Simulation and Control of Dynamic Directed Self-assembly of Nanostructures

by

Sivaraman Ramaswamy

B.Tech., Chemical Engineering
Indian Institute of Technology Madras (2010)
M.S., Chemical Engineering Practice
Massachusetts Institute of Technology (2014)

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

Signature Redacted

Author .....
Department of Chemical Engineering
Signature Redacted 5, 2015

Certified by .....
George Stephanopoulos
Arthur D. Little Professor of Chemical Engineering
Signature Redacted Thesis Supervisor

Certified by .....
Paul I. Barton
Lammot du Pont Professor of Chemical Engineering
Signature Redacted visor

Accepted by .....
Richard D. Braatz
Edwin R. Gilliland Professor of Chemical Engineering
Chairman, Committee for Graduate Students

Simulation and Control of Dynamic Directed Self-assembly of Nanostructures

by

Sivaraman Ramaswamy

Submitted to the Department of Chemical Engineering
on May 05, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Self-assembled nanoscale structures are the basis for various technological advancements in functional materials, sensors, and molecular circuits and factories. With significant progress in self-assembly of periodic nanostructures (such as monolayers), the focus is now shifting towards non-periodic structures. Control of various interaction force fields (electrostatic, Van der Waals, etc.) between the nanoparticles and external controls can result in the formation of nanostructures with desired geometry. The aim is to design the nanoparticles and the external actuators such that the desired structure can be self-assembled rapidly with high reliability and avoiding any kinetic trapping that an ill-designed energy landscape might cause.

Deterministic dynamic modeling of such self-assembled nanostructures, directed by external fields, through a Master Equation approach, leads to a set of differential equations of such large size that even the most efficient solution algorithms are overwhelmed. Thus, model reduction is a key necessity. This thesis presents a methodological approach and specific algorithms, which generate time-varying, reduced-order models for the description of directed self-assembly of nanoparticles by external fields. The approach is based on Finite State Projection and is adaptive, i.e., it generates reduced-order models that vary over time. The algorithm uses event-detection concepts to determine automatically, during simulation, suitable time points at which the projection space and thus the structure of the reduced-order model change, in such a way that the computational load remains low while the upper bound on the simulation error, resulting from model reduction, is lower than a prescribed maximum limit.

The thesis also presents an optimal control strategy that can guide any initial random configuration of nanoparticles to a final structure of desired geometry, in minimum time. It employs a multi-resolution view of the dynamically evolving configurations of nanoparticles, which are described through the simulation methodology described before. External charges, attracting or repelling the nanoparticles, are the controls, whose location and intensity are determined by the optimality conditions of the optimal control strategy. To ensure analytic consistency of the parametric

sensitivities, during the computation of the optimal controls, and thus guarantee the optimality of the resulting control policy, a priori determination of enlarged constant projection spaces is shown to be essential.

The thesis also presents a series of case studies, which illustrate how the proposed methods can be used to simulate effectively directed self-assembly of an appreciable number of nanoparticles, and reach the desired geometry. These case studies also illuminate several of its features, such as: superiority over a static optimal solution; evasion of kinetic traps; and effective handling of combinatorial complications arising for systems with large-size domains and many particles.

Thesis Supervisor: George Stephanopoulos
Title: Arthur D. Little Professor of Chemical Engineering

Thesis Supervisor: Paul I. Barton
Title: Lammot du Pont Professor of Chemical Engineering

Acknowledgments

Almost five years have passed since I came to MIT, and my transformation from being the shy, intimidated person to a more vocal, self-confident man cannot be expressed in words. My advisor, Professor George Stephanopoulos, was a constant source of motivation for me. This thesis would not have taken this shape without his encouraging words of wisdom. His stress on taking initiative, looking at the big picture, and the ability to ask the right questions provided me with the critical analysis I needed to develop the attitude towards solving independent research problems. His feedback helped me a lot in clearly communicating my ideas and explanations. I will always remember all of his quotes and play that in my head each time I am stuck in research.

Professor Paul Barton always amazed me with his extensive knowledge about every topic in this world. His guidance with the computational implementation of the numerical methods was invaluable, and I learnt a lot about optimization from him. His attention to detail, and the ability to foresee all the problems with any new idea played a very critical role in the completion of my project. A meeting with him one day lasted four and a half hours, which is a testament to his dedication and passion for the subject.

I would also like to thank my committee members. Professor Richard Braatz and Professor Arup Chakraborty for their constant guidance and feedback throughout the progress of my project.

This work could not have been completed without the help of Process Systems Engineering Laboratory (PSEL) members, especially Professor Richard Lakerveld. Richard helped me out during a time when I was new to the lab and learning how to do research. His contribution to my project and in developing my research abilities gave me a lot of confidence and propelled me in the right direction. I would also like to thank Arash for all the help he provided in understanding the optimal control literature. I have brainstormed ideas with almost all of my labmates and I would like to thank them for that. Thanks to all the staff members in the Chemical Engineering

Department, especially Joel, Suzanne and Fran in the Student Office, for helping me a lot with all the administrative issues.

A special thanks to my family and friends, who were a constant source of support for me. The Indian association at MIT has played a special part in making my stay here very memorable. Thank you all for those countless trips, gossips and enthusiasm. To Vishnu, Sayalee, and Sagar, you all were always there when I needed some company. I am glad we got to explore so many new places together. To Parnika, Radhika, Somani, Shamel, Atulya, Harshad, and Connie, thank you for being my sounding box and always listening to me. This journey was fun because of all of the friends I was fortunate enough to meet along the way.

People close to me know how passionate I am to the game of cricket. Playing with the MIT Cricket Club was something I always looked forward to every weekend, and since the introduction of competitive cricket, my passion has a new meaning. Thank you for all the fun memories.

Finally, I would like to thank National Science Foundation and the Department of Chemical Engineering at MIT for financially supporting my thesis research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Chemical engineers are now exploring nanoscale systems as a very promising technological platform for new materials, devices, and processes. The potential applications are many and can make a significant impact on human lives. Nanoscale systems provide the basis for technological advancements in various fields such as nanoelectronic devices and sensors, consumer products, multi-functional materials, biodiagnostic tools for detection of proteins and DNA, molecular computing, molecular-scale factories, and nanoscale chemical plants [65, 37, 57]. However, fabricating these complex structures is a significant challenge. The current applications employ only relatively simple or periodic and dense structures, and methods to create complex and non-periodic nanostructures have not yet been developed.

Fabrication of complex nanostructures requires fabrication of components such as nanoelectrodes, nanowires, nanoreactors and nanoscaffolding, which serve as building blocks. Molecular machinery [15] was earlier proposed, where atoms are judiciously inserted into nanoscale configurations. However, due to the lack of direct control over atoms at such a small scale, this option is not commercially viable.

For larger scales, arbitrary 2d patterns at resolutions of order 20-90 nm can be created on surfaces using techniques such as photolithography, scanning beam lithography, interferometric lithography and nanoimprinting [18]. Although these top-down

fabrication techniques work well, they have several limitations such as high cost, slow pattering rates, and most importantly, lower resolution than the desired nanostructures [65, 18, 44].

In order to fabricate the nanostructures at smaller scales, an alternate approach is to utilize the intermolecular forces between the nanoparticles to create a self-assembly process. Self-assembly [74, 75] is a spontaneous or directed process that offers fabrication routes towards nanoscale structures with unprecedented resolution. Nanoscale building blocks such as functionalized inorganic nanoparticles or fragments of DNA can bind with each other, forming nanoscale structures. This bottom-up technique is used in a wide range of structures such as nanowires and nanoreactors, with a high resolution of 1 nm.

Significant progress has been made on self-assembly of *dense* nanoscale structures (i.e., structures with a very large number of particles, theoretically tending to infinity, extending over theoretically infinite domains), with *periodic* structural patterns, through the judicious design of building blocks to induce the desired short- and long-range orderings [44]. Fabrication of nanosurfaces such as monolayers using self-assembly is an example of this methodology. Nanosurfaces with periodic patterns of dimensions 10-100 nm can be created with this technique. Such approaches fail to construct nanostructures with relatively small numbers of nanoparticles, within confined domains, and with non-periodic structural features, as required by a number of applications, e.g., nanoelectronics.

Furthermore, even for systems with dense and periodic structures, fabrication techniques produce defects, i.e., failures in achieving the desired structure. However, occasional defects in the fabrication techniques do not affect their functionality. Unlike dense and periodic structures, non-periodic structures can lose their functionality with even a small number of defects.

DNA "smart tiles", which are small, functionalized nanoparticles (5-20 nm widths), has been demonstrated as a building block, allowing self-assembly into large nanostructures [60, 77, 76, 51]. DNA tiles are formed by first chemically synthesizing single-stranded DNA oligonucleotides, which are then self-assembled into a junction

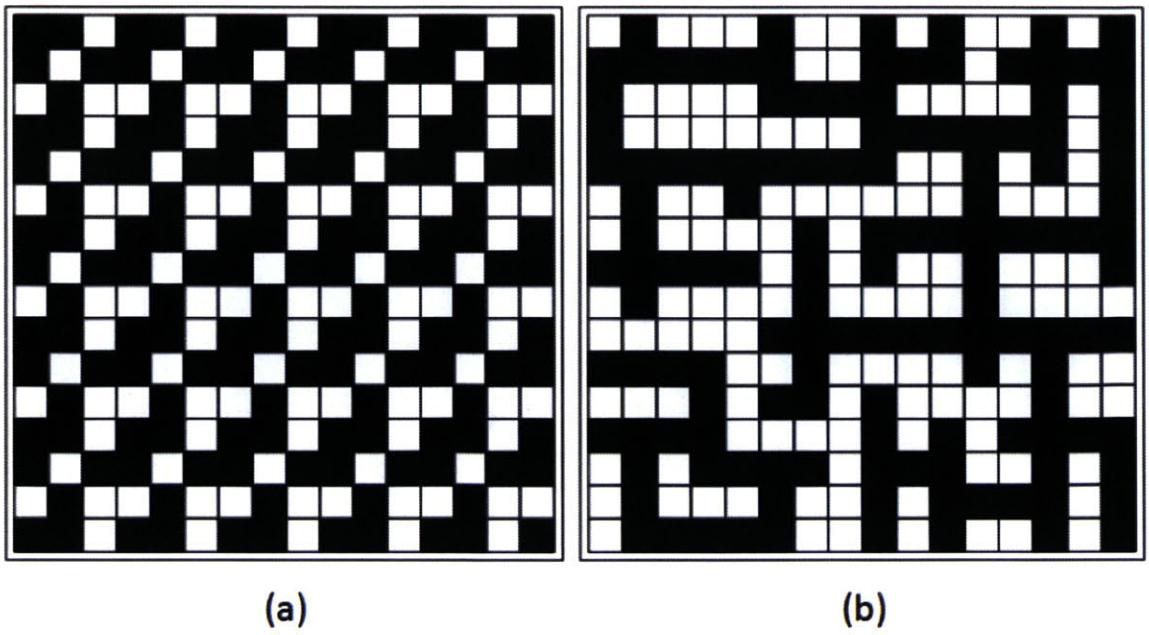

Figure 1-1: Figure showing the different kinds of nanostructures for self-assembly. The structure in (a) is a dense periodic one and the structure in (b) is more complex and non-periodic in nature [65].

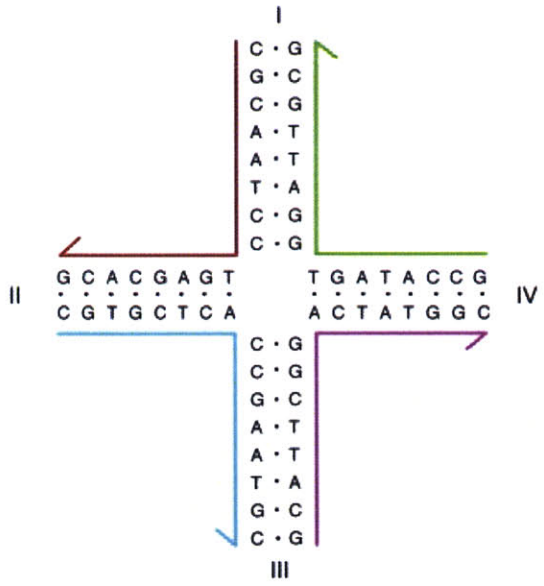

Figure 1-2: Figure of a DNA tile, illustrating the junction formed by single-stranded DNA oligonucleotides through base pairing. The figure has been reproduced from [60].

through appropriate base paring. An example of a DNA tile is shown in Figure 1-2,

which can been reproduced from [60]. DNA tiles can have functionalized edges that can preferentially attach to the functionalized edges of other particular DNA tiles. The DNA bases are negatively charged, and hence repulsive electrostatic forces will be experienced between the tiles at relatively long distances. At short distances, the tiles are aggregated due to strong hydrogen bonding that exist at the edges of these tiles because of the functionalization with specific DNA sequences.

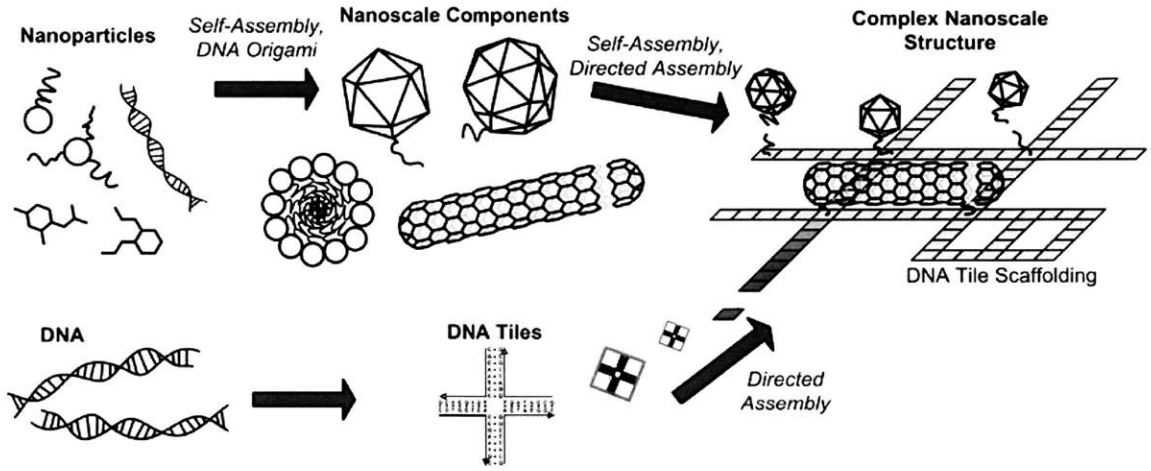

Figure 1-3: A complex nanostructure can be fabricated by placing nanoscale components onto a nanoscoffolding structure. The nanoscale components are fabricated by a self-assembly process. The nanoscaffolding is fabricated through the directed self-assembly of DNA tiles, which are formed by DNA origami.

DNA tiles can be used to create nanoscaffolds, thus providing the foundation for creating complex, non-periodic nanostructures, as shown in Figure 1-3. This forms the basis of this thesis. Some of these DNA tiles have ligands with binding affinities for certain nanoscale components such as nanotubes and nanospheres.

1.2 Research Objectives

Spontaneous self-assembly leads to free-energy structures only for systems with very large numbers of particles and unbounded domains. For systems with relatively small numbers of particles, bounded domains, and non-periodic desired structural features, external fields directing the self-assembling process are needed, such as: shear [11],

magnetic [7, 66], or electric [11, 38, 45, 3, 1, 14, 43, 39, 33]. Electric fields are of particular interest due to the tunable strength and direction [73, 72, 71, 32, 34]. The key questions that need to be answered are: (a) what are the design parameters of the nanoparticles (shape, size, distribution of surface functionality), (b) the properties of the medium in which the self-assembly is taking place, and (c) the location and intensity of external fields, so that the nanoparticles assemble, with very high probability, to a structure that possesses the desired geometry?

In this thesis, we have attempted to answer this question by focusing first on item (c) of the question, i.e., what are the locations and time-dependent intensities of the external charges, so that any initial arrangement of the nanoparticles will evolve to a final structure with desired geometry.

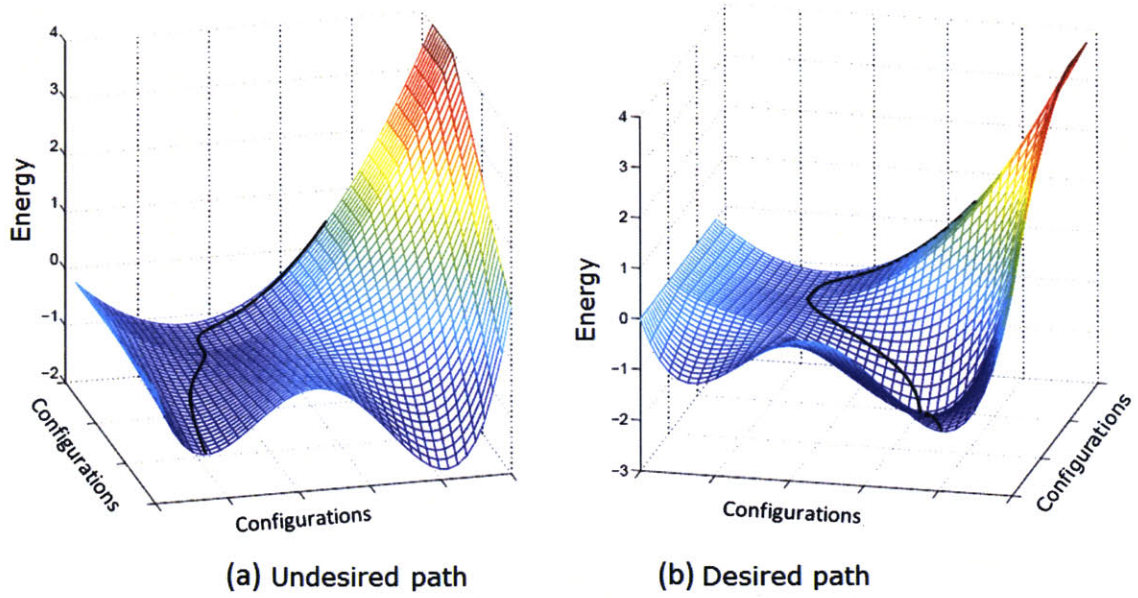

Figure 1-4: Figure showing sample energy landscapes for a self-assembly system. The desired nanostructure corresponds to the globally minimum energy level in the landscape, but the pathway towards reaching this desired nanostructure is very important. The structure in (a) is an undesired pathway leading to a metastable state and (b) is the desired pathway leading to the desired state.

In the first paper by Solis et al. [63], the authors addressed the static problem associated with the controlled formation of nanostructures. They were able to guarantee a statistically robust desired structure by solving the energy-gap maximization

problem (EMP). They showed that they could reduce the phase space combinatorics and include only the neighboring competing states, while still guaranteeing a robust desired final structure. The work done by Solis et al. [63] can be visualized using energy landscapes. The energy landscape, or the potential energy surface, is the mapping of all possible configurations that the nanoparticles can take and their corresponding energy levels. Note that this energy landscape correspond to constant intensities for the external charges. For any arbitrary initial configuration, the dynamic path of self-assembly corresponds to a "path" in the energy landscape, which may lead to the configuration that has the globally minimum energy level. In the first paper, Solis et al. [63] calculated the optimum strengths and locations of the external charges such that the global minimum of the energy landscape for these charges lie at the desired configuration. However, as illustrated in Figure 1-4, the "path" for self-assembly from any initial configuration to the desired configuration may not directly lead to the globally minimum energy level, but rather lead to a suboptimal locally minimum energy level. These locally minimum energy levels are called kinetic traps, and it becomes very difficult for such a system to move out of the kinetic traps in a reasonable time scale. In other words, starting from an arbitrary distribution of particles in the complete phase space, the probability of reaching the desired nanostructure in a reasonable time scale, under the influence of the static controls determined by the solution of the static problem in [63], can be unacceptably low and variable, since the system can be trapped in different metastable states, which are determined by the dynamic path of the self-assembly process. Thus, in their second paper, Solis et al. [64] developed a methodology that overcomes the above restriction and allows the dynamically self-assembled particles to reach the desired final geometry from any initial configuration with a pre-specified high probability, i.e., generate robust self-assembly paths. This methodology was based on the progressive reduction of the system phase space into subsets with progressively smaller numbers of locally allowable configurational states. In other words, it was based on a judicious progressive transition from ergodic to nonergodic subsystems. The subsets of allowable configurations in phase space were modeled by a spatial multi-resolution view of the

desired structure (in terms of the number of particles) and the static optimal control was computed using the methodologies of the first paper. This approach produces a sequential psuedo-steady state prescription for the optimal control, but not a truly optimal control policy, based on the dynamics of the evolving assembly process.

Therefore, the focus of this thesis is to find the dynamic profile for the strengths and locations of the external charges such that final desired structure is not only robust, but also the dynamic path for self-assembly from any initial configuration of nanoparticles does not lead to any kinetic traps.

1.3 Methods in the Literature

The analysis of energy landscapes in order to find the dynamic path has been studied in the protein folding literature. Thus, following the literature on protein folding and macromolecules, we can broadly classify the simulation methods used into two kinds of approaches - energy landscape approach and master equations approach.

1.3.1 Energy Landscape Approach

In this approach, the energy landscape of the system is explored and pathways with low energy are obtained from any initial state to the final state. One of the proposed approaches is dimensionality reduction of the energy landscape. The ISOMAP algorithm is a nonlinear dimensionality reduction technique which defines a low-dimensional embedding that tries to preserve all pairwise distances between the states [67]. Das et al. [12] developed a new method called Scalable ISOMAP (ScIMAP) which is computationally scalable and hence can analyze large data sets. However low-dimensional projections are deceptive, as they shrink the distances between points, and hence can make energy barriers disappear. Removal of barriers can result in a relatively smooth energy surface, making kinetic traps disappear too [40]. Such an approach does not represent the real dynamics of the system.

Noé et al. [49] discussed various methods with respect to conformational changes in macromolecules. In their paper, a pathway connecting the start (initial state) and

the end states is created using a continuous chain of segments. This chain is then path optimized using different techniques such as conjugate peak refinement (CPR) [16] or the nudged elastic band method (NEB) [31]. Another method developed by Noé et al. [50] is based on an adaptive approach using graph theory. In this method, only those connections are considered that go over energy barriers that contribute to the network properties. Although the above mentioned methods are effective in reducing the energy landscape, they are not efficient and cannot handle the size of the system under study in this thesis.

1.3.2 Master Equation Approach

In this approach, the dynamics of self-assembly are described using a system of master equations. Master equations enable us to describe time-dependent continuous-time Markov processes. Each state in the master equations represent the probability of being in a particular configuration of the system.

In the case of macromolecules, Noé et al. [49] explain how molecular dynamics simulations are useful in generating the master equations. The master equation based approach is very popular in a variety of systems as it helps to identify the kinetic traps [46, 8, 35, 78]. Molecular dynamics simulations help in finding the rates (or propensities) involved in movement from one configuration to another [62, 61]. These master equations can be sampled using Monte Carlo based methods such as Gillespie's stochastic simulation algorithm [20, 23]. The Gillespie algorithm samples only a portion of the possible dynamic pathways, and thus samples the probability distribution given by the master equation. There are also approximations to Gillespie's algorithm, such as time-leaping [22] and system partitioning methods [55]. However, these Monte Carlo based methods rely on the computation of large numbers of pathways to estimate certain statistical properties. Though this may seem advantageous as one does not need to solve stiff ODEs, it becomes a disadvantage when the system is too large to explore all connections. Also, because these methods cannot provide gradients with respect to parameters, the parametric sensitivities need to be approximated using finite differences for Monte Carlo based methods [24, 56], which can be

a problem when the model structure can change due to a change in the model parameters [69]. The existing optimization algorithms for Monte Carlo based methods are mostly limited to black-box strategies, such as Nelder-Mead, Divided Rectangles, multi-level coordinate search, differential evolution, and kriging techniques [13].

Lakerveld et al. [42] derived a dynamic model, involving master equations, to account explicitly for any kinetic traps in the course of a dynamic directed self-assembly process. The master equations describe a probability-based continuous-time Markov processes that capture the stochasticity of the system. Continuous-time models are essential for this system as discrete-time models would require very fine discretization to accurately model the stiffness of the system. Moreover, empirical discrete-time models require data from full simulations of the system, which are difficult to obtain as the system size increases. The master equation based model accounts for all possible configurations of the self-assembled nanoparticles and simulates the probabilities of observing the system in each specific configuration as a function of time, for given values of the control parameters. It also produces sensitivities of the dynamic profiles with respect to the values of the parameters, which play a very important role in computing optimal control policies. Although the favorable computational complexity of their method allows for the dynamic simulation of systems with fairly large numbers of configurations, it inevitably reaches its limits, as the number of configurations increases. Model-reduction techniques are needed to proceed.

An alternate model for a continuous-time Markov process is via the Langevin equations [21], which are stochastic differential equations with real-valued states [27]. There are various model reduction techniques used for Langevin equations, such as dividing the equations into fast and slow dynamics [10], and the method of adiabatic elimination [68].

Reduction of models involving master equations has received considerable attention. The potential energy surface (PES) of the system can be simplified by aggregating configurations into shared basins of attraction, which in turn can form the basis for a reduced set of master equations, describing only the rates of rare events within a time scale of interest [4, 2]. However, for optimal control problems, separation of time

scales is complicated since the features of the PES are a function of the unknown, as yet, optimal control policies. Other methods for model reduction include wavelets [28, 30, 29], decomposition of ergodicity [63, 64], diffusion maps [9, 41], and finite state projection (FSP) [47, 48, 53]. In the case of FSP methods, only those states which make a significant contribution to the dynamics of the system, known as the state projection space, are simulated. An FSP-based approach provides a bound on the maximum error that is being generated from the reduction of the state space, as a function of time. The advantage of using an FSP-based approach is that the mathematical structure of the model is preserved after model reduction, which permits exploitation of the favorable computational complexity, demonstrated by Lakerveld et al. [42].

1.4 Thesis Overview

The thesis has six chapters. In Chapter 2, dynamic modeling of self-assembled nanoparticles through master equations is discussed. The model consists of ODEs for the probabilities of the system being in any configuration. The propensity rates for the system to go from one configuration to another configuration to which it is connected are obtained using the theory of transition states. In addition to that, the Finite State Projection method for solving large system of master equations is also explained, along with its shortcomings.

In Chapter 3, a simulation strategy for the master equations is developed. This strategy uses the FSP method and makes it adaptive in order to tackle the need for different reduced models along a trajectory and the stiffness of the ODEs. This is followed by a case study that illustrates deployment of the adaptive FSP algorithm and demonstrates the benefits from its use in dynamic simulations of complex self-assembled systems.

In Chapter 4, the multi-resolution approach of decomposing the system is proposed, where the external charges are moved in space to enable this strategy. A case study demonstrates the multi-resolution deployment of the adaptive FSP algorithm

within the scope of a control strategy for the manufacturing of desired nanostructures.

In Chapter 5, the theoretical generation and numerical implementation of optimal control strategies for self-assembled systems are presented. This strategy combines the adaptive FSP algorithm and multi-resolution approach to find an optimal control policy. This is demostrated through a case study.

In Chapter 6, nanoparticles with functionalized edges are used as building blocks for the nanostructures, where edge binding is enabled. A case study identifies the important factors that play a role in the rotational orientation of the nanoparticles.

Chapter 7 discusses the scope for future work in optimal control.

Chapter 2

Dynamic Modeling

2.1 Master Equations

The model for directed self-assembly is based on an Ising lattice model and the dynamics are described by a discrete-state continuous-time Markov process. The probability of observing the particles in a particular configuration α as a function of time is given by the master equation [70]:

$$\frac{dp_\alpha}{dt}(t) = -\sum_{\beta\neq\alpha} \epsilon_{\beta\leftarrow\alpha}(t)p_\alpha(t) + \sum_{\beta\neq\alpha} \epsilon_{\alpha\leftarrow\beta}(t)p_\beta(t), \quad \forall t \in (t_0, t_f], \quad p_\alpha(t_0) = p_{\alpha,0}, \tag{2.1}$$

where p_α is the probability of observing the system in configuration α and $\epsilon_{\beta\leftarrow\alpha}$ is a propensity factor, which is the rate of transitions from configuration α to β. Equation (2.1) can be written in matrix form as

$$\frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(t)\mathbf{p}(t), \quad \forall t \in (t_0, t_f], \quad \mathbf{p}(t_0) = \mathbf{p}_0, \tag{2.2}$$

where $\mathbf{p}$ is a vector with probabilities for each configuration and $\mathbf{A}(t)$ is a propensity matrix with elements

$$x_{\alpha\beta}(t) = \epsilon_{\alpha\leftarrow\beta}(t) \quad \text{if} \quad \alpha \neq \beta, \quad x_{\alpha\alpha} = -\sum_{\beta\neq\alpha} \epsilon_{\beta\leftarrow\alpha}(t). \tag{2.3}$$

A propensity factor $\epsilon_{\beta\leftarrow\alpha}$ is obtained via transition-state-theory from the difference in potential energy between configuration α and β and normalizing it with an appropriate reciprocal time constant:

$$\epsilon_{\alpha\leftarrow\beta} = \exp\left(\frac{-\Delta E_{B,\alpha\leftarrow\beta}(t)}{k_B T(t)}\right), \tag{2.4}$$

where $\Delta E_{B,\alpha\leftarrow\beta}$ represents the height of the potential energy barrier encountered when the system moves from configuration β to α. The potential energy wells around the configurations α and β are assumed to be parabolic with the following form:

$$\begin{aligned} E_\beta(t,\xi) &= k_F\xi^2/2 + E_{0,\beta}(t), \\ E_\alpha(t,\xi) &= k_F(\xi - d)^2/2 + E_{0,\alpha}(t), \end{aligned} \tag{2.5}$$

where k_F is a force constant, ξ is the surface diffusion coordinate, d is the lattice constant, and E_0 is the minimum of the energy well. The potential energy barrier is obtained as follows:

$$\Delta E_{B,\alpha\leftarrow\beta} = E_{A,\alpha\leftrightarrow\beta}(t) - E_{0,\beta}(t), \tag{2.6}$$

where $E_{A,\alpha\leftrightarrow\beta}$ is the energy at the intersection of the two energy wells,

$$\xi^* \in (0,d) \quad : \quad E_{A,\alpha\leftrightarrow\beta}(t) = E_\beta(t,\xi^*) = E_\alpha(t,\xi^*). \tag{2.7}$$

The Hamiltonian of the system is constructed assuming only pair-wise interactions between the particles and interactions between the particles and the external controls. The pair-wise interactions between the particles are calculated by the long-range repulsive electrostatic forces and short-range attractive forces such as Van der Waals. The external potential energy field is caused by the electrstatic point charges (or controls) on the grid:

$$\mathcal{H}_\alpha = \sum_{i=1}^{V}\sum_{k=1}^{N_{cp}} \frac{q_k q_p}{|r_{i,k}|} z_{\alpha,i} + \sum_{i=1}^{V}\sum_{j\neq i}^{V} z_{\alpha,i}\left(\frac{q_p^2}{|r_{i,j}|} - \frac{a}{|r_{i,j}|^6}\right) z_{\alpha,j}, \tag{2.8}$$

where α indicates a particular configuration, N_{cp} is the number of external controls, q_k is the strength of external control k, q_p is the strength of the coloumb charge of each nanoparticle, a is a parameter which lumps the effects of the close-range interactions, $z_{\alpha,i}$ is a binary variable which takes values 0 or 1 based on the abscence or presence of a particle in cell i of the configuration α, $|r_{i,j}|$ is the distance between nanoparticles i and j, and $|r_{i,k}|$ is the distance from the centre of cell i to point charge k. The model is a phenomenological one, which captures the inherent randomness in the movement of the nanoparticles, and can be extended to more complicated cases.

The structure of the model and an algorithm to simulate large instances of this model are described in more detail by Lakerveld et al. [42]. The algorithm used by Lakerveld et al. consists of two steps. In the first step, the structure of the matrix **A** is determined by looking at all the possible configurations and their connections. In the second step, the variable-coefficient ODE solver DVODPK [6] is used to solve this large system of sparse and stiff ODEs. DVODPK uses the preconditioned Krylov method based GMRES [59] to solve the linear system. This methodology exhibits a favorable computational complexity for the master equation model.

2.1.1 Finite State Projection

In Chapter 1, various methods for solving master equations are discussed. One of the possible methods discussed is the Finite State Projection (FSP) method. The advantage of using a FSP-based method is that the mathematical structure of the model, explained in Section 2.1 is preserved and hence, the methodology used by Lakerveld at al. to solve the ODEs remain valid.

Munsky et al. [47] had developed the Finite State Projection algorithm (FSP) to solve a large system of master equation. The finite state projection method uses the fact that only a certain subset of equations (called the projection space), which correspond to configurations in this case, are enough to find the dynamic profiles for the probabilities. This projection space is formulated in the same manner as the original space, i.e. using the master equation. It was shown by Munsky et al. [47] that for a suitable subset of configurations, the sum of the probabilities of the

configurations lie within an acceptable tolerance. If the original system is of the form

$$\dot{\mathbf{p}}(\mathbf{X};t) = \mathbf{A}\mathbf{p}(\mathbf{X};t), \tag{2.9}$$

then it can be written as

$$\begin{bmatrix} \dot{\mathbf{p}}(\mathbf{X}_R;t) \\ \dot{\mathbf{p}}(\mathbf{X}_{R'};t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_R & \mathbf{A}_{RR'} \\ \mathbf{A}_{R'R} & \mathbf{A}_{R'} \end{bmatrix} \begin{bmatrix} \mathbf{p}(\mathbf{X}_R;t) \\ \mathbf{p}(\mathbf{X}_{R'};t) \end{bmatrix}, \tag{2.10}$$

where the subscript R denotes the configurations in the projection space and R' denotes the remaining configurations. Here the configuration space has been divided into R and R'. The solution to the above equation can be approximated as:

$$\mathbf{p}(\mathbf{X}_R;t_f) = \exp(\mathbf{A}_R t_f)\mathbf{p}(\mathbf{X}_R;0). \tag{2.11}$$

This solution satisfies the equation

$$\mathbf{1}^{\mathrm{T}} \exp(\mathbf{A}_R t_f)\mathbf{p}(\mathbf{X}_R;0) \geq 1-\epsilon, \tag{2.12}$$

for $\epsilon > 0$. $\mathbf{1}$ indicates a column vector of ones. For a particular value of ϵ, the configuration in the projection space are chosen such that Equation (2.12) is satisfied.

The configurations in R are selected based on their reachability from the initial configuration. The term reachability needs to be quantified, and this will be done for our case in Chapter 3. The probability of the configurations in the unreachable space (called truncated configurations) are assumed to be zero as they are small enough to be neglected. This generates an error in the reduced model prediction which grows as the system is simulated. This is because in the projection matrix $\mathbf{A}_R$, all the outgoing connections to the truncated configurations are included, i.e. the diagonal terms, but the incoming connections from truncated configurations to the projection space are not included. This is illustrated in Figure 2-1, which has been reproduced from the paper by Munsky et al. [47]. Therefore, there is a continuous leakage of probability

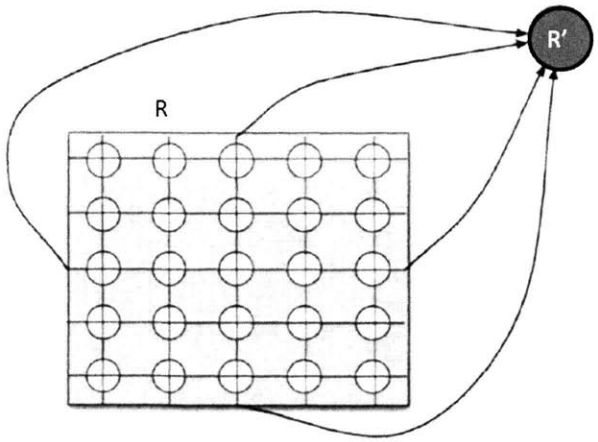

Figure 2-1: Figure showing the projection space of the large system to a subset of states in the Finite State Projection method [47]. In the FSP method, only the connections from projection space to the unreachable space are retained. This causes the sum of the probabilities of the configurations in the projection to deviate from one.

from the projection space. However Munsky et al. [47] proved two theorems: (a) as the size of the projection space increases, the error in the solution, which is the deviation of the sum of the probabilities of configurations in the projection space from 1, monotonically decreases and (b) the approximate solution can be brought arbitrarily close to the true solution based on the number of configurations in the projected space. These theorems help keep a check on the error. This technique forms a strong foundation on which model reduction of large systems of master equations can be based.

In their next paper, Munsky et al. [48] showed how the projection space can be expanded as time progresses and hence new configurations were included as they became important with time. The authors divided the time horizon into equal intervals (time steps) and fix a bound on the error for each time step based on the number of time steps and the total allowable error. For each time step, the projection space is chosen by iterating until the error requirement is met. The projection space is simulated multiple times in each iteration and every choice of the projection space is expanded based on the direction of the leakage of the probability measure. The

theorems in [47] are used to justify the method.

Although the favorable computational complexity of Lakerveld et al.'s method [42] allows for the dynamic simulation of systems with fairly large numbers of configurations, it inevitably reaches its limits, as the number of configurations increases. Model-reduction techniques are needed to proceed. In this thesis, we will focus on using the Finite State Projection method to solve the large system of ODEs modeled in Section 2.1. In the interval FSP method developed by Munsky et al. [48], the simulations are repeated multiple times and the projection space is always expanded. In the next Chapter, the Adaptive Finite State Projection (AFSP) method is proposed to overcome these shortcomings, and in the process, a reduced system of ODEs in the self-assembly of nanostructures is successfully simulated.

Chapter 3

Simulation through Adaptive Finite State Projection

In the previous chapter, the master equation based model for simulation of directed self-assembly was discussed. The method of Finite State Projection was also explored as a viable option to solve a large system of master equations. An FSP-based algorithm has two important properties that are essential for our purposes:

(i) As we will show in this thesis, it can be made to be adaptive, by changing the projection space in time.

(ii) The logic of an adaptive FSP-based algorithm aligns well with the multi-resolution approach developed by Solis et al. [64], which uses a systematic decomposition of the system's ergodicity.

It should be noted though that dynamic models for self-assembly are known to be stiff, which could call for fast necessary changes in the projection space in only a small fraction of the total simulated time, when using FSP. Furthermore, in order to minimize total CPU time, one would like to anticipate when the maximum error, resulting from model reduction, grows too large, rather than taking a posteriori corrective action, through iterations backwards in time. In this regard, the combination of an FSP-based algorithm and event detection [52] is of significant potential to improve algorithms that simulate dynamic directed self-assembly processes.

Consequently, an adaptive FSP-based algorithm is well suited for the needs of our problem and the key questions that need to be answered are:

(a) When should the projection space be adapted, and what is the logic for its adaptation?

(b) How should the projection space be adapted, i.e., what configurations should exit the projection space, and what configurations should enter?

The answer to these two questions forms the essential scope and content of this Chapter with the following elements:

1. The dynamic evolution of directed self-assembly of nanoparticles is modeled by master equations, describing the dynamic evolution of the probability of each configuration in projection space, i.e., the space of selected configurations.

2. A variable-step ODE solver is used to integrate the master equations in time.

3. An event detection logic is employed to signal the need for adaptation of the projection space. This event-detection logic is integrated with the adaptive ODE solver.

4. The event detection logic employs an algorithm, which computes an upper bound on the error that is being generated, as a function of time, simultaneously with the probability of all configurations. Once this upper bound on the error crosses a certain pre-specified threshold, the algorithm will precisely locate the point in time at which the threshold was crossed and, subsequently, adjust the projection space to reduce the rate at which the upper bound on the error is increasing so that the threshhold is not crossed.

The resulting methodology modifies the projection space only when necessary, maintains control over the error in dynamic simulation, and allows the dynamic simulation of self-assembled nanoparticles at various spatial scales (resolutions), as will be described in subsequent sections of this Chapter.

3.1 Modified Model

Simulation of the system via FSP will result in a reduced model given by

$$\frac{dp_\alpha}{dt}(t) = -\sum_{\beta\in A(t):\beta\neq\alpha} \epsilon_{\beta\leftarrow\alpha}(t)p_\alpha(t) + \sum_{\beta\in A(t):\beta\neq\alpha} \epsilon_{\alpha\leftarrow\beta}(t)p_\beta(t), \quad \forall\alpha\in A(t),$$
$$p_\alpha(t_0) = p_{\alpha,0}, \quad \forall\alpha\in A(t_0). \tag{3.1}$$

This reduced model will inevitably result in an error compared to a simulation of the full system, because the probability of observing unlikely configurations will be neglected. An upper bound on the total error as a result of model reduction is given by [47]

$$\varepsilon(t) = 1 - p_{\mathrm{T}}(t), \tag{3.2}$$

where p_{T} is the sum of the probabilities of all the configurations in the projection space $A(t)$, described by

$$\frac{d}{dt}p_{\mathrm{T}} = \sum_{\alpha\in A(t)} \frac{dp_\alpha(t)}{dt}, \quad \forall t\in(t_0,t_f], \quad p_{\mathrm{T}}(t_0) = \sum_{\alpha\in A(t_0)} p_{\alpha,0}. \tag{3.3}$$

The projection space $A(t)$ contains the likely configurations of the system at time t. Note that this reduced model contains the connections that result in the out flux of probability from the projection space to the rest of the configuration space, but does not contain any connections that result in the influx of probability to the projection space. Therefore, detailed balance is not satisfied by the FSP model. There is no "steady state" in the reduced model and the error keeps increasing. In the next section, an adaptive algorithm will be presented to modify the projection space $A(t)$ during the simulation of directed self-assembly, whenever $\varepsilon(t)$ reaches an undesirable value. The aim is to include in the projection space the most relevant configurations (i.e., configurations of high probability), and thus approximate the dynamic behavior of the whole system with the lowest possible error, while avoiding the need to predefine suitable time-points at which the projection space should be modified; a task that is very difficult to complete a priori for directed self-assembly.

3.2 Algorithm

The key challenge for the design of an adaptive FSP-based algorithm is to define rules that identify the suitable time points during a simulation, at which points the structure of the projection space $A(t)$ should be changed. A trade-off exists between frequent changes in projection space to lower $\varepsilon(t)$ and avoiding a large computational load associated with each reformulation of Equation (3.1), whenever the projection space has been changed. Therefore, time varying maximum and minimum allowed values for $\varepsilon(t)$ are introduced:

$$\varepsilon_{MIN}(t) \leq \varepsilon(t) \leq \varepsilon_{MAX}(t), \quad \varepsilon_{MIN}(t) < \varepsilon_{MAX}(t). \tag{3.4}$$

During a simulation, the upper bound on the error generated will be monitored and an algorithm will be designed such that Equation (3.4) holds. A minimum value for the error generated allows for configurations, which can be truncated without sacrificing significant accuracy, to be removed from the projection space to reduce computational load. Furthermore, a maximum value for the upper bound on the error generated ensures that relevant configurations will be added to the projection space to achieve sufficient accuracy of the reduced model. The points in time at which the error threshholds described by Equation (3.4) are crossed have to be identified and rules have to be defined to change the projection space at those times. The former requirement is addressed by implementation of an event detection scheme [52] and the latter requirement is addressed by implementation of the following rules:

1. Configuration α will be removed from the projection space $A(t)$ if and only if the following two conditions are met: 1) $p_\alpha(t) < p_{trunc}$ and, 2) $\frac{dp_\alpha}{dt}(t) < f_{trunc}$. These two conditions will assure that only configurations with a low probability and low increase in probability will be removed.

2. Candidate configuration β will be added to the projection space $A(t)$ if and only if $\sum_{\alpha \in A(t)} \epsilon_{\beta \leftarrow \alpha}(t) p_\alpha(t) > f_{add}$, where α refers to all configurations that are part of the projection space and β refers to a single candidate configuration. This

condition will add those configurations to the projection space that are likely to be important in the near future.

The resulting mechanism for addition and removal of configurations is schematically illustrated in Figure 5-1.

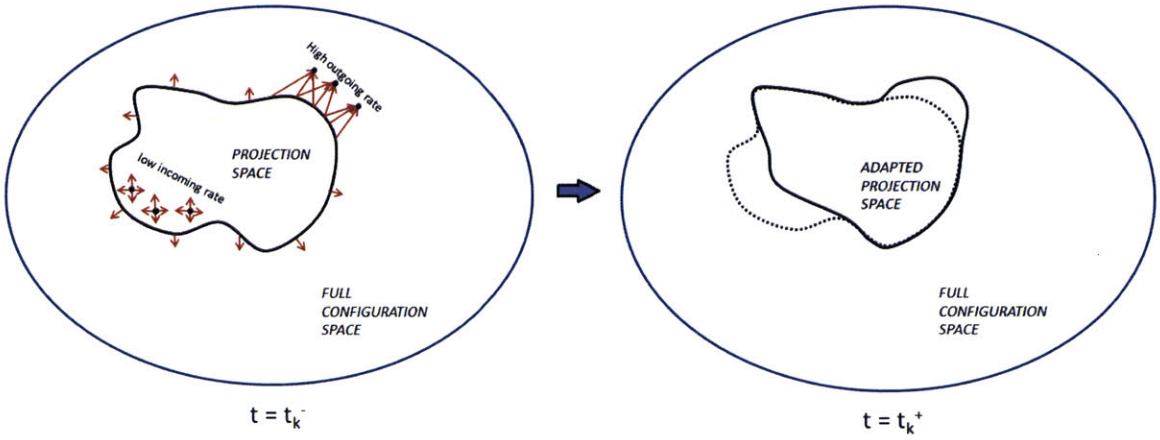

Figure 3-1: Illustration of the criteria for projection space adjustment in the proposed Adaptive Finite State Projection method. Configurations outside the projection space receiving high transition rates will be added in the next time step. Similarly, configurations inside the projection space with low probability value and transition rates are removed at the projection space adjustment event.

The application of the proposed algorithm is illustrated in Figure 3-2. The maximum and minimum allowed values for the upper bound on the error generated ($\varepsilon_{MAX}(t)$, $\varepsilon_{MIN}(t)$ respectively) are chosen to increase linearly with time so that an error budget at the final time is not exceeded. In this case, the initial probability distribution is assumed to be known and completely included in the projection space. As time progresses, the upper bound on the error generated by the simulation increases until it surpasses the imposed maximum value $\varepsilon_{MAX}(t)$. Event detection [52] will locate the point A where $\varepsilon(t) = \varepsilon_{MAX}(t)$. Subsequently, configurations will be added to the projection to reduce the rate at which the upper bound on the error generated increases. After re-initialization of the system, the simulation of directed self-assembly proceeds until, in this case, the minimum allowed value is crossed. This indicates that configurations may be truncated to reduce computational load while still achieving an acceptable accuracy. Again, an event detection scheme precisely

locates the crossing (point B in Figure 3-2). The truncation of configurations will result instantaneously in an increased upper bound on the error (point C) due to the non-zero value of the probabilities of the configurations removed from the projection space. Again, after re-initialization, the simulation proceeds until the upper bound on the error hits the limits again (point D in the figure), and the process is repeated. The simulation provides the probability of observing self-assembled structures and a bound on the error resulting from model reduction, which is readily achieved without the need of repeating (part of the) the simulation or an arbitrary *a priori* discretization of the time interval.

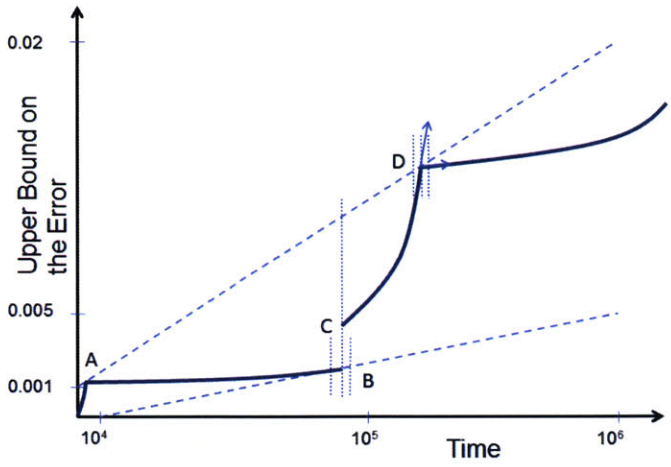

Figure 3-2: Example of simulation of directed self-assembly using an adaptive algorithm for finite state projection and event detection. The upper dashed line represents the maximum value allowed for the upper bound on the error resulting from model reduction ($\varepsilon_{MAX}(t)$) and the lower dashed line represents the minimum value allowed for the upper bound on the error ($\varepsilon_{MIN}(t)$). The solid line represents the upper bound on the error during simulation of a case study of directed self-assembly. The points A, B, and D are determined by event detection and indicate those points where the projection space is adjusted. The point C is the upper bound on the error after re-initialization of the model with the reduced projection space compared to point B.

The full algorithm proceeds as follows:

Step 0. Set $k = 1$ and set $t_k = 0$. Choose an initial projection space $A(0)$ that incorporates the initial probability distribution for the FSP.

Step 1. Solve Equation (2.2) for one step with a variable-step ODE solver. Let that

time interval be $(t_k, t_{k+1}]$.

Step 2. Compute $\varepsilon(t_{k+1}) = 1 - \sum_\alpha p_\alpha(t_{k+1})$.

Step 3. If $\varepsilon(t_{k+1}) > \varepsilon_{MAX}(t_{k+1})$

(a) Find the coefficients of the polynomial fit of the data points computed by the ODE solver.

(b) Use Newton's method to locate the time t'_{k+1} of the crossing of the allowable limits on the error between t_k and t_{k+1} via interpolation.

(c) If $\sum_{\alpha \in A(t'_{k+1})} \epsilon_{\beta \leftarrow \alpha}(t'_{k+1}) p_\alpha(t'_{k+1}) > f_{add}$, then add configuration β. Repeat this procedure for all candidate configurations. A candidate configuration β is obtained by exploring all possible one-step transitions that lead to a new configuration outside the projection space.

(d) Augment $A(t'_{k+1})$ with the configurations added under step 3(c).

(e) Calculate $\frac{d\varepsilon}{dt}(t'_{k+1}) = 1 - \sum_{\alpha \in A(t'_{k+1})} \frac{dp_\alpha}{dt}(t'_{k+1})$.

(f) If $\frac{d\varepsilon}{dt}(t'_{k+1}) > \frac{d\varepsilon_{MAX}}{dt}(t'_{k+1})$, reduce f_{add} and go back to step 3(c).

Step 4. If $\varepsilon(t_{k+1}) < \varepsilon_{MIN}(t_{k+1})$

(a) Find the polynomial coefficients of the BDF method used by the ODE solver.

(b) Use Newton's method to locate the time t'_{k+1} of the crossing of the allowable limits on the error between t_k and t_{k+1} via interpolation.

(c) For all configurations in the projection space $A(t'_{k+1})$: truncate configuration α if $p_\alpha(t'_{k+1}) < p_{trunc}$ and $\frac{dp_\alpha}{dt}(t'_{k+1}) < f_{trunc}$.

(d) If $\sum_{\alpha \in A(t'_{k+1})} \epsilon_{\beta \leftarrow \alpha}(t'_{k+1}) p_\alpha(t'_{k+1}) > f_{add}$, then add configuration β. Repeat this procedure for all candidate configurations. A candidate configuration β is obtained by exploring all possible one-step transitions that lead to a new configuration outside the projection space.

(e) Delete from $A(t'_{k+1})$ all configurations identified under step 3(c) and augment $A(t'_{k+1})$ with the configurations added under step 3(d).

(f) Calculate $\varepsilon(t'_{k+1}) = 1 - \sum_{\alpha \in A(t'_{k+1})} p_\alpha(t'_{k+1})$.

(g) If ε is not within the error limits, go to (c) after adjusting p_{trunc}, f_{trunc} and f_{add}.

Step 5. Set $t_k = t'_{k+1}$ and go to step 1 if $t_k < t_f$.

This algorithm is different from the one by Park et al. [52]. In this algorithm, intermediate crossings of the allowable error limits after t_k are ignored if the upper bound on the error is within the error limits at t_{k+1}. This is because the algorithm aims to keep the upper bound on the error within the error budget only at the final time. Also, the final adjusted values of f_{add}, p_{trunc} and f_{trunc} depend on the error limits, and are determined automatically by the algorithm. The number of iterations required to include all the important configurations depends on the initial values of f_{add}, p_{trunc} and f_{trunc}.

An important point to note is that the inclusion of any configuration depends on the time constant of dynamic change in its probability value. A missing configuration that achieves a significant increase in its probability values within the time horizon of simulation will cause the upper bound on the error to increase rapidly, thus hitting the maximum allowed value and this will result in the addition of that missing configuration. Hence, no important configurations can be missed in the proposed method.

Figure 3-3 illustrates schematically how the proposed algorithm responds to typical features of the potential energy surface over which the dynamics of self-assembly evolve. Suppose a system is initially trapped in a metastable state and the projection space of the reduced model involves configurations that have a common point of attraction (i.e., point A in Figure 3-3). The error as a result of model reduction in that case is caused by all transitions towards configurations that are not part of the projection space. Depending on the time interval of interest and the desired accuracy of the simulation, sufficient configurations may be added to include transition

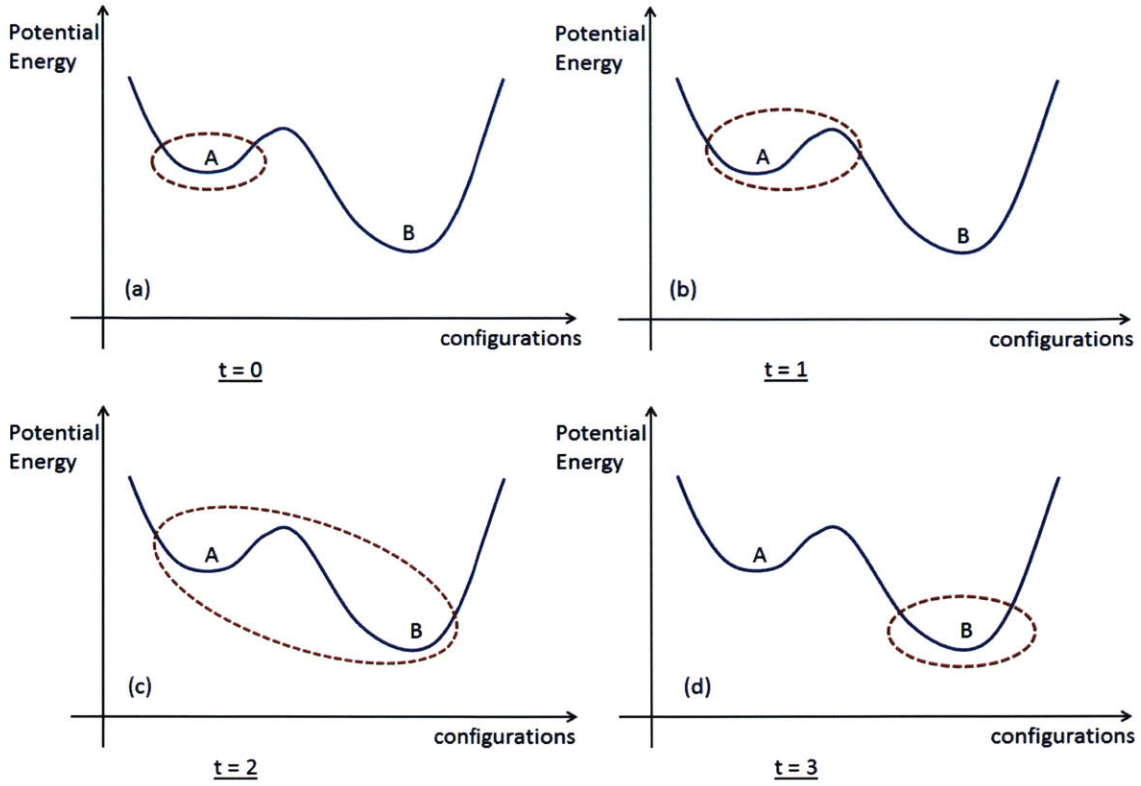

Figure 3-3: Schematic illustration of the proposed method, where the projection space (red dotted line) initially contains the configurations in (meta)stable state A. Eventually due to the transitions towards other (meta)stable states, the projection space expands to include the transition configurations and the stable configurations. However, as the probability evolves, the (meta)stable states lower their probability values and the projection space has only the configurations of B.

configurations between metastable state A to stable state B (i.e., Figure 3-3b,c). As the simulation proceeds in time, the probability of observing the system in a configuration with common attractor A will decrease and the algorithm will remove the configurations that enable the transition from A to B. Note that at this point the simulated system will not be ergodic anymore. In the next section, a case study will be presented in which the features of the potential energy surface are changed dynamically.

3.3 Case Study

Dynamic simulation through Adaptive Finite State Projection

The system under study consists of a 2D square domain with 36 lattice cells and 6 negatively charged nanoparticles. Self-assembly is driven by interactions between particles and interactions of the particles with an external electrostatic field shaped by various point charges. The total number of potential configurations for this system is 1,947,792. This total number of potential configurations is calculated by considering all possible ways in which 6 nanoparticles can be distributed on 36 lattice cells. The aim of the proposed algorithm is to simulate only a portion of those 1,947,792 configurations while maintaining sufficient accuracy in the dynamic evolution of the self-assembled system. The point charges are modified over time in such a way that the nanoparticles are directed from the top-right corner of the domain and eventually self-assemble with high probability into a structure with the nanoparticles in the bottom-left corner of the domain. During this transition, the majority of possible configurations will be kinetically restricted. The AFSP algorithm for simulation of this process will automatically take advantage of these kinetic restrictions, by modifying the projection space in time during the simulation.

The conditions for the simulation of the case study are given in Table 3.1. Note that a high value of (q_k/q_p) was used, which ensures that the electrostatic interactions with the external field are dominant.

Table 3.1: Parameter values for the simulation in the case study. N_t is the number of configurations in the projection space.

Strength of external charges $((q_k)/(q_p))$	+50 or -50
$k_C q_p^2$ [kcal mol^{-1}nm]	1
$k_B T$ [kcal mol^{-1}]	0.7
a [kcal mol^{-1}nm^6]	10^5
f_{add} or f_{trunc} [ν]	$\frac{P_t(t)}{N_t(t)} \times 10^{-3}$
p_{trunc}	$\frac{P_t(t)}{N_t(t)} \times 10^{-3}$

The parameter values of f_{add}, p_{trunc} and f_{trunc} are provided in Table 3.1 as well. For each iteration in Steps 3 and 4 of the algorithm (see the section on the algorithm of AFSP), the values of f_{add}, p_{trunc} and f_{trunc} were reduced or increased by 50%, as required by the algorithm. Reasonable initial estimates were chosen for these parameters.

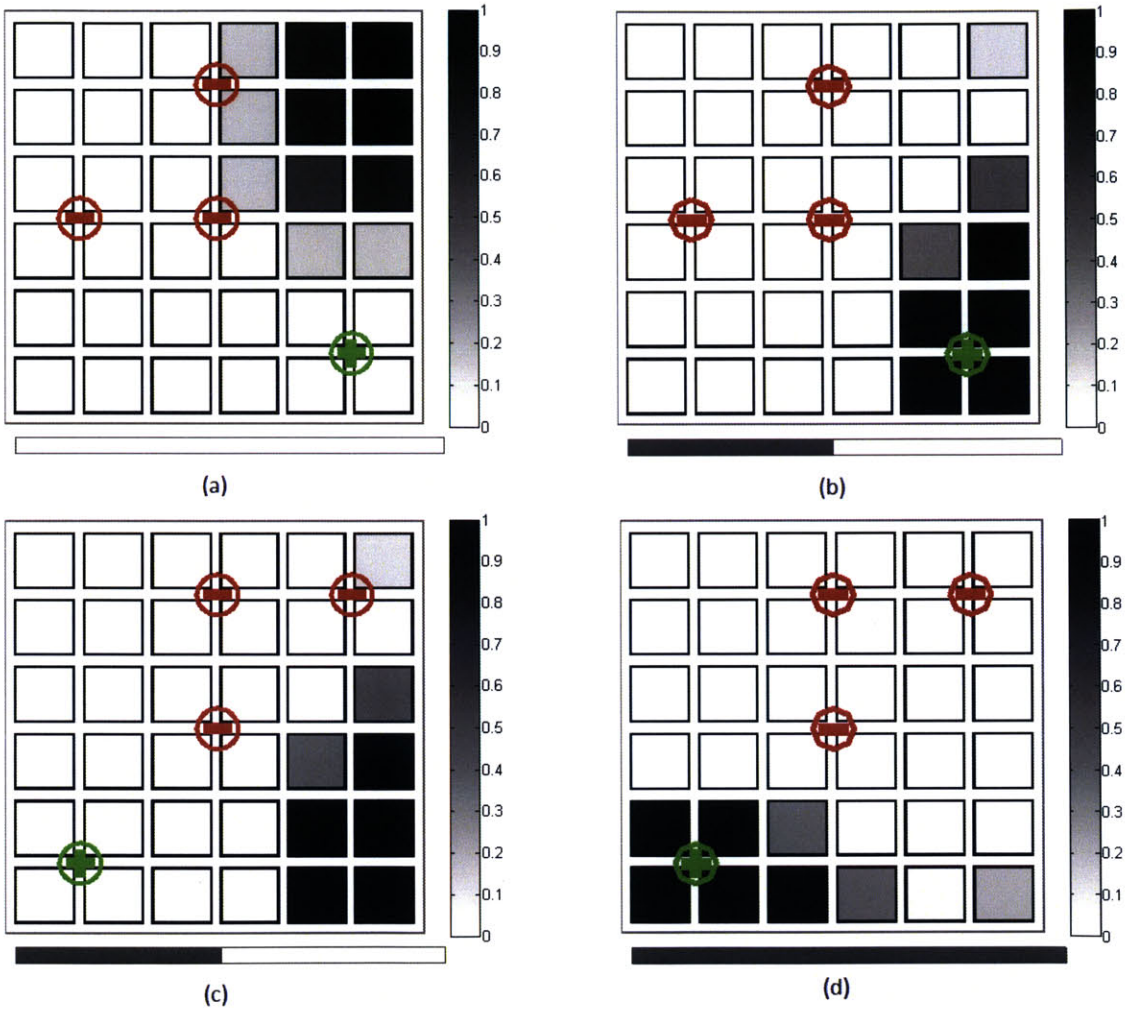

Figure 3-4: Snapshot of the dynamic evolution of self-assembly in the simulation at various times. The rectangle at the bottom of each grid is the progress bar which indicates the simulation time. The rectangle at the right of the grids shows the scale measuring the expected number of nanoparticles in a particular cell. (a) shows the initial probability distribution in each grid cell along with the external charges used. After $t = 2.8 \times 10^6 \nu^{-1}$, the probability distribution is as shown in (b). At that instant, the external charges are changed as shown in (c). (d) shows the final probability distribution of the nanoparticles at $t = 6.1 \times 10^6 \nu^{-1}$.

The initial probability distribution of the studied case, for illustrative purposes, consists of 6 configurations with equal probability, while the probability of observing the system in any other configuration is initially zero. The expected number of particles per lattice cell corresponding to the initial distribution is illustrated in Figure 3-4(a). The gray scale to the right of the physical domain indicates the expected number of particles that will occupy a cell in the domain. This number varies from 0 to 1, where 0 (white) indicates a particle will not occupy that grid cell and 1 (black) indicates a particle will occupy that grid cell with 100% probability. The expected number of particles in each cell is calculated by the sum of the probabilities of all the configurations that have a particle in that cell. Furthermore, Figure 3-4(a) illustrates the initial arrangement of the external charges. Note that this arrangement of the external charges will direct the nanoparticles to the bottom-right part of the domain as illustrated in Figure 3-4(b). At $t = 1.0 \times 10^5 \nu^{-1}$, the strengths of the external charges are changed instantaneously as illustrated in Figure 3-4c, which in turn will direct the nanoparticles towards the bottom-left corner of the domain. At the end of the simulation, a steady state is approached in which self-assembled structures with nanoparticles in the bottom-left corner of the domain have a high probability (Figure 3-4(d)).

The dynamics of the directed self-assembly process, as described above, clearly indicate the need to change the projection space during the simulation. Suitable time points to change the projection space are informed by the upper bound on the error resulting from model reduction, as explained in the section on the algorithm of AFSP. Figure 3-5 illustrates the upper bound on the error as a function of time, including the chosen maximum and minimum values allowed. The initial projection space contains only those 6 configurations for which $p_\alpha(t_0) > 0$ holds. The upper bound on the error initially increases rapidly and hits the maximum value allowed after a short time interval. Event detection locates the point in time where the upper bound on the error hits the maximum value and, subsequently, additional configurations are added after which the simulation proceeds. Around $t = 0.9 \times 10^5 \nu^{-1}$ the minimum allowed value is crossed, which indicates that configurations may be removed from the

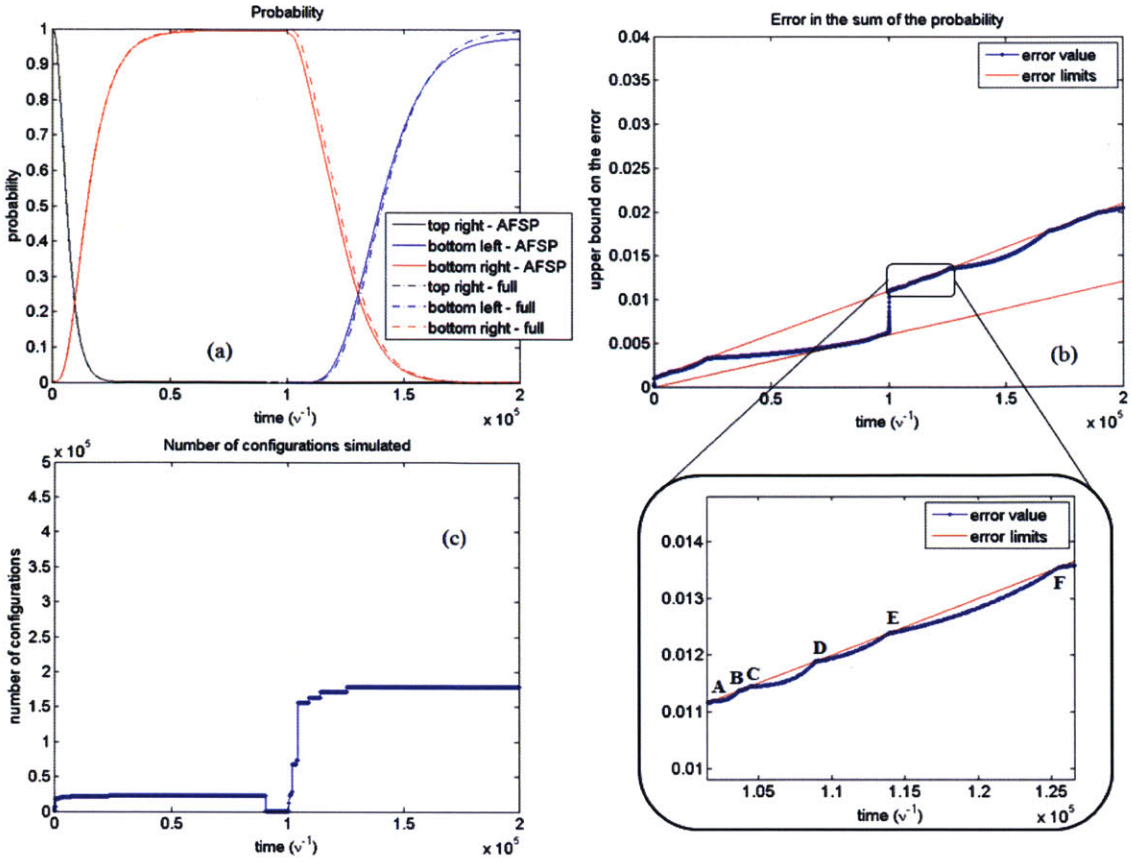

Figure 3-5: Graphs showing the performance of the proposed method for a case study. The probability of finding nanoparticles in various parts of the domain as a function of time is shown in (a). The upper bound on the error resulting from model reduction is shown in (b), where initially events are detected due to the upper bound on the error hitting the maximum value allowed. Reaching steady state results in event detection on the minimum value allowed. After sometime, the upper bound on the error increases rapidly due to the change in the arrangement of external charges and the same procedure as before is repeated. The size of the projection space is shown in (c), which shows that around 1% of the total configurations need to be simulated.

projection space. After the update of the projection space, only a small number (500) configurations are left, which are configurations with nanoparticles in the bottom-right part of the domain (Figure 3-4(c)). Simulation is continued until time $t = 1.0 \times 10^5 \nu^{-1}$. At this point, the external charges are changed instantaneously to direct the nanoparticles to the bottom-left part of the domain. The modified external field causes the upper bound on the error to increase rapidly with the prevalent projection space and the maximum value is crossed rapidly (Figure 3-5(b)). Again,

event detection locates the point of intersection and the projection space is modified subsequently to add new configurations based on the analysis of the propensity rates as described in the section on the AFSP algorithm. For the studied case, the sequence of event detection and expansion of the projection space is repeated until eventually, the added configurations reduce significantly the rate at which the upper bound on the error increases. This can be seen in the magnification of a small region of Figure 3-5(b), where A,B,...,F show the time points when the upper bound on the error crosses the maximum value repeatedly. Figure 3-5(a) shows that simulation of the reduced model is indeed a close approximation of a simulation of the full system with all possible configurations and transitions.

In summary, the presented case study demonstrates the effectiveness of the proposed algorithm to simulate directed self-assembly by using an adaptive version of FSP for model reduction with event detection. Automated identification of a changing and limited number of relevant configurations and transitions (approximately 1% compared to the full system) is sufficient to obtain a desired accuracy on the maximum error resulting from model reduction. In the presented case, the time scales of desired changes in the state projection vary significantly, which does not pose restrictions since the approach does not require a priori information on desired changes of the state projection. It is expected that tuning of the parameters that quantitatively determine which configurations to add or truncate (i.e., f_{add}, p_{trunc}, f_{trunc}) can further enhance the performance of the algorithm for the studied case. Finally, it is expected that the proposed algorithm is particularly effective for systems where the potential energy surface is strongly rugged, which is typical for directed self-assembly. Such systems can be much more complicated compared to the illustrated case. However, the automated nature of the algorithm allows such systems to be approached as well.

Chapter 4

The Multi-Resolution Approach

In Chaper 3, the dynamic modeling and simulation of directed self-assembly of nanoparticles was addressed. External point charges were used to move the nanoparticles and guide the self-assembly process. The dynamics of guided self-assembly were modeled by master equations, denoting the time-evolution of the probability of each configuration of nanoparticles. An adaptive version of the finite state projection approach [47, 48, 53] was developed, as an adaptive model reduction technique, and it was shown to be very effective in handling the combinatorial explosion in the number of configurations, as the size of the problem increased, by retaining a small number of configurations with significant levels of probabilistic presence. Event-detection logic was used to determine the points in time, when the projection space required modification (adaptation). The approach was shown to be especially effective in handling dynamic self-assembly systems with varying stiffness, since the time scales for the adaptation of the projection space perfectly conform to the varying time scale of the dynamic evolution of the system.

However, when the system size grows, the AFSP method may not be sufficient in reducing the system size. When optimal control is implemented for this process to find the optimal dynamic profile for the strengths of the externel charges, the number of potential locations for the external charges increases combinatorially. Since effectiveness of the AFSP method strongly depends on the locations of the external charges, the reduction in the system size may not be sufficient. For such a case, a

multi-resolution decomposition of the system, proposed by Solis et al. [64], allows us to produce tractable computational approaches.

The multi-resolution approach is used to tame the complexities arising during the dynamic simulation of self-assembly processes with large domains and many particles. In this approach, a desired nanostructure can be developed in phases, where at each phase the nanoparticles are arranged with progressively increasing resolution.

For example, in Figure 4-1, the self-assembly process starts with 32 nanoparticles which occupy any randomly selected initial configuration. In Phase 1, the external charges are positioned at appropriate points, and their intensities are selected in such a way as to maximize the probability of finding 17 and 15 nanoparticles in the top and bottom halves of the domain, respectively. Once this distribution has been achieved, transitions between the domains are restricted by strong repulsive external charges along their boundary. Subsequently, each of the domains of Phase 1 is treated independently in Phase 2. Each domain of Phase 2 is treated the same way as Phase 1 itself and further divided in Phase 3. This process continues until the final structure is reached. This is an illustrative example and the method can be extended to a system of any size.

There is no unique strategy for positioning the external charges and selecting their strengths, in order to achieve the progressive multi-resolution distribution of the particles in the domain, as shown in Figure 4-1. In the next Chapter, we will develop an optimal control policy, which achieves the multi-resolution objectives, by minimizing the required total time. For the dynamic simulation purposes of the present Chapter, in order to achieve the desired number of nanoparticles in each half, the locations of the external charges need to be suitably chosen and changed with time in such a way that the system does not get trapped in any kinetic traps. This will be demonstrated in the next section.

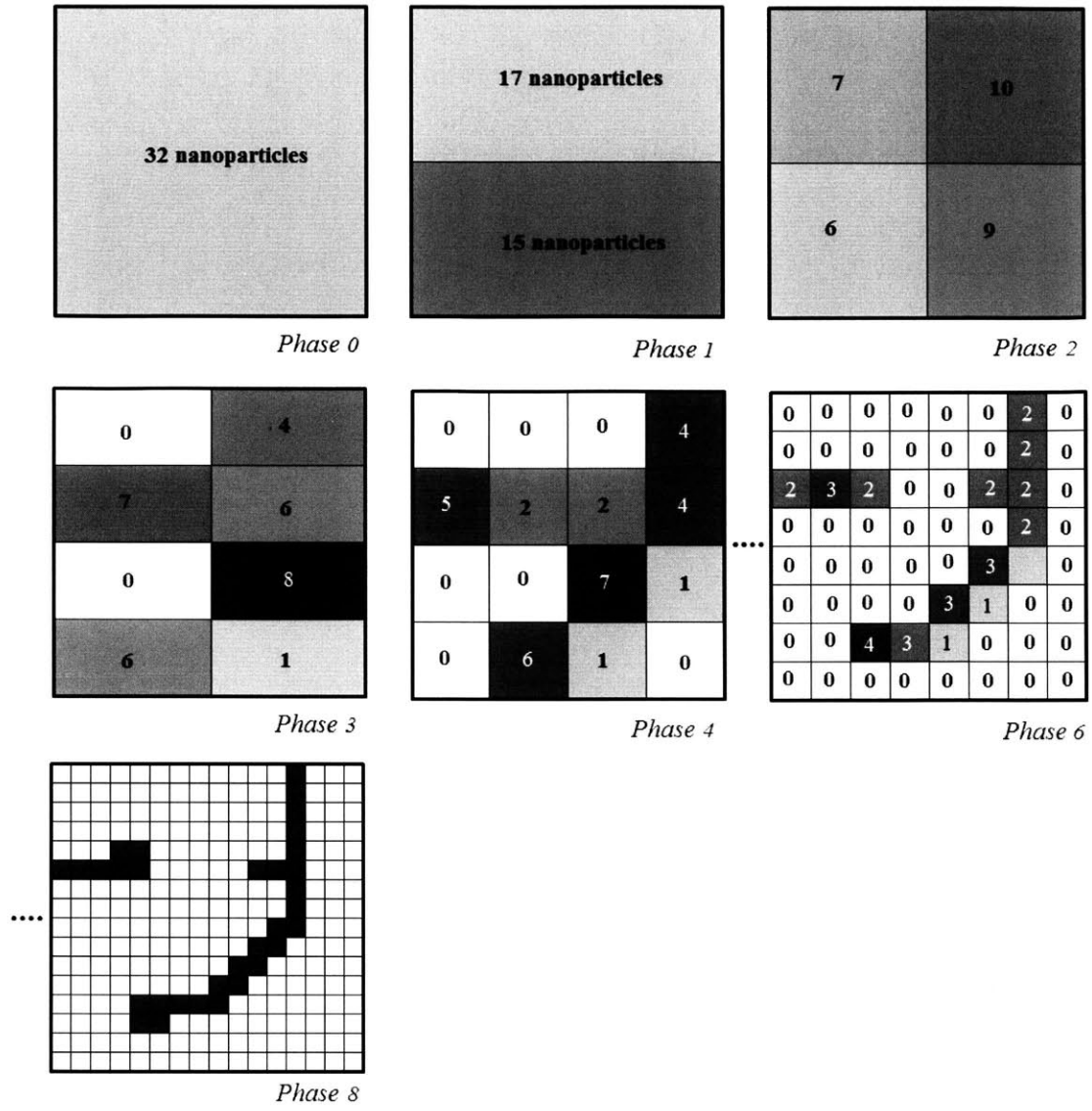

Figure 4-1: Illustration of the multi-resolution approach used for self-assembly of nanoparticles. At every phase, the domain is divided into halves and subsets of the nanoparticles are isolated in various parts of the domain. Self-assembly progressing through these multi-resolution phases enables simulation of large domains.

4.1 Designing the energy landscape for the multi-resolution decomposition of the state space

During the self-assembly process, the nanoparticles can be prevented from getting stuck in kinetic traps by suitably shaping the energy landscape of the system. Form-

ing large energy wells in the energy landscape close to the desired location of the nanoparticles causes the nanoparticles to move towards these wells. These nanoparticles can then be moved to different locations of the domain by slowly moving the energy wells. Moving these wells slowly with time forms the basis of the proposed strategy. The energy wells are moved by progressively modifying the locations and strengths of the external charges and hence attract the nanoparticles along with them. This strategy is applied in combination with the AFSP method.

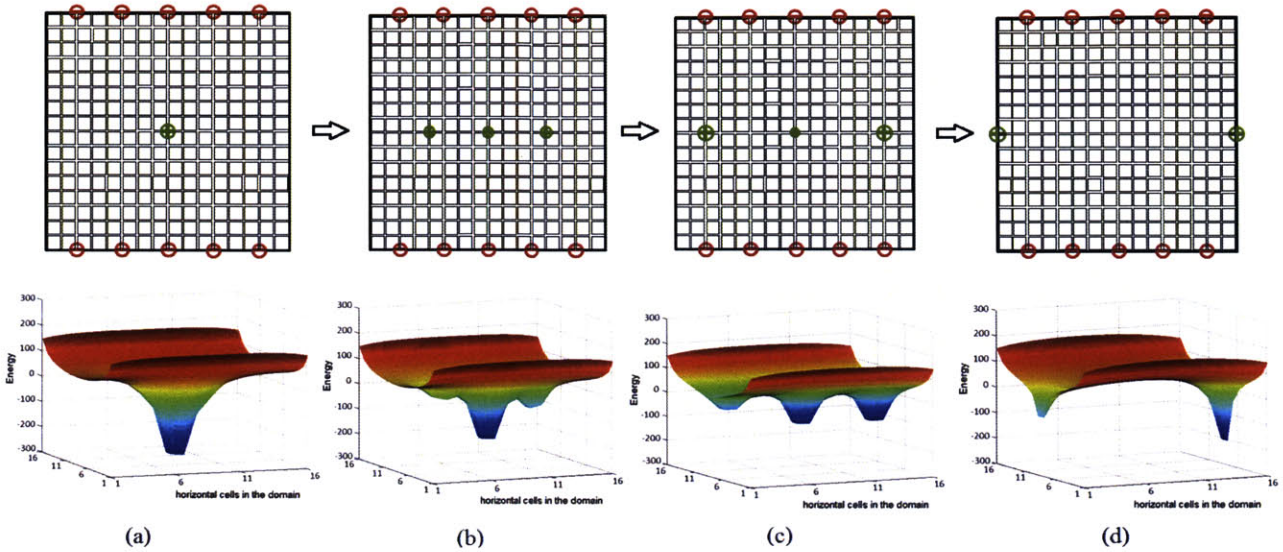

Figure 4-2: Illustation of the various steps of the proposed strategy for self-assembly in a particular step of the multi-resolution approach. Each arrangement of external charges corresponds to an energy landscape, which gives an insight into the path the nanoparticles will follow. The first step, shown in (a), has a strong attractive charge in the center that attracts the nanoparticles towards it. The subsequent steps, shown in (b), (c) and (d), two additional charges are introduced, on either side of the charge in the center, that slowly increases in strength, while the strength of the center decreases. The net result is a progressive weakening of the center well and strengthening of two new wells, which move towards the edges of the domain, pulling along the corresponding number of particles, thus creating two distinct and weakly interacting sets of particles.

The dynamic simulation of the system, to achieve the first phase of the multi-resolution approach, is started from an initial state where all the particles are near the center of the grid. This arrangement of nanoparticles is quite simple to achieve in practice - by placing a large attractive external charge in the center of the grid,

as shown in Figure 4-2(a). Once the probability of observing nanoparticles near the center of the grid is high, the arrangement of external charges is slightly modified. Two more external charges are added on either side of the charge in the center (left and right side in this case). These two external charges have a small attractive strength and the strength of the charge in the center is reduced by a small amount. This step creates two small energy wells on either side of the large energy well at the center (Figure 4-2(b)) and hence the probability of the desired separation of the nanoparticles will increase (for example, 17 particles above and 15 particle below in Phase 1 of Figure 4-1). After simulating the system for a fixed time, the two additional attractive external charges will be moved further away from the center, on either side of the charge in the center, with their strengths increased while the strength of the charge in the center is further reduced (Figure 4-2(c)). The simulation continues with this setup for a fixed time. This step causes the two energy wells on either side of the center well to deepen, with the energy well in the center becoming less deep, further increasing the probability of separation of the nanoparticles. This process will be continued until the additional external charges reach the edge of the grid, at which point, the arrangement of the external charges will be that given by the static solution (Minimum Tiling Approach) proposed by Solis et al.[63]. This solution is computed for a sample configuration of nanoparticles with the desired separation between them. The implementation of the static solution towards the end of the strategy will ensure that the probability of the desired separation of nanoparticles is quite high. The strengths of the external charges at the center will eventually go to zero or become negative (Figure 4-2(d)).

The above strategy of changing the locations and strengths of external charges is used at all phases of the multi-resolution approach. Charges located at the boundary may be used to enhance or inhibit movement of particles to the boundary of the domain. As the multi-resolution strategy decomposes the state space into non-ergodic subsets of configurations, the effectiveness of the AFSP-based algorithm, described in the section on the algorithm, increases significantly and makes feasible the dynamic simulation of guided self-assembly processes with large-sized domains and large num-

bers of nanoparticles.

In the system used here for illustrative purposes, there are no energy wells in the energy landscape other than the ones specifically formed by the attractive external charges. This is due to the fact that the external charges used have high strengths compared to the strengths of interactions between particles. Since kinetic trapping occurs when the number of nanoparticles trapped in a particular energy well is different than the desired, this can be avoided by suitably choosing the strengths of the external charges and monitoring the probability of the desired separation of the nanoparticles with time. During dynamic simulation, if the system is in a kinetic trap the rate of increase of the desired probability is very low, and the configurations in the projection space do not possess the desired final geometries. In such cases, the locations and intensities of the external charges must be changed in order to generate favorable energy landscapes that allow further evolution of the system towards the desired state. How to achieve this goal in an optimal manner, through the expanded capabilities of AFSP for the dynamic simulation of large systems, will be shown in Chapter 5 of this thesis.

4.2 Case Study

Control strategy for self-assembly

This case study consists of a 2D square domain with 64 lattice cells and 8 negatively charged nanoparticles. As in the previous case, self-assembly is driven by interactions between particles and interactions of the particles with an external electrostatic field shaped by the various point charges. The total number of configurations for this system is 4,426,165,368 and it's not practical to simulate the full system with the algorithm of Lakerveld et al. [42]. The aim of the proposed algorithm is to move the point charges and vary their strengths with time to facilitate self-assembly to a structure with desired geometry, while avoiding kinetic traps. Only a fraction of the 4,426,165,368 configurations need to be simulated. The control of the dynamic self-

assembly process towards the desired final configuration will be studied through the multi-resolution approach, described in the section on designing the energy landscape.

Phase 1

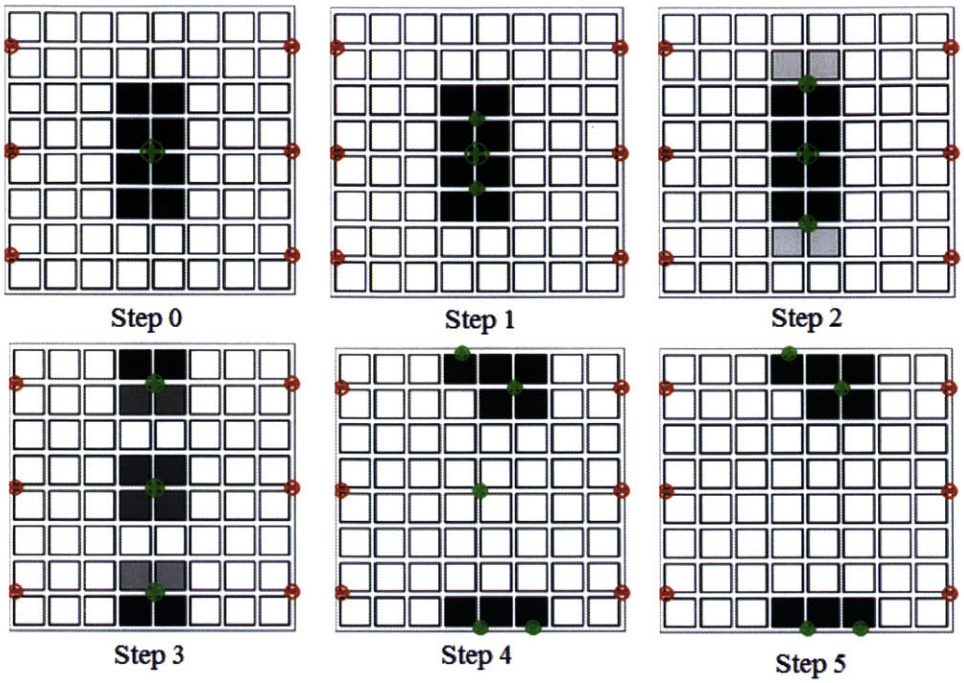

Figure 4-3: Figure illustrating the location of the external charges in the various steps for Phase 1 of the multi-resolution approach. The locations of nanoparticles at the end of each step are also shown in the figure. The gray scale of each cell indicates the expected number of particles that will occupy that cell, which varies from 0 (white) to 1 (black).

The general strategy for selecting the positions and intensities of external charges to ensure a high probability of having 5 nanoparticles in the top half and 3 nanoparticles in the bottom half of the domain was described in the section on designing the energy landscape. As shown in Figure 4-3, the first Phase is divided into five steps, where the positions and strengths of the external charges change at every step. In the zeroth step, a strong attractive charge is placed in the center of the grid, causing a deep potential well attracting all the nanoparticles. The simulation starts at the end of the zeroth step. The subsequent steps follow the proposed strategy, described in the section on designing the energy landscape: two point charges are introduced on

either side of the center charge; new charges move away from the center and their intensities increase; the center charge's intensity decreases with time. In the final step, the external charges in the corner are decomposed into two in order to implement the static solution [63], which stabilizes the steady state configurations and maximizes the probability of the desired arrangement of nanoparticles. Table 4.1 contains the strengths of the attractive external charges used during each stage.

Table 4.1: Parameter values for the simulation of Phase 1 of the multi-resolution approach. The strengths of the repulsive charges used are -300 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

Step	Strength of attractive external charges $((q_k)/(q_p))$ (clockwise)
0	200
1	24, 160, 20
2	48, 120, 40
3	72, 80, 60
4	18.2, 100, 40, 86.5, 39.9
5	18.2, 100, 86.5, 39.9

The probability of observing the system in the desired multi-resolution Phase, from the simulations is shown in Figure 4-4. The simulation starts with the initial configuration of particles as shown in step 0 of Figure 4-3 and the structure of external charges as shown in Step 1. The probability of the desired arrangement of nanoparticles (5 in the top domain and 3 in the bottom domain) increases slowly during the simulation of Steps 1 and 2, and hence, the system is quickly moved to Step 3 of the control strategy. This is shown as a magnification in the same figure. In Step 3, a more rapid increase in probability is observed. The simulation at each step is continued until the rate of increase in probability becomes low, after which point the structure of external charges is changed and all the unimportant configurations from the projection space are removed. Note that the removal of unimportant configurations occurs each time the structure of external charges changes. This is in addition to the case when the upper bound on the error of the system hits the minimum allowed value. The probability of the desired arrangement of nanoparticles

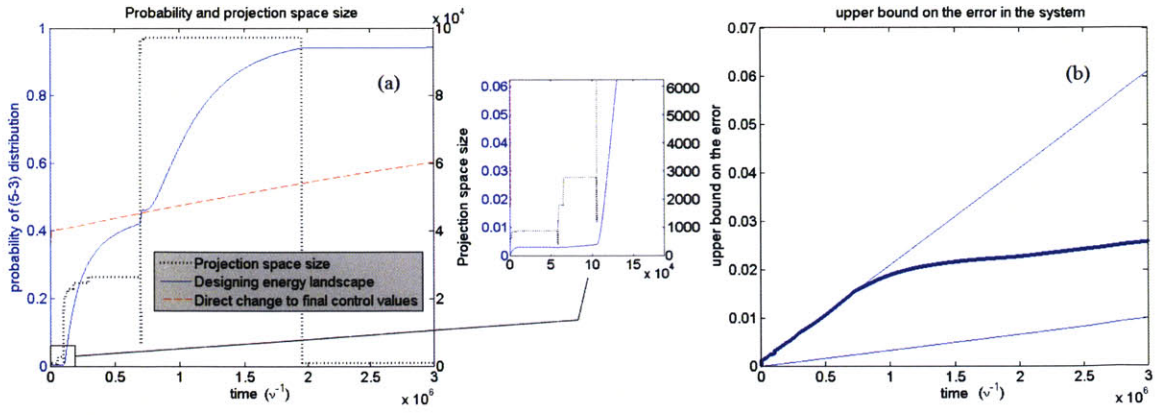

Figure 4-4: Graphs showing the performance of the proposed strategy for self-assembly in Phase 1 of the multi-resolution approach. (a) The blue solid line shows the probability of observing 5 nanoparticles in the top half and 3 nanoparticles in the bottom half of the domain. The red dotted line is the probability of the system if we used the control structure of step 5 at time $t = 0$, indicating that a time-dependent strategy is essential for reaching the desired decomposition with high probability. The black dotted line shows the number of configurations simulated (projection space size) by the AFSP at every instant (secondary axis). (b) Shows the graph of the upper bound on the error of the system.

is shown by a blue line in the graphs and the number of configurations simulated is shown by the black dotted line. The discrete steps in the size of the projection space correspond to the various steps, i.e., the changes in charge structure. Figure 4-4(b) contains the upper bound on the error of the system that results from the use of the adaptive finite state projection method while performing the simulation, which stays within its limits.

In order to demonstrate the main advantage of this proposed control strategy, an additional simulation was done, where the external charges were directly changed to those of Step 5 from Step 0. The probability of the desired arrangement of nanoparticles for that case is plotted as a red dotted line in Figure 4-4(a). As seen in the graph, this procedure yields a self-assembly process with much slower dynamics, and underlines the strength of the proposed control strategy to simulate the dynamics of directed self-assembly in an efficient way.

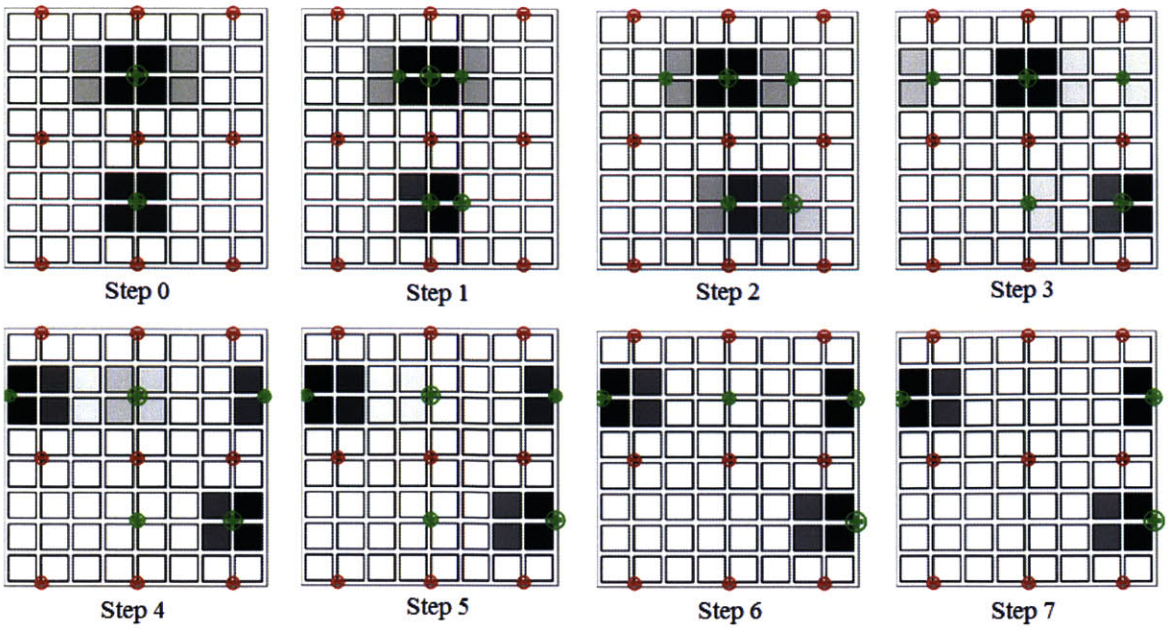

Figure 4-5: Figure illustrating the location of the external charges in the various steps for Phase 2 of the multi-resolution approach. The locations of nanoparticles at the end of each step are also shown in the figure. A horizontal line of repulsive charges is added along the middle of the domain to separate the top and bottom halves of the domain, and thus the two parts are simulated independently.

Phase 2

The aim of this phase is to achieve, with a high probability, an arrangement of 3 nanoparticles on the top-left part of the grid, 2 nanoparticles on the top-right part of the grid and 3 nanoparticles in the bottom-right part of the grid. At the end of the first Phase of the multi-resolution approach, repulsive charges are placed at the center of the domain, dividing the top and bottom domains (Figure 4-5). This is done to create an energy barrier to prevent transitions from the top domain to the bottom domain, and thus break the ergodicity of the system over the whole domain. This allows the two domains to be simulated independently. Similarly as for Step 0 of Phase 1, strong attractive charges are added in the center of each domain (top and bottom half) and simulations are carried from the end of the Step 0, when all the nanoparticles have reached the center of the domain. From Step 1 onwards, the proposed control strategy is used to facilitate self-assembly. The strengths of the attractive external charges can be found in Table 4.2. The strengths of the external

charges, when they reach the boundary of the domain, are determined by the static solution [63].

Table 4.2: Parameter values for the simulation Phase 2 of the multi-resolution approach. The strengths of the repulsive external charges used are -100 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

Step	Strength of attractive external charges $((q_k)/(q_p))$ (clockwise)
0	150, 100
1	24, 120, 10, 20, 80
2	30, 110, 15, 40, 60
3	40, 100, 20, 60, 40
4	120, 30, 57.1, 60, 40
5	120, 30, 57.1, 100, 20
6	120, 30, 57.1, 100
7	120, 57.1, 100

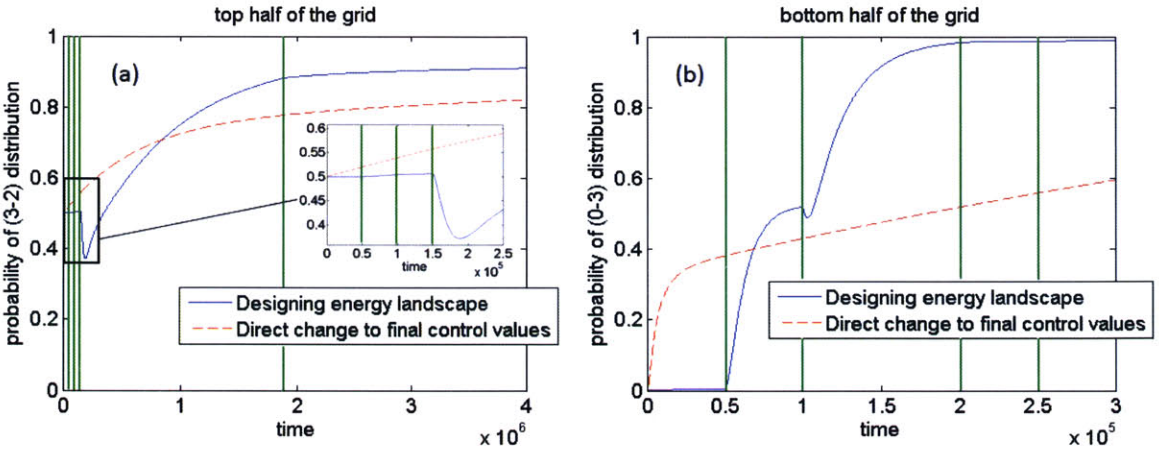

Figure 4-6: Graphs showing the performance of Phase 2 of the multi-resolution approach for the case study. (a) Probabilities of finding 3 particles and 2 particles on the left and right sides of the top half of the domain respectively. The green vertical lines indicate the time points when the external charges switch to the arrangements shown in Steps 2, 3, 4 and 7, respectively. (b) Probability of finding 3 particles on the bottom-right quadrant of the domain. The green vertical lines indicate the time points when the external charges switch to the arrangements shown in Steps 2, 3, 5 and 6, respectively.

The probability of observing the system in the desired multi-resolution Phase, from the simulation is shown in Figure 4-6. The blue line shows the probability of

the desired arrangement of nanoparticles and the green vertical lines in Figure 4-6(a) are indicative of the various steps in the control strategy. Note that the charges in the top domain from Steps 1 to 4 (Figure 4-5) move rapidly, because of the slow rate of increase in probability of the desired arrangement of nanoparticles (3 and 2 particles in the top left and right parts). Since this movement of external charges is rapid, the time points of these changes in steps are very close (Figure 4-6(a)). The magnified time points of these step changes are shown as an inset in the same figure. These step changes seem farther apart in Figure 4-6(b) because of the difference in time scale. The step changes shown in Figure 4-6(a) correspond to Steps 2, 3, 4 and 7, respectively, and Steps 2, 3, 5 and 6, respectively, in Figure 4-6(b). As in the previous Phase, the red dotted lines in the plot are the probabilities of the desired arrangements of nanoparticles in the top and bottom halves, when the controls of Step 7 are applied after Step 0, an indication of faster self-assembly through the multi-resolution control strategy.

The implementation of various steps in the top half of the grid is done by including all the configurations of the top half of the domain (201376 configurations). Similarly, the bottom half of the domain is assumed to be an independent system and the simulation is done by considering all possible configurations (4960 configurations).

Phase 3

The final desired configuration of the nanoparticles is obtained with high probability through the implementation of the steps shown in Figure 4-7. At the end of Phase 2, repulsive external charges are added in the middle of the grid to create an energy barrier and restrict the system's ergodicity in their respective parts of the grid. Therefore, this creates four 4x4 grids that can be simulated independently. Each of these grids are small enough to be simulated with the full model without having to use the AFSP method. The strengths of the various external charges used in all the steps can be found in Table 4.3.

The probability of observing the system in the desired multi-resolution Phase, from the simulations is shown in Figure 4-8. The probability indicated in the graphs

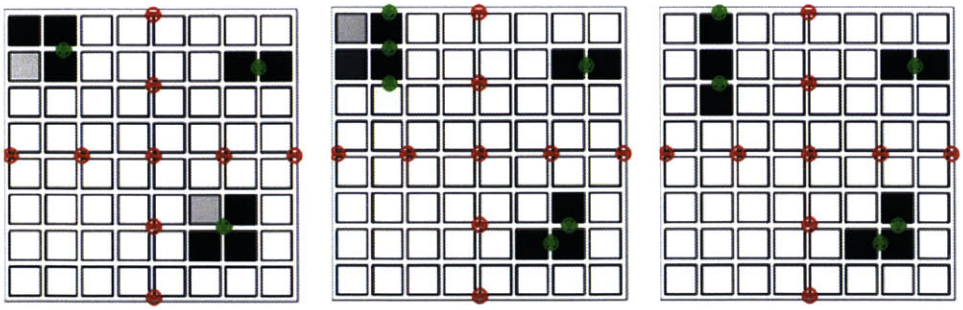

Figure 4-7: Figure illustrating the location of the external controls in the various steps for Phase 3 of multi-resolution approach. The locations of nanoparticles at the end of each step are also shown in the figure. A vertical line of repulsive charges is now added along the middle of the domain to separate the domain in to four parts, and thus each one is simulated independently.

Table 4.3: Parameter values for the simulation of Phase 3 of the multi-resolution approach. The strengths of the repulsive external charges used are -50 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

Step	Strength of attractive external charges $((q_k)/(q_p))$ (clockwise)
0	150, 150, 150
1	45, 40, 100, 150, 75, 75
2	45, 100, 150, 75, 75

are that of the desired configuration of the nanoparticles in each quadrant. The simulation is started from the end of Step 0, where there is an attractive charge in each quadrant of the system. In Step 1, two attractive charges are added to the top-left quadrant and two attractive charges replace the one attractive charge in the bottom-right quadrant. These locations of the attractive external charges are determined based on the positions of the final external charges (Step 2 of Figure 4-7). This final external charge structure can be determined using the Minimum Tiling Approach proposed by Solis et al. [63] As the simulation begins, the system quickly changes to Step 2 of the arrangement of charges, due to the slow rate of increase in the probability of the desired configuration in the top-left quadrant. The green vertical lines in Figure 4-8(a) indicate the times when the system changes to Step 2.

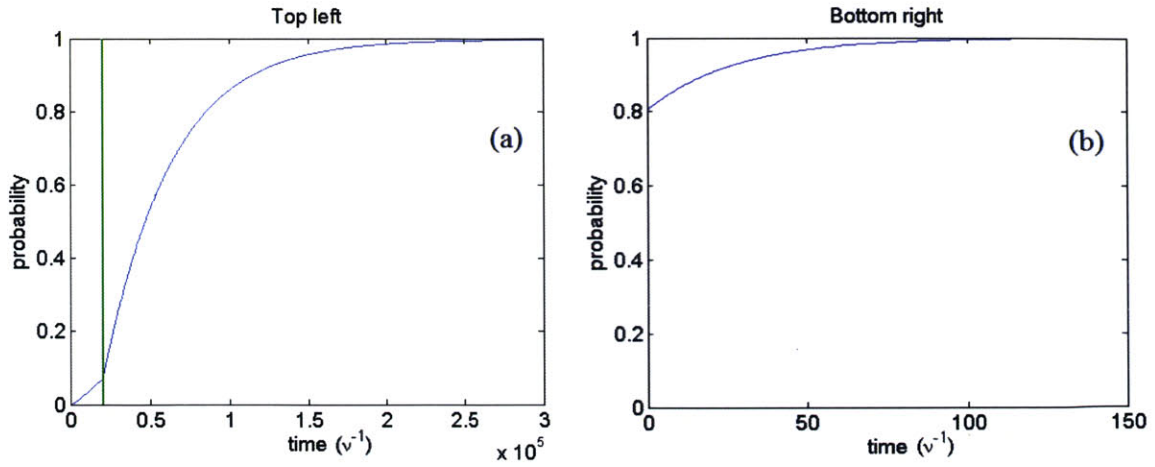

Figure 4-8: Graphs showing the performance of Phase 3 of the multi-resolution approach for the case study. (a) and (b) show the probability of the desired configuration of nanoparticles in the top-left and bottom-right quadrants of the domain. This desired configuration is shown in Step 2 of this Phase. The green vertical line in (a) indicates the time the external charges switch to the arrangement shown in Step 2.

In summary, a control strategy based on multi-resolution approach of self-assembly is presented in this Chapter. The strategy decomposes the space into non-ergodic subspace by creating and shifting energy wells. The strategy not only provides a dynamic path for the nanoparticles towards self-assembly avoiding any kinetic traps, but also integrates well with the AFSP method of Chapter 3 by helping it reduce the model size. This was demostrated through a case study. However, the resulting policy is not optimal, as the strengths of the charges used were determined by trial and error. In the next Chapter, we will present an optimal control strategy, where optimum time profile for the locations and strengths of the external charges will be calculated. This will overcome the lack of optimality used in this case study, and allows the fabrication of the desired final structure in minimum time.

Chapter 5

Multi-Resolution Based Optimal Control

In Chapter 4 we studied the dynamic modeling and simulation of guided self-assembly under the influence of a control strategy that was based on a multi-resolution view of the evolving system. The multi-resolution view modeled the various configurations by allocating the nanoparticles into spatial regions of varying resolution. For example, initially the configurations were modeled by allocating the particles in the entire domain of the 2-d space, and then progressively to the two halves of the domain, the four quadrants of the domain and so on. At each phase the position and intensity of the external controls formed potential wells that guided the movement of the nanoparticles to the desired spatial sub-regions of the 2-d domain [54]. As the resolution of the spatial sub-regions and consequently of the description of the configurations increased, the repositioning of external controls with the potential wells they created guided the nanoparticles to finer and finer resolution structures, until the desired one was achieved. Furthermore, at every phase of the multi-resolution approach, the time domain was divided into multiple steps, where each step represented a different arrangement of the external control charges. The time duration and the strengths of the external charges for each time step were chosen in such a way that the probability of the system reaching that multi-resolution structure approached a pre-specified high value.

The control strategy employed in Chapter 4 was an ad hoc strategy, possessing no specific attributes. In this chapter, the control strategy will be designed to possess certain optimality properties. Although the approach is applicable for a variety of optimization objectives, in this chapter we will focus the attention to minimum time optimal control strategies, i.e. control strategies which guide the self-assembly of nanoparticles from any initial random (and unknown) configuration to a configuration of desired geometry in minimum time. The mathematically rigorous optimal control strategy will determine the positioning and strength of the external charges at each time step.

5.1 Formulation of the Optimal Control Problem

Selecting a control strategy that leads the nanoparticles from any initial distribution to the structure of desired geometry in minimum time, is an objective that conforms to the needs of efficient manufacturing. In addition, it ensures that the nanoparticles will avoid kinetic traps, where they are expected to stay for a long time, before they can gradually escape from the unwanted local minimum potential well and continue on to the globally minimum one of the desired structure. In addition, there are also technological limitations that constrain the placement of external charges on a physical domain. These limitations are an artifact of the various top-down approaches used for the creation of these external charges. Thus, the continuous-time optimal control problem is formulated, at every phase of the multi-resolution strategy, as

follows:

$$
\begin{aligned}
\min_{\mathbf{q},\mathbf{r},t_f} \quad & t_f \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(\mathbf{q}(t),\mathbf{r}(t))\mathbf{p}(t), \quad \forall t \in (0,t_f], \quad \mathbf{p}(0) = \mathbf{p}_0, \\
& p_d(t_f) \geq p_{min} = p_{ss} - \delta, \\
& \mathbf{q}_m \leq \mathbf{q}(t) \leq \mathbf{q}_M, \quad \forall t \in [0,t_f], \\
& \mathbf{r}(t) \in R, \quad \forall t \in [0,t_f], \\
& \mathbf{q}(t_f) = \mathbf{q}_f; \quad \mathbf{r}(t_f) = \mathbf{r}_f.
\end{aligned} \tag{5.1}
$$

$\mathbf{q}$ is the vector of controls representing the ratio of the strengths of the external charges to the strength of the charge of a nanoparticle, $\mathbf{q}_s/q_p$. $\mathbf{r}$ is the vector containing the locations of all controls, i.e. of the external charges. $\mathbf{q}_m$ and $\mathbf{q}_M$ are the lower and upper bounds, respectively, on the controls. R is a discrete set of locations on the domain where the external charges can be placed.

The second constraint provides the termination criterion for the optimal control policy, where p_d is the probability of the desired configuration of the nanoparticles. The value of p_{min} is determined by using the locations and strengths of the external charges determined by the static solution found by Solis et al. [63], which are captured by the last two equality constraints. The static solution produced by Solis et al. [63], ensures robustness of the desired configuration at the steady-state, i.e. it ensures a high probability that the steady-state configuration remains at the structure with the desired geometry for a long time. As a consequence, Solis et al. showed that the static solution produces a very narrow probability distribution for the configurations, and this steady-state probability, say p_{ss}, can be easily estimated from samples generated by Metropolis Monte Carlo simulations. Fixing the locations and strengths of the charges at the final time to the static solution values ensures that the probability of the system being in the desired configuration approaches p_{ss} with time, and thus ensures feasibility of the optimal control policy. This is analogous to the use of steady-state constraints on the state (or, output), during the computation of the open-loop

optimal control over the prediction horizon, within the Model Predictive Control framework, in order to ensure stability of the feedback control law [36].

For numerical solution, the controls can then be discretized in time, keeping the locations and strengths of the external charges constant within a time interval. Note, however, that the state variables, i.e. the probabilities, remain continuous in time. Each of these discretized time intervals corresponds to a step of the control strategy, as described in Chapter 4. The new formulation is as follows:

$$
\begin{aligned}
\min_{\mathbf{q}_k, \mathbf{r}_k, t_k} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(\mathbf{q}_k, \mathbf{r}_k)\mathbf{p}(t), \quad \forall t_{k-1} < t \leq t_k, \quad \forall k = 1, 2, ..., N, \quad t_0 = 0, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& t_{k-1} \leq t_k, \quad \forall k = 1, 2, ..., N, \\
& p_d(t_N) \geq p_{min}, \\
& \mathbf{q}_m \leq \mathbf{q}_k \leq \mathbf{q}_M, \quad \forall k = 1, 2, ..., N-1, \\
& \mathbf{r}_k \in R, \quad \forall k = 1, 2, ..., N-1, \\
& \mathbf{q}_N = \mathbf{q}_f; \quad \mathbf{r}_N = \mathbf{r}_f,
\end{aligned}
\tag{5.2}
$$

where N is the number of steps in the control strategy.

The solution of the above control problem would produce the optimum dynamic profile for the external charges for the guided (controlled) self-assembly process. However, it should be noted that, as the size of the physical 2-d domain increases, the number of potential locations for the external charges and the number of configurations increases combinatorially, and the solution of problem (5.2) becomes intractable. The Multi-Resolution Decomposition of the configuration space allows us to produce tractable computational solutions to problem (5.2), as will be shown in the next paragraph.

5.1.1 Multi-Resolution Decomposition of the Configuration Space

In Chaper 3, the Adaptive Finite State Projection (AFSP) method was developed in order to provide an effective dynamic model-reduction technique for handling the large sets of differential equations describing the probabilities of all configurations in an evolving nanostructure. However, it should be noted that the AFSP approach could fail in maintaining a manageable small number of necessary configurations in the projection space. For example, inappropriately positioned controls could lead to a potential energy landscape that is relatively flat with a very large number of local minima distributed throughout the 2-d domain. In such case, the system is essentially ergodic over the whole space of configurations and we need to include in the projection space a very large number of configurations in order to model the self-assembly system with high accuracy. To overcome this difficulty, in Chapter 4 we introduced a multi-resolution decomposition of the configuration space. In Phase-1, the physical domain was decomposed into two halves with the corresponding desired numbers of particles. In Phase-2 each half is further decomposed into two halves, thus further decomposing the corresponding configuration spaces (see Chapter 4 and [54]). In early phases of the multi-resolution strategy, such as Phase-1 and Phase-2, the clustering of particles around deep potential wells effectively reduces the number of configurations that need to be included in the projection space and AFSP works very well in taming the number of configurations that need to be considered during the simulation of the self-assembling system. In later phases, the size of the independent physical domains and the number of particles in them has been significantly reduced, so that AFSP works very effectively. Actually, these reductions may allow complete enumeration of all configurations in the master equation, as it was demonstrated in an example in Chapter 4.

This multistep process constrains the movement of external charges in a way such that the basins of attractions in the energy landscape are well-defined. The optimal control problem is now formulated for every phase of the multi-resolution approach, and the positions of the external charges are no longer optimization variables in the

control problem (5.2), thus making it more tractable. The resulting formulation of the optimal control problem for an arbitrary phase j is given by (5.3):

$$
\begin{aligned}
\min_{\mathbf{q}_k, t_k} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(\mathbf{q}_k)\mathbf{p}(t), \quad \forall t_{k-1} < t \leq t_k, \quad \forall k = 1, 2, ..., N, \quad t_0 = 0, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& t_{k-1} \leq t_k, \quad \forall k = 1, 2, ..., N, \\
& \sum_{i \in \Omega_j} p_i(t_N) \geq p_{min}, \\
& \mathbf{q}_m \leq \mathbf{q}_k \leq \mathbf{q}_M, \quad \forall k = 1, 2, ..., N-1, \\
& \mathbf{q}_N = \mathbf{q}_f.
\end{aligned} \tag{5.3}
$$

Here Ω_j is the set of goal configurations for the nanoparticles in multi-resolution phase j, among all possible configurations of the domain under consideration in phase j. The value of p_{min} is determined by implementation of the static solution at the final step of the multi-resolution phase, which is determined using Solis et al.'s work [63]. This solution is calculated for an arbitrarily chosen configuration, among the set of goal configurations for the multi-resolution phase j.

As described in Chapter 4 and in [54], every phase of the multi-resolution approach starts with the placement of a strong attractive charge at the center of the domain under consideration and repulsive charges at the boundaries of this domain. This procedure attracts the nanoparticles towards the center of the domain. The time is initialized to zero at the end of this procedure, and the initial probability vector $\mathbf{p}_0$ is calculated by Metropolis Monte Carlo simulations. It will be later shown in the case study that the time duration of this initial step is much less than the total time of self-assembly in that phase.

This optimal control formulation in (5.3) is modified into the following equivalent

formulation (5.4):

$$
\begin{aligned}
\min_{\mathbf{q}_k, w_k, t_N} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(\mathbf{q}_k)\mathbf{p}(t), \quad \forall t_N w_{k-1} < t \leq t_N w_k, \quad \forall k = 1, 2, ..., N, \;\; w_0 = 0, \;\; w_N = 1, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& w_{k-1} \leq w_k, \quad w_k \in [0, 1], \quad \forall k = 1, 2, ..., N, \\
& \sum_{i \in \Omega_j} p_i(t_N) \geq p_{min}, \\
& \mathbf{q}_m \leq \mathbf{q}_k \leq \mathbf{q}_M, \quad \forall k = 1, 2, ..., N-1, \\
& \mathbf{q}_N = \mathbf{q}_f,
\end{aligned} \tag{5.4}
$$

where w_k are the ratios of the simulation times of all steps in the control strategy to the final time, i.e. $w_k \equiv t_k / t_N$. Formulation (5.4) can be made dimensionless in time by scaling it with t_N.

$$
\begin{aligned}
\min_{\mathbf{q}_k, w_k, t_N} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{d\tau}(\tau) = t_N \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau), \quad \forall w_{k-1} \leq \tau \leq w_k, \quad \forall k = 1, 2, ..., N, \;\; w_0 = 0, \;\; w_N = 1, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& w_{k-1} \leq w_k, \quad w_k \in [0, 1], \quad \forall k = 1, 2, ..., N, \\
& \sum_{i \in \Omega_j} p_i(1) \geq p_{min}, \\
& \mathbf{q}_m \leq \mathbf{q}_k \leq \mathbf{q}_M, \quad \forall k = 1, 2, ..., N-1, \\
& \mathbf{q}_N = \mathbf{q}_f.
\end{aligned} \tag{5.5}
$$

5.2 Implementation

The optimal control problem (5.5) can be solved using SNOPT [19], a sequential quadratic programming solver. The SNOPT algorithm requires derivatives of the objective function and the constraint; $\sum_{i \in \Omega_j} p_i(1) \geq p_{min}$.

5.2.1 Examining the differentiability of the formulation

In Equation (2.2), under the AFSP method, the vector $\mathbf{p}$ is defined such that it contains the probabilities of all possible configurations, and the probability values of the configurations not included in the current projection space are set to zero. Similarly, $\mathbf{A}(t)$ is a full-sized matrix, where only the block diagonal submatrix corresponding to the configurations included in the current projection space is nonzero: $\epsilon_{\beta\leftarrow\alpha}(t) \neq 0, \forall\alpha \in A(t), \beta \neq \alpha$. However, during any simulation of the problem, only the configurations in the current projection space are simulated in order to reduce the computational time, yielding a system that varies in size during the course of a simulation.

When the upper bound on the error hits the maximum or minimum allowed values, the AFSP method modifies the subset of configurations that are included in the reduced model of the system. If it hits the maximum allowed value, the subset of configurations expands with new configurations, and when it hits the minimum allowed value, it eliminates certain configurations from the projection space. In these instances, the state projection space changes, leading to a change in the structure of matrix $\mathbf{A}(t)$ in Equation (2.2).

Note that the error crosses the maximum allowed value when the sum of the probabilities of the configurations included in the projection space falls below a certain level, or equivalently, when the difference of the sum from 1 exceeds a maximum value. Similarly, the error crosses the minimum allowed value when the sum of the probabilities of the configurations included in the projection space exceeds a certain level, or equivalently, when the difference of the sum from 1 falls below a minimum value. In other words, the structure of the matrix, $\mathbf{A}(t)$, is effectively a function of the probabilities of the configurations in the projection space, leading to the following modification of the master equations:

$$\frac{d\mathbf{p}}{dt}(t) = \mathbf{A}(\mathbf{q}_k, \mathbf{p}(t))\mathbf{p}(t) = \mathbf{f}(\mathbf{q}_k, \mathbf{p}(t)). \tag{5.6}$$

To compute the optimal control strategy using SNOPT we need the sensitivities

(i.e. the derivatives) of the objective function and of the state variables (i.e. the probabilities in the master equations) with respect to the control parameters. In addition, these sensitivities need to be continuous functions of the control paramters. As can be seen from Equations (5.11), the derivatives of the probabilities with respect to the control (optimization) parameters are calculated by differentiating the ordinary differential equations. The classical result for existence of sensitivities requires $\mathbf{f}$ to be continuously differentiable with respect to its arguments (see Theorem 3.1, Chapter V [25]). However, whenever the error hits the maximum or minimum allowed values, the state projection space changes and so does the structure of the matrix, $\mathbf{A}(t)$, potentially leading to the state variables not being differentiable with respect to the control parameters. Specifically, while simulating the dynamic system with particular values of $\mathbf{q}_k$, if the error crosses the maximum allowed value, a configuration is added if the dp/dt value of that configuration is above a cutoff value and a configuration is removed if its dp/dt value is lower than another cutoff value. When the simulation is repeated with a small change in the value of $\mathbf{q}_k$, as the error hits its bounds, the change in the dp/dt value may not result in the addition/removal of the same configurations that were added/removed for the previous value of $\mathbf{q}_k$. This is due to the fact that the addition/removal of configurations depends on how far the dp/dt values are from the cutoff value.

To illustrate diagrammatically this point, consider Figure 5-1. At the time point $t = t^*$, a change from $\mathbf{q}_k$ to $\mathbf{q}_k + \delta\mathbf{q}$ does not change the way the projection space changes in time. However a change from $\mathbf{q}_k + \delta\mathbf{q}$ to $\mathbf{q}_k + 2\delta\mathbf{q}$ increases the value of dp/dt beyond the cutoff value and hence the projection space is modified in a different manner. Therefore, for small changes in the value of $\mathbf{q}_k$, one cannot guarantee the same sequence of projection spaces during the dynamic simulation. Since the structure of $\mathbf{A}(t)$ changes based on the projection space, $\mathbf{A}(t)$ is not a continuous function. Thus, $\mathbf{f}$ in Equation (5.6) is not continuous in its arguments, potentially making the sensitivities undefined as well.

Galán et al. [17] have dealt with sensitivity analysis in the case when the right-hand side function of a system of ordinary differential equations is discontinuous, by

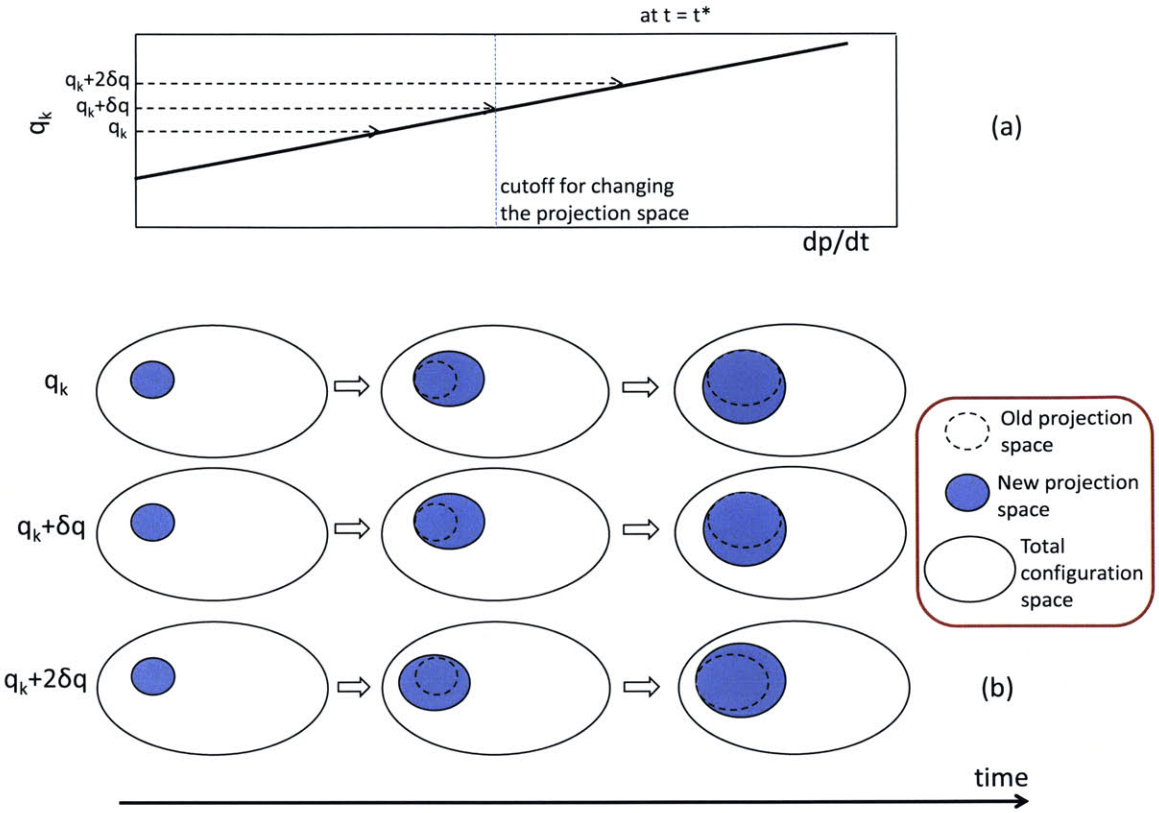

Figure 5-1: Figure illustrates the discontinuity in the right-hand side function of the ODE. At some time $t = t^*$, (a) shows the graph of dp/dt vs $\mathbf{q}_k$ for any configuration, and the corresponding change in the projection space that results from different values of $\mathbf{q}_k$ is shown in (b).

calculating a jump in the sensitivities resulting from the discontinuities. However, their result is valid only when the sequence of discontinuities in the time domain is invariant with respect to the control parameters. Unfortunately, these requirements are not always satisfied by the AFSP strategy; the number and nature of discontinuities varies with $\mathbf{q}_k$. Furthermore, the method of Galan et al. [17], requires that the error does not hit the error bounds tangentially, but the AFSP imposes no such restrictions on the evolution of the error in time.

It is clear from the above discussion, that the AFSP, as presented in Chapter 3, cannot be used as is for the computation of optimal control strategies, with provably correct optimality properties, and needs to be modified. The modified AFSP for the computation of optimal control strategies is based on the following requirements:

1. To ensure that $\mathbf{f}$ is continuously differentiable with respect to its arguments during every step of the control strategy, the projection space for every step is kept constant. The set of important configurations for the projection space is obtained by a one-time simulation of the system, using the AFSP method, for particular values for the strengths of the external charges. The potential variation in the projection space for when the strengths of the external charges are varied by the optimization procedure can be mitigated by having additional constraints on those strengths. The strengths used to determine this fixed projection space are chosen from the above additional constraints.

2. For any strengths of the external charges, the ratio of the simulation time of any step in the control strategy to the final time is kept constant. This is done for two reasons:

   (a) When time is non-dimensionalized in the formulation, this feature will force any particular step of the control strategy to start at the same time point in the scaled time domain for different values of $\mathbf{q}_k$. Hence, it ensures that the right-hand side of the differential equation becomes continuous in $\mathbf{q}_k$.

   (b) The constant projection space to be used in the optimization will remain valid. This is because the optimization would then adjust the values of $\mathbf{q}_k$ such that the simulation reaches a similar physical state at the end of any particular step of the control strategy across different iterations.

In other words, w_k is selected a priori and kept constant in the optimization problem. The new optimization problem formulation incorporating the above features is given by:

$$
\begin{aligned}
\min_{\mathbf{q}_k, t_N} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{d\tau}(\tau) = t_N \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau), \quad \forall w_{k-1} < \tau \leq w_k, \quad \forall k = 1, 2, ..., N, \;\; w_0 = 0, \;\; w_N = 1, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& \sum_{i \in \Omega_j} p_i(1) \geq p_{min}, \\
& \mathbf{q}_{kmin} \leq \mathbf{q}_k \leq \mathbf{q}_{kmax}, \quad \forall k = 1, 2, ..., N-1; \quad \mathbf{q}_N = \mathbf{q}_f.
\end{aligned} \tag{5.7}
$$

Additional constraints are added to the problem formulation in order to help keep the *a priori* determined projection space valid. Note that since the projection space is determined *a priori* for each step, $\mathbf{A}(\mathbf{q}_k)$ no longer depends on $\mathbf{p}(t)$. The validity of the projection space can be easily tested *a posteriori*, by calculating the upper bound on the error resulting from using the projection space. This error should be less than some desired threshold.

The right-hand side of the differential equations becomes:

$$
\frac{d\mathbf{p}}{d\tau}(\tau) = t_N \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau) = \mathbf{f}(t_N, \mathbf{q}_k, \mathbf{p}(\tau)), \quad \forall \tau \in (w_{k-1}, w_k]. \tag{5.8}
$$

In this new formulation, the dynamic simulation during any particular step in the control strategy starts at the same time point for all values of $\mathbf{q}_k$. In addition, the structure of $\mathbf{A}(t)$ remains constant and its entries are continuously differentiable with respect to $\mathbf{q}_k$. Hence, $\mathbf{f}$ is continuously differentiable with respect to its arguments.

Furthermore, in the new formulation the only potential discontinuities occur at the transition times between the various steps in the control strategy, i.e. at $\tau = w_k$. Moreover, the number of discontinuities remains fixed and they always occur in the same order. At any of the transition times, all new configurations added to the projection space start with probability values of zero, which are same as their values prior to being added to the projection space. Therefore, for all the new configurations, and the configurations that continue in the projection space, their probabilities, and

hence their sensitivities, remain continuous at the transition times. This is because the time points of the discontinuities are independent of the optimization parameters, as pointed out by Galán et al. [17]. However, the probabilities of the configurations that are removed from the projection space at the transition times jump to zero, thus causing their sensitivities to jump to zero as well, as can be shown by applying the results of Galán et al. [17].

For the sake of simplicity, it can be assumed that the transitions between various steps in the control strategy occur at equal intervals in the scaled time domain. Hence, the final formulation becomes:

$$
\begin{aligned}
\min_{\mathbf{q}_k, t_N} \quad & t_N \\
\text{s.t.} \quad & \frac{d\mathbf{p}}{dt}(\tau) = t_N \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau), \quad \forall \frac{k-1}{N} < \tau \leq \frac{k}{N}, \quad \forall k = 1, 2, ..., N, \\
& \mathbf{p}(0) = \mathbf{p}_0, \\
& \sum_{i \in \Omega_j} p_i(1) \geq p_{min}, \\
& \mathbf{q}_{kmin} \leq \mathbf{q}_k \leq \mathbf{q}_{kmax}, \quad \forall k = 1, 2, ..., N-1; \quad \mathbf{q}_N = \mathbf{q}_f.
\end{aligned} \tag{5.9}
$$

Let $\hat{\mathbf{q}}$ be the vector of optimization variables. Hence the objective function derivatives and the derivatives of the state variables are:

$$
\begin{aligned}
& \hat{\mathbf{q}} = \left[\mathbf{q}_1 \ \cdots \ \mathbf{q}_N \ t_N\right], \\
& \frac{\partial t_N}{\partial \hat{\mathbf{q}}} = \left[\frac{\partial t_N}{\partial \mathbf{q}_1} \ \cdots \ \frac{\partial t_N}{\partial \mathbf{q}_N} \ \frac{\partial t_N}{\partial t_N}\right] = \left[\mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{0}^{\mathrm{T}} \ 1\right], \\
& \frac{\partial}{\partial \hat{\mathbf{q}}} \sum_{i \in \Omega_j} p_i(1) = \sum_{i \in \Omega_j} \frac{\partial p_i(1)}{\partial \hat{\mathbf{q}}} = \left[\sum_{i \in \Omega_j} \frac{\partial p_i(1)}{\partial \mathbf{q}_1} \ \cdots \ \sum_{i \in \Omega_j} \frac{\partial p_i(1)}{\partial \mathbf{q}_N} \ \sum_{i \in \Omega_j} \frac{\partial p_i(1)}{\partial t_N}\right].
\end{aligned} \tag{5.10}
$$

The sensitivities, shown in Equation (5.10), can be calculated in the following manner.

$$\begin{aligned}
\frac{d}{d\tau}\left(\frac{\partial \mathbf{p}}{\partial \mathbf{q}_k}\right)(\tau) &= t_N \mathbf{A}(\mathbf{q}_k)\left(\frac{\partial \mathbf{p}}{\partial \mathbf{q}_k}(\tau)\right) + t_N \frac{\partial \mathbf{A}}{\partial \mathbf{q}_k}(\mathbf{q}_k)\mathbf{p}(\tau), && \text{when } w_{k-1} < \tau \leq w_k, \\
\frac{d}{d\tau}\left(\frac{\partial \mathbf{p}}{\partial \mathbf{q}_k}\right)(\tau) &= t_N \mathbf{A}(\mathbf{q}_{k'})\left(\frac{\partial \mathbf{p}}{\partial \mathbf{q}_k}(\tau)\right), && \text{when } w_{k'-1} < \tau \leq w_{k'},\ \forall k' \neq k, \\
\frac{d}{d\tau}\left(\frac{\partial \mathbf{p}}{\partial t_N}\right)(\tau) &= t_N \mathbf{A}(\mathbf{q}_k)\left(\frac{\partial \mathbf{p}}{\partial t_N}(\tau)\right) + \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau). && (5.11)
\end{aligned}$$

Equations (5.11) are obtained by differentiating the master equations, $\frac{d\mathbf{p}}{d\tau}(\tau) = t_N \mathbf{A}(\mathbf{q}_k)\mathbf{p}(\tau)$, with respect to the optimization (control) variables. Equations (5.11) are then appended to the state equations to create a larger system of ODEs, which are then solved using DVOPK [6], as shown by Lakerveld et al. [42]. Also, only sensitivities corresponding to the configurations in the current projection space can be non-zero. So only those sensitivity equations are simulated, as done for the probabilities of the configurations in the current projection space.

5.3 Case Study

The case study used in this section to compute the optimal control strategy is the same as the one used in Chapter 4: a 2-d square physical domain with 64 lattice cells and 8 negatively charged nanoparticles. The total number of possible configurations for this system is 4,426,165,368. The objective is to compute the minimum-time optimal control strategy that brings any initial distribution of the 8 nanoparticles to a structure with desired geometry.

Phase-1

For Phase-1 of the multi-resolution approach, the aim is to have 5 nanoparticles in the top half and 3 nanoparticles in the bottom half of the domain, as prescribed by the desired final geometry. As shown in Figure 5-2, directed self-assembly is achieved via the optimal control strategy in five time periods (steps). The gray scale to the right of the figure indicates the expected number of particles that occupy any cell in

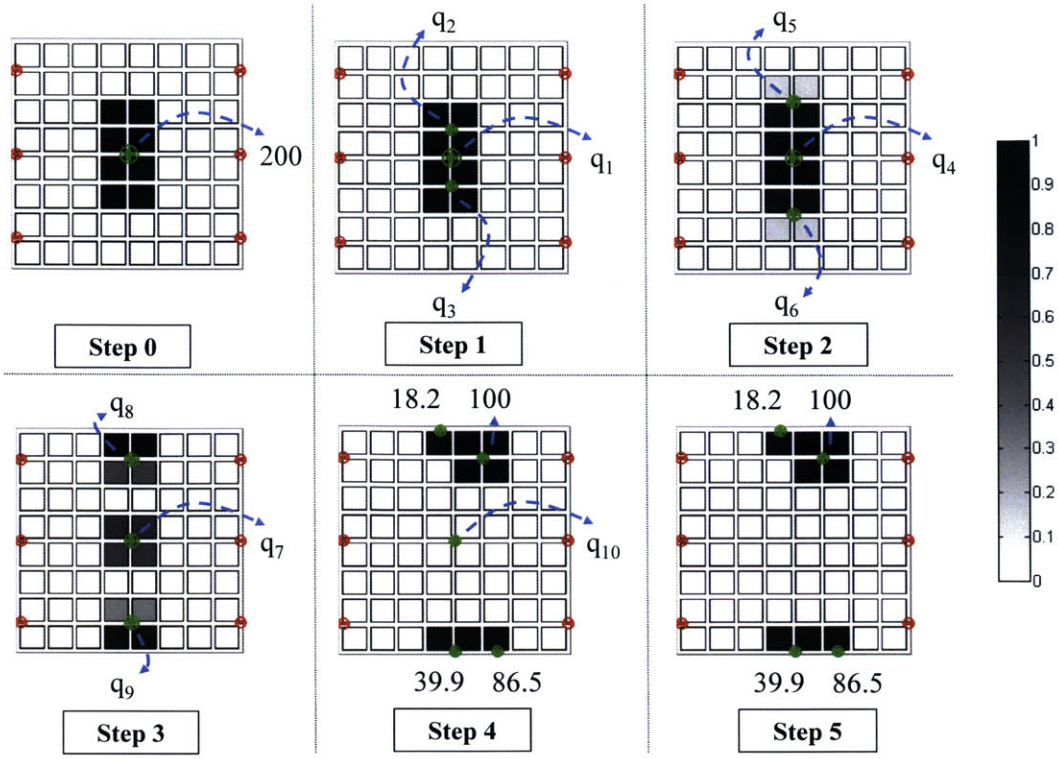

Figure 5-2: Locations and strengths of the external charges in the various steps for Phase-1 of the multi-resolution strategy. The expected locations of nanoparticles at the end of each step are also shown in the figure.

the domain. This number varies from 0 to 1, where 0 (white) indicates a particle will not occupy that grid cell and 1 (black) indicates a particle will occupy that grid cell with 100% probability. The expected number of particles in any cell is obtained by calculating the sum of the probabilities of all the configurations that have a particle in that cell. In Step 0, a high strength attractive external charge is placed in the center, which attracts all the nanoparticles towards the center. The simulation is then started with the arrangement of external charges shown in Step 1 of Figure 5-2. The strengths of all the external charges are indicated with the help of blue dashed arrows; $q_1, q_2, ..., q_{10}$ are the optimization variables and $N = 5$.

The strengths of the boundary charges in Step 4 and Step 5 are fixed *a priori* and correspond to the optimal static solution computed by Solis et al. [63] in their first paper, for the configuration shown in Step 5. The configuration in Step 5 is chosen arbitrarily among the desired possible configurations for the multi-resolution phase.

The steady-state probability (p_{ss}) is estimated to be 0.99, using the samples generated by Metropolis Monte Carlo simulations. Hence, a desired probability (p_{min}) of 0.95 is used as the termination criterion.

Since there is no prior information on w_k, equally spaced steps are used, i.e. Formulation (5.9) is used. The problem has 11 optimization variables: $q_1, q_2, ..., q_{10}$ and t_N. Table 5.1 shows the value obtained from the optimization. It also contains the constraints used for each of the charges (control variables). During Step 1 of the simulation, only q_1, q_2 and q_3 are active, and the remaining variables are assigned zero. Similarly during Step 2, the only active variables are q_4, q_5 and q_6, and so on. The value of t_N obtained from the optimization was 5,394,902 ν^{-1}.

Table 5.1: Parameter values from optimization of Phase-1 of the multi-resolution approach. The strengths of the repulsive charges used are -300 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

	Step 1	Step 2	Step 3	Step 4	Step 5
Value	200 (q_1)	173.85 (q_4)	88.61 (q_7)	43.06 (q_{10})	-
Range	$100 \leq q_1 \leq 200$	$100 \leq q_4 \leq 200$	$50 \leq q_7 \leq 100$	$0 \leq q_{10} \leq 50$	-
Value	50 (q_2)	70.97 (q_5)	139.90 (q_8)	18.2, 100	18.2, 100
Range	$0 \leq q_2 \leq 50$	$50 \leq q_5 \leq 100$	$100 \leq q_8 \leq 150$	-	-
Value	50 (q_3)	66.14 (q_6)	100 (q_9)	39.9, 86.5	39.9, 86.5
Range	$0 \leq q_3 \leq 50$	$50 \leq q_6 \leq 100$	$100 \leq q_9 \leq 150$	-	-

Figure 5-3 shows the simulation results, using the optimal values obtained from the optimal control problem. In Figure 5-3(a), the blue line shows the probability of the desired arrangement of the nanoparticles, i.e. finding 5 nanoparticles in the top half and 3 nanoparticles in the bottom half, produced by the optimal control policy. The red dashed line shows the probability of the desired arrangement of nanoparticles, when the external charges were assigned the steady-state values at the $t = 0$, i.e. a non-optimal policy. As seen from the plot, at $t =$ 5,394,902 ν^{-1}, the probability of the desired arrangement due to the optimal control policy reaches 0.95, whereas the static solution yields a probability for the desired arrangement of only around 0.75. This is due to the fact that the static solution, imposed at the $t = 0$ and maintained for all times, has led the system into a kinetic trap, a situation that the optimal

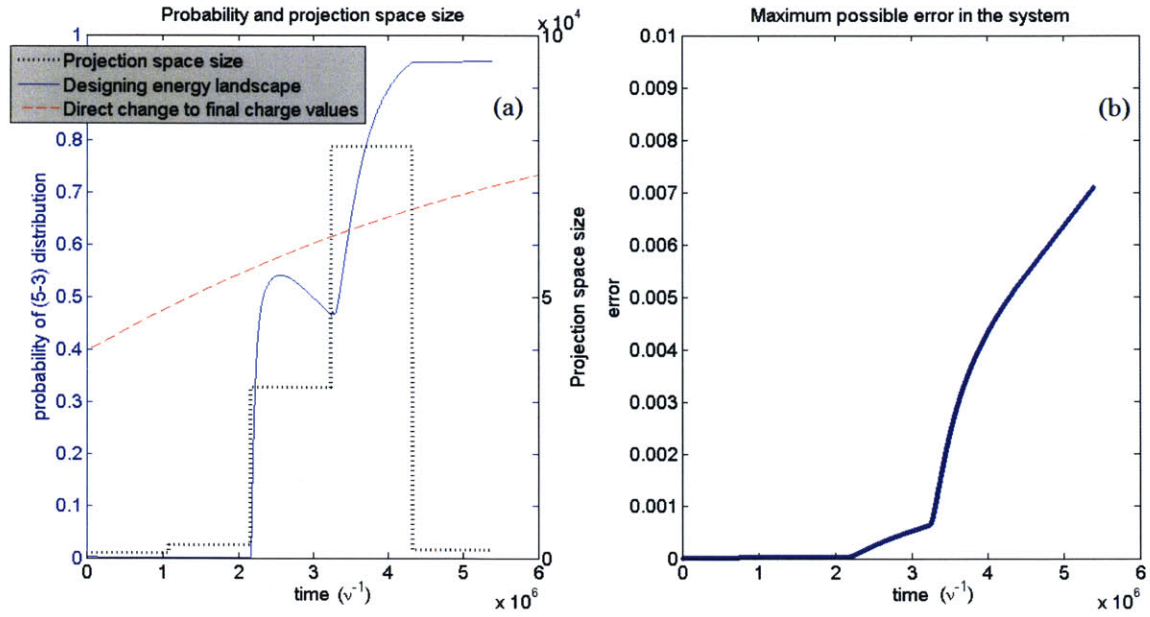

Figure 5-3: Graphs showing the simulation results of Phase-1 of the multi-resolution approach using the values obtained from the optimal control. (a) Probability of the desired arrangement of nanoparticles in blue line and red dashed line. Black dotted lines show the size of the projection space. (b) Upper bound on the error from the dynamic simulation of the assembly process, due to the use of the projection space.

control policy has evaded.

The black dotted line indicates the size of the projection space in each step, determined *a priori*, using simulations with AFSP approximation of the system. The projection size plot shows that the simulation does not need to simulate more than 80,000 configurations at any point in time. The error resulting from the projection space is not more than 0.008, as shown in Figure 5-3(b), which was less than the maximum allowed value used while generating the projection space, indicating that the chosen constant projection spaces for all steps of Phase-1 are sufficiently accurate.

Phase-2

At the end of Phase-1, a horizontal line of repulsive external charges is placed in the middle of the domain, dividing the domain into two halves. The high repulsive charges form an energy barrier, not allowing the nanoparticles to pass, and thus allowing the two halves to be modeled and simulated independently. In addition to those charges,

repulsive external charges are also added on the top and bottom boundary of the domains, thus concentrating the nanoparticles in the middle of each half domain. This reduces the projection space obtained from AFSP.

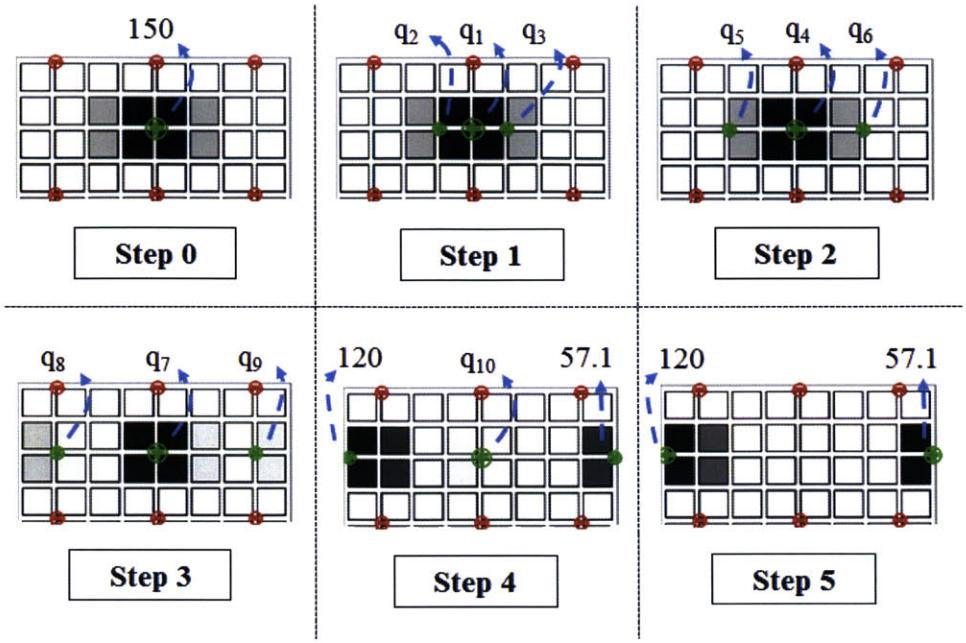

Figure 5-4: Figure illustrating the locations of the external charges in the top half of the domain, in the various steps for Phase-2 of the multi-resolution control strategy. The expected locations of nanoparticles at the end of each step are also shown in the figure. A horizontal line of repulsive charges is added along the middle of the domain to separate the top and bottom halves of the domain, and thus the two parts are simulated independently.

For Phase-2 of the control strategy, Figure 5-4 shows the control strategy for finding the optimal external charge strengths in the top half of the domain. In this phase, the aim is to have 3 nanoparticles in the top left part of the domain and 2 nanoparticles in the top-right half of the domain. The steps here are similar to the ones shown in Phase-1, above. The steady-state probability (p_{ss}) for the external charges shown in Step 5 is estimated to be 0.98 using the samples generated by Metropolis Monte Carlo simulations. Hence, a desired probability (p_{min}) of 0.95 is used as the termination criterion.

Table 5.2 shows the values obtained for the optimal control. It also contains the constraints used for each of the control variables (external charges). As in the previous

phase, only q_1, q_2 and q_3 are active in Step 1, q_4, q_5 and q_6 are active in Step 2, and so on. The optimal value of t_N was 987,973 ν^{-1}.

Table 5.2: Parameter values from the optimization of the top half of the domain in the Phase-2 of the multi-resolution control strategy. The strengths of the repulsive external charges used are -100 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

	Step 1	Step 2	Step 3	Step 4	Step 5
Value	149.42 (q_1)	137.26 (q_4)	41.27 (q_7)	29.53 (q_{10})	-
Range	$100 \leq q_1 \leq 150$	$100 \leq q_4 \leq 150$	$0 \leq q_7 \leq 100$	$0 \leq q_{10} \leq 100$	-
Value	24.93 (q_2)	25.11 (q_5)	40 (q_8)	120	120
Range	$0 \leq q_2 \leq 25$	$25 \leq q_5 \leq 40$	$40 \leq q_8 \leq 100$	-	-
Value	20 (q_3)	39.63 (q_6)	55 (q_9)	57.1	57.1
Range	$0 \leq q_3 \leq 20$	$20 \leq q_6 \leq 40$	$40 \leq q_9 \leq 55$	-	-

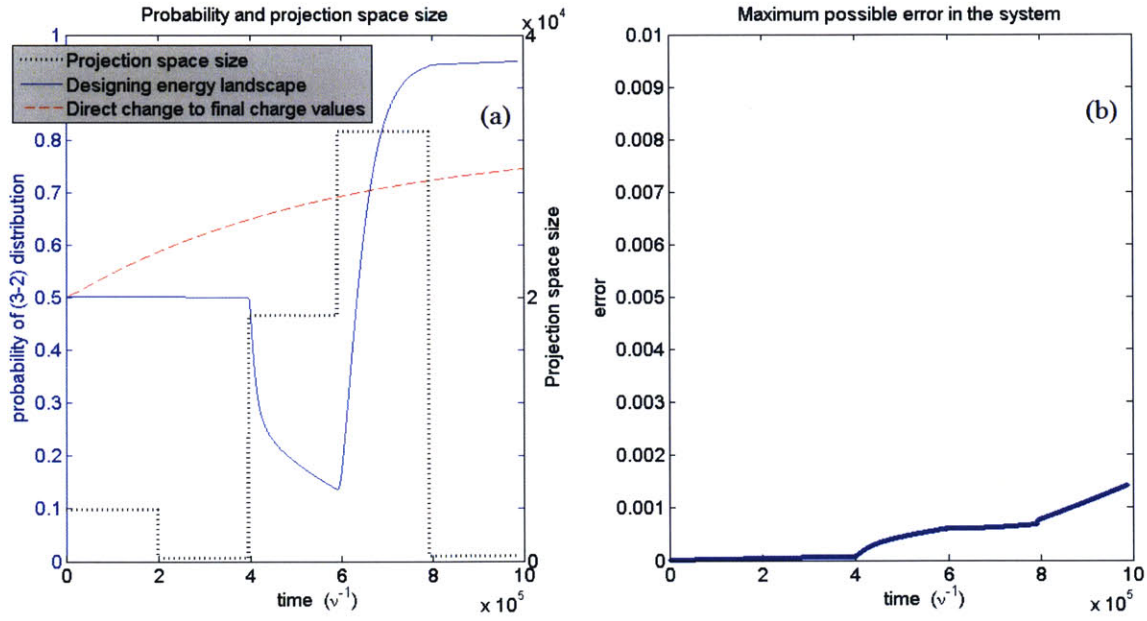

Figure 5-5: Graphs showing the simulation results of the top half in Phase-2 of the multi-resolution control strategy, using the values obtained from the optimal control policy. (a) Probability of the desired arrangement of nanoparticles in blue line and red dashed line. Black dotted lines show the size of the projection space. (b) Upper bound on the error due to the use of a state projection space.

Figure 5-5 shows the simulation results using the values obtained from the optimal control policy. In Figure 5-5(a), the blue line shows the probability of the desired

arrangement of the nanoparticles, i.e. finding 3 nanoparticles in the top-left quadrant and 2 nanoparticles in the top-right quadrant, produced by the optimal control policy. The red dashed line shows the probability of the desired arrangement of nanoparticles when the final steady-state optimal control actions were applied to the system at time, $t = 0$, and were kept constant for the duration, i.e. a non-optimal control policy. As in Phase-1, the probability of the desired arrangement reaches a high probability of around 0.95, at $t = 987{,}973\ \nu^{-1}$, whereas the use of the optimal static control policy yields a probability of only around 0.75; the result of a kinetic trap. It is noteworthy that the optimal control policy does not produce a monotonically increasing probability of the desired arrangement. In Step 3 the probability is reduced, but in Step 4 it rises sharply towards the final value of around 0.95.

The black dotted line indicates the size of the projection space in each step, determined a priori using the AFSP algorithm. The plot with the size of the state projection space shows that the simulation does not need to simulate more than 35,000 configurations at any point in time, and the maximum error resulting for the reduced projection space is less than 0.002, as shown in Figure 5-5(b), implying the projection space was well selected and did not need to be modified.

Figure 5-6 shows the control strategy for finding the optimal external charge strengths in the bottom half of the domain. In this phase, the aim is to have 3 nanoparticles in the bottom-right part of the domain. The steps here are similar to the ones shown in earlier phases of the optimal control strategy. The steady-state probability (p_{ss}) of the desired arrangement, using the optimal values of the external charges shown in Step 5, is estimated to be around 0.99, using the samples generated by Metropolis Monte Carlo simulations. Hence, a desired probability (p_{min}) of 0.98 is used as the termination criterion.

Table 5.3 shows the values obtained from the optimization. It also contains the constraints used for each of the control variables. Similar to the previous case, only q_1 and q_2 are active in Step 1, q_3 and q_4 are active in Step 2, and so on. The value of minimum time, t_N, obtained from the optimization was 166,698 ν^{-1}.

Figure 5-7 shows the probability of the desired configuration (i.e. finding 3

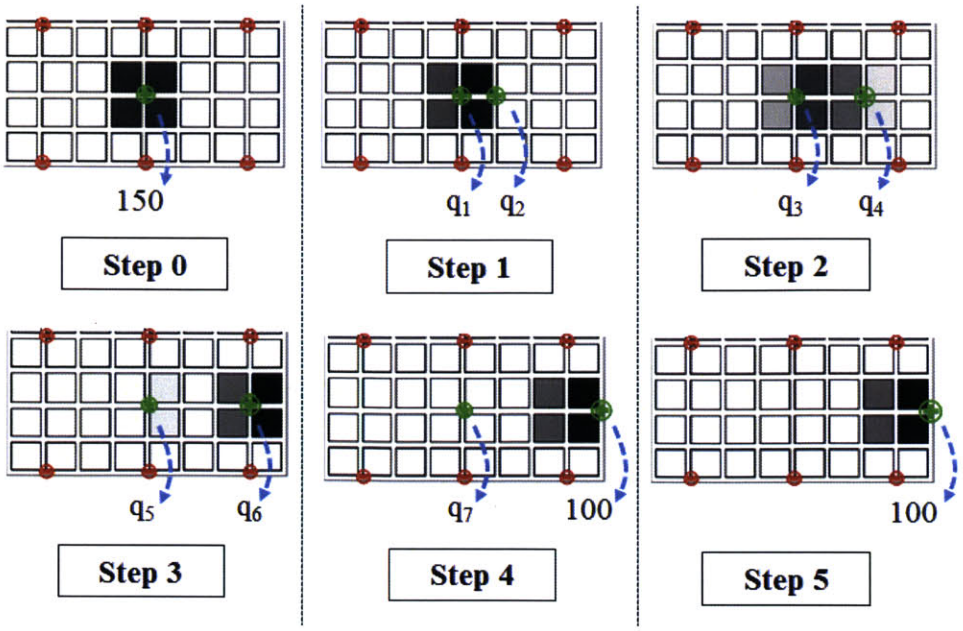

Figure 5-6: Figure illustrating the locations of the external charges in the bottom half of the domain, in the various steps for Phase-2 of the multi-resolution control strategy. The expected locations of nanoparticles at the end of each step are also shown in the figure. A horizontal line of repulsive charges is added along the middle of the domain to separate the top and bottom halves of the domain, and thus the two parts are simulated independently.

Table 5.3: The values of parameters used from the optimization of the bottom half of the domain in Phase-2 of the multi-resolution control strategy. The strengths of the repulsive external charges used are -100 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

	Step 1	Step 2	Step 3	Step 4	Step 5
Value	100 (q_1)	100 (q_3)	38.26 (q_5)	18.82 (q_7)	-
Range	$50 \leq q_1 \leq 100$	$50 \leq q_3 \leq 100$	$0 \leq q_5 \leq 50$	$0 \leq q_7 \leq 50$	-
Value	50 (q_2)	50 (q_4)	100 (q_6)	100	100
Range	$0 \leq q_2 \leq 50$	$0 \leq q_4 \leq 50$	$50 \leq q_6 \leq 100$	-	-

nanoparticles in the bottom right quadrant), obtained from the dynamic optimal control policy (blue line), while the red dashed line shows the probability of the desired arrangement of nanoparticles when the optimal steady-state control was applied to the system at time, $t = 0$, for the entire period. At $t = 166{,}698\ \nu^{-1}$, the probability of the desired arrangement due to the optimal external charges strengths reaches

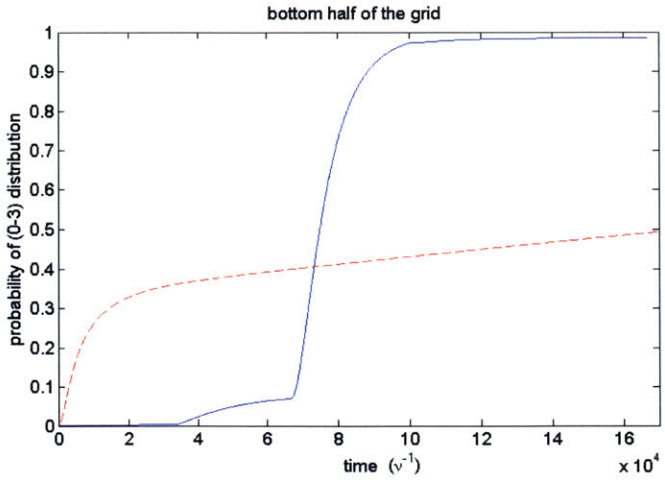

Figure 5-7: Graph showing the simulation results for the bottom half in Phase-2 of the multi-resolution control strategy, using the values obtained from the optimal control policy. The blue line shows the probability of the desired arrangement of nanoparticles under dynamic optimal control, while the red line shows the probability of the desired configuration under the application of the optimal steady-state control.

0.98, whereas that of the static solution, as computed in Step 5, yielded a probability of around 0.50.

Intermediate equilibration step between Phase-1 and Phase-2

At the end of Phase-1, the positions of the nanoparticles are shown in Step 5 of Figure 5-2. At that point, as mentioned before, a horizontal line of repulsive charges are added in the center of the domain, in addition to the other repulsive charges. Attractive charges are also added at the center of each of the top and bottom half, thus attracting the nanoparticles towards them, as shown in Step 0 of Figures 5-4 and 5-6.

The time scale required to achieve this intermediate step between the two phases is assumed to be much smaller than the total time of self-assembly in Phase-1 or Phase-2. This is because all the repulsive charges are at the "boundaries" of the reduced domain, and with a strong attractive charge in the middle, the nanoparticles are immediately attracted towards the middle of this domain. To verify this, the intermediate step for the top half was simulated with 5 nanoparticles from the end of

Step 5 in Figure 5-2 to Step 0 in Figure 5-4.

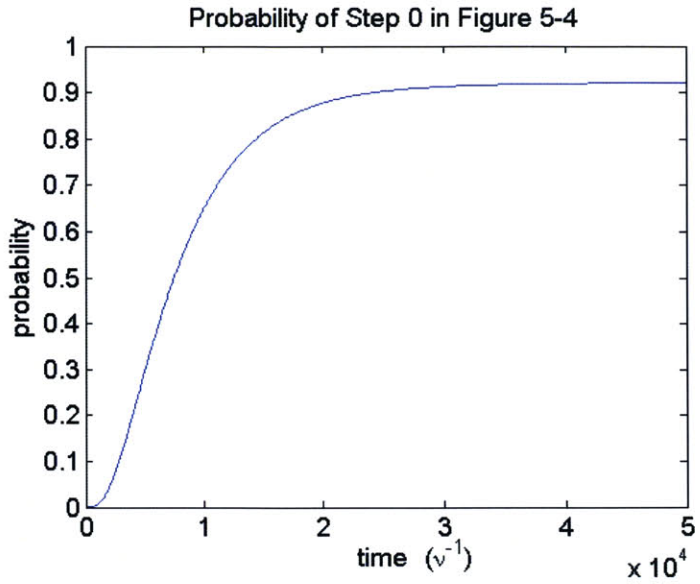

Figure 5-8: Graph showing the simulation results for the top half in the intermediate equilibration step between Step 5 of Phase-1 (Figure 5-2) and Step 0 of Phase-2 top half (Figure 5-4) of the multi-resolution control strategy.

As seen from Figure 5-8, the time taken to reach Step 0 of Figure 5-4 is around $4 \times 10^4\ \nu^{-1}$, whereas the time of self-assembly in both Phase-1 and Phase-2 top half are of $\mathcal{O}(10^6)\ \nu^{-1}$. Therefore, this process can be assumed to be instantaneous.

Phase-3

At the end of Phase-2, a vertical line of repulsive external charges is placed in the middle of the domain, dividing it into four quadrants. As in Phase-2, the high repulsive charges form an energy barrier, allowing the four quadrants to be simulated independently, and yielding four independent ergodic spaces.

For Phase-3 of the multi-resolution control strategy, Figure 5-9 shows the control policy in the top-left quadrant of the domain. The aim is to bring the nanoparticles to the configuration shown in the Step 2 of the figure. The strategy starts with a strong attractive external charge in a location that is determined by the final (steady-state) locations of the external charges. In this case, the attractive charge is placed in between the locations of the final external charges. The strengths and the locations of

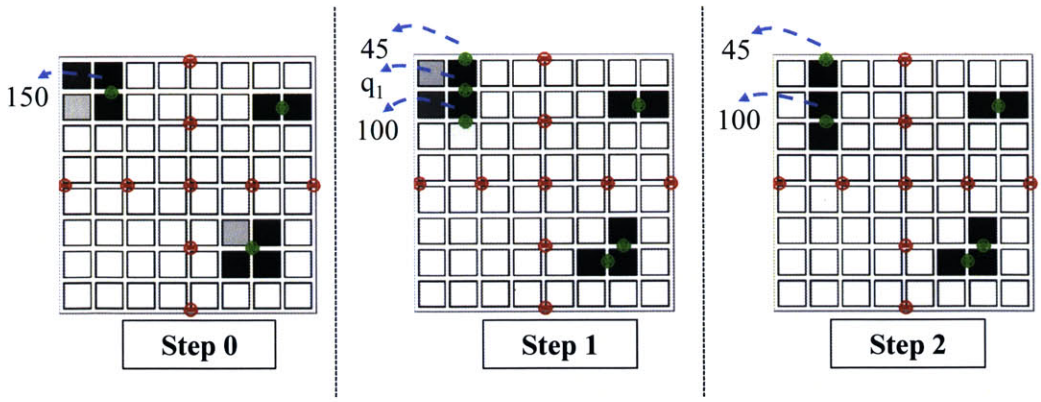

Figure 5-9: Figure illustrating the locations of the external controls in the top-left quadrant of the domain, in the various steps for Phase-3 of multi-resolution control strategy. The expected locations of nanoparticles at the end of each step are also shown in the figure. A vertical line of repulsive charges is now added along the middle of the domain to separate the domain into four parts, and treat each one independently.

the external charges in Step 2 are determined by the optimal static solution developed in Solis et al. [63] in their first paper. The simulation starts at the end of Step 0, when the initial charge is mapped on two new charges on either side (Step 1). The self-assembly in this phase of the multi-resolution control strategy requires a smaller number of steps. The optimization variables are q_1 and t_N. The final steady-state probability can be calculated easily in this case as the entire configuration space can be modeled. Based on the final steady-state probability, the termination criterion probability (p_{min}) of 0.99 is used.

Table 5.4 shows the values obtained from the optimization. It also contains the constraint used for q_1. The value of t_N obtained from the optimization was 209,207 ν^{-1}.

Table 5.4: The values of parameters used from the optimization of the top left quadrant in Phase-3 of the multi-resolution control strategy. The strengths of the repulsive external charges used are -50 in all the steps. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm, $k_B T = 0.7$ kcal mol^{-1} and $a = 10^5$ kcal mol^{-1}nm^6.

	Step 1
Value	0 (q_1)
Range	$0 \leq q_1 \leq 50$

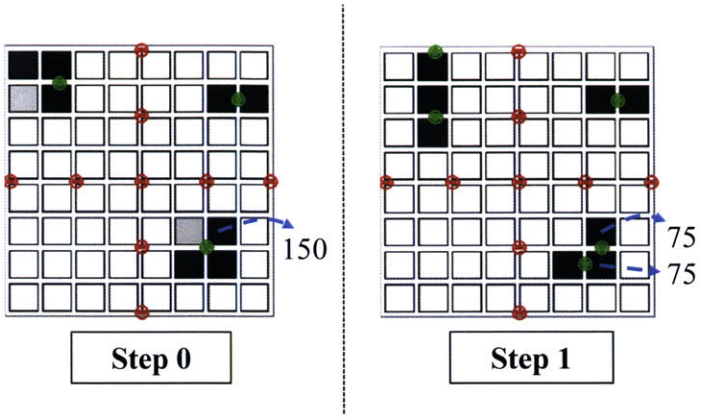

Figure 5-10: Locations of the external controls in the bottom-right quadrant of the domain, in the various steps for Phase-3 of multi-resolution control strategy. The expected locations of nanoparticles at the end of each step are also shown in the figure. A vertical line of repulsive charges has been added along the middle of the domain to separate the domain into four parts.

Figure 5-10 shows the control strategy in the bottom-right quadrant of the domain. Step 1 shows the final desired configuration that the strategy aims to achieve. The optimal control policy involves only one step; the mapping of the initial charge in the center of the domain with two charges that correspond to the optimal static solution for the final desired arrangement, as computed by the method described by Solis et al. [63] in their first paper. The optimal control policy involves one step, and the only optimization variable is t_N, whose optimal value is found to be 150 ν^{-1}.

There was no need for an optimization problem in the top-right quadrant of the domain, as only one attractive external charge is required to achieve the desired configuration of the nanoparticles (Figure 5-9 and Figure 5-10).

Figure 5-11 shows the simulation results for Phase-3 of the multi-resolution control strategy. Figure 5-11(a) shows the probability of the desired configuration in the top-left quadrant. Note that from Table 5.4, $q_1 = 0$, which implies that Step 2 is implemented directly after Step 0. Figure 5-11(b) shows the probability of the desired configuration in the bottom-right quadrant in blue.

In summary, an optimal control problem formulation and solution strategy is proposed that allows self-assembly of nanoparticles into desired structures. The methodology uses AFSP method and multi-resolution approach to make the optimization

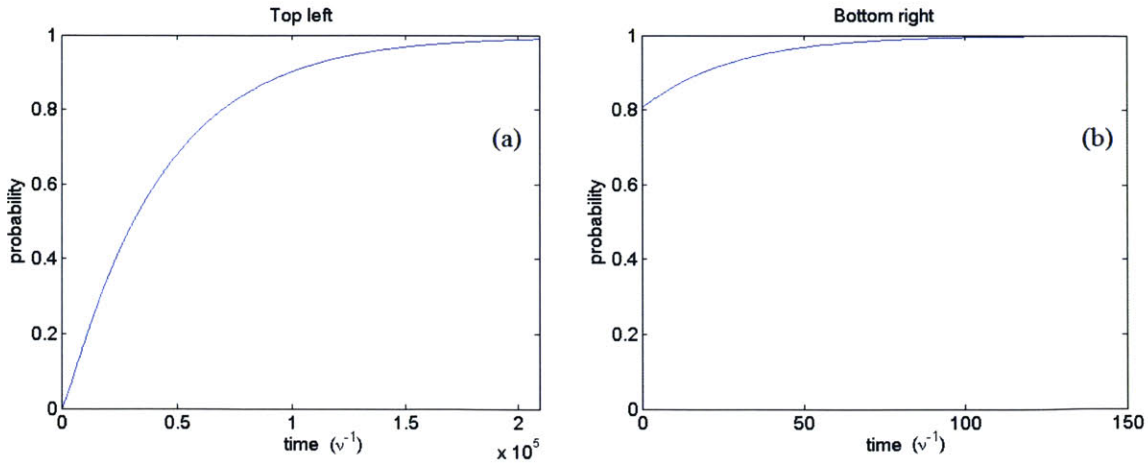

Figure 5-11: Graphs showing the probabilities of the desired arrangements in Phase-3 of the multi-resolution control strategy: (a) Top-left quadrant. (b) Bottom-right quadrant.

problem tractable, and the minimum-time objective function ensured that the system avoids any kinetic traps.

Chapter 6

Rotation of nanoparticles

In the next phase of this study, self-assembly of nanoparticles with "sticky" edges is considered. For example, as proposed in Chapter 1, a DNA tile could be a nanoparticle building block for the nanostructures. The edges of these DNA tiles can be functionalized with specific DNA sequences (e.g. ATCGAA) which have strong binding affinities for their complementary sequences present in other tiles (e.g. TAGCTT). The binding affinities between two functionalized edges can be expressed with the help of a function that characterizes the hydrogen bonding interactions between the complementary base pairs. The well-matched edges will experience strong binding affinities, and the umatched or partially matched edges will experience weak affinities. Weak binding causes errors in the self-assembly process [26] and should be taken into account.

However, the effect of the binding forces between edges is greatly dependent on temperature. At higher temperatures, the tiles move faster due to the Brownian motion and hence it is less likely for any two matched edges to remain close enough for the binding to occur. Generally, premature binding is not preferred during self-assembly, and therefore the temperature during most of the multi-resolution phases can be kept high enough to prevent any specific DNA tile binding. The system can be then cooled down in the final phase of the multi-resolution approach, allowing edge binding.

6.1 Modified Model

There are two additional considerations that need to be included in the master equation model in order to account for the functionalized edges in particles: hydrogen bonding between the functionalized edges, and the rotational configurations that are possible for each nanoparticle. For the first case, since each nanoparticle or DNA tile will have the same number of DNA bases, it can be assumed that all the particles will have the same charge. Therefore, no changes are required to the contributions from Van der Waals or electrostatic part of the energy. However, the binding forces between matched edges (or penalty for unmatched edges) will have an impact on the total energy of a configuration. To tackle this consideration, a constant factor is subtracted (or added) to the energy expression for any configuration α:

$$\mathcal{H}_\alpha = \sum_{i=1}^{V}\sum_{k=1}^{N_{cp}} \frac{q_k q_p}{|r_{i,k}|} z_{\alpha,i} + \sum_{i=1}^{V}\sum_{j\neq i}^{V} z_{\alpha,i}\left(\frac{q_p^2}{|r_{i,j}|} - \frac{a}{|r_{i,j}|^6}\right) z_{\alpha,j} - \sum_{b=1}^{B} E_b, \tag{6.1}$$

where B is the number of bonds between the edges of nanoparticles, E_b is a constant value that depends on the matched functionalized edges between adjacent nanoparticles. Since hydrogen bonding is electrostatic in nature [5], the order of magnitude of E_b is same as that of the electrostatic energy between any two nanoparticles. However, hydrogen bonding strongly depends on the solvent conditions [5]. For example, hydrogen bonding is known to be quite weak in protic solvents such as water and alcohol. Therefore, the energy contribution from hydrogen bonding can be easily varied by changing the solvent conditions under which the self-assembly process is taking place.

The second new consideration is the inclusion of rotational configurations of the nanoparticles. Nanoparticles, depending on the number of functionalized edges, may contain one, two, or four unique rotational orientations. This increases the total number of possible configurations of the nanoparticles. However, as mentioned before, these new configurations need to be considered only when the binding energy is significant enough, which happens when the temperature is low enough or when the

hydrogen bonding is high due to the presence of certain solvents.

6.2 Case Study

A case study similar to previous case studies is considered. In this case study, it is assumed that self-assembly until the last multi-resolution phase has already been achieved, and one is left with 4 particles in each top-left and top-right part of the domain, in the desired configuration. We will assume that there are only two functionalized edges possible in a nanoparticle, and they are on the adjacent edges of the nanoparticle. Therefore, there are only 2 possible rotational orientations for any nanoparticle. Every configuration of the nanoparticles has 2^4 possible rotational orientations, where each of the four nanoparticles can rotate in two ways. Since there are 1,820 possible positional configurations (4 particles in 16 lattice cells), the total number of configurations with different rotational orientations are 29,120. For the purpose of this simulation, in addition to the step-wise jump of a nanoparticle to an adjacent empty cell, we will also allow for a 90° rotational change for any nanoparticle. However, only one of these changes are possible at a time, either jump of a nanoparticle to an adjacent cell or rotation of a nanoparticle by 90°.

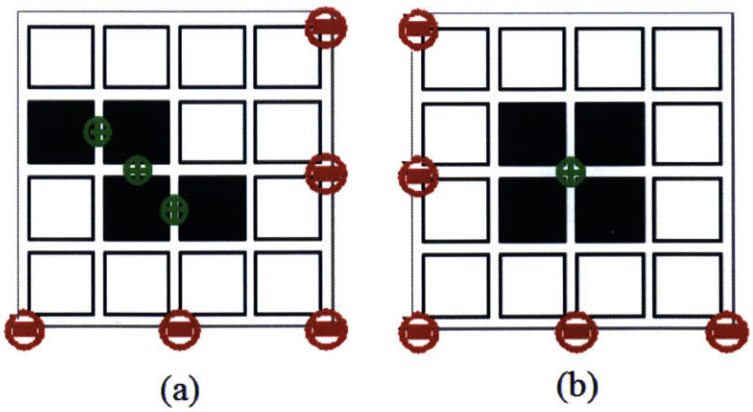

Figure 6-1: Figure showing the desired configuration of the nanoparticles at the end of the multi-resolution approach, i.e., using the procedure given in Chapter 5. The top-left quadrant has the configuration shown in (a) and the top-right quadrant has the configuration shown in (b).

Figure 6-1(a) shows the initial configuration of the top left part of the domain for this case study, and Figure 6-1(b) shows the initial configuration for the top right part. These configurations can be achieved using the multi-resolution based control policy proposed in Chapter 5. This process is assumed to have been performed at a high enough temperature so that the hydrogen bonding between matched edges does not play a significant role. Therefore the nanoparticles may not be aligned based on their edges.

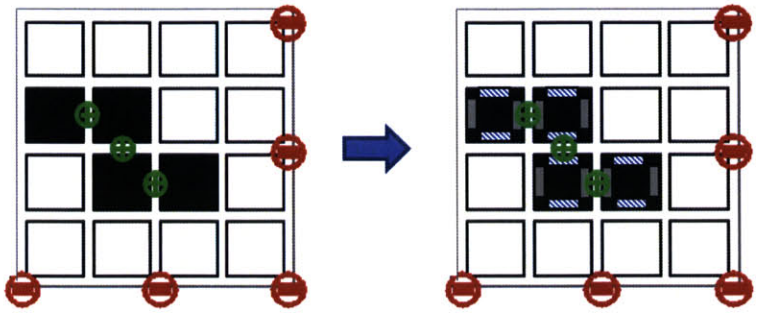

Figure 6-2: Figure showing the desired orientation of the nanoparticles in the final configuration in the top-left part of the domain. The case study aims to align the nanoparticles in any possible orientation, shown in (a), into the desired orientation shown in (b).

Figure 6-2 shows the final desired aligned configuration. Note that in Figures 6-1 and 6-2(a) the functionalized edges are present, but not displayed. This is because the functionalized edges are not important in those configurations, and the nanoparticles can be in any possible orientation. In Figure 6-2(b), the functionalized edges are important and it is desired to achieve the displayed orientation of the nanoparticles. The process of alignment of the nanoparticles into the desired orientation will be studied in this case study.

The two functionalized edges in Figure 6-2(b) are indicated with a gray box and a shaded box. With these two functionalized edges, there are three kinds of bonding affinities possible in this system - gray edge next to a gray edge, gray edge next a shaded edge and shaded edge next to a shaded edge. Hence, in Equation (6.1), E_b can take three possible values, as indicated in the Table 6.1 below.

Table 6.1: Definition of the energy parameters associated with the various binding energies that are possible between any two nanoparticles.

Type of bonding	Parameter
gray edge next a gray edge	E_{gg}
gray edge next a shaded edge	E_{sg}
shaded edge next a shaded edge	E_{ss}

The parameter values used in the simulation are contained in Table 6.2.

Table 6.2: External charge values for the simulation of top-left part of the grid. The strengths of the repulsive charges used are -50. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm and $a = 10^5$ kcal mol^{-1}nm^6.

Strength of attractive external charges $((q_k)/(q_p))$ (top to bottom)
18.2, 50, 50

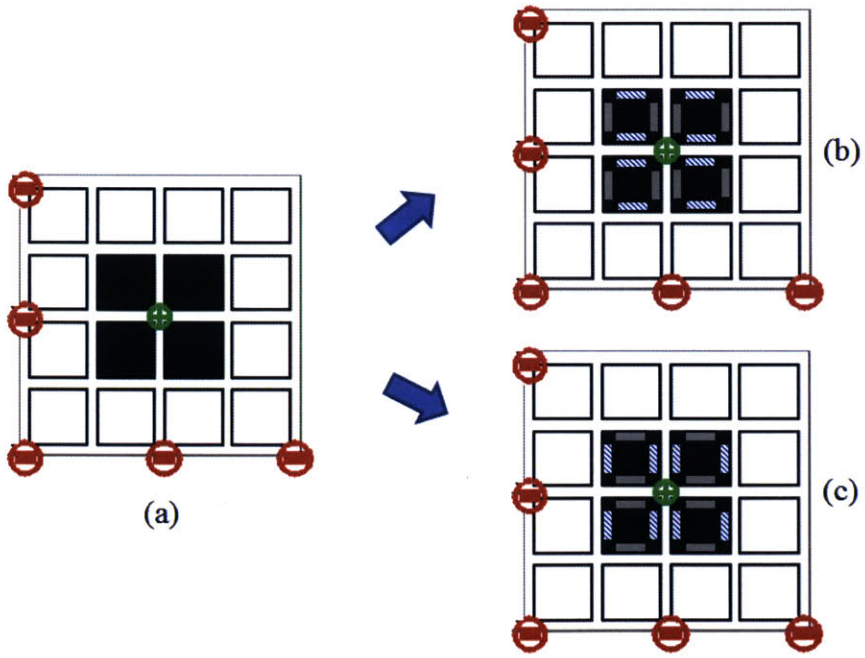

Figure 6-3: Figure showing the desired orientation of the nanoparticles in the final configuration in the top-right part of the domain. The case study aims to align the nanoparticles in any possible orientation, shown in (a), into the desired orientation shown in (b) and (c).

Similar simulations are also carried out for the top-right part of the grid. Figure

6-3(a) shows the initial configuration of the nanoparticles after the multi-resolution based self-assembly, where any orientation of the nanoparticles is possible. In top-right part of the grid, for the desired orientation of the nanoparticles, there are two gray-gray edge bonds and two shaded-shaded edge bonds. Hence, there are two orientations of the nanoparticles that have the same energies. Note that these two orientations are the same for the purposes of the self-assembly process. This is because the structure shown in Figure 6-3(b) can be obtained by just rotating the structure shown in Figure 6-3(c). This case study studies the process of alignment of the nanoparticles into the desired orientation, as shown in Figure 6-3(b) and (c).

The strength of attractive external charge values for the simulation of top right part of the grid used is +50 and the strengths of the repulsive charges used are -50. Other parameters used in the simulation are $k_C q_p^2 = 1$ kcal mol^{-1}nm and $a = 10^5$ kcal mol^{-1}nm^6.

6.2.1 Reduction in temperature

For the top-left part of the grid, the initial configuration (Figure 6-2(a)) is assumed to have been obtained using the parameter values shown in the second column of Table 6.3. The temperature is then reduced from 0.6 kcal mol^{-1} to 0.1 kcal mol^{-1}, thus enabling strong binding affinities between adjacent edges.

Table 6.3: Values for the various binding energies and the temperature. The second column contains the parameter values that were used to obtain the configurations in Figures 6-2(a) and 6-3(a). The third column contains the parameters that were used for the simulation of the case study, where the temperature was reduced.

Parameter	Value at initial time	Value used for simulation
E_{gg} [kcal mol^{-1}]	1.0	1.0
E_{sg} [kcal mol^{-1}]	-1.0	-1.0
E_{ss} [kcal mol^{-1}]	0.2	0.2
$k_B T$ [kcal mol^{-1}]	0.6	0.1

As seen in Figure 6-2(b), the desired orientation of the nanoparticles is such that the shaded edge should always face upwards. As a result, in the desired orientation,

there are two bonds between the gray edges and one bond between the shaded edges. Therefore, in order to maximize the probability of this orientation, the energy of the system reduces the most when gray-gray edge binds, i.e., by E_{gg}. It is also assumed that the gray edge does not bind with the shaded edge, and causes steric hindrance instead. Therefore, it results in the increase in the energy of the orientation, which is indicated by negative E_{sg}.

Since the model is a phenomenological one, the parameter values were chosen using trial and error. As mentioned before, these numbers are chosen such that they are the same order of magnitude as the electrostatic forces between two nanoparticles, and these numbers can be easily modified by the appropriate choice of solvent.

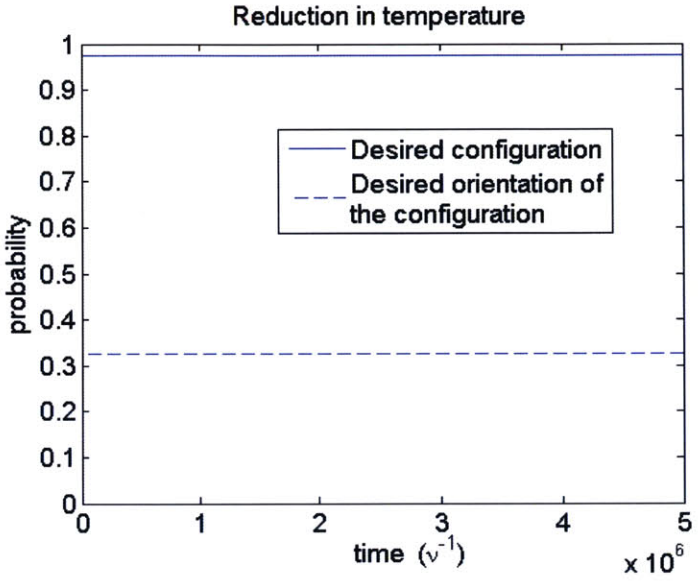

Figure 6-4: Graph showing the simulation results for the top-left part of the domain when the temperature is reduced. The blue bold line shows the probability of the desired configuration, irrespective of the orientation. The blue dashed line shows the probability of the desired orientation of the nanoparticles in that configuration.

Figure 6-4 shows the results from the simulation at reduced temperature. The blue bold line shows the probability of the desired configuration, which has a high value of 0.98 initially, and does not change at all when the temperature is reduced. The blue dashed line tracks the probability of the oriented nanoparticles in the desired configuration. Although the probability of the desired orientation of nanoparticles is expected to increase due to the increased affinity between the matched edges,

the dynamics of reaching this orientation is very slow. After $t = 5 \times 10^6\ \nu^{-1}$, the probability increases by only 1×10^{-5}.

The simulation was repeated for the top-right part of the grid (Figure 6-3).

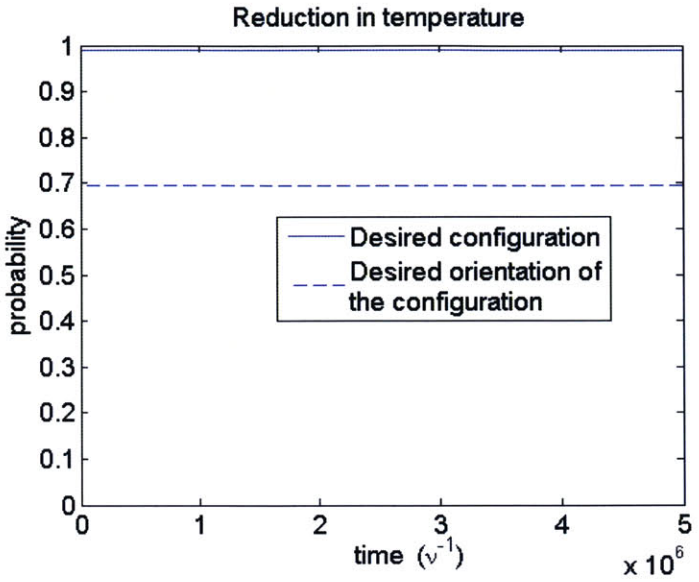

Figure 6-5: Graph showing the simulation results for the top-right part of the domain when the temperature is reduced. The blue bold line shows the probability of the desired configuration, irrespective of the orientation. The blue dashed line shows the probability of the desired orientation of the nanoparticles in that configuration.

Figure 6-5 shows the results from the simulation at reduced temperature. The blue bold line shows the probability of the desired configuration, which has a high value of 0.99 initially, and does not change at all when the temperature is reduced. The blue dashed line tracks the probability of the oriented nanoparticles in the desired configuration and it includes both the structures shown in Figure 6-3(b) and (c). Although the probability of the desired orientation of nanoparticles is expected to increase due to the increased affinity between the matched edges, the dynamics of reaching this orientation is very slow. After $t = 5 \times 10^6\ \nu^{-1}$, the probability increases by only 2×10^{-5}.

6.2.2 Changing the solvent

Since the dynamics of the alignment of nanoparticles in the desired orientation at low temperature is quite slow, an alternate methodology is proposed. In this new simulation, the temperature is kept constant, and the hydrogen bonding affinities are increased instead. The new set of parameter values are given by Table 6.4. As mentioned earlier, this can be achieved by changing the solvent conditions under which the process of self-assembly takes place.

Table 6.4: Values for the various binding energies and the temperature. The second column contains the parameter values that were used to obtain the configurations in Figures 6-2(a) and 6-3(a). The third column contains the parameters that were used for the simulation of the case study, where the binding affinities were increased.

Parameter	Value at initial time	Value used for simulation
E_{ss}	1.0	3.0
E_{sg}	-1.0	-3.0
E_{gg}	0.2	0.2
k_BT	0.6	0.6

At the new parameter settings, the bonding affinity for gray-gray edges, E_{gg}, is increased in order to force the nanoparticles to maximize the bonds between gray edges. The high negative value of E_{sg} also penalizes the unmatched edges in the system.

Figure 6-6 shows the results from the simulation for increased bonding affinities. The blue bold line shows the probability of the desired configuration, which has a high value of 0.98 initially, and does not change at all when the conditions change. The blue dashed line tracks the probability of the oriented nanoparticles in the desired configuration. Since the temperature is kept high, the fast dynamics of the system is maintained while rapidly aligning towards the desired orientation.

Figure 6-7 shows the results from the simulation for increased bonding affinities. The blue bold line shows the probability of the desired configuration, which has a high value of 0.99 initially, and does not change at all when the conditions change. The blue dashed line tracks the probability of the oriented nanoparticles in the desired

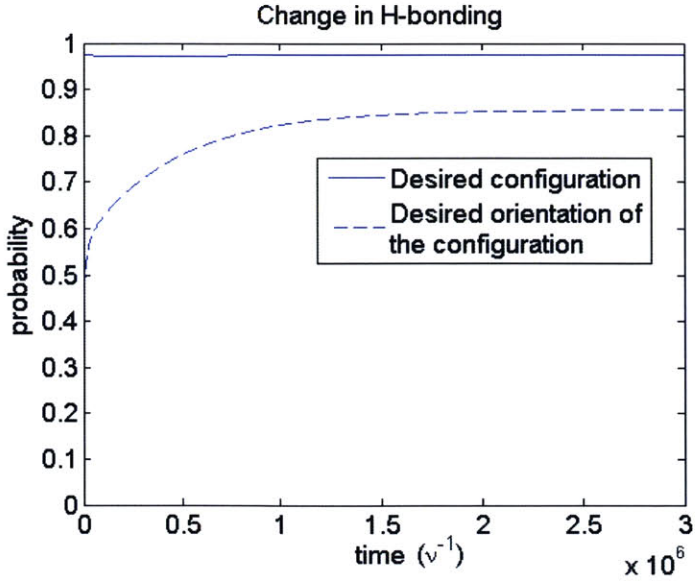

Figure 6-6: Graph showing the simulation results for the top-left part of the domain when the bonding affinities are increased. The blue bold line shows the probability of the desired configuration, irrespective of the orientation. The blue dashed line shows the probability of the desired orientation of the nanoparticles in that configuration.

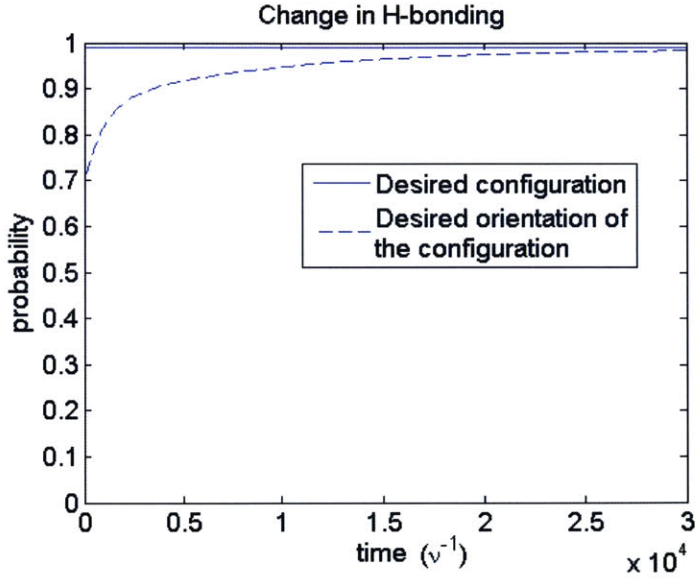

Figure 6-7: Graph showing the simulation results for the top-right part of the domain when the bonding affinities are increased. The blue bold line shows the probability of the desired configuration, irrespective of the orientation. The blue dashed line shows the probability of the desired orientation of the nanoparticles in that configuration.

configuration. Since the temperature is kept high, the fast dynamics of the system is maintained while rapidly aligning towards the desired orientation.

In conclusion, temperature and solvent conditions are important factors that influence the orientation of the nanoparticles. Preliminary results indicate that reduction in temperature causes the dynamics of the alignment of nanoparticles into a desired orientation to slow down. A better approach is to change the solvent, which will in turn influence the binding affinities. However, the binding affinities cannot be arbitrarily large, as it will interfere with the external charges, and cause the probability of the system being in desired configuration to fall. Therefore, optimal control should be used to find the best solvent conditions, such that the probability of desired orientation of nanoparticles in the desired configuration is maximized.

Chapter 7

Conclusion and future directions

In the previous chapters, a guided self-assembly of nanostructures is achieved. Suitable numerical methods and control strategies are developed to simplify the simulation and optimization of this self-assembly process. In Chapter 3, an algorithm is presented to simulate the dynamics of directed self-assembly of nanoparticles, through the model reduction of master equations. The approach is based on a novel adaptive version of the finite state projection method. Event detection is used to determine effectively those points in time at which the projection space should be modified. Consequently, the method does not require a priori information for changes in projection space, which is particularly useful when the relevant time scales for modification of the projection space vary significantly, as is typically the case for directed self-assembly. Furthermore, transition rates are analyzed systematically to anticipate the configurations that will become relevant to prevent a steep increase in the error from model reduction. A case study illustrates the effectiveness of the approach. The number of selected master equations is at least 100-fold smaller for the studied cases, compared to the simulation of the full system.

In Chapter 4, a control strategy that allows efficient implementation of AFSP is presented that provides an effective path towards reliable self-assembly. The strategy is based on a multi-resolution view of the self-assembled system. Over time, it judiciously decomposes the state space into non-ergodic subspaces, by creating new and shifting energy wells, thus moving the nanoparticles along with them. A case

study illustrates the proposed strategy and demonstrates the advantage over static strategies, which would have attempted to reach the final desired structure through implementation of the static solution at the initial time. The strategy enables the nanoparticles to follow a dynamic path towards self-assembly that avoids any kinetic traps. The resulting control policy is not optimal in any sense. The strengths of the charges employed were determined *a priori* by trial and error.

In Chapter 5, an optimal control problem formulation and a strategy for its solution have been proposed for the design of optimal control policies that bring an initial random configuration of nanoparticles to a configuration with desired geometry, in minimum time. External charges (attractive or repulsive) act as the controls, whose locations and intensities vary over time. The minimum-time objective of the optimal control policy ensures that the system does not fall into a kinetic trap. In order to guarantee the optimality of the control policy in the presence of possible discrete transitions in the projection space, an expanded projection space that includes all pertinent configurations is determined *a priori* with the help of an AFSP simulation. The validity of the projection space is maintained with the help of additional constraints on the external charges.

In Chapter 6, the nanoparticles are assumed to have been self-assembled into their respective desired locations, but are not in the desired rotational orientation. Due to the binding affinities between certain edges, various external factors, which rotate the nanoparticles to the desired orientation, are identified, namely, temperature and the external solvent. Temperature slows down the nanoparticles, enabling their edges to bind, but this results in a drastic slowing of the dynamics of the system. Changing the solvent in order to increase the hydrogen bonding energy between complementary edges produces the same effect without slowing down the dynamics of the system. Optimal control should be used to find the best solvent conditions, such that the probability of desired orientation of nanoparticles in the desired configuration is maximized.

7.1 Future directions - Sub-Assemblies

The next step in the process of self-assembly is the fabrication of complex structures. With the help of the approach described in this thesis, smaller, intermediate structures (called sub-assemblies) can be fabricated reliably. Then, those intermediate structures can be combined to form the final complex nanostructure.

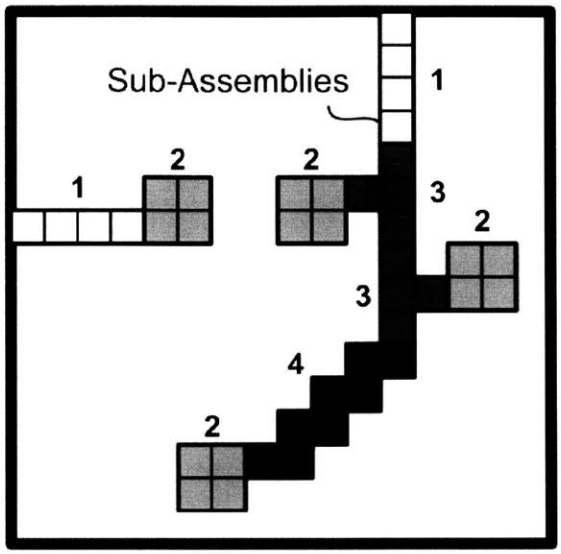

Figure 7-1: Figure showing the fabrication of the desired complex nanostructure by dividing the structure into finer sub-assemblies (labelled 1 through 4).

For example, in Figure 7-1, four types of sub-assemblies can be fabricated (labeled 1 through 4) using the methodology described in this thesis. These sub-assemblies are then combined and positioned accordingly to form the final complex nanostructure. The multi-resolution approach can be used in combination with the sub-assembly approach, thus enabling the formation of the sub-assemblies.

The sub-assemblies can be fabricated and transported to the main fabrication area. These sub-assemblies can be functionalized in such a way that when added to the domain in a specific order under the influence of dynamically controlled external charges, they bind to the exact position as in the final nanostructure. The sub-assemblies will self-assemble into the scaffolding superstructure, as shown in Figure 7-2. By controlling the rate and order at which the sub-assemblies are introduced

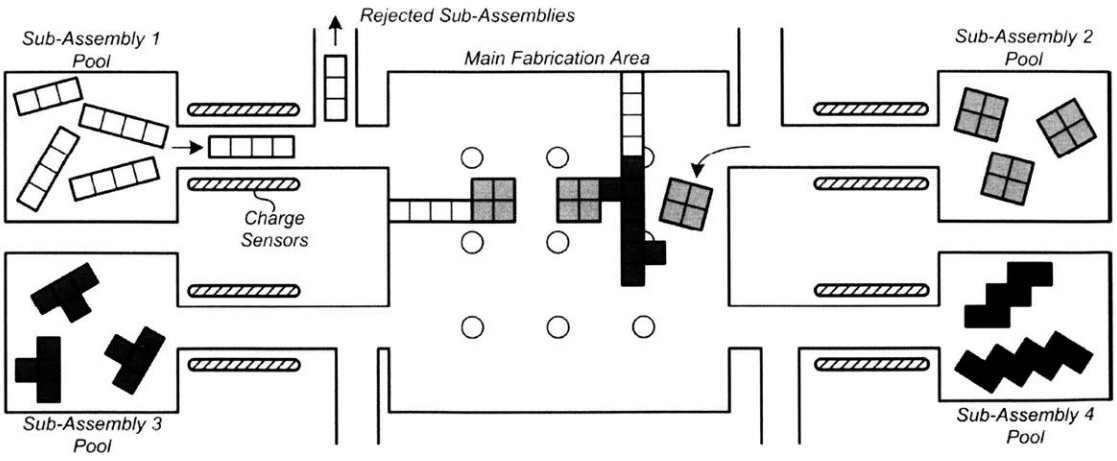

Figure 7-2: Figure showing the final stage of the self-assembly process, where the sub-assemblies are combined into the final nanostructure under the influence of the external charges.

into the fabrication area, one has complete control over the self-assembly process.

Bibliography

[1] Joanna Aizenberg, Paul V. Braun, and Pierre Wiltzius. Patterned colloidal deposition controlled by electrostatic and capillary forces. *Phys. Rev. Lett.*, 84:2997–3000, 2000.

[2] Keith D. Ball and R. Stephen Berry. Realistic master equation modeling of relaxation on complete potential energy surfaces: Kinetic results. *The Journal of Chemical Physics*, 109(19):8557–8572, 1998.

[3] Martin Z. Bazant and Todd M. Squires. Induced-charge electrokinetic phenomena: Theory and microfluidic applications. *Phys. Rev. Lett.*, 92:066101, 2004.

[4] R. Stephen Berry and Ralph Breitengraser-Kunz. Topography and dynamics of multidimensional interatomic potential surfaces. *Phys. Rev. Lett.*, 74:3951–3954, May 1995.

[5] Kyle J. M. Bishop, Christopher E. Wilmer, Siowling Soh, and Bartosz A. Grzybowski. Nanoscale forces and their uses in self-assembly. *Small*, 5(14):1600–1630, 2009.

[6] Peter N. Brown, George D. Byrne, and Alan C. Hindmarsh. VODE: A variable-coefficient ODE solver. *SIAM Journal on Scientific and Statistical Computing*, 10(5):1038–1051, 1989.

[7] Guangjun Cheng, Danilo Romero, Gerald T. Fraser, and A. R. Hight Walker. Magnetic-field-induced assemblies of cobalt nanoparticles. *Langmuir*, 21(26):12055–12059, 2005.

[8] Marek Cieplak, Malte Henkel, Jan Karbowski, and Jayanth R. Banavar. Master equation approach to protein folding and kinetic traps. *Phys. Rev. Lett.*, 80:3654–3657, 1998.

[9] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. Special Issue: Diffusion Maps and Wavelets.

[10] Marie-Nathalie Contou-Carrere, Vassilios Sotiropoulos, Yiannis N. Kaznessis, and Prodromos Daoutidis. Model reduction of multi-scale chemical Langevin equations. *Systems & Control Letters*, 60(1):75 – 86, 2011.

[11] Levente Csoka, Ingrid C. Hoeger, Perry Peralta, Ilona Peszlen, and Orlando J. Rojas. Dielectrophoresis of cellulose nanocrystals and alignment in ultrathin films by electric field-assisted shear assembly. *Journal of Colloid and Interface Science*, 363(1):206 – 212, 2011.

[12] Payel Das, Mark Moll, Hernán Stamati, Lydia E. Kavraki, and Cecilia Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences*, 103(26):9885–9890, 2006.

[13] Eddie Davis and Marianthi Ierapetritou. A centroid-based sampling strategy for kriging global modeling and optimization. *AIChE Journal*, 56(1):220–240, 2010.

[14] A.F. Demirörs, P.M. Johnson, C.M. van Kats, A. van Blaaderen, and A. Imhof. Directed self-assembly of colloidal dumbbells with an electric field. *Langmuir*, 26(18):14466–14471, 2010.

[15] Eric K. Drexler. *Nanosystems: Molecular Machinery, Manufacturing and Computation.* John Wiley & Sons, New York, 1992.

[16] Stefan Fischer and Martin Karplus. Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom. *Chemical Physics Letters*, 194(3):252 – 261, 1992.

[17] Santos Galán, William F. Feehery, and Paul I. Barton. Parametric sensitivity functions for hybrid discrete/continuous systems. *Applied Numerical Mathematics*, 31(1):17 – 47, 1999.

[18] Byron D. Gates, Qiaobing Xu, Michael Stewart, Declan Ryan, C. Grant Willson, and George M. Whitesides. New approaches to nanofabrication: Molding, printing, and other techniques. *Chemical Reviews*, 105(4):1171–1196, 2005.

[19] P. Gill, W. Murray, and M. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[20] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[21] Daniel T. Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1):297–306, 2000.

[22] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115(4):1716–1733, 2001.

[23] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, 2007.

[24] Rudiyanto Gunawan, Yang Cao, Linda Petzold, and Francis J. Doyle. Sensitivity analysis of discrete stochastic systems. *Biophysical Journal*, 88(4):2530 – 2540, 2005.

[25] Philip Hartman. *Ordinary Differential Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2002.

[26] M. Hashempour, Z.M. Arani, and F. Lombardi. Healing assessment of tile sets for error tolerance in DNA self-assembly. *Nanobiotechnology, IET*, 2(4):81–92, December 2008.

[27] Desmond J. Higham. Modeling and simulating chemical reactions. *SIAM Review*, 50(2):347–368, 2008.

[28] Ahmed E. Ismail, Gregory C. Rutledge, and George Stephanopoulos. Multiresolution analysis in statistical mechanics. I. Using wavelets to calculate thermodynamic properties. *The Journal of Chemical Physics*, 118(10):4414–4423, 2003.

[29] Ahmed E. Ismail, Gregory C. Rutledge, and George Stephanopoulos. Using wavelet transforms for multiresolution materials modeling. *Computers & Chemical Engineering*, 29(4):689 – 700, 2005.

[30] Ahmed E. Ismail, George Stephanopoulos, and Gregory C. Rutledge. Multiresolution analysis in statistical mechanics. II. The wavelet transform as a basis for Monte Carlo simulations on lattices. *The Journal of Chemical Physics*, 118(10):4424–4431, 2003.

[31] Hannes Jónsson, Greg Mills, and Karsten W. Jacobsen. Nudged elastic band method for finding minimum energy paths of transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulations*, Proceedings of the International School of Physics, pages 385–404, 1998.

[32] Jaime J. Juárez and Michael A. Bevan. Feedback controlled colloidal self-assembly. *Advanced Functional Materials*, 22(18):3833–3839, 2012.

[33] Jaime J. Juárez, Sarah E. Feicht, and Michael A. Bevan. Electric field mediated assembly of three dimensional equilibrium colloidal crystals. *Soft Matter*, 8:94–103, 2012.

[34] Jaime J. Juárez, Pramod P. Mathai, J. Alexander Liddle, and Michael A. Bevan. Multiple electrokinetic actuators for feedback control of colloidal crystal size. *Lab Chip*, 12:4063–4070, 2012.

[35] Sëma Kachalo, Hsiao-Mei Lu, and Jie Liang. Protein folding dynamics via quantification of kinematic energy landscape. *Phys. Rev. Lett.*, 96:058106, 2006.

[36] S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988.

[37] Richard A. Kiehl. DNA-directed assembly of nanocomponents for nanoelectronics, nanophotonics, and nanosensing. *Proc. SPIE*, 6768:67680Z–67680Z–7, 2007.

[38] Seong Koh. Strategies for controlled placement of nanoscale building blocks. *Nanoscale Research Letters*, 2:519–545, 2007.

[39] Joanna Kolny, Andreas Kornowski, and Horst Weller. Self-organization of cadmium sulfide and gold nanoparticles by electrostatic interaction. *Nano Letters*, 2(4):361–364, 2002.

[40] Sergei V. Krivov and Martin Karplus. Hidden complexity of free energy surfaces for peptide (protein) folding. *Proceedings of the National Academy of Sciences of the United States of America*, 101(41):14766–14770, 2004.

[41] S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1393–1403, 2006.

[42] Richard Lakerveld, George Stephanopoulos, and Paul I. Barton. A master-equation approach to simulate kinetic traps during directed self-assembly. *The Journal of Chemical Physics*, 136(18):184109, 2012.

[43] Mirjam E. Leunissen, Hanumantha Rao Vutukuri, and Alfons van Blaaderen. Directing colloidal self-assembly with biaxial electric fields. *Advanced Materials*, 21(30):3116–3120, 2009.

[44] J. Christopher Love, Lara A. Estroff, Jennah K. Kriebel, Ralph G. Nuzzo, and George M. Whitesides. Self-assembled monolayers of thiolates on metals as a form of nanotechnology. *Chemical Reviews*, 105(4):1103–1170, 2005.

[45] Manish Mittal, Pushkar P. Lele, Eric W. Kaler, and Eric M. Furst. Polarization and interactions of colloidal particles in ac electric fields. *The Journal of Chemical Physics*, 129(6):064513, 2008.

[46] Victor Munoz, Eric R. Henry, James Hofrichter, and William A. Eaton. A statistical mechanical model for β-hairpin kinetics. *Proceedings of the National Academy of Sciences*, 95(11):5872–5879, 1998.

[47] Brian Munsky and Mustafa Khammash. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124(4):044104, 2006.

[48] Brian Munsky and Mustafa Khammash. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *Journal of Computational Physics*, 226(1):818 – 835, 2007.

[49] Frank Noé and Stefan Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Current Opinion in Structural Biology*, 18(2):154 – 162, 2008.

[50] Frank Noé, Marcus Oswald, Gerhard Reinelt, Stefan Fischer, and Jeremy C. Smith. Computing best transition pathways in high-dimensional dynamical systems: Application to the $\alpha_l \leftrightharpoons \beta \leftrightharpoons \alpha_r$ transitions in octaalanine. *Multiscale Modeling & Simulation*, 5(2):393–419, 2006.

[51] Sung Ha Park, Peng Yin, Yan Liu, John H. Reif, Thomas H. LaBean, and Hao Yan. Programmable DNA self-assemblies for nanoscale organization of ligands and proteins. *Nano Letters*, 5(4):729–733, 2005. PMID: 15826117.

[52] Taeshin Park and Paul I. Barton. State event location in differential-algebraic models. *ACM Trans. Model. Comput. Simul.*, 6(2):137–165, 1996.

[53] Slaven Peleš, Brian Munsky, and Mustafa Khammash. Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of Chemical Physics*, 125(20):204104, 2006.

[54] Sivaraman Ramaswamy, Richard Lakerveld, Paul I. Barton, and George Stephanopoulos. Controlled formation of nanostructures with desired geometries: Part 3. Dynamic modeling and simulation of directed self-assembly of nanoparticles through adaptive finite state projection. *Industrial & Engineering Chemistry Research*, In Press.

[55] Christopher V. Rao and Adam P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *The Journal of Chemical Physics*, 118(11):4999–5010, 2003.

[56] Muruhan Rathinam, Patrick W. Sheppard, and Mustafa Khammash. Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks. *The Journal of Chemical Physics*, 132(3):034103, 2010.

[57] Nathaniel L. Rosi and Chad A. Mirkin. Nanostructures in biodiagnostics. *Chemical reviews*, 105(4):1547–1562, 2005.

[58] E.N. Rozenvasser. General sensitivity equations of discontinuous systems. *Automat. Remote Control*, (3):400–404, 1967.

[59] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[60] Nadrian C Seeman. DNA engineering and its application to nanotechnology. *Trends in Biotechnology*, 17(11):437 – 443, 1999.

[61] Nina Singhal and Vijay S. Pande. Error analysis and efficient sampling in Markovian state models for molecular dynamics. *The Journal of Chemical Physics*, 123(20):204909, 2005.

[62] Nina Singhal, Christopher D. Snow, and Vijay S. Pande. Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *The Journal of Chemical Physics*, 121(1):415–425, 2004.

[63] Earl O. P. Solis, Paul I. Barton, and George Stephanopoulos. Controlled formation of nanostructures with desired geometries. 1. Robust static structures. *Industrial & Engineering Chemistry Research*, 49(17):7728–7745, 2010.

[64] Earl O. P. Solis, Paul I. Barton, and George Stephanopoulos. Controlled formation of nanostructures with desired geometries. 2. Robust dynamic paths. *Industrial & Engineering Chemistry Research*, 49(17):7746–7757, 2010.

[65] N. Stephanopoulos, E. O. P. Solis, and G. Stephanopoulos. Nanoscale process systems engineering: Toward molecular factories, synthetic cells, and adaptive devices. *AIChE Journal*, 51(7):1858–1869, 2005.

[66] M. Tanase, D. M. Silevitch, A. Hultgren, L. A. Bauer, P. C. Searson, G. J. Meyer, and D. H. Reich. Magnetic trapping and self-assembly of multicomponent nanowires. *Journal of Applied Physics*, 91(10):8549 –8551, 2002.

[67] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[68] W. Theiss and U.M. Titulaer. The systematic adiabatic elimination of fast variables from a many-dimensional Fokker-Planck equation. *Physica A: Statistical Mechanics and its Applications*, 130(1-2):123 – 142, 1985.

[69] John E. Tolsma and Paul I. Barton. Hidden discontinuities and parametric sensitivity calculations. *SIAM Journal on Scientific Computing*, 23(6):1861–1874, 2002.

[70] N.G. Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. North holland, 1992.

[71] David A. Walker, Kevin P. Browne, Bartlomiej Kowalczyk, and Bartosz A. Grzybowski. Self-assembly of nanotriangle superlattices facilitated by repulsive electrostatic interactions. *Angewandte Chemie International Edition*, 49(38):6760–6763, 2010.

[72] David A. Walker, Bartlomiej Kowalczyk, Monica Olvera de la Cruz, and Bartosz A. Grzybowski. Electrostatics at the nanoscale. *Nanoscale*, 3:1316–1344, 2011.

[73] David A. Walker, Christopher E. Wilmer, Bartlomiej Kowalczyk, Kyle J. M. Bishop, and Bartosz A. Grzybowski. Precision assembly of oppositely and like-charged nanoobjects mediated by charge-induced dipole interactions. *Nano Letters*, 10(6):2275–2280, 2010.

[74] G. M. Whitesides, J. P. Mathias, and C. T. Seto. Molecular Self-Assembly and Nanochemistry: A Chemical Strategy for the Synthesis of Nanostructures. *Science*, 254:1312–1319, 1991.

[75] George M. Whitesides and Bartosz Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[76] Hao Yan, Thomas H. LaBean, Liping Feng, and John H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences*, 100(14):8103–8108, 2003.

[77] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H. Reif, and Thomas H. LaBean. DNA-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003.

[78] Huan-Xiang Zhou. A minimum-reaction-flux solution to master-equation models of protein folding. *The Journal of Chemical Physics*, 128(19):195104, 2008.