in graph theoretical terms as follows. Let the digraph consisting of nodes for all local minima and edges for each transition state be $\mathcal{G}$. The digraph consisting of all nodes except those belonging to region A is denoted by G. We assume that there are no isolated nodes in $\mathcal{G}$, so that all the nodes in A can be reached from every node in G. Suppose we start a KMC simulation from a particular node $\beta \in G$. Let $P_\alpha(n)$ be the expected occupation probability of node α after n KMC steps, with initial conditions $P_\beta(0) = 1$ and $P_{\alpha\neq\beta}(0) = 0$. We further define an escape probability for each $\alpha \in G$ as the sum of branching probabilities to nodes in A, i.e.

$$\mathcal{E}_\alpha^G = \sum_{a\in A} P_{a,\alpha}. \tag{4.12}$$

KMC trajectories terminate when they arrive at an A minimum, and the expected probability transfer to the A region at the nth KMC step is $\sum_{\alpha\in G} \mathcal{E}_\alpha^G P_\alpha(n)$. If there is at least one escape route from G to A with a non-zero branching probability, then eventually all the occupation probabilities in G must tend to zero and

$$\Sigma_\beta^G = \sum_{n=0}^{\infty} \sum_{\alpha\in G} \mathcal{E}_\alpha^G P_\alpha(n) = 1. \tag{4.13}$$

We now rewrite $P_\alpha(n)$ as a sum over all n-step paths that start from β and end at α without leaving G. Each path contributes to $P_\alpha(n)$ according to the appropriate product of n branching probabilities, so that

$$\begin{aligned}
\Sigma_\beta^G &= \sum_{\alpha\in G} \mathcal{E}_\alpha^G \sum_{n=0}^{\infty} P_\alpha(n) \\
&= \sum_{\alpha\in G} \mathcal{E}_\alpha^G \sum_{n=0}^{\infty} \sum_{\Xi(n)} P_{\alpha,i_{n-1}} P_{i_{n-1},i_{n-2}} \cdots P_{i_2,i_1} P_{i_1,\beta} \\
&= \sum_{\alpha\in G} \mathcal{E}_\alpha^G \mathcal{S}_{\alpha,\beta}^G = 1,
\end{aligned} \tag{4.14}$$

where $\Xi(n)$ denotes the set of n-step paths that start from β and end at α without leaving G, and the last line defines the pathway sum $\mathcal{S}_{\alpha,\beta}^G$.

It is clear from the last line of Equation 4.14 that for fixed β the quantities $\mathcal{E}_\alpha^G \mathcal{S}_{\alpha,\beta}^G$ define a probability distribution. However, the pathway sums $\mathcal{S}_{\alpha,\beta}^G$ are not probabilities,

and may be greater than unity. In particular, $\mathcal{S}^G_{\beta,\beta} \geqslant 1$ because the path of zero length is included, which contributes one to the sum. Furthermore, the normalisation condition on the last line of Equation 4.14 places no restriction on $\mathcal{S}^G_{\alpha,\beta}$ terms for which $\mathcal{E}^G_\alpha$ vanishes.

We can also define a probability distribution for individual pathways. Let $\mathcal{W}_\xi$ be the product of branching probabilities associated with a path ξ so that

$$\mathcal{S}^G_{\alpha,\beta} = \sum_{n=0}^{\infty} \sum_{\xi \in \Xi(n)} \mathcal{W}_\xi \equiv \sum_{\xi \in \alpha \leftarrow \beta} \mathcal{W}_\xi, \tag{4.15}$$

where $\alpha \leftarrow \beta$ is the set of all appropriate paths from β to α of any length that can visit and revisit any node in G. If we focus upon paths starting from minima in region B

$$\sum_{b \in B} \frac{P^{\mathrm{eq}}_b}{P^{\mathrm{eq}}_B} \sum_{\alpha \in G} \mathcal{E}^G_\alpha \sum_{\xi \in \alpha \leftarrow b} \mathcal{W}_\xi = \sum_{b \in B} \frac{P^{\mathrm{eq}}_b}{P^{\mathrm{eq}}_B} \sum_{\alpha \in G_A} \mathcal{E}^G_\alpha \sum_{\xi \in \alpha \leftarrow b} \mathcal{W}_\xi = 1, \tag{4.16}$$

where G_A is the set of nodes in G that are adjacent to A minima in the complete graph $\mathcal{G}$, since $\mathcal{E}^G_\alpha$ vanishes for all other nodes. We can rewrite this sum as

$$\sum_{\xi \in G_A \leftarrow B} \frac{P^{\mathrm{eq}}_b}{P^{\mathrm{eq}}_B} \mathcal{E}^G_\alpha \mathcal{W}_\xi = \sum_{\xi \in A \leftarrow B} \frac{P^{\mathrm{eq}}_b}{P^{\mathrm{eq}}_B} \mathcal{W}_\xi = \sum_{\xi \in A \leftarrow B} \mathcal{P}_\xi = 1, \tag{4.17}$$

which defines the non-zero pathway probabilities $\mathcal{P}_\xi$ for all paths that start from any node in set B and finish at any node in set A. The paths $\xi \in A \leftarrow B$ can revisit any minima in the G set, but include just one A minimum at the terminus. Note that $\mathcal{W}_\xi$ and $\mathcal{P}_\xi$ can be used interchangeably if there is only one state in set B.

The average of some property, Q_ξ, defined for each KMC trajectory, ξ, can be calculated from the $\mathcal{P}_\xi$ as

$$\langle Q_\xi \rangle = \sum_{\xi \in A \leftarrow B} \mathcal{P}_\xi Q_\xi. \tag{4.18}$$

Of course, KMC simulations avoid this complete enumeration by generating trajectories with probabilities proportional to $\mathcal{P}_\xi$, so that a simple running average can be used to calculate $\langle Q_\xi \rangle$. In the following sections we will develop alternative approaches based upon evaluating the complete sum, which become increasingly efficient at low temperature. We emphasise that these methods are only applicable to problems with a finite number of states, which are assumed to be known in advance.

The evaluation of the DPS sum defined in Equation 4.11 can also be rewritten in terms of pathway probabilities:

$$\begin{aligned} k_{A,B}^{\mathrm{DPS}} &= \sum_{n=0}^{\infty} {\sum_{\Xi(n)}}' P_{\alpha,i_1} P_{i_1,i_2} \cdots P_{i_{n-1},i_n} \frac{k_{i_n,\beta}\, P_\beta^{\mathrm{eq}}}{P_B^{\mathrm{eq}}}, \\ &= \sum_{n=0}^{\infty} {\sum_{\Xi(n)}}' P_{\alpha,i_1} P_{i_1,i_2} \cdots P_{i_{n-1},i_n} P_{i_n,\beta} \tau_\beta^{-1} \frac{P_\beta^{\mathrm{eq}}}{P_B^{\mathrm{eq}}} \\ &= {\sum_{\xi \in A \leftarrow B}}' \mathcal{P}_\xi \tau_\beta^{-1}, \end{aligned} \tag{4.19}$$

where the prime on the summation indicates that the paths are not allowed to revisit the B region. We have also used the fact that $k_{i_n,b} = P_{i_n,b}/\tau_b$.

A digraph representation of the restricted set of pathways defined in Equation 4.19 can be created if we allow sets of sources and sinks to overlap. In that case all the nodes $A \cup B$ are defined to be sinks and all the nodes in B are the sources, i.e. every node in set B is both a source and a sink. The required sum then includes all the pathways that finish at sinks of type A, but not those that finish at sinks of type B. The case when sets of sources and sinks (partially) overlap is discussed in detail in Section 4.6.

4.1.5 Mean Escape Times

Since the mean first passage time between states B and A, or the escape time from a subgraph, is of particular interest, we first illustrate a means to derive formulae for these quantities in terms of pathway probabilities.

The average time taken to traverse a path $\xi = \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{l(\xi)}$ is calculated as $\mathfrak{t}_\xi = \tau_{\alpha_1} + \tau_{\alpha_2} + \tau_{\alpha_3}, \ldots, \tau_{\alpha_{l(\xi)-1}}$, where τ_α is the mean waiting time for escape from node α, as above, α_k identifies the kth node along path ξ, and $l(\xi)$ is the length of path ξ. The mean escape time from a graph G if started from node β is then

$$\mathcal{T}_\beta^G = \sum_{\xi \in A \leftarrow \beta} \mathcal{P}_\xi \mathfrak{t}_\xi. \tag{4.20}$$

If we multiply every branching probability, $P_{\alpha,\beta}$, that appears in $\mathcal{P}_\xi$ by $\exp(\zeta\tau_\beta)$ then

the mean escape time can be obtained as:

$$\begin{aligned}
\mathcal{T}_\beta^G &= \left[\frac{d}{d\zeta}\left(\sum_{\xi\in A\leftarrow\beta} P_{\alpha_{l(\xi)},\alpha_{l(\xi)-1}} e^{\zeta\tau_{l(\xi)-1}} P_{\alpha_{l(\xi)-1},\alpha_{l(\xi)-2}} e^{\zeta\tau_{l(\xi)-2}} \dots P_{\alpha_2,\alpha_1} e^{\zeta\tau_{\alpha_1}}\right)\right]_{\zeta=0} \\
&= \left[\frac{d}{d\zeta}\left(\sum_{\xi\in A\leftarrow\beta} P_{\alpha_{l(\xi)},\alpha_{l(\xi)-1}} P_{\alpha_{l(\xi)-1},\alpha_{l(\xi)-2}} \dots P_{\alpha_2,\alpha_1} e^{\zeta \mathfrak{t}_\xi}\right)\right]_{\zeta=0} \\
&= \sum_{\xi\in A\leftarrow\beta} \mathcal{P}_\xi \mathfrak{t}_\xi.
\end{aligned} \tag{4.21}$$

This approach is useful if we have analytic results for the total probability Σ_β^G, which may then be manipulated into formulae for $\mathcal{T}_\beta^G$, and is standard practice in probability theory literature [208, 209]. The quantity $P_{\alpha,\beta}e^{\zeta\tau_\beta}$ is similar to the 'ζ probability' described in Reference [208]. Analogous techniques are usually employed to obtain $\mathcal{T}_\beta^G$ and higher moments of the first-passage time distribution from analytic expressions for the first-passage probability generating function (see, for example, References [210, 211]). We now define $\widetilde{P}_{\alpha,\beta} = P_{\alpha,\beta}e^{\zeta\tau_\beta}$ and the related quantities

$$\begin{aligned}
\widetilde{\mathcal{E}}_\alpha^G &= \sum_{a\in A} \widetilde{P}_{a,\alpha} = \mathcal{E}_\alpha^G e^{\zeta\tau_\alpha}, \\
\widetilde{\mathcal{W}}_\xi &= \widetilde{P}_{\alpha_{l(\xi)},\alpha_{l(\xi)-1}} \widetilde{P}_{\alpha_{l(\xi)-1},\alpha_{l(\xi)-2}} \dots \widetilde{P}_{\alpha_2,\alpha_1} = \mathcal{W}_\xi e^{\zeta\mathfrak{t}_\xi}, \\
\widetilde{\mathcal{P}}_\xi &= \widetilde{\mathcal{W}}_\xi P_b^{\mathrm{eq}}/P_B^{\mathrm{eq}}, \\
\widetilde{\mathcal{S}}_{\alpha,\beta}^G &= \sum_{\xi\in\alpha\leftarrow\beta} \widetilde{\mathcal{W}}_\xi, \\
\text{and}\quad \widetilde{\Sigma}_\beta^G &= \sum_{\alpha\in G} \widetilde{\mathcal{E}}_\alpha^G \widetilde{\mathcal{S}}_{\alpha,\beta}^G.
\end{aligned} \tag{4.22}$$

Note that $\left[\widetilde{\mathcal{E}}_\alpha^G\right]_{\zeta=0} = \mathcal{E}_\alpha^G$ etc., while the mean escape time can now be written as

$$\mathcal{T}_\beta^G = \left[\frac{d\widetilde{\Sigma}_\beta^G}{d\zeta}\right]_{\zeta=0}. \tag{4.23}$$

In the remaining sections we show how to calculate the pathway probabilities, $\mathcal{P}_\xi$, exactly, along with pathway averages, such as the waiting time. Chain graphs are treated in Section 4.2 and the results are generalised to arbitrary graphs in Section 4.3.

4.2 Chain Graphs

A general account of the problem of the first passage time in chemical physics was given by Weiss as early as 1965 [212]. In Reference [212] he summarised various techniques for solving such problems to date, and gave a general formula for moments of the first passage time in terms of the Green's function of the Fokker-Plank operator. Explicit expressions for the mean first passage times in terms of the basic transition probabilities for the case of a one-dimensional random walk were obtained by Ledermann and Reuter in 1954 [213], Karlin and MacGregor in 1957 [214], Weiss in 1965 [212], Gardiner in 1985 [215], Van den Broeck in 1988 [216], Le Doussal in 1989 [217], Murthy and Kehr and Matan and Havlin in 1989-1990 [211, 218, 219], Raykin in 1992 [210], Bar-Haim and Klafter in 1998 [203], Pury and Cáceres in 2003 [220], and Slutsky, Kardar and Mirny in 2004 [221, 222]. The one-dimensional random walk is therefore a very well researched topic, both in the field of probability and its physical and chemical applications. The results presented in this section differ in the manner of presentation.

A random walk in a discrete state space S, where all the states can be arranged on a linear chain in such a way that $P_{i,j} = 0$ for all $|i - j| > 1$, is called a one-dimensional or simple random walk (SRW). The SRW model attracts a lot of attention because of its rich behaviour and mathematical tractability. A well known example of its complexity is the anomalous diffusion law discovered by Sinai [223]. He showed that there is a dramatic slowing down of an ordinary power law diffusion (RMS displacement is proportional to $(\log t)^2$ instead of $t^{1/2}$) if a random walker at each site i experiences a random bias field $B_i = P_{i,i+1} - P_{i,i-1}$. Stanley and Havlin generalised the Sinai model by introducing long-range correlations between the bias fields on each site and showed that the SRW can span a range of diffusive properties [224].

Although one-dimensional transport is rarely found on the macroscopic scale at a microscopic level there are several examples, such as kinesin motion along microtubules [225, 226], or DNA translocation through a nanopore [227, 228], so the SRW is interesting not only from a theoretical standpoint. There is a number of models that build upon the SRW that have exciting applications, examples being the SRW walk with branching and annihilation [229], and the SRW in the presence of random trappings [230]. Techniques developed for the SRW were applied to study more com-

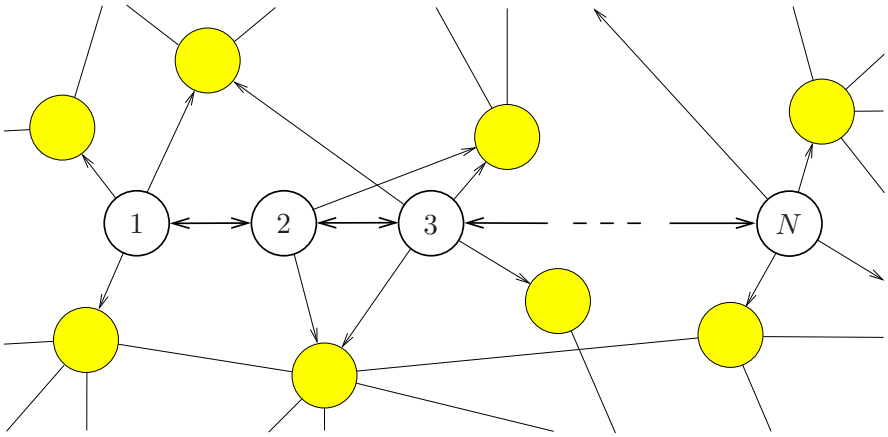


Figure 4.2: Chain graph of length N, depicted as the subgraph of a larger graph. Visible sink nodes are shaded. Double-headed arrows represent pairs of directed edges.

plex cases, such as, for example, multistage transport in liquids [231], random walks on fractals [232, 233], even-visiting random walks [234], self-avoiding random walks [235], random walks on percolation clusters [236, 237], and random walks on simple regular lattices [238, 239] and superlattices [240].

A presentation that discusses SRW first-passage probabilities in detail sufficient for our applications is that of Raykin [210]. He considered pathway ensembles explicitly and obtained the generating functions for the occupation probabilities of any lattice site for infinite, half-infinite and finite one-dimensional lattices with the random walker starting from an arbitrary lattice site. As we discuss below, these results have a direct application to the evaluation of the DPS rate constants augmented with recrossings. We have derived equivalent expressions for the first-passage probabilities independently, considering the finite rather than the infinite case, which we discuss in terms of chain digraphs below.

We define a chain as a digraph $C_N = (V, E)$ with N nodes and $2(N-1)$ edges, where $V = \{v_1, v_2, \ldots, v_N\}$ and $E = \{e_{1,2}, e_{2,1}, e_{2,3}, e_{3,2}, \ldots, e_{N-2,N-1}, e_{N-1,N-2}\}$. Adjacent nodes in the numbering scheme are connected via two oppositely directed edges, and these are the only connections. A transition probability $P_{\alpha,\beta}$ is associated with every edge $e_{\alpha,\beta}$, as described in Section 4.1.1. An N-node chain is depicted in Figure 4.2 as a subgraph of another graph. The total probability of escape from the chain via node N if started from node 1 is of interest because it has previously been used to associate

contributions to the total rate constant from unique paths in DPS studies [8, 9]. We can restrict the sampling to paths without recrossings between intermediate minima if we perform the corresponding recrossing sums explicitly [8].

We denote a pathway in C_N by the ordered sequence of node indices. The length of a pathway is the number of edges traversed. For example, the pathway $1, 2, 1, 2, 3, 2, 3$ has length 6, starts at node 1 and finishes at node 3. The indices of the intermediate nodes $2, 1, 2, 3, 2$ are given in the order in which they are visited. The product of branching probabilities associated with all edges in a path was defined above as $\mathcal{W}_\xi$. For example, the product for the above pathway is $P_{3,2}P_{2,3}P_{3,2}P_{2,1}P_{1,2}P_{2,1}$, which we can abbreviate as $\mathcal{W}_{3,2,3,2,1,2,1}$. For a chain graph C_N, which is a subgraph of $\mathcal{G}$, we also define the set of indices of nodes in $\mathcal{G}$ that are adjacent to nodes in C_N but not members of C_N as $Adj[C_N]$. These nodes will be considered as sinks if we are interested in escape from C_N.

Analytical results for C_3 are easily derived:

$$\begin{aligned}
\mathcal{S}^{C_3}_{1,1} &= \sum_{n=0}^{\infty}\left(\mathcal{W}_{1,2,1}\sum_{m=0}^{\infty}(\mathcal{W}_{2,3,2})^m\right)^n &&= \frac{1-\mathcal{W}_{2,3,2}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}},\\
\mathcal{S}^{C_3}_{2,1} &= \sum_{n=0}^{\infty}(\mathcal{W}_{2,3,2})^n\,P_{2,1}\mathcal{S}^{C_3}_{1,1} &&= \frac{P_{2,1}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}},\\
\mathcal{S}^{C_3}_{3,1} &= P_{3,2}\sum_{n=0}^{\infty}(\mathcal{W}_{2,3,2})^n\,P_{2,1}\mathcal{S}^{C_3}_{1,1} &&= \frac{\mathcal{W}_{3,2,1}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}},\\
\mathcal{S}^{C_3}_{2,2} &= \sum_{n=0}^{\infty}(\mathcal{W}_{1,2,1}+\mathcal{W}_{2,3,2})^n &&= \frac{1}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}},\\
\mathcal{S}^{C_3}_{3,2} &= P_{3,2}\mathcal{S}^{C_3}_{2,2} &&= \frac{P_{3,2}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}}.
\end{aligned} \tag{4.24}$$

These sums converge if the cardinality of the set $Adj[C_3]$ is greater than zero. To prove this result consider a factor, f, of the form

$$f = P_{\alpha,\beta}P_{\beta,\alpha}\sum_{m=0}^{\infty}(P_{\beta,\gamma}P_{\gamma,\beta})^m, \tag{4.25}$$

and assume that the branching probabilities are all non-zero, and that there is at least one additional escape route from α, β or γ. We know that $P_{\beta,\gamma}P_{\gamma,\beta} < P_{\gamma,\beta} < 1$ because $P_{\alpha,\beta} + P_{\gamma,\beta} \leqslant 1$ and $P_{\alpha,\beta} \neq 0$. Hence $f = P_{\alpha,\beta}P_{\beta,\alpha}/(1 - P_{\beta,\gamma}P_{\gamma,\beta})$. However, $P_{\alpha,\beta}P_{\beta,\alpha} + P_{\beta,\gamma}P_{\gamma,\beta} \leqslant P_{\alpha,\beta} + P_{\gamma,\beta} \leqslant 1$, and equality is only possible if $P_{\beta,\alpha} = P_{\beta,\gamma} =$

$P_{\alpha,\beta}+P_{\gamma,\beta}=1$, which contradicts the assumption of an additional escape route. Hence $P_{\alpha,\beta}P_{\beta,\alpha} < 1-P_{\beta,\gamma}P_{\gamma,\beta}$ and $f<1$. The pathway sums $\mathcal{S}_{1,2}^{C_3}$, $\mathcal{S}_{1,3}^{C_3}$, $\mathcal{S}_{2,3}^{C_3}$ and $\mathcal{S}_{3,3}^{C_3}$ can be obtained from Equation 4.24 by permuting the indices. The last two sums in Equation 4.24 are particularly instructive: the n'th term in the sum for $\mathcal{S}_{2,2}^{C_3}$ and the n'th term in the sum for $\mathcal{S}_{3,2}^{C_3}$ are the contributions from pathways of length $2n$ and $2n+1$, respectively.

The pathway sums $\mathcal{S}_{\alpha,\beta}^{C_N}$ can be derived for a general chain graph C_N in terms of recursion relations, as shown in Appendix C. The validity of our results for C_N was verified numerically using the matrix multiplication method described in Reference [8]. For a chain of length N we construct an $N\times N$ transition probability matrix $\mathbf{P}$ with elements

$$\mathbf{P}=\begin{pmatrix} 0 & P_{1,2} & 0 & \dots \\ P_{2,1} & 0 & P_{2,3} & \dots \\ 0 & P_{3,2} & 0 & \\ \vdots & \vdots & & \ddots \end{pmatrix}. \tag{4.26}$$

The matrix form of the system of Chapman-Kolmogorov equations [187] for homogeneous discrete-time Markov chains [123, 187] allows us to obtain the n-step transition probability matrix recursively as

$$\mathbf{P}(n)=\mathbf{P}\mathbf{P}(n-1)=\mathbf{P}^n. \tag{4.27}$$

$\mathcal{S}_{\alpha,\beta}^{C_N}$ can then be computed as

$$\mathcal{S}_{\alpha,\beta}^{C_N}=\sum_{n=1}^{M}\left[\mathbf{P}^n\right]_{\alpha,\beta}, \tag{4.28}$$

where the number of matrix multiplications, M, is adjusted dynamically depending on the convergence properties of the above sum. We note that sink nodes are excluded when constructing $\mathbf{P}$ and $\sum_j P_{j,i}$ can be less than unity.

For chains a sparse-optimised matrix multiplication method for $\mathcal{S}_{\alpha,\beta}^{C_N}$ scales as $\mathcal{O}(NM)$, and may suffer from convergence and numerical precision problems for larger values of N and branching probabilities that are close to zero or unity [8]. The summation method presented in this section can be implemented to scale as $\mathcal{O}(N)$ with constant memory requirements (Algorithm B.1). It therefore constitutes a faster, more robust

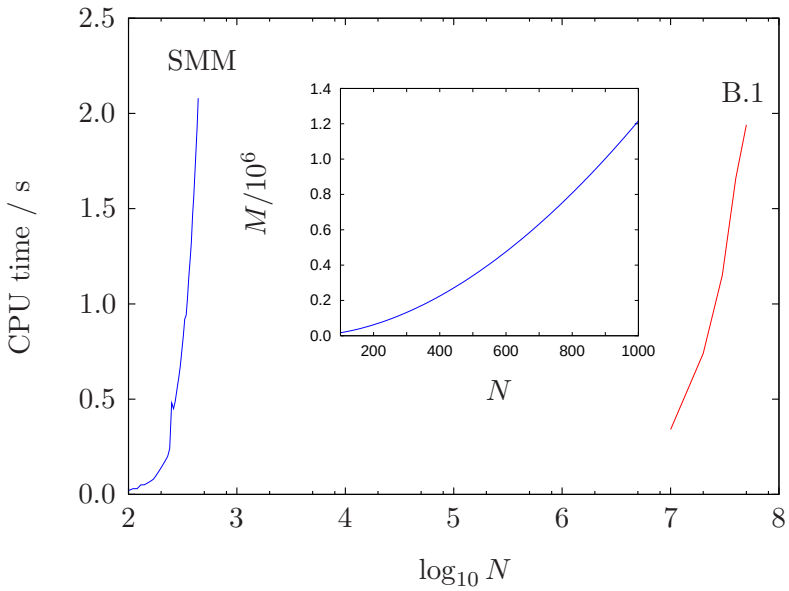


Figure 4.3: CPU time needed to calculate the total transition probabilities for a chain of length N. The data is shown for a sparse-optimised matrix multiplication (SMM) method (blue) and a sparse-optimised version of Algorithm B.1 (red). Terminal nodes of each chain were connected to a sink, one of the terminal nodes was chosen to be the source. All the branching probabilities were set to 0.5. Each SMM calculation was terminated when $1.0 - \Sigma_0^{C_N}$ was less than 10^{-5}. The inset shows the number of matrix multiplications, M, as a function of chain length. Note the $\log_{10}$ scale on the horizontal axes.

and more precise alternative to the matrix multiplication method when applied to chain graph topologies (Figure 4.3).

Mean escape times for C_3 are readily obtained from the results in Equation 4.24 by applying the method outlined in Section 4.1.5:

$$\begin{aligned} \mathcal{T}_1^{C_3} &= \frac{\tau_1(1-\mathcal{W}_{2,3,2}) + \tau_2 P_{2,1} + \tau_3 \mathcal{W}_{3,2,1}}{1 - \mathcal{W}_{1,2,1} - \mathcal{W}_{2,3,2}}, \\ \mathcal{T}_2^{C_3} &= \frac{\tau_1 P_{1,2} + \tau_2 + \tau_3 P_{3,2}}{1 - \mathcal{W}_{1,2,1} - \mathcal{W}_{2,3,2}}, \end{aligned} \tag{4.29}$$

and $\mathcal{T}_3^{C_3}$ can be obtained from $\mathcal{T}_1^{C_3}$ by permuting the subscripts 1 and 3.

The mean escape time from the C_N graph if started from node β can be calculated recursively using the results of Appendix D and Section 4.1.5 or by resorting to a first-step analysis [241].

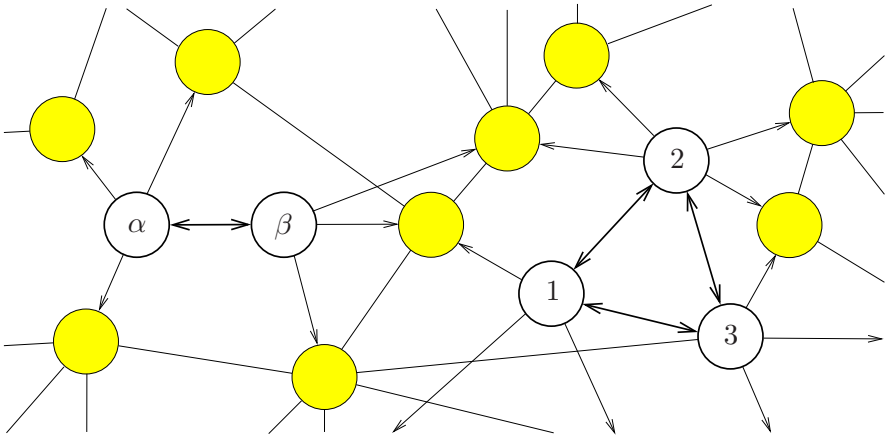


Figure 4.4: Complete graphs K_2 and K_3, depicted as the subgraphs of a larger graph. Visible sink nodes are shaded.

4.3 Complete Graphs

In a complete digraph each pair of nodes is connected by two oppositely directed edges [155]. The complete graph with N graph nodes is denoted $K_N = (V, E)$, and has N nodes and $N(N-1)$ edges, remembering that we have two edges per connection (Figure 4.4). Due to the complete connectivity we need only consider two cases: when the starting and finishing nodes are the same and when they are distinct. We employ complete graphs for the purposes of generality. An arbitrary graph G_N is a subgraph of K_N with transition probabilities for non-existent edges set to zero. All the results in this section are therefore equally applicable to arbitrary graphs.

The complete graph K_2 will not be considered here as it is topologically identical to the graph C_2. The difference between the K_3 and C_3 graphs is the existence of edges that connect nodes 1 and 3. Pathways confined to K_3 can therefore contain cycles, and for a given path length they are significantly more numerous (Figure 4.5). The $\mathcal{S}^{K_3}_{\alpha,\beta}$

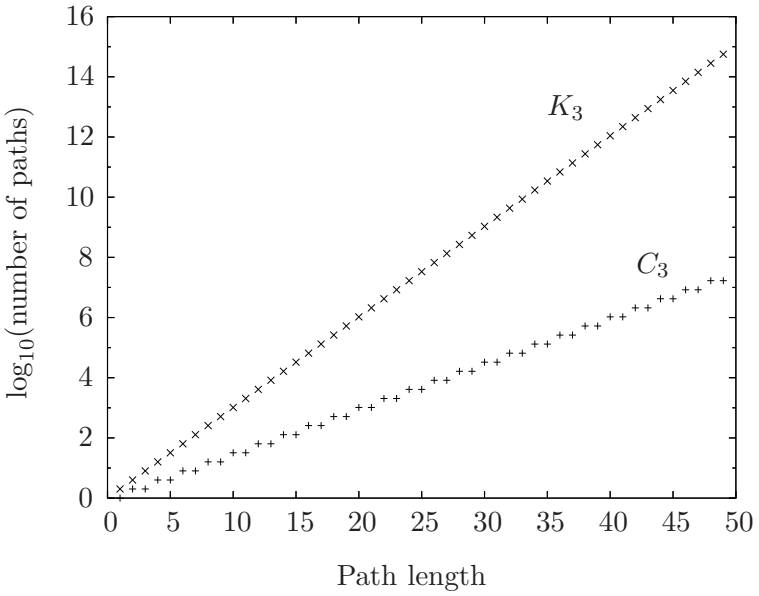


Figure 4.5: The growth of the number of pathways with the pathway length for K_3 and C_3. The starting node is chosen arbitrarily for K_3 while for C_3 the we start at one of the terminal nodes. Any node adjacent to K_3 or C_3 is a considered to be a sink and for simplicity we consider only one escape route from every node. Note the $\log_{10}$ scale on the vertical axis.

can again be derived analytically for this graph:

$$\begin{aligned}
\mathcal{S}_{1,1}^{K_3} &= \sum_{n=0}^{\infty}\left((\mathcal{W}_{1,2,1}+\mathcal{W}_{1,3,1}+\mathcal{W}_{1,2,3,1}+\mathcal{W}_{1,3,2,1})\sum_{m=0}^{\infty}(\mathcal{W}_{2,3,2})^m\right)^n \\
&= \frac{1-\mathcal{W}_{2,3,2}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}-\mathcal{W}_{1,3,1}-\mathcal{W}_{1,2,3,1}-\mathcal{W}_{1,3,2,1}}, \\
\mathcal{S}_{2,1}^{K_3} &= \sum_{n=0}^{\infty}(\mathcal{W}_{2,3,2})^n\,(P_{2,1}+\mathcal{W}_{2,3,1})\,\mathcal{S}_{1,1}^{K_3} \\
&= \frac{P_{2,1}+\mathcal{W}_{2,3,1}}{1-\mathcal{W}_{1,2,1}-\mathcal{W}_{2,3,2}-\mathcal{W}_{1,3,1}-\mathcal{W}_{1,2,3,1}-\mathcal{W}_{1,3,2,1}}.
\end{aligned} \tag{4.30}$$

The results for any other possibility can be obtained by permuting the node indices appropriately.

The pathway sums $\mathcal{S}_{\alpha,\beta}^{K_N}$ for larger complete graphs can be obtained by recursion. For $\mathcal{S}_{N,N}^{K_N}$ any path leaving from and returning to N can be broken down into a step out of N to any $i<N$, all possible paths between i and $j<N-1$ within K_{N-1}, and finally a step back to N from j. All such paths can be combined together in any order,

so we have a multinomial distribution [242]:

$$\begin{aligned}\mathcal{S}_{N,N}^{K_N} &= \sum_{n=0}^{\infty}\left(\sum_{i=1}^{N-1}\left(\sum_{j=1}^{N-1}\left(P_{N,j}\mathcal{S}_{j,i}^{K_{N-1}}P_{i,N}\right)\right)\right)^n \\ &= \left(1-\sum_{i=1}^{N-1}\sum_{j=1}^{N-1}P_{N,j}\mathcal{S}_{j,i}^{K_{N-1}}P_{i,N}\right)^{-1}.\end{aligned} \tag{4.31}$$

To evaluate $\mathcal{S}_{1,N}^{K_N}$ we break down the sum into all paths that depart from and return to N, followed by all paths that leave node N and reach node 1 without returning to N. The first contribution corresponds to a factor of $\mathcal{S}_{N,N}^{K_N}$, and the second produces a factor $P_{i,N}\mathcal{S}_{1,i}^{K_{N-1}}$ for every $i < N$:

$$\mathcal{S}_{1,N}^{K_N} = \mathcal{S}_{N,N}^{K_N}\sum_{i=1}^{N-1}\mathcal{S}_{1,i}^{K_{N-1}}P_{i,N}, \tag{4.32}$$

where $\mathcal{S}_{1,1}^{K_1}$ is defined to be unity. Any other $\mathcal{S}_{\alpha,\beta}^{K_N}$ can be obtained by a permutation of node labels.

Algorithm B.2 provides an example implementation of the above formulae optimised for incomplete graphs. The running time of Algorithm B.2 depends strongly on the graph density. (A digraph in which the number of edges is close to the maximum value of $N(N-1)$ is termed a dense digraph [202].) For K_N the algorithm runs in $\mathcal{O}(N^{2N})$ time, while for an arbitrary graph it scales as $\mathcal{O}(d^{2N})$, where d is the average degree of the nodes. For chain graphs the algorithm runs in $\mathcal{O}(N)$ time and therefore constitutes a recursive-function-based alternative to Algorithm B.1 with linear memory requirements. For complete graphs an alternative implementation with $\mathcal{O}((N!)^2)$ scaling is also possible.

Although the scaling of the above algorithm with N may appear disastrous, it does in fact run faster than standard KMC and MM approaches for graphs where the escape probabilities are several orders of magnitude smaller than the transition probabilities (Algorithm B.2). Otherwise, for anything but moderately branched chain graphs, Algorithm B.2 is significantly more expensive. However, the graph-transformation-based method presented in Section 4.4 yields both the pathway sums and the mean escape times for a complete graph K_N in $\mathcal{O}(N^3)$ time, and is the fastest approach that we have found.

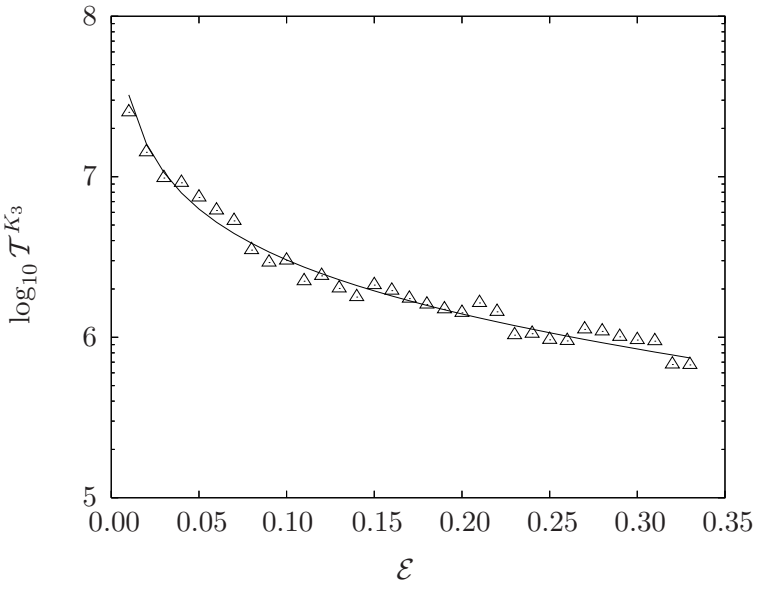


Figure 4.6: Mean escape time from K_3 as a function of the escape probability $\mathcal{E}$. The transition probabilities for the K_3 graph are parametrised by $\mathcal{E}$ for simplicity: $P_{i,j} = (1-\mathcal{E})/2$ for all $i, j \in \{1, 2, 3\}$ and $\mathcal{E}_1 = \mathcal{E}_2 = \mathcal{E}_3 = \mathcal{E}$. Kinetic Monte Carlo data (triangles) was obtained by averaging over 100 trajectories for each of 33 parameterisations. The solid line is the exact solution obtained using Equation 4.33. The units of $\mathcal{T}^{K_3}$ are arbitrary. Note the $\log_{10}$ scale on the vertical axis.

Mean escape times for K_3 are readily obtained from the results in Equation 4.30 by applying the method outlined in Section 4.1.5:

$$\mathcal{T}_1^{K_3} = \frac{\tau_1(1 - \mathcal{W}_{2,3,2}) + \tau_2(P_{2,1} + \mathcal{W}_{2,3,1}) + \tau_3(P_{3,1} + \mathcal{W}_{3,2,1})}{1 - \mathcal{W}_{1,2,1} - \mathcal{W}_{2,3,2} - \mathcal{W}_{3,1,3} - \mathcal{W}_{1,2,3,1} - \mathcal{W}_{1,3,2,1}}. \tag{4.33}$$

We have verified this result analytically using first-step analysis and numerically for various values of the parameters τ_i and $P_{\alpha,\beta}$. and obtained quantitative agreement (see Figure 4.6). Figure 4.7 demonstrates how the advantage of exact summation over KMC and MM becomes more pronounced as the escape probabilities become smaller.

4.4 Graph Transformation Method

The problem of calculation of the properties of a random walk on irregular networks was addressed previously by Goldhirsch and Gefen [208, 209]. They described a generating-function-based method where an ensemble of pathways is partitioned into 'basic walks'. A walk was defined as a set of paths that satisfies a certain restriction. As the probabil-

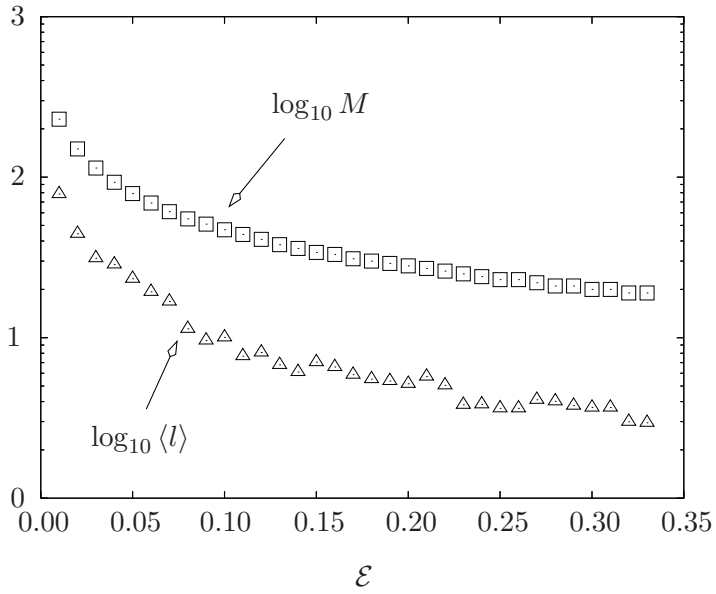


Figure 4.7: The computational cost of the kinetic Monte Carlo and matrix multiplication methods as a function of escape probability for K_3 (see caption to Figure 4.6 for the definition of $\mathcal{E}$). M is the number of matrix multiplications required to converge the value of the total probability of getting from node 1 to nodes 1, 2 and 3: the calculation was terminated when the change in the total probability between iterations was less than 10^{-3}. The number of matrix multiplications M and the average trajectory length $\langle l \rangle$ can be used as a measure of the computational cost of the matrix multiplication and kinetic Monte Carlo approaches, respectively. The computational requirements of the exact summation method are independent of $\mathcal{E}$. Note the $\log_{10}$ scale on the vertical axis.

ity generating functions corresponding to these basic walks multiply, the properties of a network as a whole can be inferred given knowledge of the generating functions corresponding to these basic walks. The method was applied to a chain, a loopless regularly branched chain and a chain containing a single loop. To the best of our knowledge only one [243] out of the 30 papers [209–211, 219–222, 240, 243–264] that cite the work of Goldhirsch and Gefen [208] is an application, perhaps due to the complexity of the method.

Here we present a graph transformation (GT) approach for calculating the pathway sums and the mean escape times for K_N. In general, it is applicable to arbitrary digraphs, but the best performance is achieved when the graph in question is dense. The algorithm discussed in this section will be referred to as DGT (D for dense). A

sparse-optimised version of the GT method (SGT) will be discussed in Section 4.5.

The GT approach is similar in spirit to the ideas that lie behind the mean value analysis and aggregation/disaggregation techniques commonly used in the performance and reliability evaluation of queueing networks [187, 265–267]. It is also loosely related to dynamic graph algorithms [268–271], which are used when a property is calculated on a graph subject to dynamic changes, such as deletions and insertions of nodes and edges. The main idea is to progressively remove nodes from a graph whilst leaving the average properties of interest unchanged. For example, suppose we wish to remove node x from graph G to obtain a new graph G'. Here we assume that x is neither source nor sink. Before node x can be removed the property of interest is averaged over all the pathways that include the edges between nodes x and $i \in Adj[x]$. The averaging is performed separately for every node i. Next, we will use the waiting time as an example of such a property and show that the mean first passage time in the original and transformed graphs is the same.

The theory is an extension of the results used to perform jumps to second neighbours in previous KMC simulations [8, 272]. Consider KMC trajectories that arrive at node i, which is adjacent to x. We wish to step directly from i to any node in the set of nodes Γ that are adjacent to i or x, excluding these two nodes themselves. To ensure that the mean first-passage times from sources to sinks calculated in G and G' are the same we must define new branching probabilities, $P'_{\gamma,i}$ from i to all $\gamma \in \Gamma$, along with a new waiting time for escape from i, τ'_i. Here, τ'_i corresponds to the mean waiting time for escape from i to any $\gamma \in \Gamma$, while the modified branching probabilities subsume all the possible recrossings involving node x that could occur in G before a transition to a node in Γ. Hence the new branching probabilities are:

$$P'_{\gamma,i} = (P_{\gamma,x}P_{x,i} + P_{\gamma,i}) \sum_{m=0}^{\infty} (P_{i,x}P_{x,i})^m = (P_{\gamma,x}P_{x,i} + P_{\gamma,i})/(1 - P_{i,x}P_{x,i}). \tag{4.34}$$

This formula can also be applied if either $P_{\gamma,i}$ or $P_{\gamma,x}$ vanishes.

It is easy to show that the new branching probabilities are normalised:

$$\sum_{\gamma\in\Gamma} \frac{P_{\gamma,x}P_{x,i} + P_{\gamma,i}}{1 - P_{i,x}P_{x,i}} = \frac{(1 - P_{i,x})P_{x,i} + (1 - P_{x,i})}{1 - P_{i,x}P_{x,i}} = 1. \tag{4.35}$$

To calculate τ_i' we use the method of Section 4.1.4:

$$\tau_i' = \left[\frac{d}{d\zeta} \sum_{\gamma \in \Gamma} \frac{P_{\gamma,x} P_{x,i} e^{\zeta(\tau_x + \tau_i)} + P_{\gamma,i} e^{\zeta \tau_i}}{1 - P_{i,x} P_{x,i} e^{\zeta(\tau_x + \tau_i)}} \right]_{\zeta=0} = \frac{\tau_i + P_{x,i} \tau_x}{1 - P_{i,x} P_{x,i}}. \tag{4.36}$$

The modified branching probabilities and waiting times could be used in a KMC simulation based upon graph G'. Here we continue to use the notation of Section 4.1.4, where sinks correspond to nodes $a \in A$ and sources to nodes in $b \in B$, and G contains all the nodes in $\mathcal{G}$ expect for the A set, as before. Since the modified branching probabilities, $P'_{\gamma,i}$, in G' subsume the sums over all path paths from i to γ that involve x we would expect the sink probability, $\Sigma^G_{a,b}$, of a trajectory starting at b ending at sink a, to be conserved. However, since each trajectory exiting from $\gamma \in \Gamma$ acquires a time increment equal to the average value, τ_i', the mean first-passage times to individual sinks, $\mathcal{T}^G_{a,b}$, are not conserved in G' (unless there is a single sink). Nevertheless, the overall mean first-passage time to A is conserved, i.e. $\sum_{a \in A} \mathcal{T}^{G'}_{a,b} = \mathcal{T}^{G'}_b = \mathcal{T}^G_b$. To prove these results more formally within the framework of complete sums consider the effect of removing node x on trajectories reaching node $i \in Adj[x]$ from a source node b. The sink probability for a particular sink a is

$$\begin{aligned} \Sigma^G_{a,b} &= \sum_{\xi \in a \leftarrow b} \mathcal{W}_\xi \\ &= \sum_{\xi_1 \in i \leftarrow b} \mathcal{W}_{\xi_1} \sum_{\gamma \in \Gamma} (P_{\gamma,i} + P_{x,i} P_{\gamma,x}) \sum_{m=0}^{\infty} (P_{i,x} P_{x,i})^m \sum_{\xi_2 \in a \leftarrow \gamma} \mathcal{W}_{\xi_2} \\ &= \sum_{\xi_1 \in i \leftarrow b} \mathcal{W}_{\xi_1} \sum_{\gamma \in \Gamma} P'_{\gamma,i} \sum_{\xi_2 \in a \leftarrow \gamma} \mathcal{W}_{\xi_2}, \end{aligned} \tag{4.37}$$

and similarly for any other node adjacent to x. Hence the transformation preserves the individual sink probabilities for any source.

Now consider the effect of removing node x on the mean first-passage time from

source b to sink a, $\mathcal{T}_{a,b}^{G'}$, using the approach of Section 4.1.4.

$$\begin{aligned}
\mathcal{T}_{a,b}^{G'} &= \left[\frac{d}{d\zeta}\sum_{\xi_1\in i\leftarrow b}\widetilde{\mathcal{W}}_{\xi_1}\sum_{\gamma\in\Gamma}\widetilde{P}'_{\gamma,i}\sum_{\xi_2\in a\leftarrow\gamma}\widetilde{\mathcal{W}}_{\xi_2}\right]_{\zeta=0} \\
&= \sum_{\xi_1\in i\leftarrow b}\left[\frac{d\widetilde{\mathcal{W}}_{\xi_1}}{d\zeta}\right]_{\zeta=0}\sum_{\gamma\in\Gamma}P'_{\gamma,i}\sum_{\xi_2\in a\leftarrow\gamma}\mathcal{W}_{\xi_2} \\
&\quad+\sum_{\xi_1\in i\leftarrow b}\mathcal{W}_{\xi_1}\sum_{\gamma\in\Gamma}\left[\frac{d\widetilde{P}'_{\gamma,i}}{d\zeta}\right]_{\zeta=0}\sum_{\xi_2\in a\leftarrow\gamma}\mathcal{W}_{\xi_2} \\
&\quad+\sum_{\xi_1\in i\leftarrow b}\mathcal{W}_{\xi_1}\sum_{\gamma\in\Gamma}P'_{\gamma,i}\sum_{\xi_2\in a\leftarrow\gamma}\left[\frac{d\widetilde{\mathcal{W}}_{\xi_2}}{d\zeta}\right]_{\zeta=0},
\end{aligned} \tag{4.38}$$

where the tildes indicate that every branching probability $P_{\alpha,\beta}$ is replaced by $P_{\alpha,\beta}e^{\xi\tau_\beta}$, as above. The first and last terms are unchanged from graph G in this construction, but the middle term,

$$\begin{aligned}
&\sum_{\xi_1\in i\leftarrow b}\mathcal{W}_{\xi_1}\sum_{\gamma\in\Gamma}\left[\frac{d\widetilde{P}'_{\gamma,i}}{d\zeta}\right]_{\zeta=0}\sum_{\xi_2\in a\leftarrow\gamma}\mathcal{W}_{\xi_2} \\
&= \sum_{\xi_1\in i\leftarrow b}\mathcal{W}_{\xi_1}\sum_{\gamma\in\Gamma}\frac{P_{\gamma,x}P_{x,i}(\tau_i+\tau_x)+P_{\gamma,i}(\tau_i+P_{i,x}P_{x,i}\tau_x)}{(1-P_{i,x}P_{x,i})^2}\sum_{\xi_2\in a\leftarrow\gamma}\mathcal{W}_{\xi_2},
\end{aligned} \tag{4.39}$$

is different (unless there is only one sink). However, if we sum over sinks then

$$\sum_{a\in A}\sum_{\xi_2\in a\leftarrow\gamma}\mathcal{W}_{\xi_2}=1 \tag{4.40}$$

for all γ, and we can now simplify the sum over γ as

$$\sum_{\gamma\in\Gamma}\frac{P_{\gamma,x}P_{x,i}(\tau_i+\tau_x)+P_{\gamma,i}(\tau_i+P_{i,x}P_{x,i}\tau_x)}{(1-P_{i,x}P_{x,i})^2}=\tau'_i=\sum_{\gamma\in\Gamma}P'_{\gamma,i}\tau'_i. \tag{4.41}$$

The same argument can be applied whenever a trajectory reaches a node adjacent to x, so that $\mathcal{T}_b^G=\mathcal{T}_b^{G'}$, as required.

The above procedure extends the BKL approach [190] to exclude not only the transitions from the current state into itself but also transitions involving an adjacent node x. Alternatively, this method could be viewed as a coarse-graining of the Markov chain. Such coarse-graining is acceptable if the property of interest is the average of the distribution of times rather than the distribution of times itself. In our simulations the average is the only property of interest. In cases when the distribution itself is

sought, the approach described here may still be useful and could be the first step in the analysis of the distribution of escape times, as it yields the exact average of the distribution.

The transformation is illustrated in Figure 4.8 for the case of a single source. Figure 4.8 (a) displays the original graph and its parametrisation. During the first iteration of the algorithm node 2 is removed to yield the graph depicted in Figure 4.8 (b). This change reduces the dimensionality of the original graph, as the new graph contains one node and three edges fewer. The result of the second, and final, iteration of the algorithm is a graph that contains source and sink nodes only, with the correct transition probabilities and mean waiting time [Figure 4.8 (c)].

We now describe algorithms to implement the above approach and calculate mean escape times from complete graphs with multiple sources and sinks. Listings for some of the algorithms discussed here are given in Appendix B. We follow the notation of Section 4.1.4 and consider a digraph $\mathcal{G}_N$ consisting of N_B source nodes, N_A sink nodes, and N_I intervening nodes. $\mathcal{G}_N$ therefore contains the subgraph $G_{N_I+N_B}$.

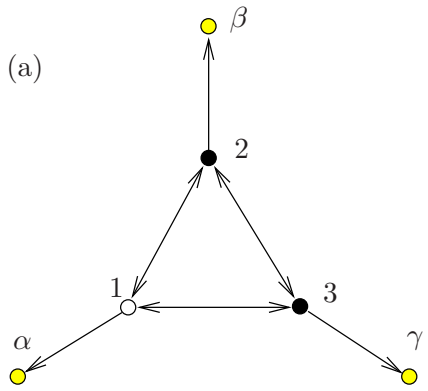
The result of the transformation of a graph with a single source b and N_A sinks using Algorithm B.3 is the mean escape time $\mathcal{T}_b^{G_{N_I+1}}$ and N_A pathway probabilities $\mathcal{P}_\xi$, $\xi \in A \leftarrow b$. Solving a problem with N_B sources is equivalent to solving N_B single source problems. For example, if there are two sources b_1 and b_2 we first solve a problem where only node b_1 is set to be the source to obtain $\mathcal{T}_{b_1}^{G_{N_I+N_B}}$ and the pathway sums from b_1 to every sink node $a \in A$. The same procedure is then followed for b_2.

The form of the transition probability matrix $\mathbf{P}$ is illustrated below at three stages: first for the original graph, then at the stage when all the intervening nodes have been removed (line 16 in Algorithm B.3), and finally at the end of the procedure:

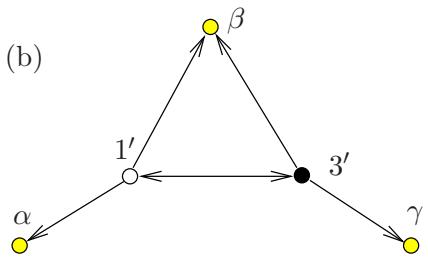
$$\left(\begin{array}{c|c|c} \mathbf{0} & A \leftarrow I & A \leftarrow B \\ \hline \mathbf{0} & I \leftrightarrows I & I \leftarrow B \\ \hline \mathbf{0} & B \leftarrow I & B \leftrightarrows B \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} \mathbf{0} & \mathbf{0} & A \leftarrow B \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & B \leftrightarrows B \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} \mathbf{0} & \mathbf{0} & A \leftarrow B \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right), \quad (4.42)$$

Each matrix is split into blocks that specify the transitions between the nodes of a particular type, as labelled. Upon termination, every element in the top right block of matrix $\mathbf{P}$ is non-zero.

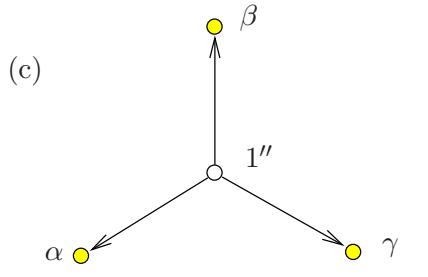
Algorithm B.3 uses the adjacency matrix representation of graph $\mathcal{G}_N$, for which the



$P_{2,1}, P_{3,1}, P_{\alpha,1}, P_{1,2}, P_{3,2}, P_{\beta,2},$
$P_{1,3}, P_{2,3}, P_{\gamma,3}, \tau_1, \tau_2, \tau_3.$



$$P_{\alpha,1'} = \frac{P_{\alpha,1}}{1 - P_{1,2,1}}, P_{\beta,1'} = \frac{P_{\beta,2}P_{2,1}}{1 - P_{1,2,1}},$$
$$P_{3',1'} = \frac{P_{3,1} + P_{3,2,1}}{1 - P_{1,2,1}}, P_{1',3'} = \frac{P_{1,3} + P_{1,2,3}}{1 - P_{2,3,2}},$$
$$P_{\beta,3'} = \frac{P_{\beta,2}P_{2,3}}{1 - P_{2,3,2}}, P_{\gamma,3'} = \frac{P_{\gamma,3}}{1 - P_{2,3,2}},$$
$$\tau_{1'} = \frac{\tau_1 + \tau_2 P_{2,1}}{1 - P_{1,2,1}}, \tau_{3'} = \frac{\tau_3 + \tau_2 P_{2,3}}{1 - P_{2,3,2}}.$$



$$P_{\alpha,1''} = P_{\alpha,1}\mathcal{S}^{K_3}_{1,1}, P_{\beta,1''} = P_{\beta,2}\mathcal{S}^{K_3}_{2,1},$$
$$P_{\gamma,1''} = P_{\gamma,3}\mathcal{S}^{K_3}_{3,1}, \tau_{1''} = \tau^{K_3}_1.$$

Figure 4.8: The graph transformation algorithm of Section 4.4 at work. (a) A digraph with 6 nodes and 9 edges. The source node is node 1 (white), the sinks are nodes α, β and γ (shaded), and the intermediate nodes are 2 and 3 (black). The waiting times and transition probabilities that parametrise the graph are given below the diagram. (b) Node 2 and all its incoming and outgoing edges are deleted from the graph depicted in (a). Two edges $\beta \leftarrow 1$ and $\beta \leftarrow 3$ are added. The parameters for this new graph are denoted by primes and expressed in terms of the parameters for the original graph. (c) Node 3 is now disconnected as well. The resulting graph is composed of source and sink nodes only. The total probability and waiting times coincide with these of K_3, as expected. The new parameters are denoted by a double prime.

average of the distribution of mean first passage times is to be obtained. For efficiency, when constructing the transition probability matrix $\mathbf{P}$ we order the nodes with the sink nodes first and the source nodes last. Algorithm B.3 is composed of two parts. The first part (lines 1-16) iteratively removes all the intermediate nodes from graph $\mathcal{G}_N$ to yield a graph that is composed of sink nodes and source nodes only. The second part (lines 17-34) disconnects the source nodes from each other to produce a graph with $N_A + N_B$ nodes and $(N_A + N_B)^2$ directed edges connecting each source with every sink. Lines 13-15 are not required for obtaining the correct answer, but the final transition probability matrix looks neater.

The computational complexity of lines 1-16 of Algorithm B.3 is $\mathcal{O}(N_I^3 + N_I^2 N_B + N_I^2 N_A + N_I N_B^2 + N_I N_B N_A)$. The second part of Algorithm B.3 (lines 17-34) scales as $\mathcal{O}(N_B^3 + N_B^2 N_A)$. The total complexity for the case of a single source and for the case when there are no intermediate nodes is $\mathcal{O}(N_I^3 + N_I^2 N_A)$ and $\mathcal{O}(N_B^3 + N_B^2 N_A)$, respectively. The storage requirements of Algorithm B.3 scale as $\mathcal{O}\left((N_I + N_B)^2\right)$.

We have implemented the algorithm in Fortran 95 and timed it for complete graphs of different sizes. The results presented in Figure 4.9 confirm the overall cubic scaling. The program is GPL-licensed [273] and available online [274]. These and other benchmarks presented in this chapter were obtained for a single Intel® Pentium® 4 3.00GHz 512 Kb cache processor running under the Debian GNU/Linux operating system [275]. The code was compiled and optimised using the Intel® Fortran compiler for Linux.

4.5 Applications to Sparse Random Graphs

Algorithm B.3 could easily be adapted to use adjacency-lists-based data structures [154], resulting in a faster execution and lower storage requirements for sparse graphs. We have implemented [274] a sparse-optimised version of Algorithm B.3 because the graph representations of the Markov chains of interest in the present work are sparse [201].

The algorithm for detaching a single intermediate node from an arbitrary graph stored in a sparse-optimised format is given in Algorithm B.4. Having chosen the node to be removed, γ, all the neighbours $\beta \in Adj[\gamma]$ are analysed in turn, as follows. Lines 3-9 of Algorithm B.4 find node γ in the adjacency list of node β. If β is not a sink, lines 11-34 are executed to modify the adjacency list of node β: lines 13-14 delete node

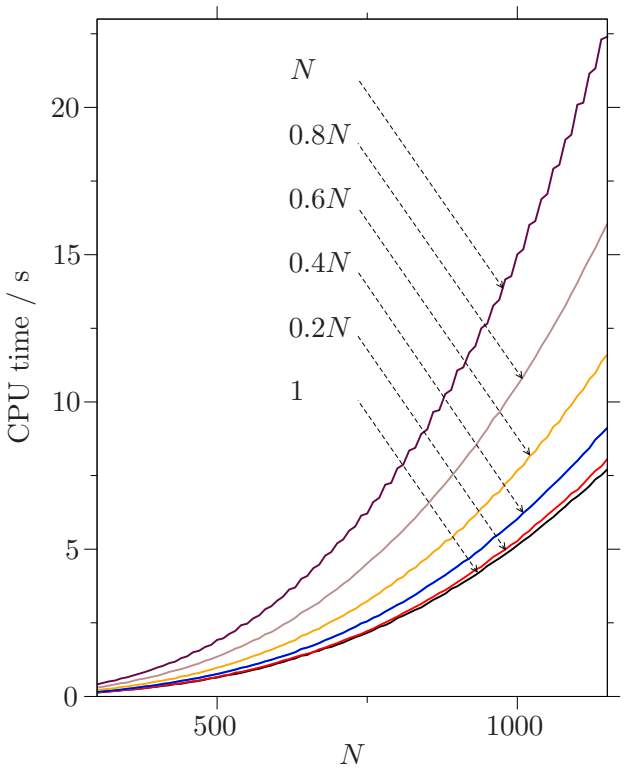


Figure 4.9: CPU time needed to transform a dense graph G_{2N} using Algorithm B.3 as a function of N. The graph G_{2N} is composed of a K_N subgraph and N sink nodes. The data is shown for six different cases, when there was a single source, and when the sources comprised 20, 40, 60, 90, and 100 percent of the number of nodes in K_N, as labelled. The data for the cases 1 and N was fitted as $5.1 \times 10^{-9} N^3$ and $1.5 \times 10^{-8} N^3$, respectively (curves not shown). For case 1 only DetachNode operations were performed while for N — only Disconnect.

γ from the adjacency list of β, while lines 15-30 make all the neighbours $\alpha \in Adj[\gamma] \ominus \beta$ of node γ the neighbours of β. The symbol $\ominus$ denotes the union minus the intersection of two sets, otherwise known as the symmetric difference. If the edge $\beta \to \alpha$ already existed only the branching probability is changed (line 21). Otherwise, a new edge is created and the adjacency and branching probability lists are modified accordingly (line 26 and line 27, respectively). Finally, the branching probabilities of node β are renormalised (lines 31-33) and the waiting time for node β is increased (line 34).

Algorithm B.4 is invoked iteratively for every node that is neither a source nor a sink to yield a graph that is composed of source nodes and sink nodes only. Then

the procedure described in Section 4.4 for disconnection of source nodes (lines 17-34 of Algorithm B.3) is applied to obtain the mean escape times for every source node. The sparse-optimised version of the second part of Algorithm B.3 is straightforward and is therefore omitted here for brevity.

The running time of Algorithm B.4 is $\mathcal{O}(d_c \sum_{i \in Adj[c]} d_i)$, where d_k is the degree of node k. For the case when all the nodes in a graph have approximately the same degree, d, the complexity is $\mathcal{O}(d^3)$. Therefore, if there are N intermediate nodes to be detached and d is of the same order of magnitude as N, the cost of detaching N nodes is $\mathcal{O}(N^4)$. The asymptotic bound is worse than that of Algorithm B.3 because of the searches through adjacency lists (lines 3-9 and lines 19-24). If d is sufficiently small the algorithm based on adjacency lists is faster.

After each invocation of Algorithm B.4 the number of nodes is always decreased by one. The number of edges, however, can increase or decrease depending on the in- and out-degree of the node to be removed and the connectivity of its neighbours. If node γ is not directly connected to any of the sinks, and the neighbours of node γ are not connected to each other directly, the total number of edges is increased by $d_\gamma(3 - d_\gamma)$. Therefore, the number of edges decreases (by 2) only when $d_\gamma \in \{1, 2\}$, and the number of edges does not change if the degree is 3. For $d_\gamma > 3$ the number of edges increases by an amount that grows quadratically with d_γ. The actual increase depends on how many connections already existed between the neighbours of γ.

The order in which the intermediate nodes are detached does not change the final result and is unimportant if the graph is complete. For sparse graphs, however, the order can affect the running time significantly. If the degree distribution for successive graphs is sharp with the same average, d, then the order in which the nodes are removed does not affect the complexity, which is $\mathcal{O}(d^3 N)$. If the distributions are broad it is helpful to remove the nodes with smaller degrees first. A Fibonacci heap min-priority queue [276] was successfully used to achieve this result. The overhead for maintaining a heap is d_γ increase-key operations (of $\mathcal{O}(\log(N))$ each) per execution of Algorithm B.4. Fortran and Python implementations of Algorithm B.4 algorithm are available online [274].

Random graphs provide an ideal testbed for the GT algorithm by providing control over the graph density. A random graph, R_N, is obtained by starting with a set of N

nodes and adding edges between them at random [33]. In this work we used a random graph model where each edge is chosen independently with probability $\langle d \rangle / (N-1)$, where $\langle d \rangle$ is the target value for the average degree.

The complexity for removal of N nodes can then be expressed as

$$\mathcal{O}\left(\log(N) \sum_{i \in \{1,2,3,\ldots,N\}} \left(d_{c(i)}^2 \sum_{j \in Adj[c(i)]} d_{j,c(i)} \right) \right), \tag{4.43}$$

where $d_{c(i)}$ is the degree of the node, $c(i)$, removed at iteration i, $Adj[c(i)]$ is its adjacency list, and $d_{j,c(i)}$ is the degree of the jth neighbour of that node at iteration i. The computational cost given in Equation 4.43 is difficult to express in terms of the parameters of the original graph, as the cost of every cycle depends on the distribution of degrees, the evolution of which, in turn, is dependent on the connectivity of the original graph in a non-trivial manner (see Figure 4.10). The storage requirements of a sparse-optimised version of GT algorithm scale linearly with the number of edges.

To investigate the dependence of the cost of the GT method on the number of nodes, N, we have tested it on a series of random graphs R_N for different values of N and fixed average degree, $\langle d \rangle$. The results for three different values of $\langle d \rangle$ are shown in Figure 4.11. The motivation for choosing $\langle d \rangle$ from the interval $[3, 5]$ was the fact that most of our stationary point databases have average connectivities for the local minima that fall into this range.

It can be seen from Figure 4.11 that for sparse random graphs R_N the cost scales as $\mathcal{O}(N^4)$ with a small $\langle d \rangle$-dependent prefactor. The dependence of the computational complexity on $\langle d \rangle$ is illustrated in Figure 4.12.

From Figure 4.10 it is apparent that at some point during the execution of the GT algorithm the graph reaches its maximum possible density. Once the graph is close to complete it is no longer efficient to employ a sparse-optimised algorithm. The most efficient approach we have found for sparse graphs is to use the sparse-optimised GT algorithm until the graph is dense enough, and then switch to Algorithm B.3. We will refer to this approach as SDGT. The change of data structures constitutes a negligible fraction of the total execution time. Figure 4.13 depicts the dependence of the CPU time as a function of the switching parameter R_s. Whenever the ratio $d_{c(i)}/n(i)$, where the $d_{c(i)}$ is the degree of intermediate node c detached at iteration i,

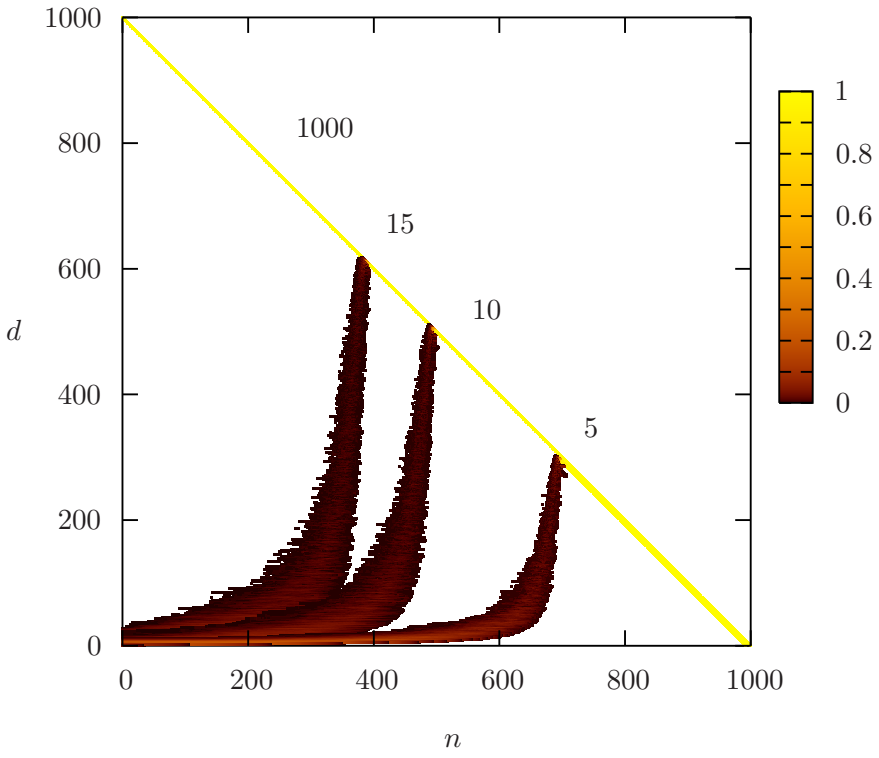


Figure 4.10: Evolution of the distribution of degrees for random graphs of different expected degree, $\langle d \rangle = 5, 10, 15$, as labelled. This is a colour-coded projection of the probability mass function [277, 278], $P(d)$, of the distribution of degrees as a function of the number of the detached intermediate nodes, n. The straight line shows $P(d, n)$ for complete graph K_{1000}. All four graphs contain a single source, 999 intermediate nodes and a single sink. The transformation was done using sparse-optimised version of Algorithm B.3 with Fibonacci-heap-based min-priority queue. It can be seen that as the intermediate nodes are detached the density of the graph that is being transformed grows. The expected degree of the initial graph determines how soon the maximum density will be reached.

and $n(i)$ is the number of the nodes on a heap at iteration i, is greater than R_s, the partially transformed graph is converted from the adjacency list format into adjacency matrix format and the transformation is continued using Algorithm B.3. It can be seen from Figure 4.10 that for the case of a random graphs with a single sink, a single source and 999 intermediate nodes the optimal values of R_s lie in the interval $[0.07, 0.1]$.

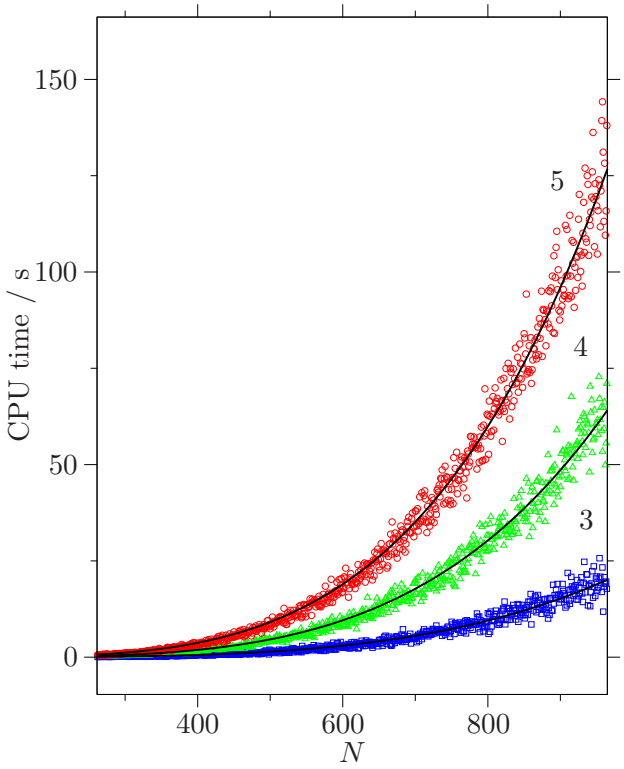


Figure 4.11: CPU time needed to transform a sparse random graph R_{2N} using the GT approach described in Section 4.4 as a function of the number of intermediate nodes, N. R_{2N} is composed of a single source node, N sink nodes and $N-1$ intermediate nodes. For each value of N the data for three different values of the expected degree, $\langle d \rangle = 3, 4, 5$, is shown, as labelled. Solid lines are analytic fits of the form cN^4, where $c = 2.3 \times 10^{-11}, 7.4 \times 10^{-11}, 1.5 \times 10^{-10}$ for $\langle d \rangle = 3, 4, 5$, respectively. CPU time is in seconds.

4.6 Overlapping Sets of Sources and Sinks

We now return to the digraph representation of a Markov chain that corresponds to the DPS pathway ensemble discussed in Section 4.1.4. A problem with (partially) overlapping sets of sources and sinks can easily be converted into an equivalent problem where there is no overlap, and then the GT method discussed in Section 4.4 and Section 4.5 can be applied as normal.

As discussed above, solving a problem with n sources reduces to solving n single-source problems. We can therefore explain how to deal with a problem of overlapping sets of sinks and sources for a simple example where there is a single source-sink i and,

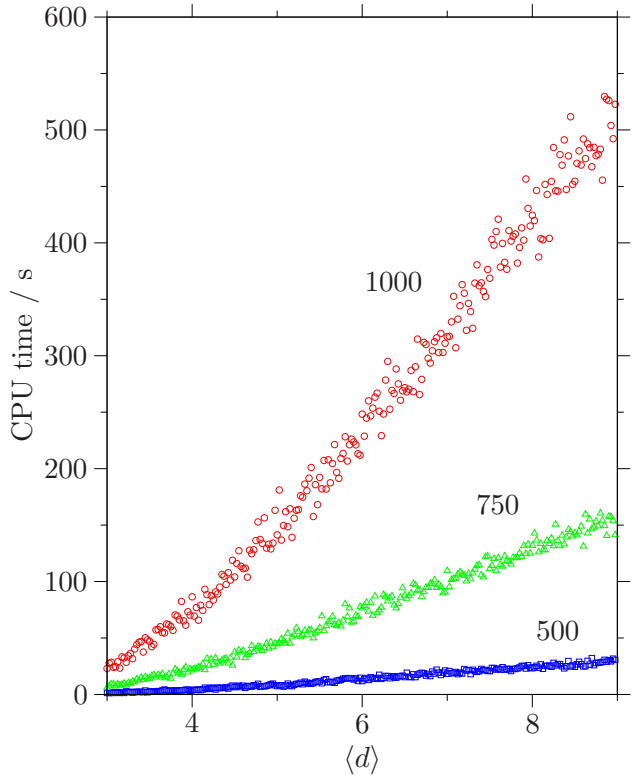


Figure 4.12: CPU time needed to transform a sparse random graph R_{2N} using the GT approach as a function of the expected degree, $\langle d \rangle$. The data is shown for three graphs with $N = 500$, 750 and 1000, as labelled. R_{2N} is composed of a single source node, N sink nodes and $N - 1$ intermediate nodes.

optionally, a number of sink nodes.

First, a new node i' is added to the set of sinks and its adjacency lists are initialised to $AdjOut[i'] = \emptyset$ and $AdjIn[i'] = AdjIn[i]$. Then, for every node $j \in AdjOut[i]$ we update its waiting time as $\tau_j = \tau_j + \tau_i$ and add node j to the set of sources with probabilistic weight initialised to $P_{j,i}W_i$, where W_i is the original probabilistic weight of source i (the probability of choosing source i from the set of sources). Afterwards, the node i is deleted.

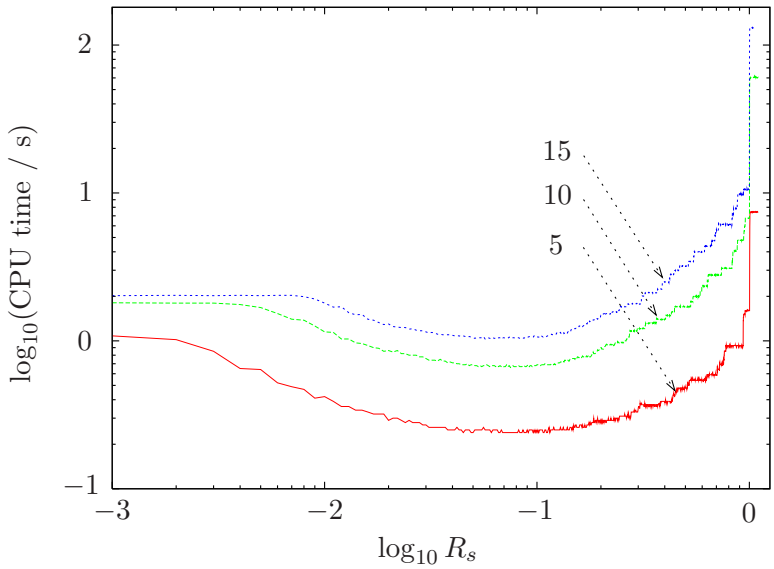


Figure 4.13: CPU time as a function of switching ratio R_s shown for random graphs of different expected degree, $\langle d \rangle = 5, 10, 15$, as labelled. All three graphs contain a single source, 999 intermediate nodes and a single sink. The transformation was performed using the sparse-optimised version of Algorithm B.3 until the the ratio $d_{c(i)}/n(i)$ became greater than R_s. Then a partially transformed graph was converted into adjacency matrix format and the transformation was continued with Algorithm B.3. The optimal value of R_s lies in the interval $[0.07, 0.1]$. Note the $\log_{10}$ scale on both axes.

4.7 Applications to Lennard-Jones Clusters

4.7.1 $O_h \leftrightarrow I_h$ Isomerisation of LJ_{38}

We have applied the GT method to study the temperature dependence of the rate of $O_h \leftrightarrow I_h$ interconversion of 38-atom Lennard-Jones cluster. The PES sample was taken from a previous study [8] and contained 1740 minima and 2072 transition states. Only geometrically distinct structures were considered when generating this sample because this way the dimensionality of the problem is reduced approximately by a factor of $2N!/h$, where h is the order of the point group. Initial and final states in this sample roughly correspond to icosahedral-like and octahedral-like structures on the PES of this cluster. The assignment was made in Reference [8] by solving master equation numerically to find the eigenvector that corresponds to the smallest non-zero eigenvalue. As simple two-state dynamics are associated with exponential rise

and decay of occupation probabilities there must exist a time scale on which all the exponential contributions to the solution of the master equation decay to zero except for the slowest [9]. The sign of the components of the eigenvector corresponding to the slowest mode was used to classify the minima as I_h or O_h in character [8].

The above sample was pruned to ensure that every minimum is reachable from any other minimum to create a digraph representation that contained 759 nodes including 43 source nodes and 5 sink nodes, and 2639 edges. The minimal, average and maximal degree for this graph were 2, 3.8 and 84, respectively, and the graph density was 4.6×10^{-3}. We have used the SDGT algorithm with the switching ratio set to 0.08 to transform this graph for several values of temperature. In each of these calculations 622 out of 711 intermediate nodes were detached using SGT, and the remaining 89 intermediate nodes were detached using the GT algorithm optimised for dense graphs (DGT).

An Arrhenius plot depicting the dependence of the rate constant on temperature is shown in Figure 4.14 (a). The running time of SDGT algorithm was 1.8×10^{-2} seconds [this value was obtained by averaging over 10 runs and was the same for each SDGT run in Figure 4.14 (a)]. For comparison, the timings obtained using the SGT and DGT algorithms for the same problem were 2.0×10^{-2} and 89.0×10^{-2} seconds, respectively. None of the 43 total escape probabilities (one for every source) deviated from unity by more than 10^{-5} for temperatures above $T = 0.07$ (reduced units). For lower temperatures the probability was not conserved due to numerical imprecision.

The data obtained using SDGT method is compared with results from KMC simulation, which require increasingly more CPU time as the temperature is lowered. Figure 4.14 also shows the data for KMC simulations at temperatures 0.14, 0.15, 0.16, 0.17 and 0.18. Each KMC simulation consisted of the generation of an ensemble of 1000 KMC trajectories from which the averages were computed. The cost of each KMC calculation is proportional to the average trajectory length, which is depicted in Figure 4.14 (b) as a function of the inverse temperature. The CPU timings for each of these calculations were (in the order of increasing temperature, averaged over five randomly seeded KMC simulations): 125, 40, 18, 12, and 7 seconds. It can be seen that using GT method we were able to obtain kinetic data for a wider range of temperatures

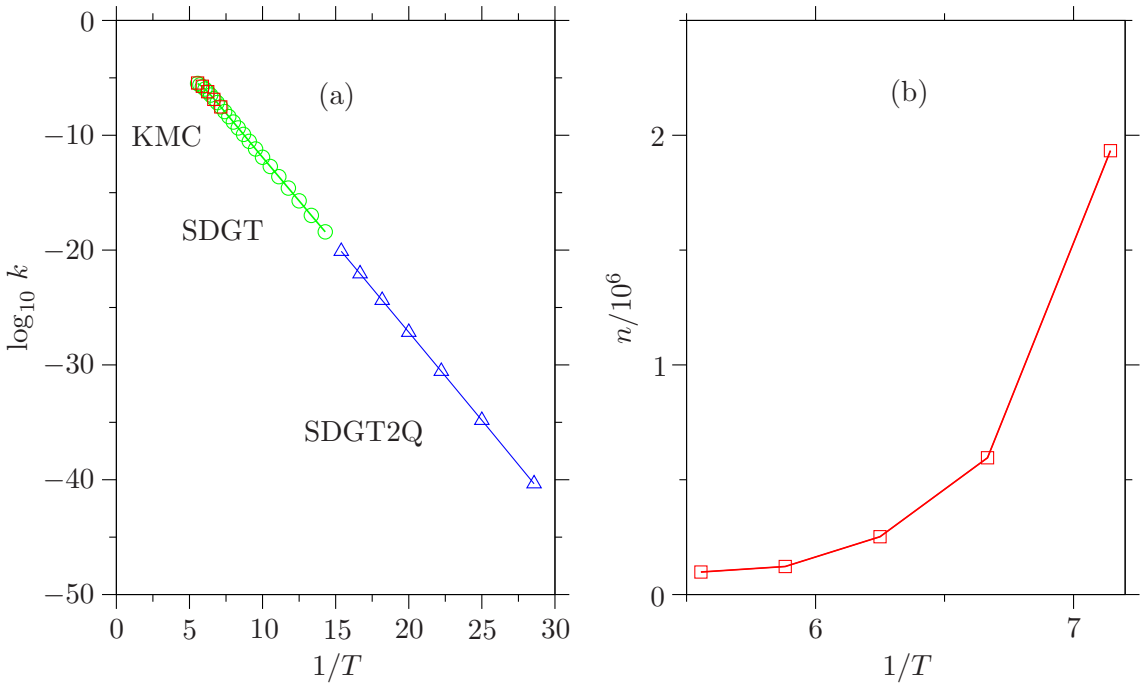


Figure 4.14: (a) Arrhenius plots for the LJ_{38} cluster. k is the rate constant corresponding to transitions from icosahedral-like to octahedral-like regions. Green circles represent the data obtained from 23 SDGT runs at temperatures $T \in \{0.07, 0.075, \ldots, 0.18\}$. The data from five KMC runs is also shown (red squares). The data shown in blue corresponds to temperatures $T \in \{0.035, 0.04, \ldots, 0.065\}$ and was obtained using the SDGT2 method (discussed in Section 4.7.2) with quadruple precision enabled. In all SDGT runs the total escape probabilities calculated for every source at the end of the calculation deviated from unity by no more then 10^{-5}. For this PES sample the lowest temperature for which data was reported in previous works was $T = 0.08$. (b) The average KMC trajectory length [data in direct correspondence with KMC results shown in (a)]. A solid line is used to connect the data points to guide the eye.

and with less computational expense.

4.7.2 Internal Diffusion in LJ_{55}

We have applied the graph transformation method to study the centre-to-surface atom migration in 55-atom Lennard-Jones cluster. The global potential energy minimum for LJ_{55} is a Mackay icosahedron, which exhibits special stability and 'magic number' properties [279, 280]. Centre-to-surface and surface-to-centre rates of migration of a tagged atom for this system were considered in previous work [10]. In Reference [10]

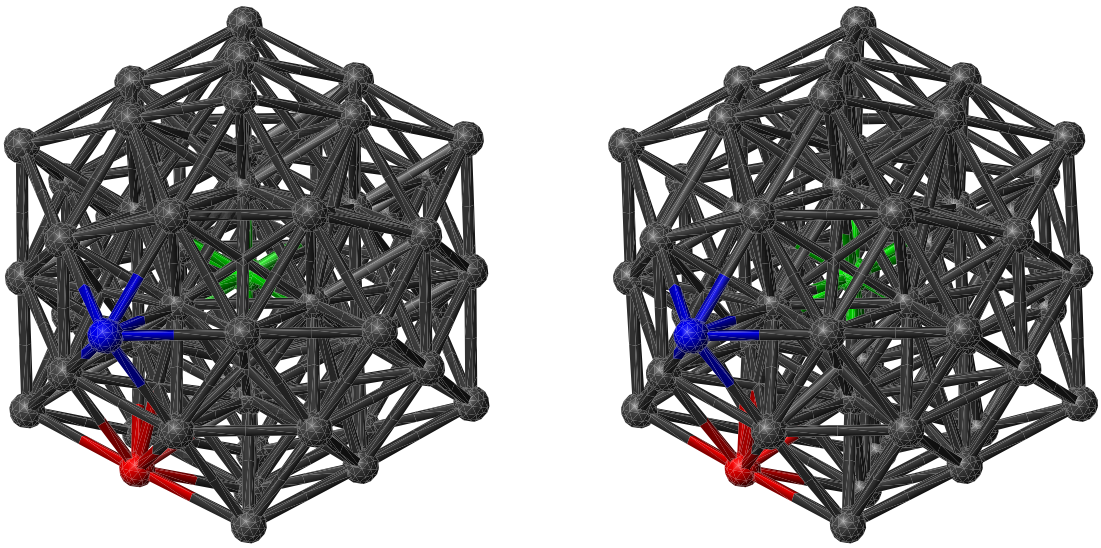


FIGURE 4.15: Global minimum of the LJ_{55} cluster (shown in stereo). The tagged atom can occupy the central position (green) or one of the two different surface sites (red and blue).

the standard DPS procedure was applied to create and converge an ensemble of paths linking the structure of the global minimum with the tagged atom occupying the central position and structures where tagged atom is placed in sites that lie on fivefold and twofold symmetry axes (Figure 4.15). We have reused this sample in the present work.

The sample contained 9907 minima and 19384 transition states. We excluded transition states that facilitate degenerate rearrangements from consideration. For minima interconnected by more than one transition state we added the rate constants in each direction to calculate the branching probabilities. Four digraph representations were created with minimum degrees of 1, 2, 3 and 4 via iterative removal of the nodes with degrees that did not satisfy the requirement. These digraphs will be referred to as digraphs 1, 2, 3 and 4, respectively. The corresponding parameters are summarised in Table 4.1. Since the cost of the GT method does not depend on temperature we also quote CPU timings for the DGT, SGT and SDGT methods for each of these graphs in the last three columns of Table 4.1. Each digraph contained two source nodes labelled 1 and 2 and a single sink. The sink node corresponds to the global minimum with the tagged atom in the centre (Figure 4.15). It is noteworthy that the densities of

Table 4.1: Properties of four digraphs corresponding to the LJ_{55} PES sample from an internal diffusion study. $|V|$ is the number of nodes; $|E|$ is the number of directed edges; d_{min}, $\langle d \rangle$ and d_{max} are the minimum, average and maximum degrees, respectively; ρ is the graph density, defined as a ratio of the number of edges to the maximum possible number of edges; r and d are the graph radius and diameter, defined as the maximum and minimum node eccentricity, respectively, where the eccentricity of a node v is defined as the maximum distance between v and any other node. $\langle l \rangle$ is the average distance between nodes. The CPU time, t, necessary to transform each graph using the DGT, SGT and SDGT methods is given in seconds for a single 32-bit Intel® Pentium® 4 3.00 GHz 512 Kb cache processor.

$\|V\|$	$\|E\|$	d_{min}	$\langle d \rangle$	d_{max}	$\rho/10^{-4}$	r	$\langle l \rangle$	d	t_{DGT}	t_{SGT}	t_{SDGT}
9843	34871	1	3.9	983	3.6	10	5.71	20	2346.1	39.6	1.36
6603	28392	2	4.8	983	6.5	9	4.86	17	1016.1	38.9	1.33
2192	14172	3	7.9	873	29.5	4	3.63	8	46.9	5.9	0.49
865	7552	4	1.9	680	101.0	4	3.07	7	3.1	0.8	0.12

the graphs corresponding to both our samples (LJ_{38} and LJ_{55}) are significantly lower than the values predicted for a complete sample [115], which makes the use of sparse-optimised methods even more advantageous. From Table 4.1 it is clear that the SDGT approach is the fastest, as expected; we will use SDGT for all the rate calculations in the rest of this section.

For this sample KMC calculations are unfeasible at temperatures lower than about $T = 0.3$ (Here T is expressed in the units of ϵ/k_B). Already for $T = 0.4$ the average KMC trajectory length is 7.5×10^6 (value obtained by averaging over 100 trajectories). In previous work it was therefore necessary to use the DPS formalism, which invokes a steady-state approximation for the intervening minima, to calculate the rate constant at temperatures below 0.35 [10]. Here we report results that are in direct correspondence with the KMC formulation of the problem, for temperatures as low as 0.1.

Figure 4.16 presents Arrhenius plots that we calculated using the SDGT method for this system. The points in the green dataset are the results from seven SDGT calculations at temperatures $T \in \{0.3, 0.35, \ldots, 0.6\}$ conducted for each of the digraphs. The total escape probabilities, Σ_1^G and Σ_2^G, calculated for each of the two sources at

the end of the calculation deviated from unity by no more than 10^{-5}. For higher temperatures and smaller digraphs the deviation was smaller, being on the order of 10^{-10} for digraph 4 at $T = 0.4$, and improving at higher temperatures and/or smaller graph sizes.

At temperatures lower than 0.3 the probability deviated by more than 10^{-5} due to numerical imprecision. This problem was partially caused by the round-off errors in evaluation of terms $1 - P_{\alpha,\beta}P_{\beta,\alpha}$, which increase when $P_{\alpha,\beta}P_{\beta,\alpha}$ approaches unity. These errors can propagate and amplify as the evaluation proceeds. By writing

$$\begin{aligned} P_{\alpha,\beta} &= 1 - \sum_{\gamma\neq\alpha} P_{\gamma,\beta} \equiv 1 - \epsilon_{\alpha,\beta} \\ \text{and} \quad P_{\beta,\alpha} &= 1 - \sum_{\gamma\neq\beta} P_{\gamma,\alpha} \equiv 1 - \epsilon_{\beta,\alpha}, \end{aligned} \tag{4.44}$$

and then using

$$1 - P_{\alpha,\beta}P_{\beta,\alpha} = \epsilon_{\alpha,\beta} - \epsilon_{\alpha,\beta}\epsilon_{\beta,\alpha} + \epsilon_{\beta,\alpha} \tag{4.45}$$

we were able to decrease $1 - \Sigma_\alpha^G$ by several orders of magnitude at the expense of doubling the computational cost. The SDGT method with probability denominators evaluated in this fashion will be referred to as SDGT1.

Terms of the form $1 - P_{\alpha,\beta}P_{\beta,\alpha}$ approach zero when nodes α and β become 'effectively' (i.e. within available precision) disconnected from the rest of the graph. If this condition is encountered in the intermediate stages of the calculation it could also mean that a larger subgraph of the original graph is effectively disconnected. The waiting time for escape if started from a node that belongs to this subgraph tends to infinity. If the probability of getting to such a node from any of the source nodes is close to zero the final answer may still fit into available precision, even though some of the intermediate values cannot. Obtaining the final answer in such cases can be problematic as division-by-zero exceptions may occur.

Another way to alleviate numerical problems at low temperatures is to stop round-off errors from propagation at early stages by renormalising the branching probabilities of affected nodes $\beta \in Adj[\gamma]$ after node γ is detached. The corresponding check that the updated probabilities of node β add up to unity could be inserted after line 33 of Algorithm B.4 (see Appendix B), and similarly for Algorithm B.3. A version of SDGT method with this modification will be referred to as SDGT2.

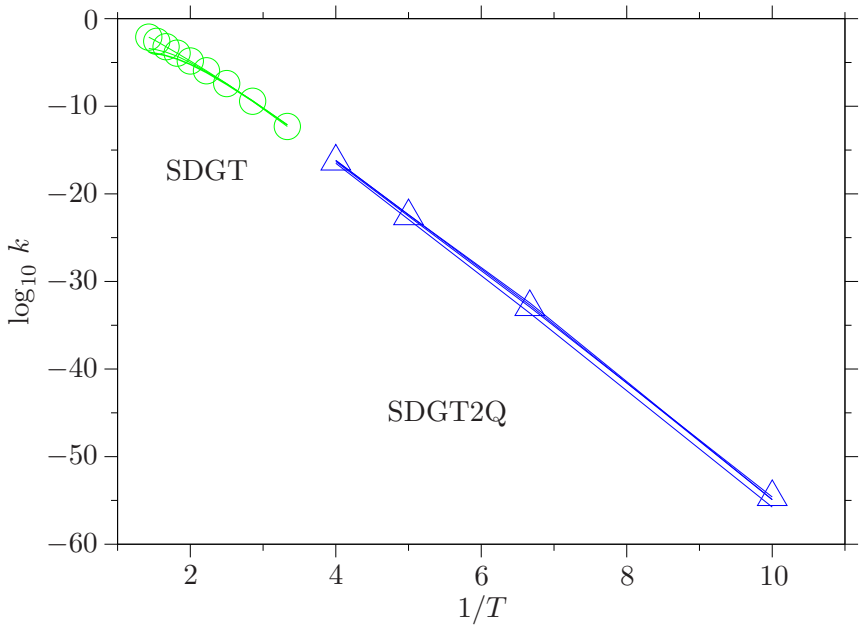


Figure 4.16: Arrhenius plots for four digraphs of varying sizes (see Table 4.1) created from a sample of the PES for the LJ_{55} cluster. k is the rate constant corresponding to surface-to-centre migration of a tagged atom. Calculations were conducted at $T \in \{0.3, 0.35, \ldots, 0.7\}$ using the SDGT method (green) and $T \in \{0.1, 0.15, \ldots, 0.25\}$ using SDGT2Q (blue). For each of the digraphs the calculations yielded essentially identical results so data points for only one of them are shown.

Both SDGT1 and SDGT2 have similarly scaling overheads relative to the SDGT method. We did not find any evidence for superiority of one scheme over another. For example, the SDGT calculation performed for digraph 4 at $T = 0.2$ yielded $\mathcal{T}^G = \mathcal{T}_1^G W_1 + \mathcal{T}_2^G W_2 = 6.4 \times 10^{-18}$, and precision was lost as both Σ_1^G and Σ_2^G were less than 10^{-5}. The SDGT1 calculation resulted in $\mathcal{T}^G = 8.7 \times 10^{-22}$ and $\Sigma_1^G = \Sigma_2^G = 1.0428$, while the SDGT2 calculation produced $\mathcal{T}^G = 8.4 \times 10^{-22}$ with $\Sigma_1^G = \Sigma_2^G = 0.99961$. The CPU time required to transform this graph using our implementations of the SDGT1 and SDGT2 methods was 0.76 and 0.77 seconds, respectively.

To calculate the rates at temperatures in the interval $[0.1, 0.3]$ reliably we used an implementation of the SDGT2 method compiled with quadruple precision (SDGT2Q) (note that the architecture is the same as in other benchmarks, i.e. with 32 bit wide registers). The points in the blue dataset in Figure 4.16 are the results from 4 SDGT2Q calculations at temperatures $T \in \{0.10, 0.35, \ldots, 0.75\}$.

By using SDGT2Q we were also able to improve on the low-temperature results for LJ_{38} presented in the previous section. The corresponding data is shown in blue in Figure 4.14.

4.8 Summary

The most important result of this chapter is probably the graph transformation (GT) method. The method works with a digraph representation of a Markov chain and can be used to calculate the first moment of a distribution of the first-passage times, as well as the total transition probabilities for an arbitrary digraph with sets of sources and sinks that can overlap. The calculation is performed in a non-iterative and non-stochastic manner, and the number of operations is independent of the simulation temperature.

We have presented three implementations of the GT algorithm: sparse-optimised (SGT), dense-optimised (DGT), and hybrid (SDGT), which is a combination of the first two. The SGT method uses a Fibonacci heap min-priority queue to determine the order in which the intermediate nodes are detached to achieve slower growth of the graph density and, consequently, better performance. SDGT is identical to DGT if the graph is dense. For sparse graphs SDGT performs better then SGT because it switches to DGT when the density of a graph being transformed approaches the maximum. We find that for SDGT method performs well both for sparse and dense graphs. The worst case asymptotic scaling of SDGT is cubic.

We have also suggested two versions of the SDGT method that can be used in calculations where a greater degree of precision is required. The code that implements SGT, DGT, SDGT, SDGT1 and SDGT2 methods is available for download [274].

The connection between the DPS and KMC approaches was discussed in terms of digraph representations of Markov chains. We showed that rate constants obtained using the KMC or DPS methods can be computed using graph transformation. We have presented applications to the isomerisation of the LJ_{38} cluster and the internal diffusion in the LJ_{55} cluster. Using the GT method we were able to calculate rate constants at lower temperatures than was previously possible, and with less computational expense.

We also obtained analytic expressions for the total transition probabilities for arbitrary digraphs in terms of combinatorial sums over pathway ensembles. It is hoped

that these results will help in further theoretical pursuits, e.g. these aimed at obtaining higher moments of the distribution of the first passage times for arbitrary Markov chains.

Finally, we showed that the recrossing contribution to the DPS rate constant of a given discrete pathway can be calculated exactly. We presented a comparison between a sparse-optimised matrix multiplication method and a sparse-optimised version of Algorithm B.1 and showed that it is beneficial to use Algorithm B.1 because it is many orders of magnitude faster, runs in linear time and has constant memory requirements.

Chapter 5

Conclusions

Still round the corner there may wait
A new road or a secret gate,
And though we pass them by today,
Tomorrow we may come this way
And take the hidden paths that run
Towards the Moon or to the Sun.

J. R. R. Tolkien, A walking song

5.1 Summary of Contributions

In this thesis I have tried to improve upon existing double-ended methods for finding rearrangement pathways as well as methods for extracting kinetic information from pathway ensembles. The main accomplishments are as follows:

(I) We presented a graph transformation (GT) method, which can be used to calculate the total transition probabilities and mean escape times for arbitrary digraphs with arbitrary sets of sources and sinks that are allowed to overlap. At low temperatures the GT method becomes the method of choice, outperforming kinetic Monte Carlo and matrix multiplication methods.

(II) We have suggested a version of the GT method (SDGT) that can take full advantage of the sparsity of the problem. Apart from switching to the standard sparse-optimised adjacency-list-based data structure, the modifications were the

implementation of Fibonacci-heap-based min-priority queue to ensure that nodes with smaller degrees are detached first, and an algorithm that monitors the graph density and switches to the dense-optimised version of the method when it becomes computationally cost-effective.

(III) The stability of the NEB method was improved by introducing a portion of the spring gradient component perpendicular to the path back into the NEB gradient.

(IV) The efficiency of the DNEB method was improved by eliminating the removal of the overall rotation and translation and employing a quasi-Newton method (L-BFGS) for minimisation of the band.

(V) The efficiency and stability of the QVV minimiser was increased by finding the optimal point in time of quenching the velocity.

(VI) We have devised a method for finding rearrangement pathways between distant local minima, which is based on the consecutive DNEB searches and uses the Euclidean distance as a measure of separation in configuration space. Part of this work that concerned the Dijkstra-based selector was performed in collaboration with Dr. Joanne M. Carr [167].

(VII) A new cooperativity index, introduced in Chapter 3, enabled us to find a correlation between the cooperativity of an atomic rearrangement and the energy barrier. We showed that cooperative rearrangements of LJ clusters and the BLJ liquid have lower energy barriers irrespective of the degree of localisation.

(VIII) We have demonstrated that it is possible to control the overall cooperativity of the pathway sample, and outlined a technique for sampling cooperative pathways using single-ended transition state searching methods.

(IX) We have described the edge weight function that allows us to find the path with the largest DPS non-recrossing rate constant using the Dijkstra algorithm. We have also described an algorithm for sampling for the fastest paths.

(X) We have devised a method for computing a recrossing contribution to the DPS rate constant exactly in linear time with constant memory requirements.

(XI) We have obtained recursive expressions for the total transition probabilities from an arbitrary digraph by considering the corresponding pathway ensembles.

Because random walk has applications in many areas of sciences, from Brownian motion [120] and diffusion [121] in physics to dynamics of stock markets in economics [281] and tumour angiogenesis in medicine [282], we expect methods developed in Chapter 4 to be relevant to a much wider domain.

5.2 Future Research

Avenues for future research based on the results of this thesis may be opened by trying to answer the following questions:

(I) If a pathway sample that accurately reproduces the kinetics of the complete pathway ensemble is sought what is the best sampling strategy to use?

(II) Do cooperative rearrangements start to dominate the relaxation processes at lower temperatures?

(III) What is the relationship between the number of cooperative pathways supported by a PES and the form of the potential?

(IV) Can cooperative moves improve existing methods for global optimisation and evolution of kinetic databases?

As a number of alternative double-ended approaches have been developed in the past few years, such as, for example, the string method [283–287], the growing string method [109, 288], and a super-linear NEB based on adopted basis Newton-Raphson minimiser [289], it would be interesting to make a detailed comparison of these methods on a set of problems we are likely to be solving in the future.

The connection algorithm and the algorithm for sampling the fastest path presented in Chapter 2 and Appendix E, respectively, are far from optimum because they operate on evolving databases but use the static Dijkstra algorithm to build the shortest path tree. When applying these methods to databases larger than these discussed in this thesis the use of dynamic graph algorithms may be of benefit.

It would be exciting to see more applications of the GT method that would allow us to come to a more detailed understanding of its strengths and weaknesses. Comparisons with sparse-optimised numerical approaches for solving the master equation and iterative solvers based on first-step analysis are also desirable.

From a theoretical point of view, it would be interesting to extend the approach of Section 4.3 to obtain higher moments of the escape time distribution function and maybe even the distribution function itself.

Although the DNEB method and connection algorithm presented in this thesis have seen a number of successful applications already the scope for further applications and development is ample. Potential areas include the design of better initial pathway guessing strategies for proteins, reduction of memory requirements of the connection algorithm, better understanding of the relationship of the optimal edge weight function and the form of the potential, and parallelisation of these methods for distributed-memory computing, to name but a few. Much ongoing work is now focused on attempts to construct an initial folding pathway for a medium-sized protein, barnase. Preliminary results showed that further improvements (with emphasis on the large number of degrees of freedom) to both double-ended and single-ended transition state searching methods are required to complete this task.

5.3 Outlook

There is a need for future research to address the gap between the force field designers and the energy landscapes community. Success in the application of many methods discussed in this thesis is contingent on the potential energy function being continuous and well-defined. Some of the most promising potential energy functions developed recently can only be used with methods described here after serious modifications.

Ultimately, our ability to make valid predictions about the properties of any system is limited by the accuracy of the force field. It is thus hoped that more research will be performed in the direction of the development of realistic, cheap and friendly force fields.

Appendix A

Rotamers in the CHARMM19 Force Field

> The people may be made to follow a path of action,
> but they may not be made to understand it.
>
> Confucius (551 BC - 479 BC)

$V\Big(\mathbf{r}_1(j), \mathbf{r}_2(j), \mathbf{r}_3(j), \ldots, \mathbf{r}_N(j)\Big)$ is the energy functional that describes a system with N atoms, where $\mathbf{r}_i(j)$ is the three-dimensional vector that contains the Cartesian coordinates of atom i for structure j, i.e. $\mathbf{r}_i(j) = \Big(X_i(j), Y_i(j), Z_i(j)\Big)$, where $X_i(j)$ is the X coordinate of atom i in structure j, etc. Structure A is fully specified by a set of coordinates $\Big\{\mathbf{r}_1(A), \mathbf{r}_2(A), \mathbf{r}_3(A), ...\mathbf{r}_N(A)\Big\}$. Ideally, the energy functional is invariant with respect to any permutations within a subset of atoms of the same element. For example, if atoms 1 and 2 are both hydrogens, swapping their coordinates around should not make any difference for the evaluation of energy and its derivatives. This effectively means that 'numbering' or 'labelling' of the atoms within such a subset is completely arbitrary. Swapping the coordinates and relabelling are therefore equivalent operations.

The CHARMM19 energy functional [4] is capable of describing some of the permutational isomers. While it is not possible to swap two atoms of the same type that belong to different residues (a rearrangement of this type would involve bond breaking and is thus impossible to model with CHARMM), rotations of the side chains can map

an initial structure onto a permutational isomer. Such rotational isomers (rotamers) can be thought of as structures where symmetry-related atoms are relabelled. For example, in the phenylalanine side chain a rotation around a bond CB–CG by 180 degrees generates a structure that could otherwise be obtained via permutation CD1 $\Leftrightarrow$ CD2 followed by CE1 $\Leftrightarrow$ CE2 [see Figure A.1(Phe)]. CB, CG, CD1 etc. are CHARMM19 atom types as specified in CHARMM19 topology and parameter files, toph19.inp and param19.inp.

Landscape searching using the DPS method [8, 10] involves precise identification of the stationary points, which is best done by comparing their energies and the principal components of the inertia tensor. Recently Evans and Wales applied DPS to met-enkephalin and the B1 domain of protein G [133, 197, 290]. They noted that permutational isomers of stationary points using the CHARMM19 force field can have slightly different energies and geometries even if tightly optimised.*

There are eight amino acids that have side chain rotamers in CHARMM19. The aromatic rings in tyrosine and phenylalanine have two pairs of carbon-hydrogen united atoms in the ring that are equivalent, namely, CD1, CD2 and CE1, CE2. Aspartic and glutamic acids each have a pair of equivalent oxygens — OD1, OD2 and OE1, OE2, respectively. Ball-and-stick models of Phe, Tyr, Asp and Glu amino acids are shown in Figure A.1. Asparagine, glutamine and arginine have equivalent hydrogens in amino groups. Lysine has three equivalent hydrogens in the NH_3^+– group. Asn, Gln, Arg and Lys amino acids are depicted in Figure A.2. Standard CHARMM19 N- and C-terminal capping groups 'Nter' and 'Cter' also have rotational isomers. Finally, methyl groups in valine and leucine side chains can be swapped without bond breaking. Cter, Nter, Val and Leu are shown in Figure A.3.

Rotamers of tyrosine, asparagine, glutamine and the C-terminal residue Cter can have different energies in the CHARMM19 force field. The energy difference can be as large as 10^{-3} kcal/mol for tightly optimised structures and is due to the asymmetry in the definition of the dihedral angle in the case of tyrosine, asparagine and glutamine, and in the definition of the improper dihedral angle in the case of Cter.

It is easy to demonstrate this effect for tyrosine with a structure for which the axis of

*The authors mentioned rotation of tyrosine ring system and $-COO^-$ groups as well as swapping of the two $-CH_3$ groups in valine side chains [290].

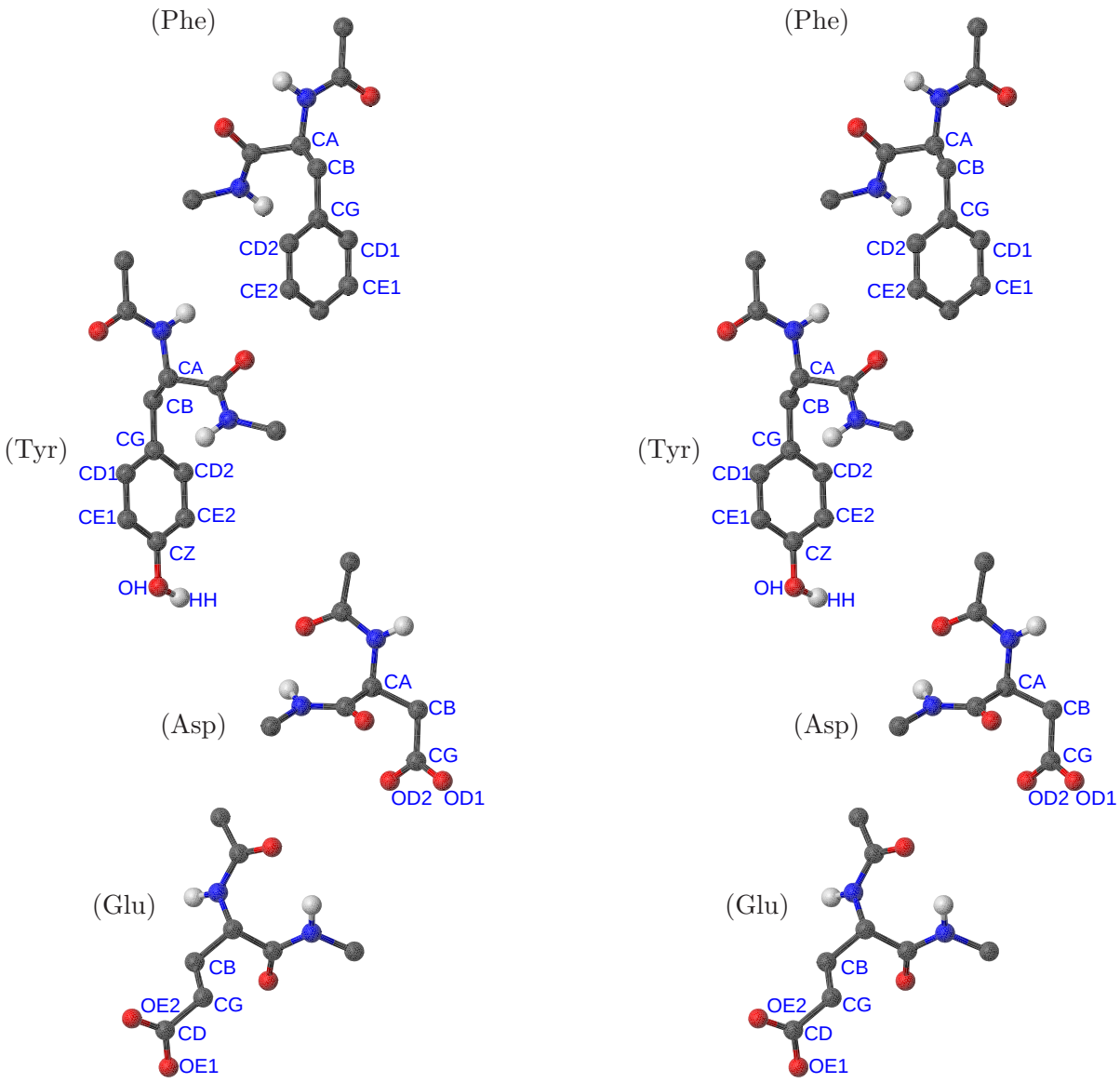


FIGURE A.1: Amino acids that have ring- and carboxylic-based rotameric sidechains. Stereo ball-and-stick models of CHARMM19 representations of phenylalanine, tyrosine, aspartic and glutamatic acids are shown. For the description of the atom colour-coding see the caption to Figure 2.10. Each molecule was optimised using CHARMM19, and has standard Ace and Cbx capping groups blocking N- and C-termini, respectively. Atoms involved in the definition of asymmetric torsions are labelled with their respective CHARMM19 atom names.

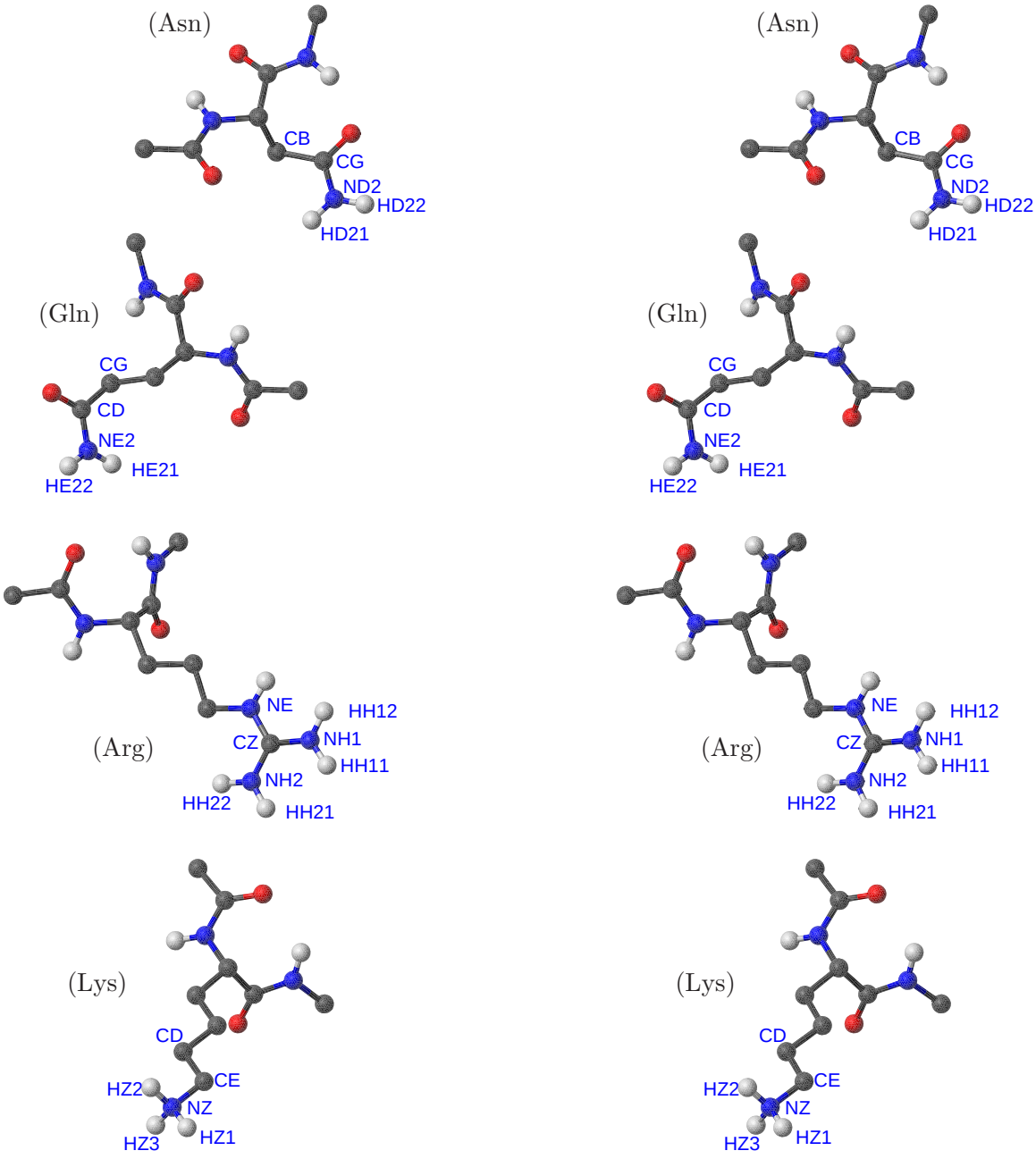


FIGURE A.2: Amino acids that have isomers due to rotation of $-NH_2$ and $-NH_3^+$ groups. Stereo ball-and-stick models of CHARMM19 representations of asparagine, glutamine, arginine and lysine are shown. For the description of the atom colour-coding see the caption to Figure 2.10. Each molecule was optimised using CHARMM19, and has standard Ace and Cbx capping groups blocking N- and C-termini, respectively. Atoms involved in the definition of asymmetric torsions are labelled with their respective CHARMM19 atom names.

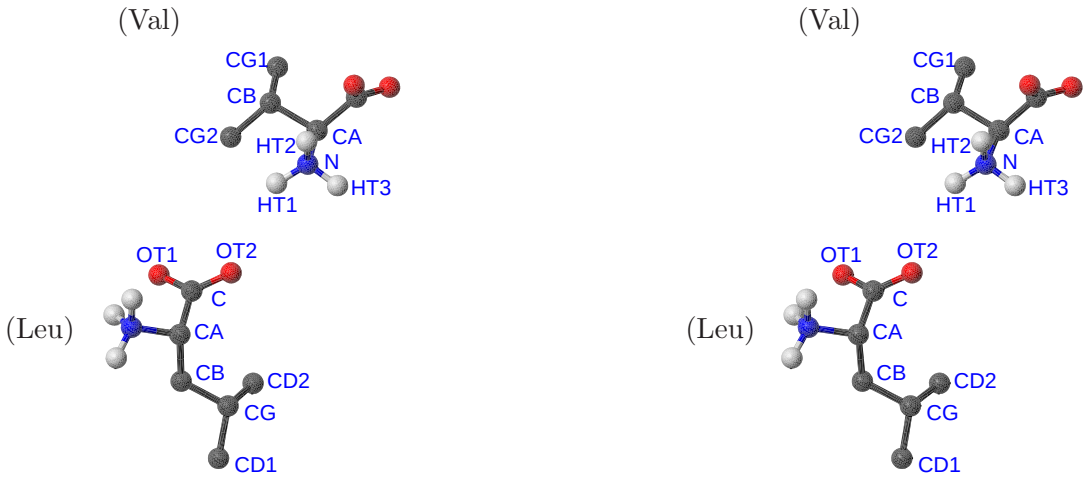


FIGURE A.3: Amino acids that have isomers that are related by an operation of methyl group swapping. Stereo ball-and-stick models of CHARMM19 representations of valine and leucine are shown. For the description of the atom colour-coding see the caption to Figure 2.10. Each molecule was optimised using CHARMM19, and has standard Nter and Cter capping groups blocking N- and C-termini, respectively. Atoms involved in the definition of asymmetric torsions are labelled with their respective CHARMM19 atom names.

rotation of the phenyl ring (which passes through atoms CG and CZ) does not coincide with the torsion axis (which passes through atoms CZ and OH). In this case the angles of the torsion CE1–CZ–OH–HH in the two rotamers are not complementary, which results in a slightly different energy. A similar situation is found in asparagine and glutamine where rotation of aminogroups is described by asymmetric CB–CG–ND2–HD21 and CG–CD–NE2–HE21 dihedral angles, respectively.

The same asymmetry is present in another tyrosine torsion, CA–CB–CG–CD1, as well as the CA–CB–CG–CD1 torsion in phenylalanine, the CB–CG–CD–OE1 torsion in glutamic acid, and the CA–CB–CG–OD1 torsion in aspartic acid, but they are not noticeable when a standard parameter file is used because the corresponding force constants are zero.

The improper torsion C–CA–OT2–OT1 is the origin of the energy difference for the Cter residue [see Figure A.1(Leu)]. In the general case it is difficult to symmetrise improper torsions in CHARMM19 without changing the form of the energy functional and/or reparametrisation. Fortunately, when the equilibrium angle is zero the improper torsion term is a symmetric function of the angle and, provided symmetry-related atoms

are given in 'correct' order, the improper torsion energy and its derivatives will be the same in both rotamers. Specifically, for improper torsion I–J–K–L, where I is the central atom and the torsion angle is the angle between the plane defined by atoms I–J–K and atom L, if the symmetry related atoms are J and K then ϕ(I–J–K–L)=$-\phi$(I–K–J–L). While tyrosine improper torsions CG–CD1–CD2–CB and CZ–CE1–CE2–OH, phenylalanine CG–CD1–CD2–CB improper torsion, the CD–OE1–OE2–CG improper torsion for glutamic acid, and the CG–OD1–OD2–CB improper torsion for aspartic acid are all symmetrically specified, the Cter's C–CA–OT2–OT1 improper torsion is not.

The valine and leucine situation is special because if one of the two structures with side chains that are mirror images of one another is a stationary point the other one is not. In other words, because the rearrangement where the –C–C(CH_3)$_2$ group undergoes an inversion is not describable by the force field, only one of the two isomers can exist on the CHARMM19 potential energy surface as a stationary point.

We have fixed the dihedral asymmetries in Asn, Gln, Tyr, Phe, Glu, Asp, Val and Leu by adding multiple dihedral terms between bonds with similar symmetry-related positions and halving the corresponding force constants, as was advised by Prof. Charles L. Brooks III. We also 'symmetrised' the Cter improper torsion by changing the topology for that residue. Patches for toph19.inp, param19.inp, toph19_eef1.inp and param19_eef1.inp are available online [165].

Appendix B

Algorithms

Here we adopt a standard notation (pseudocode) for defining algorithms [33, 154]. Assignment of a value α to a variable β is denoted by $\beta \leftarrow \alpha$. The branching probability $P_{\alpha,\beta}$ is represented as $P[\alpha, \beta]$ if it is stored in matrix form, and as $P[\beta][\alpha]$ if it is stored in the form of adjacency lists.

Algorithm B.1 Calculate the pathway sum $\mathcal{S}_{\alpha,\beta}^{C_N}$

Require: Chain nodes are numbered $0, 1, 2, \ldots, N-1$
Require: $1 < N$
Require: $\alpha, \beta \in \{0, 1, 2, \ldots, N-1\}$
Require: $P[i, j]$ is the probability of branching from node j to node i

1: **if** $\alpha < \beta$ **then**
2: $h \leftarrow 1$; $t \leftarrow N-2$; $s \leftarrow 1$
3: **else**
4: $h \leftarrow N-2$; $t \leftarrow 1$; $s \leftarrow -1$
5: **end if**
6: $L \leftarrow 1$
7: **for all** $i \in \{h, h+s, h+2s, \ldots, \alpha\}$ **do**
8: $L \leftarrow 1/(1 - P[i-s, i]P[i, i-s]L)$
9: **end for**
10: $\Pi \leftarrow 1$
11: **for all** $i \in \{\alpha+s, \alpha+2s, \alpha+3s, \ldots, \beta\}$ **do**
12: $\Pi \leftarrow P[i-s, i]L\Pi$
13: $L \leftarrow 1/(1 - P[i-s, i]P[i, i-s]L)$
14: **end for**
15: $R \leftarrow 1$
16: **for all** $i \in \{t, t-s, t-2s, \ldots, \beta\}$ **do**
17: $R \leftarrow 1/(1 - P[i+s, i]P[i, i+s]R)$
18: **end for**
19: **return** $LR\Pi/(L - LR + R)$

Algorithm B.2 Calculate the pathway sum $\mathcal{S}_{a,b}^{G_N}$

Require: $1 < N$
Require: $a, b \in \{0, 1, 2, \ldots, N-1\}$
Require: W is a boolean array of size N with every element initially set to True
Require: N_W is the number of True elements in array W (initialised to N)
Require: $P[i, j]$ is the probability of branching from node j to node i
Require: $AdjIn[i]$ and $AdjOut[i]$ are the lists of indices of all nodes connected to node i via incoming and outgoing edges, respectively
Recursive function $F(\alpha, \beta, W, N_W)$

$W[\beta] \leftarrow$ False
$N_W \leftarrow N_W - 1$
if $\alpha = \beta$ and $N_W = 0$ **then**
  $\Sigma \leftarrow 1$
else
  $\Sigma \leftarrow 0.0$
  for all $i \in AdjOut[\beta]$ **do**
    for all $j \in AdjIn[\beta]$ **do**
      if $W[i]$ and $W[j]$ **then**
        $\Sigma \leftarrow \Sigma + P[\beta, j]F(j, i, W, N_W)P[i, \beta]$
      end if
    end for
  end for
  $\Sigma \leftarrow 1/(1 - \Sigma)$
  if $\alpha \neq \beta$ **then**
    $\Lambda \leftarrow 0.0$
    for all $i \in AdjOut[\beta]$ **do**
      if W[i] **then**
        $\Lambda \leftarrow \Lambda + F(\alpha, i, W, N_W)P(i, \beta)$
      end if
    end for
    $\Sigma \leftarrow \Sigma\Lambda$
  end if
end if
$W[\beta] \leftarrow$ True
$N_W \leftarrow N_W + 1$
return Σ

Algorithm B.3 Calculate the pathway sum $\mathcal{S}_{\alpha,\beta}^{\mathcal{G}_N}$ from every source to every sink, and the mean escape time for every source in a dense graph $\mathcal{G}_N$

Require: Nodes are numbered $0, 1, 2, \ldots, N-1$
Require: Sink nodes are indexed first, source nodes - last
Require: i is the index of the first intermediate node
Require: s is the index of the first source node
Require: In case there are no intermediate nodes $i = s$, otherwise $i < s$
Require: $1 < N$
Require: $i, s \in \{0, 1, 2, \ldots, N-1\}$
Require: $\tau[\alpha]$ is the waiting time for node α, $\alpha \in \{i, i+1, i+2, \ldots, N-1\}$
Require: $P[i, j]$ is the probability of branching from node j to node i

for all $\gamma \in \{i, i+1, i+2, \ldots, s-1\}$ **do**
  for all $\beta \in \{\gamma+1, \gamma+2, \ldots, N-1\}$ **do**
    if $P[\gamma, \beta] > 0$ **then**
      $\tau[\beta] \leftarrow (\tau[\beta] + \tau[\gamma]P[\gamma, \beta])/(1 - P[\beta, \gamma]P[\gamma, \beta])$
      for all $\alpha \in \{0, 1, 2, \ldots, N-1\}$ **do**
        if $\alpha \neq \beta$ and $\alpha \neq \gamma$ **then**
          $P[\alpha, \beta] \leftarrow (P[\alpha, \beta] + P[\alpha, \gamma]P[\gamma, \beta])/(1 - P[\beta, \gamma]P[\gamma, \beta])$
        end if
      end for
      $P[\gamma, \beta] \leftarrow 0.0$
    end if
  end for
  for all $\alpha \in \{0, 1, 2, \ldots, N-1\}$ **do**
    $P[\alpha, \gamma] \leftarrow 0.0$
  end for
end for
for all $\alpha \in \{s, s+1, s+2, \ldots, N-1\}$ **do**
  for all $\beta \in \{s, s+1, s+2, \ldots, N-1\}$ **do**
    if $\alpha \neq \beta$ and $P[\alpha, \beta] > 0$ **then**
      $P_{\alpha,\beta} \leftarrow P[\alpha, \beta]$
      $P_{\beta,\alpha} \leftarrow P[\beta, \alpha]$
      $T \leftarrow \tau[\alpha]$
      $\tau[\alpha] \leftarrow (\tau[\alpha] + \tau[\beta]P_{\beta,\alpha})/(1 - P_{\alpha,\beta}P_{\beta,\alpha})$
      $\tau[\beta] \leftarrow (\tau[\beta] + TP_{\alpha,\beta})/(1 - P_{\alpha,\beta}P_{\beta,\alpha})$
      for all $\gamma \in \{0, 1, 2, \ldots, i-1\} \cup \{s, s+1, s+2, \ldots, N-1\}$ **do**
        $T \leftarrow P[\gamma, \alpha]$
        $P[\gamma, \alpha] \leftarrow (P[\gamma, \alpha] + P[\gamma, \beta]P_{\beta,\alpha})/(1 - P_{\alpha,\beta}P_{\beta,\alpha})$
        $P[\gamma, \beta] \leftarrow (P[\gamma, \beta] + TP_{\alpha,\beta})/(1 - P_{\alpha,\beta}P_{\beta,\alpha})$
      end for
      $P[\alpha, \beta] \leftarrow 0.0$
      $P[\beta, \alpha] \leftarrow 0.0$
    end if
  end for
end for

Algorithm B.4 Detach node γ from an arbitrary graph $\mathcal{G}_N$

Require: $1 < N$
Require: $\gamma \in \{0, 1, 2, \ldots, N-1\}$
Require: $\tau[i]$ is the waiting time for node i
Require: $Adj[i]$ is the ordered list of indices of all nodes connected to node i via outgoing edges
Require: $|Adj[i]|$ is the cardinality of $Adj[i]$
Require: $Adj[i][j]$ is the index of the jth neighbour of node i
Require: $P[i]$ is the ordered list of probabilities of leaving node i via outgoing edges, $|P[i]| = |Adj[i]|$
Require: $P[i][j]$ is the probability of branching from node i to node $Adj[i][j]$

for all $\beta_\gamma \in \{0, 1, 2, \ldots, |Adj[\gamma]| - 1\}$ **do**
  $\beta \leftarrow Adj[\gamma][\beta_\gamma]$
  $\gamma_\beta \leftarrow -1$
  for all $i \in \{0, 1, 2, \ldots, |Adj[\beta]| - 1\}$ **do**
    if $Adj[\beta][i] = \gamma$ **then**
      $\gamma_\beta \leftarrow i$
      break
    end if
  end for
  if not $\gamma_\beta = -1$ **then**
    $P_{\beta,\beta} \leftarrow 1/(1 - P[\beta][\gamma_\beta]P[\gamma][\beta_\gamma])$
    $P_{\beta,\gamma} \leftarrow P[\beta][\gamma_\beta]$
    $Adj[\beta] \leftarrow \{Adj[\beta][0], Adj[\beta][1], \ldots, Adj[\beta][\gamma_\beta - 1], Adj[\beta][\gamma_\beta + 1], \ldots\}$
    $P[\beta] \leftarrow \{P[\beta][0], P[\beta][1], \ldots, P[\beta][\gamma_\beta - 1], P[\beta][\gamma_\beta + 1], \ldots\}$
    for all $\alpha_\gamma \in \{0, 1, 2, \ldots, |Adj[\gamma]| - 1\}$ **do**
      $\alpha \leftarrow Adj[\gamma][\alpha_\gamma]$
      if not $\alpha = \beta$ **then**
        if exists edge $\beta \to \alpha$ **then**
          for all $i \in \{0, 1, 2, \ldots, |Adj[\beta]| - 1\}$ **do**
            if $Adj[\beta][i] = \alpha$ **then**
              $P[\beta][i] \leftarrow P[\beta][i] + P_{\beta,\gamma}P[\gamma][\alpha_\gamma]$
              break
            end if
          end for
        else
          $Adj[\beta] \leftarrow \{\alpha, Adj[\beta][0], Adj[\beta][1], Adj[\beta][2], \ldots\}$
          $P[\beta] \leftarrow \{P_{\beta,\gamma}P[\gamma][\alpha_\gamma], P[\beta][0], P[\beta][1], P[\beta][2], \ldots\}$
        end if
      end if
    end for
    for all $i \in \{0, 1, 2, \ldots, |P[\beta]| - 1\}$ **do**
      $P[\beta][i] \leftarrow P[\beta][i]P_{\beta,\beta}$
    end for
    $\tau_\beta \leftarrow (\tau_\beta + P_{\beta,\gamma}\tau_\gamma)\, P_{\beta,\beta}$
  end if
end for

Appendix C

Pathway Sums for Chain Graphs, $\mathcal{S}^{C_N}_{\alpha,\beta}$

To obtain the total probability of leaving the chain C_N via node α if started from node β, i.e. $\mathcal{E}^{C_N}_{\alpha}\mathcal{S}^{C_N}_{\alpha,\beta}$, we must calculate the pathway sum $\mathcal{S}^{C_N}_{\alpha,\beta}$. We start with the case $\alpha = \beta$ and obtain $\mathcal{S}^{C_N}_{\beta,\beta}$. Consider any path that has reached node $N-1$. The probability factor due to all possible $N-1$ to N recrossings is simply $R_{N-1} = 1/(1 - P_{N-1,N}P_{N,N-1})$. We need to include this factor every time we reach node $N-1$ during recrossings of $N-2$ to $N-1$. The corresponding sum becomes

$$R_{N-2} = \sum_{m=0}^{\infty} (P_{N-2,N-1}P_{N-1,N-2}R_{N-1})^m = \frac{1}{1 - P_{N-2,N-1}P_{N-1,N-2}R_{N-1}}. \quad \text{(C.1)}$$

Similarly, we can continue summing contributions in this way until we have recrossings of β to $\beta+1$, for which the result of the nested summations is $R_\beta = 1/(1 - P_{\beta,\beta+1}P_{\beta+1,\beta}R_{\beta+1})$. Hence, R_β is the total transition probability for pathways that return to node β and are confined to nodes with index greater than β without escape from C_N.

We can similarly calculate the total probability for pathways returning to β and confined to nodes with indices smaller than β. The total probability factor for recrossings between nodes 1 and 2 is $L_2 = 1/(1 - P_{1,2}P_{2,1})$. Hence, the required probability for recrossings between nodes 2 and 3 including arbitrary recrossings between 1 and 2 is $L_3 = 1/(1 - P_{2,3}P_{3,2}L_2)$. Continuing up to recrossings between nodes $\beta-1$ and β we obtain the total return probability for pathways restricted to this side of β as

$L_\beta = 1/(1 - P_{\beta-1,\beta}P_{\beta,\beta-1}L_{\beta-1})$. The general recursive definitions of L_j and R_j are:

$$\begin{aligned} L_j &= \begin{cases} 1, & j = 1, \\ 1/(1 - P_{j-1,j}P_{j,j-1}L_{j-1}), & j > 1, \end{cases} \\ \text{and} \quad R_j &= \begin{cases} 1, & j = N, \\ 1/(1 - P_{j,j+1}P_{j+1,j}R_{j+1}), & j < N. \end{cases} \end{aligned} \tag{C.2}$$

We can now calculate $\mathcal{S}^{C_N}_{\beta,\beta}$ as

$$\begin{aligned} \mathcal{S}^{C_N}_{\beta,\beta} &= \sum_{m=0}^{\infty}\sum_{n=0}^{m} \frac{n!}{m!(n-m)!} \left(P_{\beta-1,\beta}P_{\beta,\beta-1}L_{\beta-1}\right)^n \left(P_{\beta,\beta+1}P_{\beta+1,\beta}R_{\beta+1}\right)^{m-n} \\ &= \sum_{m=0}^{\infty} (P_{\beta-1,\beta}P_{\beta,\beta-1}L_{\beta-1} + P_{\beta,\beta+1}P_{\beta+1,\beta}R_{\beta+1})^m \\ &= (1 - P_{\beta-1,\beta}P_{\beta,\beta-1}L_{\beta-1} - P_{\beta,\beta+1}P_{\beta+1,\beta}R_{\beta+1})^{-1} \\ &= \left(1 - \frac{L_\beta - 1}{L_\beta} - \frac{R_\beta - 1}{R_\beta}\right)^{-1} \\ &= \frac{L_\beta R_\beta}{L_\beta - L_\beta R_\beta + R_\beta}, \end{aligned} \tag{C.3}$$

where we have used Equation C.2 and the multinomial theorem [242].

We can now derive $\mathcal{S}^{C_N}_{\alpha,\beta}$ as follows. If $\alpha > \beta$ we can write

$$\mathcal{S}^{C_N}_{\alpha,\beta} = \mathcal{S}^{C_N}_{\alpha-1,\beta}P_{\alpha,\alpha-1}R_\alpha. \tag{C.4}$$

$\mathcal{S}^{C_N}_{\alpha-1,\beta}$ gives the total transition probability from β to $\alpha - 1$, so the corresponding probability for node α is $\mathcal{S}^{C_N}_{\alpha-1,\beta}$ times the branching probability from $\alpha - 1$ to α, i.e. $P_{\alpha,\alpha-1}$, times R_α, which accounts for the weight accumulated from all possible paths that leave and return to node α and are restricted to nodes with indexes greater than α. We can now replace $\mathcal{S}^{C_N}_{\alpha-1,\beta}$ by $\mathcal{S}^{C_N}_{\alpha-2,\beta}P_{\alpha-1,\alpha-2}R_{\alpha-1}$ and so on, until $\mathcal{S}^{C_N}_{\alpha,\beta}$ is expressed in terms of $\mathcal{S}^{C_N}_{\beta,\beta}$. Similarly, if $\alpha < \beta$ we have

$$\mathcal{S}^{C_N}_{\alpha,\beta} = \mathcal{S}^{C_N}_{\alpha+1,\beta}P_{\alpha,\alpha+1}L_\alpha, \tag{C.5}$$

and hence

$$\mathcal{S}^{C_N}_{\alpha,\beta} = \begin{cases} \mathcal{S}^{C_N}_{\beta,\beta} \prod_{i=\alpha}^{\beta-1} P_{i,i+1}L_i, & \alpha < \beta, \\ \mathcal{S}^{C_N}_{\beta,\beta} \prod_{i=\beta+1}^{\alpha} P_{i,i-1}R_i, & \alpha > \beta. \end{cases} \tag{C.6}$$

Appendix D

Total Escape Probability for Chain Graphs, $\Sigma_\beta^{C_N}$

So long as there is at least one escape route from C_N the total escape probability must be unity:

$$\Sigma_\beta^{C_N} = \sum_{j=1}^{N} \mathcal{E}_j^{C_N} \mathcal{S}_{j,\beta}^{C_N} = 1, \tag{D.1}$$

otherwise, if $Adj[C_N]$ is the empty set, we have

$$\Sigma_\beta^{C_N} = \sum_{j=1}^{N} \mathcal{S}_{j,\beta}^{C_N} = 1. \tag{D.2}$$

For example, to show that the formulae in Appendix C are consistent with the first result we expand

$$\mathcal{E}_j^{C_N} = 1 - P_{j-1,j} - P_{j+1,j} \tag{D.3}$$

and define

$$P_{0,1} = P_{N+1,N} = 0 \tag{D.4}$$

for convenience, so that

$$\begin{aligned}
\Sigma_\beta^{C_N} &= \mathcal{S}_{1,\beta}^{C_N} - P_{0,1}\mathcal{S}_{1,\beta}^{C_N} - P_{2,1}\mathcal{S}_{1,\beta}^{C_N} \\
&+ \mathcal{S}_{2,\beta}^{C_N} - P_{1,2}\mathcal{S}_{2,\beta}^{C_N} - P_{3,2}\mathcal{S}_{2,\beta}^{C_N} \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
&+ \mathcal{S}_{N-1,\beta}^{C_N} - P_{N-2,N-1}\mathcal{S}_{N-1,\beta}^{C_N} - P_{N,N-1}\mathcal{S}_{N-1,\beta}^{C_N} \\
&+ \mathcal{S}_{N,\beta}^{C_N} - P_{N-1,N}\mathcal{S}_{N,\beta}^{C_N} - P_{N+1,N}\mathcal{S}_{N,\beta}^{C_N}.
\end{aligned} \tag{D.5}$$

Using the recursion relations in Equation C.2 (which assume that there is an escape route from C_N) we can show that

$$\mathcal{S}^{C_N}_{j,\beta} - \mathcal{S}^{C_N}_{j-1,\beta} P_{j,j-1} - \mathcal{S}^{C_N}_{j+1,\beta} P_{j,j+1} = 0, \tag{D.6}$$

for $j \neq \beta$. We can now group together terms in Equation D.5 into sets of three that sum to zero. The terms that do not immediately cancel are as follows. From the first and second lines of Equation D.5 we have

$$\mathcal{S}^{C_N}_{1,\beta} - P_{2,1}\mathcal{S}^{C_N}_{2,\beta} = 0 \tag{D.7}$$

because

$$\mathcal{S}^{C_N}_{1,\beta} = \mathcal{S}^{C_N}_{2,\beta} P_{2,1} L_1 = \mathcal{S}^{C_N}_{2,\beta} P_{2,1}. \tag{D.8}$$

Similarly, on the last two lines we find

$$\mathcal{S}^{C_N}_{N,\beta} - P_{N,N-1}\mathcal{S}^{C_N}_{N-1,\beta} = 0 \tag{D.9}$$

because

$$\mathcal{S}^{C_N}_{N,\beta} = P_{N,N-1}\mathcal{S}^{C_N}_{N-1,\beta} R_N = P_{N,N-1}\mathcal{S}^{C_N}_{N-1,\beta}. \tag{D.10}$$

The final remaining terms are:

$$\begin{aligned}
& \mathcal{S}^{C_N}_{\beta,\beta} - \mathcal{S}^{C_N}_{\beta-1,\beta} P_{\beta,\beta-1} - \mathcal{S}^{C_N}_{\beta+1,\beta} P_{\beta,\beta+1} \\
= \;& \mathcal{S}^{C_N}_{\beta,\beta} - \mathcal{S}^{C_N}_{\beta,\beta}\left(1 - \frac{1}{L_\beta}\right) - \mathcal{S}^{C_N}_{\beta,\beta}\left(1 - \frac{1}{R_\beta}\right) \\
= \;& \mathcal{S}^{C_N}_{\beta,\beta}\left(\frac{1}{L_\beta} + \frac{1}{R_\beta} - 1\right) \\
= \;& \mathcal{S}^{C_N}_{\beta,\beta}\left(\frac{L_\beta + R_\beta - L_\beta R_\beta}{L_\beta R_\beta}\right) \\
= \;& 1,
\end{aligned} \tag{D.11}$$

which proves Equation D.1.

Appendix E

Finding the Shortest and the Fastest Paths

> There are many paths to the top of the mountain,
> but the view is always the same.
>
> Chinese proverb

Techniques for finding the shortest and the fastest pathways can be useful during sampling for pathways, in post-processing and analysis of DPS databases, and in various optimisations of pathway ensembles [8, 10, 192, 290, 291]. Given a digraph representation of a connected database of minima and transition states it is possible to identify the shortest path between any two minima using a breadth-first search algorithm [154], which runs in linear time. The shortest path, however, is not necessarily the fastest. Recently Evans suggested using Dijkstra's algorithm [153] for finding the path with the largest non-recrossing DPS rate [290]. It was pointed out by Carr, however, that the weight function used in Reference [290] is not positively defined, and the use of the Bellman-Ford-Moore (BFM) algorithm [292–294] was suggested instead [291]. Because Dijkstra's algorithm scales better than BFM algorithm (see Figure E.1) we describe below a weight function that enables us to use the Dijkstra's algorithm for finding the fastest path.

As detailed in Section 4.1.1, within a digraph representation of a coarse-grained PES each minimum is represented by a single node and each transition state is represented by two oppositely directed edges. Here we ignore degenerate rearrangements since they

do not affect the rates, and for cases where more than one transition state links the same pair of minima only the one with the fastest forward and backward rates is used. Adopting the non-recrossing DPS rate definition from Equation 4.11 we can associate forward and backward rate constants

$$\begin{aligned} k_{a,b} &= \frac{P_b^{\mathrm{eq}}}{P_B^{\mathrm{eq}}} \frac{k_{a,i_1}\, k_{i_1,i_2} \cdots k_{i_n,b}}{\sum_{\gamma_1} k_{\gamma_1,i_1} \sum_{\gamma_2} k_{\gamma_2,i_2} \cdots \sum_{\gamma_n} k_{\gamma_n,i_n}}, \\ k_{b,a} &= \frac{P_a^{\mathrm{eq}}}{P_A^{\mathrm{eq}}} \frac{k_{b,i_1}\, k_{i_1,i_2} \cdots k_{i_n,a}}{\sum_{\gamma_1} k_{\gamma_1,i_1} \sum_{\gamma_2} k_{\gamma_2,i_2} \cdots \sum_{\gamma_n} k_{\gamma_n,i_n}} \end{aligned} \tag{E.1}$$

with a pathway $\xi = a, i_1, i_2, \ldots, i_n, b$.

For a detailed description of Dijkstra and BFM algorithms we refer the reader to Reference [154]. A suitable choice of edge weight function to be used with either of the shortest path algorithms must be made. We define the weight of each directed edge $\alpha \rightarrow \beta$ as

$$w(\beta, \alpha) = \ln\left(\frac{\sum_\gamma k_{\gamma,\alpha}}{k_{\beta,\alpha}}\right). \tag{E.2}$$

Note that $w(\beta, \alpha) \neq w(\alpha, \beta)$ and $0 \leqslant w < \infty$. Since the weight is non-negative it is possible to apply Dijkstra's algorithm to find the fastest pathway [154].

It makes sense to include only minima with more than one connection when searching for the fastest pathway. In our calculations we first iteratively identified a connected subset of minima with the number of connections greater than one, and then run the above algorithm for this subset.

To find the fastest path connecting minima a and b in the direction $b \rightarrow a$ it is necessary to solve either a single-source shortest paths problem with minimum b chosen as a source, or a single destination shortest paths problem with minimum a chosen as the destination. In the latter case all the edges in the graph need to be reversed, which can be accomplished by simply swapping the arguments to the weight function in Equation E.2. This reversal is possible because our graph is a symmetric digraph, i.e. if there is an edge $a \rightarrow b$ then there is also an edge $b \rightarrow a$.

A useful check for correct implementation of this algorithm is to verify that the weight of the shortest path from minimum b to minimum a, $W(a, b)$, is related to the

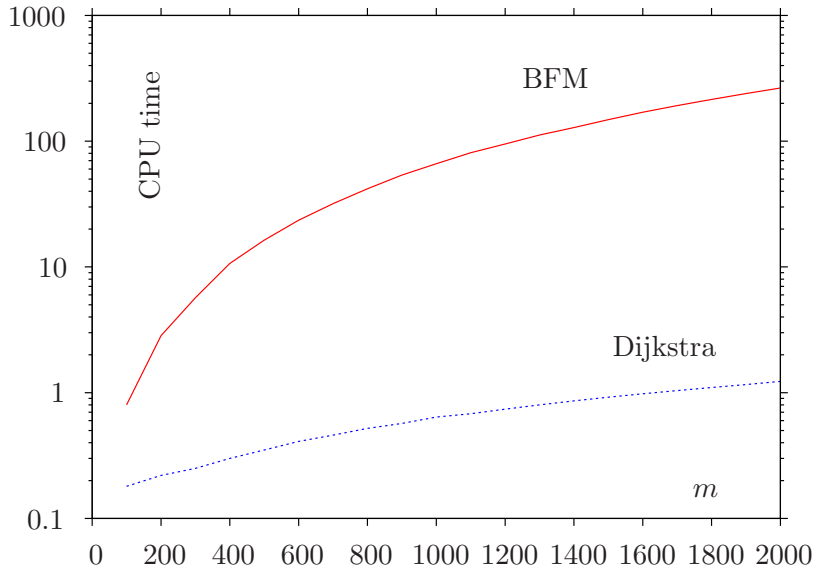


FIGURE E.1: Performance comparison of BFM and static Dijkstra algorithms for an evolving database of minima and transition states for tryptophan zipper 2. CPU time as a function of the number of minima in the database, m, is shown. The number of transition states, t, scales linearly with m, and for the plotted data set the dependence can be approximated as $t = 1.59m - 46.98$ with a correlation coefficient of 1.000. The units of time are arbitrary.

rate $k_{a,b}$ calculated for that pathway by

$$k_{a,b} = \frac{P_b^{\mathrm{eq}}}{P_B^{\mathrm{eq}}} \exp\Big(-W\left(a,b\right)\Big) \sum_{\gamma} k_{\gamma,b}. \tag{E.3}$$

The BFM algorithm runs in $\mathcal{O}(|V|\,|E|)$ time while Dijkstra's algorithm as described in Reference [153] scales as $\mathcal{O}(|V|^2 + |E|) = \mathcal{O}(|V|^2)$ [154]. Although both algorithms appear to have similar (quadratic) asymptotic complexity for the case of sparse graphs (e.g. $|E| = \mathcal{O}(|V|)$) the scaling prefactor is smaller for Dijkstra's algorithm. In addition, for sparse graphs with $|E| = o(|V|^2/\lg|V|)$ an implementation of Dijkstra's algorithm that utilises a priority queue of some sort can improve the asymptotic bounds further. Priority queues based on binary min-heap and Fibonacci heap [276] result in $\mathcal{O}((|E|+|V|)\lg|V|) = \mathcal{O}(|E|\lg|V|)$ and $\mathcal{O}(|V|\lg|V|+|E|)$ scaling of the algorithm, respectively [154]. In the present work we have used a priority queue based on Fibonacci heap. A performance comparison of our implementations of BFM and static Dijkstra's algorithms is shown in Figure E.1.

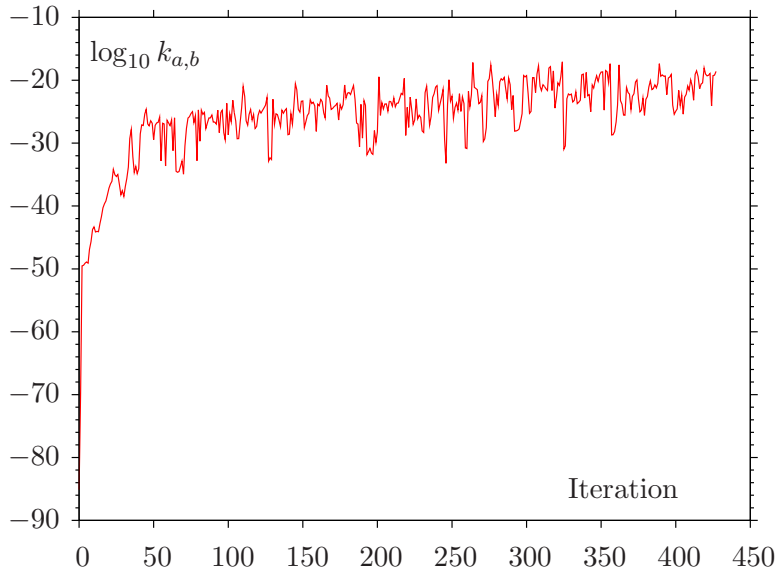


FIGURE E.2: The rate of the fastest pathway, $k_{a,b}$, as a function of the iteration number for the evolving database of tryptophan zipper 3-I. Note $\log_{10}$ scale on vertical axis. As the generation of new connections can affect the rates of the pathways explicitly recorded in the pathway database (see Equation E.1), these rates were recomputed at the end of the calculation. This result, and the fact that $k_{a,b}$ includes recrossing contributions, explains the wiggles in the data. The recrossing contribution was calculated using Algorithm B.1, which is discussed in Chapter 4.

Since knowledge of all the pathways on a complicated PES is out of the question a representative sample must be obtained to properly describe the property of interest. It is unclear what is the best strategy of evolving the pathway database if the proper description of kinetics is sought, as sampling techniques may introduce a bias towards pathways of particular type.

In previous DPS studies a set of perturbations was applied to the fastest known path to generate new ones [8–10]. New pathways were constructed from a subset of stationary points featured in the pathway being perturbed, and from these that were found as a result of perturbation. Because building the shortest paths tree with Dijkstra's algorithm is relatively cheap it is possible to analyse the whole database for the presence of new faster pathways rather than just the subset. Results for an example application are presented in Figure E.2. After a set of perturbations was applied to the fastest currently known pathway $a \leftarrow b$ Dijkstra's algorithm was used to solve the

single-source shortest paths problem to build the shortest paths tree rooted at minimum b. For path ξ the set of perturbations constituted $l(\xi) - \Delta + 1$ attempts to shortcut the path, where Δ is a shortcut parameter, $\Delta \in \{1, 2, \ldots, l(\xi)\}$.

Each attempt to shortcut the path was realised as a single connect run (see Chapter 2) seeded with two minima $\alpha, \beta \in \xi$ separated by a gap of length Δ along the path. If the fastest path $a \leftarrow b$ found by Dijkstra's algorithm was different (not necessarily new) from the one that was currently being perturbed the iteration was completed, this pathway was perturbed in the next iteration and parameter Δ was reset to 1. Otherwise, Δ was incremented by one, unless $\Delta = l(\xi)$ in which case the calculation was terminated.

References

[1] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp. **35**, 773 (1980).

[2] D. C. Liu and J. Nocedal, *On limited memory BFGS method for large scale optimization*, Math. Prog. **45**, 503 (1989).

[3] J. Nocedal, *Software for large-scale unconstrained optimization*, http://www.ece.northwestern.edu/∼nocedal/lbfgs.html (2000).

[4] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan and M. Karplus, *CHARMM: A program for macromolecular energy, minimization, and dynamics calculations*, J. Comp. Chem. **4**, 187 (1983).

[5] E. Neria, S. Fischer and M. Karplus, *Simulation of activation free energies in molecular systems*, J. Chem. Phys. **105**, 1902 (1996).

[6] A. D. Mackerell Jr., D. Bashford, M. Bellott, R. L. D. Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin and M. Karplus, *All-atom empirical potential for molecular modelling and dynamics studies of proteins*, J. Phys. Chem. B **102**, 3586 (1998).

[7] S. A. Trygubenko and D. J. Wales, *A doubly nudged elastic band method for finding transition states*, J. Chem. Phys. **120**, 2082 (2004).

[8] D. J. Wales, *Discrete path sampling*, Mol. Phys. **100**, 3285 (2002).

[9] D. J. Wales, *Energy landscapes: Applications to clusters, biomolecules and glasses*, Cambridge University Press, Cambridge (2003).

[10] D. J. Wales, *Some further applications of discrete path sampling to cluster isomerization*, Mol. Phys. **102**, 883 (2004).

[11] T. Lazaridis and M. Karplus, *Effective energy function for protein dynamics and thermodynamics*, Proteins: Struct., Func. and Gen. **35**, 133 (1999).

[12] G. M. Crippen and H. A. Scheraga, *Minimization of polypeptide energy. X. A global search algorithm*, Arch. Biochem. Biophys. **144**, 453 (1971).

[13] J. Pancíř, *Calculation of the least energy path on the energy hypersurface*, Coll. Czech. Chem. Comm. **40**, 1112 (1974).

[14] R. L. Hilderbrandt, *Application of Newton-Raphson optimization techniques in molecular mechanics calculations*, Comput. Chem **1**, 179 (1977).

[15] C. J. Cerjan and W. H. Miller, *On finding transition states*, J. Chem. Phys. **75**, 2800 (1981).

[16] J. Simons, P. Jørgenson, H. Taylor and J. Ozment, *Walking on potential energy surfaces*, J. Phys. Chem. **87**, 2745 (1983).

[17] A. Banerjee, N. Adams, J. Simons and R. Shepard, *Search for stationary points on surfaces*, J. Phys. Chem. **89**, 52 (1985).

[18] J. Baker, *An algorithm for the location of transition states*, J. Comp. Chem. **7**, 385 (1986).

[19] D. J. Wales, *Rearrangements of 55-atom Lennard-Jones and $(C_{60})_{55}$ clusters*, J. Chem. Phys. **101**, 3750 (1994).

[20] D. J. Wales and J. Uppenbrink, *Rearrangements in model face-centred-cubic solids*, Phys. Rev. B **50**, 12342 (1994).

[21] L. J. Munro and D. J. Wales, *Rearrangements of bulk face-centred cubic nickel modelled by a Sutton-Chen potential*, Faraday Discuss. **106**, 409 (1997).

[22] L. J. Munro and D. J. Wales, *Defect migration in crystalline silicon*, Phys. Rev. B **59**, 3969 (1999).

[23] Y. Kumeda, L. J. Munro and D. J. Wales, *Transition states and rearrangement mechanisms from hybrid eigenvector-following and density functional theory. Application to $C_{10}H_{10}$ and defect migration in crystalline silicon*, Chem. Phys. Lett. **341**, 185 (2001).

[24] D. J. Wales and T. R. Walsh, *Theoretical study of the water pentamer*, J. Chem. Phys. **105**, 6957 (1996).

[25] K. Müller and L. D. Brown, *Location of saddle points and minimum energy paths by a constrained simplex optimization procedure*, Theor. Chim. Acta **53**, 75 (1979).

[26] M. A. Novotny, *Monte Carlo algorithms with absorbing Markov chains: Fast local algorithms for slow dynamics*, Phys. Rev. Lett. **74**, 1 (1995).

[27] H. Jónsson, G. Mills and W. Jacobsen, *Nudged elastic band method for finding minimum energy paths of transitions*, in *Classical and quantum dynamics in condensed phase simulations*, edited by B. J. Berne, G. Ciccotti and D. F. Coker, p. 385, World Scientific (1998).

[28] G. Henkelman and H. Jónsson, *Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points*, J. Chem. Phys. **113**, 9978 (2000).

[29] G. Henkelman, B. P. Uberuaga and H. Jónsson, *A climbing image nudged elastic band method for finding saddle points*, J. Chem. Phys. **113**, 9901 (2000).

[30] G. Henkelman, G. Johannesson and H. Jónsson, *Methods for finding saddle points and minimum energy paths*, in *Progress in theoretical chemistry and physics*, edited by S. D. Schwartz, p. 269, Kluwar Academic Publishers, Dordrecht (2000).

[31] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical recipes in Fortran: The art of scientific computing*, Cambridge University Press, Cambridge (1992).

[32] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*, Clarendon Press, Oxford (1989).

[33] *Wikipedia, the free encyclopedia*, http://www.wikipedia.org/ (2006).

[34] J. N. Murrell, S. Carter, S. C. Farantos, P. Huxley and A. J. C. Varandas, *Molecular potential energy functions*, Wiley, Chichester (1984).

[35] J. N. Murrell, *The many-body expansion of the potential energy function for elemental clusters*, Int. J. Quant. Chem. **37**, 95 (1990).

[36] R. L. Johnston, *Atomic and molecular clusters*, Taylor & Francis, New York (2002).

[37] H. B. Schlegel, *Exploring potential energy surfaces for chemical reactions: An overview of some practical methods*, J. Comp. Chem. **24**, 1514 (2003).

[38] J. N. Murrell and K. J. Laidler, *Symmetries of activated complexes*, Trans. Faraday Soc. **64**, 371 (1968).

[39] D. J. Wales and J. P. K. Doye, *Stationary points and dynamics in high-dimensional systems*, J. Chem. Phys. **119**, 12409 (2003).

[40] J. P. K. Doye and D. J. Wales, *Saddle points and dynamics of Lennard-Jones clusters, solids, and supercooled liquids*, J. Chem. Phys. **116**, 3777 (2002).

[41] F. H. Stillinger, *Exponential multiplicity of inherent structures*, Phys. Rev. E **59**, 48 (1999).

[42] F. H. Stillinger and T. A. Weber, *Hidden structures in liquids*, Phys. Rev. A **25**, 978 (1982).

[43] F. H. Stillinger and T. A. Weber, *Packing structures and transitions in liquids and solids*, Science **225**, 983 (1984).

[44] H. Pelzer and E. Wigner, *The speed constants of the exchange reactions*, Z. Phys. Chem. **B15**, 445 (1932).

[45] H. Eyring, *The activated complex and the absolute rate of chemical reactions*, Chem. Rev. **17**, 65 (1935).

[46] M. G. Evans and M. Polanyi, *Some applications of the transition state method to the calculation of reaction velocities, especially in solution*, Trans. Faraday Soc. **31**, 875 (1935).

[47] W. Forst, *Theory of unimolecular reactions*, Academic Press, New York (1973).

[48] K. J. Laidler, *Chemical kinetics*, Harper & Row, New York (1987).

[49] P. Pulay, G. Fogorasi, F. Pang and J. E. Boggs, *Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives*, J. Am. Chem. Soc. **101**, 2550 (1979).

[50] G. Fogorasi, X. Zhou, P. W. Taylor and P. Pulay, *The calculation of ab initio molecular geometries: efficient optimization by natural internal coordinates and empirical correction by offset forces*, J. Am. Chem. Soc. **114**, 8191 (1992).

[51] P. Pulay and G. Fogorasi, *Geometry optimization in redundant internal coordinates*, J. Chem. Phys. **96**, 2856 (1996).

[52] J. Baker, A. Kessi and B. Delley, *The generation and use of delocalized internal coordinates in geometry optimization*, J. Chem. Phys. **105**, 192 (1996).

[53] C. Peng, P. Y. Ayala, H. B. Schlegel and M. J. Frisch, *Using redundant internal coordinates to optimize equilibrium geometries and transition states*, J. Comp. Chem. **17**, 49 (1996).

[54] J. Baker, D. Kinghorn and P. Pulay, *Geometry optimization in delocalized internal coordinates: An efficient quadratically scaling algorithm for large molecules*, J. Chem. Phys. **110**, 4986 (1999).

[55] S. R. Billeter, A. J. Turner and W. Thiel, *Linear scaling geometry optimisation and transition state search in hybrid delocalised internal coordinates*, Phys. Chem. Chem. Phys. **2**, 2177 (2000).

[56] B. Paizs, J. Baker, S. Suhai and P. Pulay, *Geometry optimization of large biomolecules in redundant internal coordinates*, J. Chem. Phys. **113**, 6566 (2000).

[57] V. Bakken and T. Helgaker, *The efficient optimization of molecular geometries using redundant internal coordinates*, J. Chem. Phys. **117**, 9160 (2002).

[58] J. W. McIver and A. Komornicki, *Structure of transition states in organic reactions. General theory and an application to the cyclobutene-butadiene isomerization using a semiempirical molecular orbital method*, J. Am. Chem. Soc. **94**, 2625 (1972).

[59] D. O'Neal, H. Taylor and J. Simons, *Potential surface walking and reaction paths for* C_{2v} $Be + H_2 \leftarrow BeH_2 \rightarrow Be + 2H$ ($^1A_1$), J. Phys. Chem. **88**, 1510 (1984).

[60] S. Bell and J. S. Crighton, *Locating transition states*, J. Chem. Phys. **80**, 2464 (1984).

[61] D. T. Nguyen and D. A. Case, *On finding stationary states on large-molecule potential energy surfaces*, J. Phys. Chem. **89**, 4020 (1985).

[62] J. Baker, *An algorithm for geometry optimization without analytical gradients*, J. Comp. Chem. **8**, 563 (1987).

[63] M. C. Smith, *How to find a saddle point*, Int. J. Quant. Chem. **37**, 773 (1990).

[64] J. Nichols, H. Taylor, P. Schmidt and J. Simons, *Walking on potential-energy surfaces*, J. Chem. Phys. **92**, 340 (1990).

[65] J. Baker and W. H. Hehre, *Geometry optimization in Cartesian coordinates: the end of the Z-matrix?*, J. Comp. Chem. **12**, 606 (1991).

[66] J. Baker, *Geometry optimization in Cartesian coordinates: constrained optimization*, J. Comp. Chem. **13**, 240 (1992).

[67] C. J. Tsai and K. D. Jordan, *Use of an eigenmode method to locate the stationary points on the potential energy surfaces of selected argon and water clusters*, J. Phys. Chem. **97**, 11227 (1993).

[68] D. J. Wales, *Locating stationary points for clusters in Cartesian coordinates*, J. Chem. Soc., Faraday Trans. **89**, 1305 (1993).

[69] S. F. Chekmarev, *A simple gradient method for locating saddles*, Chem. Phys. Lett. **227**, 354 (1994).

[70] J.-Q. Sun and K. Ruedenberg, *Locating transition states by quadratic image gradient descent on potential energy surfaces*, J. Chem. Phys. **101**, 2157 (1994).

[71] J.-Q. Sun and K. Ruedenberg, *A simple prediction of approximate transition states on potential energy surfaces*, J. Chem. Phys. **101**, 2168 (1994).

[72] F. Jensen, *Locating transition structures by mode following: a comparison of six methods on the Ar_8 Lennard-Jones potential*, J. Chem. Phys. **102**, 6706 (1995).

[73] J. M. Bofill and M. Comajuan, *Analysis of the updated Hessian matrices for locating transition structures*, J. Comp. Chem. **16**, 1326 (1995).

[74] Q. Zhao and J. B. Nicholas, *Transition state optimization using the divide-and-conquer method: Reaction of trans-2-butene with HF*, J. Chem. Phys. **104**, 767 (1996).

[75] W. Quapp, *A gradient-only algorithm for tracing a reaction path uphill to the saddle of a potential energy surface*, Chem. Phys. Lett. **253**, 286 (1996).

[76] J. Baker and F. Chan, *The location of transition states: a comparison of Cartesian, Z-matrix and natural internal coordinates*, J. Comp. Chem. **17**, 888 (1996).

[77] T. R. Walsh and D. J. Wales, *Rearrangements of the water trimer*, J. Chem. Soc., Faraday Trans. **92**, 2505 (1996).

[78] P. Y. Ayala and H. B. Schlegel, *A combined method for determining reaction paths, minima, and transition state geometries*, J. Chem. Phys. **107**, 375 (1997).

[79] A. Ulitsky and D. Shalloway, *Finding transition states using contangency curves*, J. Chem. Phys. **106**, 10099 (1997).

[80] B. Paizs, G. Fogarasi and P. Pulay, *An efficient direct method for geometry optimization of large molecules in internal coordinates*, J. Chem. Phys. **109**, 6571 (1998).

[81] O. Farkas and H. B. Schlegel, *Methods for geometry optimization of large molecules. I. An O(N-2) algorithm for solving systems of linear equations for the transformation of coordinates and forces*, J. Chem. Phys. **109**, 7100 (1998).

[82] N. Mousseau and G. T. Barkema, *Travelling through potential energy landscapes of disordered materials: The activation-relaxation technique*, Phys. Rev. E **57**, 2419 (1998).

[83] H. Goto, *A frontier mode-following method for mapping saddle points of conformational interconversion in flexible molecules starting from the energy minimum*, Chem. Phys. Lett. **292**, 254 (1998).

[84] W. Quapp, M. Hirsch, O. Imig and D. Heidrich, *Searching for saddle points of potential energy surfaces by following a reduced gradient*, J. Comp. Chem. **19**, 1087 (1998).

[85] O. Farkas and H. B. Schlegel, *Methods for optimizing large molecules. II. Quadratic search*, J. Chem. Phys. **111**, 10806 (1999).

[86] D. J. Wales, J. P. K. Doye, M. A. Miller, P. N. Mortenson and T. R. Walsh, *Energy landscapes: From clusters to biomolecules*, Adv. Chem. Phys. **115**, 1 (2000).

[87] L. R. Pratt, *A statistical method for identifying transition states in high dimensional problems*, J. Chem. Phys. **85**, 5045 (1986).

[88] R. Elber and M. Karplus, *A method for determining reaction paths in large molecules: Application to myoglobin*, Chem. Phys. Lett. **139**, 375 (1987).

[89] R. S. Berry, H. L. Davis and T. L. Beck, *Finding saddles on multidimensional potential surfaces*, Chem. Phys. Lett. **147**, 13 (1988).

[90] R. Czerminski and R. Elber, *Reaction path study of conformational transitions in flexible systems*, J. Chem. Phys. **92**, 5580 (1990).

[91] S. Fischer and M. Karplus, *Conjugate peak refinement — an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom*, Chem. Phys. Lett. **194**, 252 (1992).

[92] L. L. Stachó and M. I. Bán, *A global strategy for determining reaction paths. 1. General theory of a procedure finding Fukui intrinsic reaction coordinate*, Theor. Chim. Acta **83**, 433 (1992).

[93] I. V. Ionova and E. A. Carter, *Ridge method for finding saddle points on potential energy surfaces*, J. Chem. Phys. **98**, 6377 (1993).

[94] L. L. Stachó and M. I. Bán, *An algorithm for determining dynamically defined reaction paths (DDRP)*, Theor. Chim. Acta **84**, 535 (1993).

[95] G. Dömötör, M. I. Bán and L. L. Stachó, *Experiences and practical hints on using the DDRP method, illustrated by the example of the H(2)+H reaction*, J. Comput. Chem. **14**, 1491 (1993).

[96] C. Y. Peng and H. B. Schlegel, *Combining synchronous transit and quasi-Newton methods to find transition states*, Israeli J. Chem. **33**, 449 (1993).

[97] A. Matro, D. L. Freeman and J. D. Doll, *Locating transition states using double-ended classical trajectories*, J. Chem. Phys. **101**, 10458 (1994).

[98] O. S. Smart, *A new method to calculate reaction paths for conformational transitions of large molecules*, Chem. Phys. Lett. **222**, 503 (1994).

[99] M. I. Bán, G. Dömötör and L. L. Stachó, *Dynamically defined reaction-path (DDRP) method*, J. Mol. Struct. (Theochem) **311**, 29 (1994).

[100] G. Mills and H. Jónsson, *Quantum and thermal effects in H2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems*, Phys. Rev. Lett. **72**, 1124 (1994).

[101] I. V. Ionova and E. A. Carter, *Direct inversion in the iterative subspace-induced acceleration of the ridge method for finding transition states*, J. Chem. Phys. **103**, 5437 (1995).

[102] L. L. Stachó, G. Dömötör and M. I. Bán, *On the Elber-Karplus reaction path-following method and related procedures*, Chem. Phys. Lett. **311**, 328 (1999).

[103] R. Elber and M. Karplus, *Reply to a paper by Stachó et al.*, Chem. Phys. Lett. **311**, 335 (1999).

[104] G. Henkelman and H. Jónsson, *A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives*, J. Chem. Phys. **111**, 7010 (1999).

[105] G. Henkelman and H. Jónsson, *Long time scale kinetic Monte Carlo simulations without lattice approximation and predefined event table*, J. Chem. Phys. **115**, 9657 (2001).

[106] R. A. Miron and K. A. Fichthorn, *The step and slide method for finding saddle points on multidimensional potential surfaces*, J. Chem. Phys. **115**, 8742 (2001).

[107] P. Maragakis, S. A. Andreev, Y. Brumer, D. R. Reichman and E. Kaxiras, *Adaptive nudged elastic band approach for transition state calculation*, J. Chem. Phys. **117**, 4651 (2002).

[108] L. Xie, H. Liu, and W. Yang, *Adapting the nudged elastic band method for determining minimum-energy paths of chemical reactions in enzymes*, J. Chem. Phys. **120**, 8039 (2004).

[109] B. Peters, A. Heyden, A. T. Bell and A. Chakraborty, *A growing string method for determining transition states: Comparison to the nudged elastic band and string methods*, J. Chem. Phys. **120**, 7877 (2004).

[110] R. Crehuet and M. J. Field, *A temperature-dependent nudged elastic band algorithm*, J. Chem. Phys. **118**, 9563 (2003).

[111] M. E. J. Newman, *The structure and function of complex networks*, SIAM Review **45**, 167 (2003).

[112] R. S. Berry and R. Breitengraser-Kunz, *Topography and dynamics of multidimensional interatomic potential surfaces*, Phys. Rev. Lett. **74**, 3951 (1995).

[113] M. A. Miller, J. P. K. Doye and D. J. Wales, *Structural relaxation in atomic clusters: Master equation dynamics*, Phys. Rev. E **60**, 3701 (1999).

[114] J. P. K. Doye and D. J. Wales, *Structural transitions and global minima of sodium chloride clusters*, Phys. Rev. B **59**, 2292 (1999).

[115] J. P. K. Doye and C. P. Massen, *Characterizing the network topology of the energy landscapes of atomic clusters*, J. Chem. Phys. **122**, 084105 (2005).

[116] S. H. Strogatz, *Exploring complex networks*, Nature **410**, 268 (2001).

[117] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks*, Nature **393**, 440 (1998).

[118] A. L. Barabási and R. Albert, *Emergence of scaling in random networks*, Science **286**, 509 (1999).

[119] J. S. Andrade, H. J. Herrmann, R. F. S. Andrade and L. R. da Silva, *Apollonian networks: Simultaneously scale-free, small world, Euclidean, space filling, and with matching graphs*, Phys. Rev. Lett. **94**, 018702 (2005).

[120] E. Nelson, *Dynamical theories of Brownian motion*, Princeton University Press, Princeton (1967).

[121] H. C. Berg, *Random walks in biology*, Princeton University Press, Princeton (1993).

[122] W. Forst, *Unimolecular reactions: A concise introduction*, Cambridge University Press, Cambridge (2003).

[123] G. R. Grimmett and D. R. Stirzaker, *Probability and random processes*, Oxford University Press, Oxford (2005).

[124] I. Todhunter, *History of the mathematical theory of probability*, Cambridge University Press, Cambridge (1965).

[125] T. A. Halgren and W. N. Lipscomb, *The synchronous-transit method for determining reaction pathways and locating molecular transition states*, Chem. Phys. Lett. **49**, 225 (1977).

[126] M. J. S. Dewar, E. F. Healy and J. J. P. Stewart, *Location of transition states in reaction mechanisms*, J. Chem. Soc., Faraday Trans. 2 **80**, 227 (1984).

[127] C. Cardenas-Lailhacar and M. C. Zerner, *Searching for transition-states — the line-then-plane (LTP) approach*, Int. J. Quant. Chem. **55**, 429 (1995).

[128] D. A. Liotard, *Algorithmic tools in the study of semiempirical potential surfaces*, Int. J. Quant. Chem. **44**, 723 (1992).

[129] C. Choi and R. Elber, *Reaction path study of helix formation in tetrapeptides — effect of side chains*, J. Chem. Phys. **94**, 751 (1991).

[130] F. Jensen, *Introduction to computational chemistry*, John Willey & Sons (1999).

[131] CSEP, *Electronic book on mathematical optimization*, http://csep1.phy.ornl.gov/mo/mo.html (1995).

[132] R. Fletcher, *Practical methods of optimization*, John Willey & Sons (2001).

[133] D. A. Evans and D. J. Wales, *The free energy landscape and dynamics of met-enkephalin*, J. Chem. Phys. **119**, 9947 (2003).

[134] H. Goldstein, *Classical mechanics*, Addison-Wesley, Reading, Massachusetts (1980).

[135] F. S. Acton, *Numerical methods that work*, Mathematical Association of America, Washington (1990).

[136] D. R. Alfonso and K. D. Jordan, *A flexible nudged elastic band program for optimisation of minimum energy pathways using ab initio electronic structure methods*, J. Comp. Chem. **24**, 990 (2003).

[137] D. R. Alfonso, *Driver for performing minimum energy path optimizations using the nudged elastic band algorithm*, http://www.pitt.edu/∼alfonso/NEB/neb.html (2002).

[138] D. J. Wales, *OPTIM: A program for optimizing geometries and calculating reaction pathways*, http://www-wales.ch.cam.ac.uk/OPTIM/ (2006).

[139] M. P. Hodges, *Xmakemol: A program for visualizing atomic and molecular systems, version 5.14*, http://www.nongnu.org/xmakemol/ (2006).

[140] *Mathematica, version 5.0*, Wolfram Research, Inc., Champaign, Illinois (1996).

[141] D. J. Wales, *A Mathematica notebook containing some simple programs for making and manipulating triangulated polyhedra*, http://www-wales.ch.cam.ac.uk/software.html (2002).

[142] Y. M. Rhee, *Construction of an accurate potential energy surface by interpolation with Cartesian weighting coordinates*, J. Chem. Phys. **113**, 6021 (2000).

[143] C. Eckart, *Some studies concerning rotating axes and polyatomic molecules*, Phys. Rev. **47**, 552558 (1935).

[144] A. D. McLachlan, *A mathematical procedure for superimposing atomic coordinates of proteins*, Acta. Crystallogr. **A 28**, 656 (1972).

[145] A. Y. Dymarsky and K. N. Kudin, *Computation of the pseudorotation matrix to satisfy the Eckart axis conditions*, J. Chem. Phys. **122**, 124103 (2005).

[146] M. Dierksen, *Comment on "Computation of the pseudorotation matrix to satisfy the Eckart axis conditions"*, J. Chem. Phys. **122**, 227101 (2005).

[147] A. Y. Dymarsky and K. N. Kudin, *Response to "Comment on 'Computation of the pseudorotation matrix to satisfy the Eckart axis conditions' "*, J. Chem. Phys. **122**, 227102 (2005).

[148] R. E. Leone and P. von R. Schleyer, *Degenerate carbonium ions*, Angew. Chem. Int. Ed. Engl. **9**, 860 (1970).

[149] J. P. K. Doye, M. A. Miller and D. J. Wales, *Evolution of the potential energy surface with size for Lennard-Jones clusters*, J. Chem. Phys. **111**, 8417 (1999).

[150] J. P. Neirotti, F. Calvo, D. L. Freeman and J. D. Doll, *Phase changes in 38-atom Lennard-Jones clusters. I. A parallel tempering study in the canonical ensemble*, J. Chem. Phys. **112**, 10340 (2000).

[151] F. Calvo, J. P. Neirotti, D. L. Freeman and J. D. Doll, *Phase changes in 38-atom Lennard-Jones clusters. II. A parallel tempering study of equilibrium and dynamic properties in the molecular dynamics and microcanonical ensembles*, J. Chem. Phys. **112**, 10350 (2000).

[152] F. H. Stillinger and T. A. Weber, *Dynamics of structural transitions in liquids*, Phys. Rev. A **28**, 2408 (1983).

[153] E. W. Dijkstra, *A note on two problems in connection with graphs*, Numerische Math. **1**, 269 (1959).

[154] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to algorithms*, The MIT Press, Cambridge, Massachusetts , Cambridge, Mass., 2nd edn. (2001).

[155] E. W. Weisstein, *'Complete graph'. From Mathworld — A Wolfram web resource*, http://mathworld.wolfram.com/CompleteGraph.html (2005).

[156] A. G. Cochran, N. J. Skelton and M. A. Starovasnik, *Tryptophan zippers: Stable, monomeric β-hairpins*, Proc. Natl. Acad. Sci. USA **98**, 5578 (2001).

[157] C. D. Snow, L. Qiu, D. Du, F. Gai, S. J. Hagen and V. S. Pande, *Trp zipper folding kinetics by molecular dynamics and temperature-jump spectroscopy*, Proc. Natl. Acad. Sci. USA **101**, 4077 (2004).

[158] D. Du, Y. Zhu, C. Huang and F. Gai, *Understanding the key factors that control the rate of β-hairpin folding*, Proc. Natl. Acad. Sci. USA **101**, 15915 (2004).

[159] J. D. Bloom, *Computer simulations of protein aggregation*, Master's thesis, University of Cambridge (July 2002).

[160] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne, *The protein data bank*, Nucleic Acids Research **28**, 235 (2000).

[161] Z. Li and H. A. Scheraga, *Structure and free energy of complex thermodynamic systems*, Proc. Natl. Acad. Sci. USA **84**, 6611 (1987).

[162] D. J. Wales and J. P. K. Doye, *Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms*, J. Phys. Chem. A. **101**, 5111 (1997).

[163] D. J. Wales and H. A. Scheraga, *Global optimization of clusters, crystals and biomolecules*, Science **285**, 1368 (1999).

[164] C. Branden and J. Tooze, *Introduction to protein structure*, Taylor & Francis, New York (1999).

[165] S. A. Trygubenko, *Rotamers in CHARMM19 force field*, http://www.trygub.com/charmm/rotamers/ (2006).

[166] R. Koradi, M. Billeter and K. Wüthrich, *MOLMOL: A program for display and analysis of macromolecular structures*, J. Mol. Graphics **14**, 51 (1996).

[167] J. M. Carr, S. A. Trygubenko and D. J. Wales, *Finding pathways between distant local minima*, J. Chem. Phys. **122**, 234903 (2005).

[168] J. P. K. Doye and D. J. Wales, *Calculation of thermodynamic properties of small Lennard-Jones clusters incorporating anharmonicity*, J. Chem. Phys. **102**, 9659 (1995).

[169] F. Calvo, J. P. K. Doye and D. J. Wales, *Quantum partition functions from classical distributions: Application to rare-gas clusters*, J. Chem. Phys. **114**, 7312 (2001).

[170] P. G. Mezey, *Catchment region partitioning of energy hypersurfaces. I.*, Theor. Chim. Acta **58**, 309 (1981).

[171] T. F. Middleton and D. J. Wales, *Energy landscapes of some model glass formers*, Phys. Rev. B **64**, 24205 (2001).

[172] S. Sastry, P. G. Debenedetti and F. H. Stillinger, *Signatures of distinct dynamical regimes in the energy landscape of a glass-forming liquid*, Nature **393**, 554 (1998).

[173] S. D. Stoddard and J. Ford, *Numerical experiments on the stochastic behavior of a Lennard-Jones gas system*, Phys. Rev. A **8**, 1504 (1973).

[174] M. Vogel, B. Doliwa, A. Heuer and S. C. Glotzer, *Particle rearrangements during transitions between local minima of the potential energy landscape of a binary Lennard-Jones liquid*, J. Chem. Phys. **120**, 4404 (2004).

[175] M. Arndt, R. Stannarius, H. Groothues, E. Hempel and F. Kremer, *Length scale of cooperativity in the dynamic glass transition*, Phys. Rev. Lett. **79**, 2077 (1997).

[176] L. Berthier, *Time and length scales in supercooled liquids*, Phys. Rev. E **69**, 20201 (2004).

[177] D. J. Wales, *A microscopic basis for the global appearance of energy landscapes*, Science **293**, 2013 (2001).

[178] T. V. Bogdan and D. J. Wales, *New results for phase transitions from catastrophe theory*, J. Chem. Phys. **120**, 11090 (2004).

[179] M. G. Bulmer, *Principles of statistics*, Dover Publications, New York (1979).

[180] K. D. Ball, R. S. Berry, R. E. Kunz, F. Y. Li, A. Proykova and D. J. Wales, *From topographies to dynamics on multidimensional potential energy surfaces of atomic clusters*, Science **271**, 963 (1996).

[181] S. A. Trygubenko, *Rearrangements of LJ_{75} cluster localized on two atoms*, http://www.trygub.com/ExtremeRearrangement/LJ75/2atom.html (2006).

[182] Y. Gebremichael, M. Vogel and S. C. Glotzer, *Particle dynamics and the development of string-like motion in a simulated monoatomic supercooled liquid*, J. Chem. Phys. **120**, 4415 (2004).

[183] T. S. Jain and J. J. de Pablo, *Influence of confinement on the vibrational density of states and the boson peak in a polymer glass*, J. Chem. Phys. **120**, 9371 (2004).

[184] T. F. Middleton and D. J. Wales, *Energy landscapes of model glasses. II. Results for constant pressure*, J. Chem. Phys. **118**, 4583 (2003).

[185] B. Doliwa and A. Heuer, *Energy barriers and activated dynamics in a supercooled Lennard-Jones liquid*, Phys. Rev. E **67**, 031506 (2003).

[186] A. Saksaengwijit, B. Doliwa and A. Heuer, *Description of the dynamics in complex energy landscapes via metabasins: A simple model study*, J. Phys. Cond. Matt. **15**, S1237 (2003).

[187] G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi, *Queueing networks and Markov chains*, John Wiley and Sons, New York (1998).

[188] G. R. Grimmett and D. R. Stirzaker, *One thousand exercises in probability*, Oxford University Press, Oxford (2005).

[189] N. G. van Kampen, *Stochastic processes in physics and chemistry*, Elsevier, Amsterdam (1981).

[190] A. B. Bortz, M. H. Kalos and J. L. Lebowitz, *A new algorithm for Monte Carlo simulation of Ising spin systems*, J. Comput. Phys. **17**, 10 (1975).

[191] K. A. Fichthorn and W. H. Weinberg, *Theoretical foundations of dynamical Monte Carlo simulations*, J. Chem. Phys **95**, 1090 (1991).

[192] M. A. Miller, *Energy landscapes and dynamics of model clusters*, Ph.D. thesis, University of Cambridge (March 1999).

[193] M. Block, R. Kunert, E. Schöll, T. Boeck and T. Teubner, *Kinetic Monte Carlo simulation of formation of microstructures in liquid droplets*, New J. Phys. **6**, 166 (2004).

[194] D. Mukherjee, C. G. Sonwane and M. R. Zachariah, *Kinetic Monte Carlo simulation of the effect of coalescence energy release on the size and shape evolution of nanoparticles grown as an aerosol*, J. Chem. Phys. **119**, 3391 (2003).

[195] F. M. Bulnes, V. D. Pereyra and J. L. Riccardo, *Collective surface diffusion: n-fold way kinetic Monte Carlo simulation*, Phys. Rev. E **58**, 8692 (1998).

[196] R. E. Kunz, *Dynamics of first-order phase transitions*, Deutsch, Thun (1995).

[197] D. A. Evans and D. J. Wales, *Folding of the GB1 hairpin peptide from discrete path sampling*, J. Chem. Phys. **121**, 1080 (2004).

[198] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu and J.-C. Latombe, *Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion*, in *Recomb '02: Proceedings of the sixth annual international conference on computational biology*, pp. 12–21. ACM Press, New York, NY (2002).

[199] M. S. Apaydin, C. E. Guestrin, C. Varma, D. L. Brutlag and J.-C. Latombe, *Stochastic roadmap simulation for the study of ligand-protein interactions*, Bioinf. **18**, S18 (2002).

[200] N. Singhal, C. D. Snow and V. S. Pande, *Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin*, J. Chem. Phys. **121**, 415 (2004).

[201] G. Chartrand, *Introductory graph theory*, Dover Publications, New York (1977).

[202] NIST, *Dictionary of algorithms and data structures*, http://www.nist.gov/dads/ (2005).

[203] A. Bar-Haim and J. Klafter, *On mean residence and first-passage times in finite one-dimensional systems*, J. Chem. Phys. **109**, 5187 (1998).

[204] B. E. Trumbo, *Relationship between the Poisson and exponential distributions*, http://www.sci.csuhayward.edu/statistics/Resources/Essays/PoisExp.htm (1999).

[205] T. F. Middleton, *Energy landscapes of model glasses*, Ph.D. thesis, University of Cambridge (March 2003).

[206] D. A. Reed and G. Ehrlich, *Surface diffusivity and the time correlation of concentration fluctuations*, Surf. Sci. **105**, 603 (1981).

[207] A. F. Voter, *Introduction to the kinetic Monte Carlo method*, in *Radiation effects in solids*, pp. 1–22. Springer-Verlag, New York (2005).

[208] I. Goldhirsch and Y. Gefen, *Analytic method for calculating properties of random walks on networks*, Phys. Rev. A **33**, 2583 (1986).

[209] I. Goldhirsch and Y. Gefen, *Biased random walk on networks*, Phys. Rev. A **35**, 1317 (1987).

[210] M. Raykin, *First-passage probability of a random walk on a disordered one-dimensional lattice*, J. Phys. A **26**, 449 (1992).

[211] K. P. N. Murthy and K. W. Kehr, *Mean first-passage time of random walks on a random lattice*, Phys. Rev. A **40**, 2082 (1989).

[212] G. H. Weiss, *First-passage time problems in chemical physics*, Adv. Chem. Phys. **13**, 1 (1967).

[213] W. Ledermann and G. E. H. Reuter, *Spectral theory for the differential equations of simple birth and death processes*, Philos. T. Roy. Soc. A **A246**, 321 (1954).

[214] S. Karlin and J. L. MacGregor, *The differential equations of birth and death processes*, Trans. Am. Math. Soc. **85**, 489 (1957).

[215] C. W. Gardiner, *Handbook of stochastic methods in physics, chemistry and the natural sciences*, Springer, Berlin (1985).

[216] C. V. den Broeck, *A glimpse into the world of random walks*, in *Proceedings of NATO conference on noise and nonlinear phenomena in nuclear systems*, edited by J. L. Munoz-Cobo and F. C. Difilippo, pp. 3–18. Plenum, New York (1988).

[217] P. L. Doussal, *First-passage time for random walks in random environments*, Phys. Rev. Lett. **62**, 3097 (1989).

[218] K. P. N. Murthy and K. W. Kehr, *Erratum: Mean first-passage time of random walks on a random lattice*, Phys. Rev. A **41**, 1160 (1990).

[219] O. Matan and S. Havlin, *Mean first-passage time on loopless aggregates*, Phys. Rev. A **40**, 6573 (1989).

[220] P. A. Pury and M. O. Cáceres, *Mean first-passage and residence times of random walks on asymmetric disordered chains*, J. Phys. A **36**, 2695 (2003).

[221] M. Slutsky, M. Kardar and L. A. Mirny, *Diffusion in correlated random potentials, with applications to DNA*, Phys. Rev. E **69**, 061903 (2004).

[222] M. Slutsky and L. A. Mirny, *Kinetics of protein-DNA interaction: Facilitated target location in sequence-dependent potential*, Biophys. J. **87**, 4021 (2004).

[223] Ya. G. Sinai, *The limiting behaviour of a one-dimensional random walk in random environment*, Theor. Prob. and Appl. **27**, 256 (1982).

[224] H. E. Stanley and S. Havlin, *Generalisation of the Sinai anomalous diffusion law*, J. Phys. A **20**, L615 (1987).

[225] R. D. Vale and R. A. Milligan, *The way things move: Looking under the hood of molecular motor proteins*, Science **288**, 88 (2000).

[226] R. D. Astumian, *Thermodynamics and kinetics of a Brownian motor*, Science **276**, 917 (1997).

[227] P. G. de Gennes, *Problems of DNA entry into a cell*, Physica A **274**, 1 (1999).

[228] A. Meller, L. Nivon and D. Branton, *Voltage-driven DNA translocations through a nanopore*, Phys. Rev. Lett. **86**, 3435 (2001).

[229] K. Mussawisade, J. E. Santos and G. M. Schütz, *Branching-annihilating random walks in one dimension: Some exact results*, J. Phys. A **31**, 4381 (1998).

[230] C. Sire, *Analytical results for random walks in the presence of disorder and traps*, Phys. Rev. E **60**, 1464 (1999).

[231] M. F. Schlesinger and J. Klafter, *Random walks in liquids*, J. Phys. Chem. **93**, 7023 (1989).

[232] I. Herbut and S. Milosevic, *Hopping on hierarchical structures and random walking on deterministic fractals*, J. Phys. A **23**, 99 (1990).

[233] L. Acedo and S. B. Yuste, *Territory covered by N random walkers on fractal media: The Sierpinski gasket and the percolation aggregate*, Phys. Rev. E **63**, 011105 (2000).

[234] M. Bauer, D. Bernard and J. M. Luck, *Even-visiting random walks: exact and asymptotic results in one dimension*, J. Phys. A **34**, 2659 (2001).

[235] V. I. Alkhimov, *The problem of a self-avoiding random walk*, Uspekhi Fizicheskikh Nauk **161**, 133 (1991).

[236] I. Majid, D. Ben-Avraham, S. Havlin and H. E. Stanley, *Exact-enumeration approach to random walks on percolation clusters in two dimensions*, Phys. Rev. B **30**, 1626 (1984).

[237] B. L. Trus, S. Havlin and D. Stauffer, *Distribution of first-passage times for diffusion at the percolation threshold*, J. Phys. A **20**, 6627 (1987).

[238] E. W. Montroll and G. H. Weiss, *Random walks on lattices. II.*, J. Math. Phys. **6**, 167 (1965).

[239] J. M. Luck, *A numerical study of diffusion and conduction in a 2D random medium*, J. Phys. A **17**, 2069 (1984).

[240] D. Zheng, Y. Liu and Z. D. Wang, *Matrix method for random walks on lattices*, J. Phys. A **28**, L409 (1995).

[241] H. M. Taylor and S. Karlin, *An introduction to stochastic modeling*, Academic Press, New York (1998).

[242] S. Goldberg, *Probability: An introduction*, Dover Publications, New York (1960).

[243] B. Kahng and S. Redner, *Scaling of the first-passage time and the survival probability on exact and quasi-exact self-similar structures*, J. Phys. A **22**, 887 (1989).

[244] S. K. Kim and H. H. Lee, *Crystallinity and average grain size of films grown by chemical vapor deposition*, J. Appl. Phys. **78**, 3809 (1995).

[245] P. C. Bressloff, V. M. Dwyer and M. J. Kearney, *A 'sum-over-paths' approach to diffusion on trees*, J. Phys. A **29**, 1881 (1996).

[246] S. Revathi, V. Balakrishnan, S. Lakshmibala and K. P. N. Murthy, *Validity of the mean-field approximation for diffusion on a random comb*, Phys. Rev. E **54**, 2298 (1996).

[247] K. Kim, J. S. Choi and Y. S. Kong, *Multifractals of normalized first passage time in Sierpinski gasket*, J. Phys. Soc. Jpn. **67**, 1583 (1998).

[248] K. Kim, G. H. Kim and Y. S. Kong, *Multifractal measures characterized by the iterative map with two control parameters*, Fractals **8**, 181 (2000).

[249] J. Asikainen, J. Heinonen and T. Ala-Nissila, *Exact and efficient discrete random walk method for time-dependent two-dimensional environments*, Phys. Rev. E **66**, 066706 (2002).

[250] D. ben Avraham, S. Redner and Z. Cheng, *Random walk in a random multiplicative environment*, J. Stat. Phys. **56**, 437 (1989).

[251] Y. Gefen and I. Goldhirsch, *The building blocks of random walks*, Physica D **38**, 119 (1989).

[252] H. Haucke, S. Washburn, A. D. Benoit, C. P. Umbach and R. A. Webb, *Universal scaling of nonlocal and local resistance fluctuations in small wires*, Phys. Rev. B **41**, 12454 (1990).

[253] S. Revathi and V. Balakrishnan, *Diffusion coefficient for random walks on strips with spatially inhomogeneous boundaries*, J. Phys. A **26**, 467 (1993).

[254] S. H. Noskowicz and I. Goldhirsch, *First-passage-time distribution in a random random walk*, Phys. Rev. A **42**, 2047 (1990).

[255] S. Revathi and V. Balakrishnan, *Analytic calculation of the diffusion coefficient for random walks on strips of finite width — dependence on size and nature of boundaries*, Phys. Rev. E **47**, 916 (1993).

[256] V. Balakrishnan and C. Vandenbroeck, *Transport properties on a random comb*, Physica A **217**, 1 (1995).

[257] A. R. Kerstein and R. B. Pandey, *Conductivity exponent for stirred superconductor-insulator mixtures*, Phys. Rev. B **35**, 3575 (1987).

[258] Y. Gefen and I. Goldhirsch, *Relation between the classical resistance of inhomogeneous networks and diffusion*, Phys. Rev. B **35**, 8639 (1987).

[259] Y. Gefen and I. Goldhirsch, *Biased diffusion on random networks: mean first-passage time and DC conductivity*, J. Phys. A **18**, L1037 (1985).

[260] J. W. Haus and K. W. Kehr, *Diffusion in regular and disordered lattices*, Phys. Reports **150**, 263 (1987).

[261] I. G. S. H. Noskowicz, *Distribution functions for random walk processes on networks: An analytic method*, J. Stat. Phys. **48**, 255 (1987).

[262] R. Landauer and M. Buttiker, *Diffusive traversal time — effective area in magnetically induced interference*, Phys. Rev. B **36**, 6255 (1987).

[263] R. Tao, *Studies of the spectral dimension for branched Koch curves*, J. Phys. A **20**, 6151 (1987).

[264] J. Koplik, S. Redner and D. Wilkinson, *Transport and dispersion in random networks with percolation disorder*, Phys. Rev. A **37**, 2619 (1988).

[265] N. M. van Dijk, *Queueing networks and product forms*, John Wiley and Sons, New York (1993).

[266] E. Gelenbe and G. Pujolle, *Introduction to queueing networks*, John Wiley and Sons, New York (1998).

[267] A. E. Conway and N. D. Georganas, *Queueing networks - exact computational algorithms*, The MIT Press, Cambridge, Massachusetts (1989).

[268] D. Eppstein, Z. Galil and G. F. Italiano, *Dynamic graph algorithms*, in *Algorithms and theory of computation handbook*, edited by M. J. Atallah, chap. 8, CRC Press (1999).

[269] B. V. Cherkassky, A. V. Goldberg and T. Radzik, *Shortest paths algorithms: Theory and experimental evaluation*, in *Soda '94: Proceedings of the fifth annual ACM-SIAM symposium on discrete algorithms*, pp. 516–525, Society for Industrial and Applied Mathematics, Philadelphia, PA (1994).

[270] G. Ramalingam and T. Reps, *An incremental algorithm for a generalization of the shortest-path problem*, J. Algorithms **21**, 267 (1996).

[271] G. Ramalingam and T. Reps, *On the computational complexity of dynamic graph problems*, Theor. Comput. Sci. **158**, 233 (1996).

[272] S. A. Trygubenko and D. J. Wales, *Kinetic analysis of discrete path sampling stationary point databases*, Mol. Phys., in press (2006).

[273] *GNU general public license*, http://www.gnu.org/copyleft/gpl.html (2006).

[274] S. A. Trygubenko, *Graph transformation program*, http://www.trygub.com/gt/ (2006).

[275] *Debian — the universal operating system*, http://www.debian.org/ (2006).

[276] M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimisation algorithms*, Journal of the ACM **34**, 596 (1987).

[277] J. Pitman, *Probability*, Springer-Verlag, New York (1993).

[278] N. A. Weiss, *A course in probability*, Pearson, Addison Wesley, Boston (2006).

[279] P. Labastie and R. L. Whetten, *Statistical mechanics of the cluster solid-liquid transition*, Phys. Rev. Lett. **65**, 1567 (1990).

[280] R. S. Berry, T. L. Beck, H. L. Davis and J. Jellinek, *Solid-liquid phase behavior in microclusters*, Adv. Chem. Phys. **70B**, 75 (1988).

[281] P. Chen, *A random walk or color chaos on the stock market? Time-frequency analysis of S&P indexes*, Stud. in Nonlin. Dyn. and Econ. **1**, 87 (1996).

[282] M. J. Plank and B. D. Sleeman, *A reinforced random walk model of tumour angiogenesis and anti-angiogenic strategies*, Math. Med. Bio. **20**, 135 (2003).

[283] W. E, W. Ren and E. Vanden-Eijnden, *String method for the study of rare events*, Phys. Rev. B **66**, 052301 (2002).

[284] W. E, W. Ren and E. Vanden-Eijnden, *Energy landscape and thermally activated switching of submicron-sized ferromagnetic elements*, J. Appl. Phys. **93**, 2275 (2003).

[285] W. Quapp, *Reaction pathways and projection operators: Application to string methods*, J. Comp. Chem. **25**, 1277 (2004).

[286] W. E, W. Ren and E. Vanden-Eijnden, *Finite temperature string method for the study of rare events*, J. Phys. Chem. A. **109**, 6688 (2005).

[287] W. Ren, E. Vanden-Eijnden, P. Maragakis and W. E, *Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide*, J. Chem. Phys. **123**, 134109 (2005).

[288] W. Quapp, *A growing string method for the reaction pathway defined by a Newton trajectory*, J. Chem. Phys. **122**, 174106 (2005).

[289] J. W. Chu, B. L. Trout and B. R. Brooks, *A super-linear minimization scheme for the nudged elastic band method*, J. Chem. Phys. **119**, 12708 (2003).

[290] D. A. Evans, *Energy landscapes and dynamics of model peptides*, Ph.D. thesis, University of Cambridge (December 2003).

[291] J. M. Carr, *Energy landscapes and dynamics of model proteins*, Ph.D. thesis, University of Cambridge (April 2005).

[292] R. E. Bellman, *On a routing problem*, Quat. Appl. Math **16**, 87 (1958).

[293] L. R. Ford and D. R. Fulkerson, *Flows in networks*, Princeton University Press, Princeton (1962).

[294] E. F. Moore, *The shortest path through a maze*, in *Proceedings of the international symposium on the theory of switching*, pp. 285–292. Harvard University Press, Harvard (1959).

Publications

Some of the work presented in this thesis has been published in the following papers:

Chapter 2

S. A. Trygubenko and D. J. Wales, *A doubly nudged elastic band method for finding transition states*, J. Chem. Phys. **120**, 2082-2094 (2004).

J. M. Carr, S. A. Trygubenko and D. J. Wales, *Finding pathways between distant local minima*, J. Chem. Phys. **122**, 234903-234910 (2005).

Chapter 3

S. A. Trygubenko and D. J. Wales, *Analysis of cooperativity and localisation for atomic rearrangements*, J. Chem. Phys. **121**, 6689-6697 (2004).

Chapter 4

S. A. Trygubenko and D. J. Wales, *Kinetic analysis of discrete path sampling stationary point databases*, Mol. Phys., in press (2006).