AFRL-SR-AR-TR-10-0195

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
26/02/2010	FINAL	NOV 2006 - FEB 2010

4. TITLE AND SUBTITLE
ROBUST UNCERTAINTY MANAGEMENT

5a. CONTRACT NUMBER
FA9550-07-C-0024

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
BANASZUK, ANDRZEJ
PASINI, JOSE MIGUEL
BECZ, SANDOR

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNITED TECHNOLOGIES RESEARCH CENTER
411 SILVER LANE, EAST HARTFORD, CT 06108

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
875 NORTH RANDOLPH STREET ROOM 3112
ARLINGTON, VA, 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for public release: distribution is unlimited.

13. SUPPLEMENTARY NOTES
This work was completed in cooperation with Yale, UCSB, AIMdyn, Stanford, Caltech, Princeton, UC Berkeley and MIT.

14. ABSTRACT
The Robust Uncertainty Management (RUM) team created methods and tools for the development of new defense systems composed of complex, nonlinear, multi-scale, and uncertain dynamic networks. Challenge problems included acceleration of Uncertainty Quantification (UQ) in molecular dynamics and design of robust search algorithms for UAV swarms. The UQ methods evaluated include: polynomial chaos-based collocation, kriging-based response surface, and dynamical systems sampling quasi-Monte Carlo. For scalability, these methods were combined with graph-theoretic techniques for network decomposition into weakly connected sub-networks and with wave-form relaxation methods. For search and surveillance, research includes new algorithms for: self-localization of sensor networks, stochastic consensus for distributed estimation, Bayesian estimation, chaotic search, UAV mission planning using pre-computed locally-optimal trajectory elements, and multi-scale optimization. In addition to RUM, a seedling activity for Abstraction Based Complexity Management was conducted as an add-on program. This research focused on cost and complexity drivers in cyber-physical defense systems, and described a notional design paradigm to manage this complexity.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jose Miguel Pasini
U	U	U			

20100622252

ev. 8/98)
Std. Z39.18
ssional 7.0

DYNAMIC NETWORK ANALYSIS FOR ROBUST UNCERTAINTY MANAGEMENT

CONTRACT FA9550–07–C–0024

FINAL REPORT

Andrzej Banaszuk
José Miguel Pasini
United Technologies Research Center
411 Silver Lane, MS 129–15
East Hartford, CT 06108
tel. 860 610 7381, banasza@utrc.utc.com

March 1, 2010

Contents

Chapter 1

Summary

1.1 Objectives

The objective of this project was to develop and demonstrate, in challenge problems selected by DARPA DSO, techniques for managing uncertainty in complex dynamical systems. Out of the original three-year program, the first two phases were executed. Each of these phases was divided into tool development as well as meeting challenge problems to demonstrate the convergence of these tools in concerted efforts.

More explicitly, the overarching goal of the project was to develop tools and workflows for quantifying and managing uncertainty in ways that would perform orders of magnitude faster than Monte Carlo sampling with controlled, provable scaling (preferably linear in the system size). The challenge problems were designed to demonstrate progress toward this ultimate goal.

The focus of Phase I was to show that the techniques selected and developed could be applied correctly to systems of many particles. The two challenge problems for this phase were:

- Self-assembly: Obtain an interaction potential such that a system of particles in a box would spontaneously assemble into a honeycomb structure and compare this to a benchmark solution from the literature [9, 10].
- Phase diagram: Obtain the phase transition temperature of a noble gas physisorbed on a graphite substrate, demonstrating that the team could correctly extract complex emergent behavior of a system of 10,000 particles.

Apart from the further development, selection, and implementation of mathematical tools for uncertainty quantification, Phase II included the following challenge problems:

- Phase diagram with uncertainty: Show orders-of-magnitude speed-up over Monte Carlo sampling in the quantification of uncertainty in a complex, uncertain system. The system chosen for this challenge was a monolayer of carbon monoxide (CO) on graphite in the presence of an

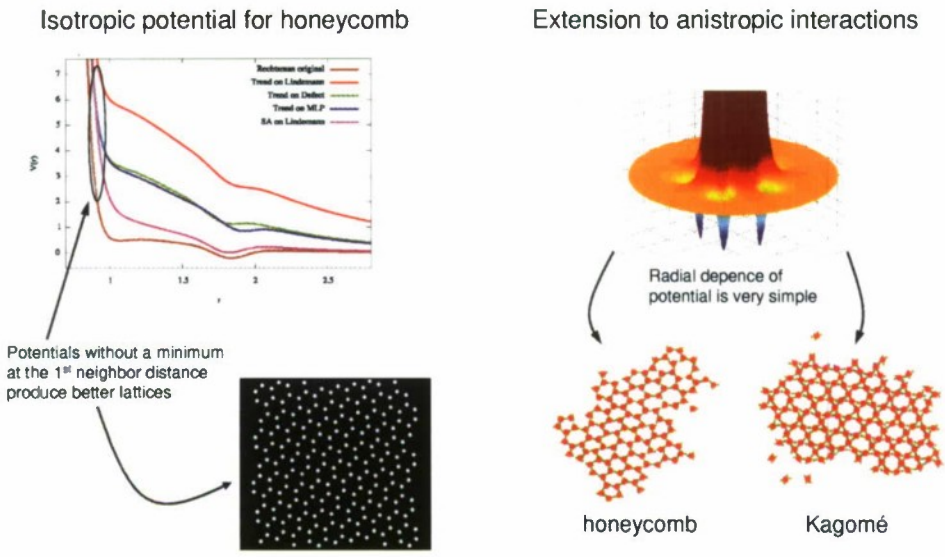


Figure 1.1: Summary of main results for the self-assembly problem.

uncertain level of argon impurities. The system exhibits a complex phase transition [12] and the task was to calculate the transition temperature, including its uncertainty, as a function of the uncertain argon concentration.

- Surveillance: Design search strategies for 50 simulated unmanned aerial vehicles (UAVs) looking for a stationary target in a complex terrain using noisy sensors with uncertain footprint. The strategies had to exhibit shorter search times than straightforward lawnmower patterns while still satisfying constraints on detection (lower bound) and false alarm (upper bound) probabilities.

1.2 Summary of accomplishments

The performance requirements from the Phase I and II challenge problems were met, and in some cases surpassed by orders of magnitude above the required acceleration. Here we summarize the accomplishments directly related to the challenge problems. Chapter 2 summarizes the tools developed, organized by themes.

Phase I self-assembly challenge

- As shown in appendix C.1 and reference [J3], the team first developed several relevant metrics for lattice quality and then applied trend optimization using ridge regression ([J3] and [11]) to obtain solutions superior (in terms of robustness of the self-assembly) to the benchmark [10].

The most robust solutions were particularly counterintuitive in that the resulting interaction potential was purely repulsive. The team also showed that extending the interaction to anisotropic potentials can yield much more robust self-assembly, even of structures that were identified in the literature as impossible to obtain through central potentials ([J9] and [10]).

- The self-assembly problem also exposed the team to the state of the art in molecular dynamics simulation algorithms and, in particular, to the different approaches for simulation of systems of particles with noise, typically due to contact with a thermal bath. This spurred the development of a completely novel approach to simulating noisy systems that preserve the specific structure of the noise in a controlled manner [J2]. In other words, this is the stochastic equivalent of variational integrators, with the difference that instead of preserving energy (as in the widely used Verlet algorithm [4]) they preserve the invariant measure.

- Other mathematical results related to the self-assembly challenge are a proof that central potentials cannot yield certain structures when the system is not confined to a fixed-volume box [J9] and several provably-correct metrics for quantifying the distance between simulation results and target lattices (appendix C.2 and [J11]). Finally, several *tunable* lattice quality measures were developed (appendix D.2 and [J12]). These can be selected to emphasize different desired qualities in the target lattice (e.g., shape vs. density) and can be used for self-assembly as well as for phase diagram computations.

Figure 1.1 summarizes the main results for the self-assembly challenge problem.

Phase I phase diagram challenge

- In appendix E.1 [J17] the team developed a new class of Hidden Markov Models, the finite-rank optimal-prediction (FRO) model, for quickly learning the dynamics of a system. This new tool was used to learn from MD simulations of helium atoms physisorbed on graphite the dynamics of a coarse variable relevant to the phase transition (the potential energy per atom). The transition is then associated with metastability in the spectrum of the Markov model. The team showed that it is faster to directly learn when the spectrum exhibits metastability than to directly simulate the system until it settles into its stationary distribution. The method was later extended into reference [C4], where it was applied to fast decentralized control over networks through the construction of multiple local Markov models.

- Model order reduction was approached from the point of view of data clustering and stochastic modeling. A Markov matrix whose state space is the possible size of clusters can be learned from the simulation of molecular systems at specific conditions, such as temperature, density and pressure with prior belief. The expectation value of an invariant distribution of learned Markov matrix indicates the phase transition of the molecular dynamics system qualitatively while the second largest eigenvalue modulus can be used as a quantitative indicator. As a consequence, the stochastic reduced order model not only reduces the order of the system based on the choice of coarse variable but also provides an insight of macroscopic properties.

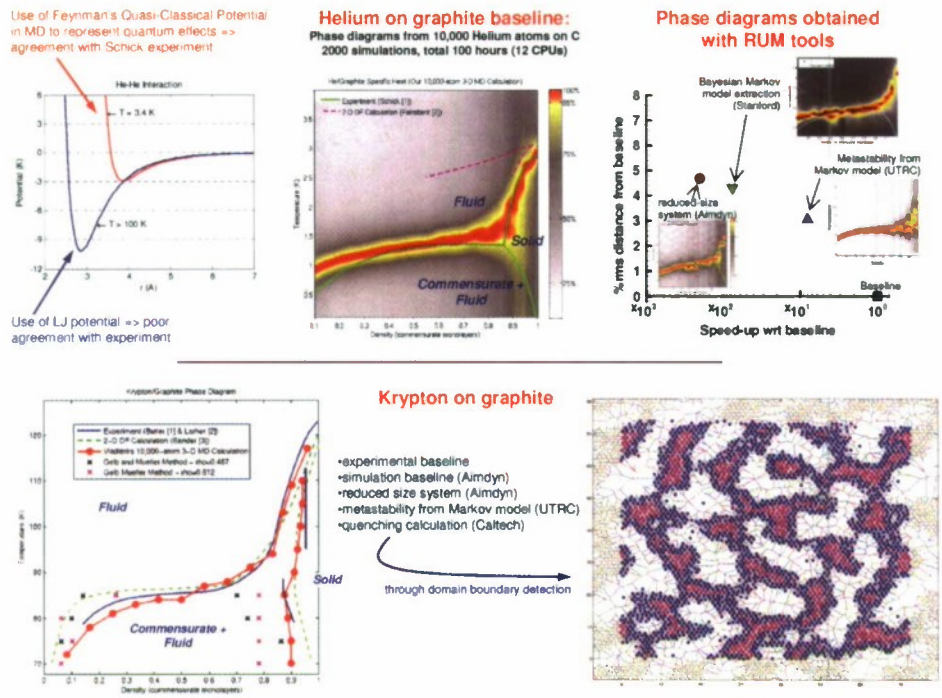


Figure 1.2: Summary of phase I results for the phase diagram problem.

Heuristic graph decomposition Bayesian estimation is shown to be more reliable and robust than maximum likelihood estimation because it reflects prior information on the system.

- Two orders of magnitude acceleration over Molecular Dynamics 10,000 atom baseline was achieved by running a reduced size system of only 100 particles. To assess the error in the phase transition temperature determined using lower number of atoms in MD simulation, convergence of the phase transition temperature was studied using numerical and analytical methods.

- In reference [J1] the team builds on the Coarse Molecular Dynamics technique [1] to obtain the order-to-disorder transition temperature of krypton physisorbed on graphite. The team obtains 5× acceleration compared to standard MD measurements of fluctuations of the total energy. The CMD technique falls under the umbrella of the more general *equation-free* methods, in which the macroscopic evolution of a system is simulated by doing short bursts of microscopic-level simulations compatible with the required macroscopic state. Initializing such microscopic systems is called lifting, and doing it efficiently is an area of active development (see appendix B.7 and [J8]).

- A parallel effort for the krypton problem (appendix D.1) was to extend the use of quenching simulations [5] by developing pattern boundary detection methods to separate high and low

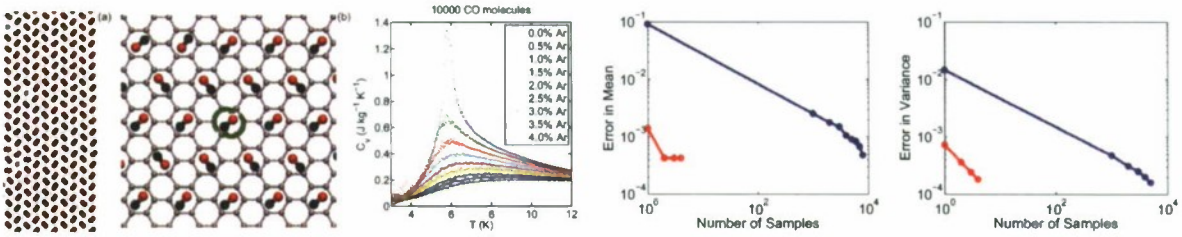


Figure 1.3: Summary of Phase II results for the phase diagram problem. From left to right: system snapshot showing pinwheel structures around argon impurities (blue); flip moves used in Ising-type model; heat capacity curves for different impurity concentrations; acceleration in calculation of the transition temperature for Monte Carlo (blue) and PCM (red).

density regions that appear spontaneously in first order phase transitions.

Figure 1.2 summarizes these results. Note that, in order to obtain a good match with the experimental results from the literature for the case of helium, the team had to add a quantum-mechanical correction to the classical potential used in the simulations. This correction, based on Feynman's quasi-classical potential, went beyond the asymptotic approximations from [14] that were used in the theoretical calculations of reference [2].

Phase II phase diagram with uncertainty challenge Appendix D.3 [J21] focuses on the low temperature phase transition for carbon monoxide (CO) physisorbed on a graphite substrate.

- The team first developed an Ising-type model for the system that accurately captures the phase transition in the presence of an uncertain concentration of argon impurities.
- Since in the simulations the number of argon impurities had to be an integer, the team had to extend the Polynomial Chaos-based Probabilistic Collocation Method [3] to the case where the uncertain parameters can only take integer values, leading to a rare application of Krawtchouk polynomials. PCM allowed the team to calculate the mean and variance of the phase transition temperature 2000 times faster than Monte Carlo sampling [J21].
- The team's result settled the scientific question of whether the ground state of CO on graphite is head-to-head ordered or head-to-tail ordered in favor of the latter. Furthermore, the team showed that formation of pinwheel regions of CO around argon atoms are at the origin of the anomalous effect of stabilization of the low-temperature phase [J21].

Figure 1.3 shows the system, a typical configuration in the Ising-type model, a representative flip move (molecule rotation) in the computational procedure, the variations in the specific heat vs. temperature curves, and the comparative acceleration of PCM over Monte Carlo.

Phase II surveillance challenge Appendix A as well as references [J4, J6, J13, J18, J19], [C5], and [6] contain extended reports on results relevant to the surveillance challenge problem. Figure 1.4 shows the search area used by the team to compare the performance of the algorithms developed in the project with that of both standard and "smart" lawnmower search patterns that take into account the prior distribution for the single target. The challenge included the following constraints, so as to make the different algorithms comparable:

- There is a single, immobile target to be found.
- The vehicles' sensors have a small footprint compared to the total search area (0.1% for one sensor, 5% for the whole swarm) and their dynamics must be constrained (speed and acceleration limits).
- The sensors are noisy: at each observation, a sensor has a probability of detection $s_{\mathrm{d}} < 1$ and false alarm $s_{\mathrm{fa}} > 0$.
- The terrain includes foliage. If the target is in the foliage, it is undetectable. The algorithms must be able to conclude that the target is undetectable after a finite time.
- The algorithms as a whole must exhibit a global probability of detection above a given threshold $P_{\mathrm{d,global}}$ and false alarm rate below a given threshold $P_{\mathrm{fa,global}}$.

Under these constraints, the algorithms compete for lowest median detection time.

As shown in figure 1.4, two different lines of attack yielded successful practical search strategies: Spectral Multiscale Search (SMS), Greedy Spirals, and Dynamic Greedy Search (DyGS). Each uses a very different approach and has its own strengths. These strengths treat different axes of the problem and could in the future be combined into a unified approach to control a swarm of UAVs performing autonomous search missions. Both methods achieved almost 2× reduction in median search time compared with *smart* lawnmower.

Spectral Multiscale Search, or SMS (appendix A.1 and [J13]), which combines a novel application of the Neyman-Pearson lemma [J6] with a Lyapunov method, is a fully-autonomous approach that flexibly dictates the required control forces on the whole swarm at every time step. Given the prior distribution for the single target, the method evaluates how much the time-integrated coverage differs from the prior, using a specially-designed weighted measure that yields a naturally multiscale approach. The method spontaneously spreads out the vehicles, initially covering the large-scale features of the prior and then filling in the smaller scale details. As shown in figure 1.4, this method can take into consideration both uniform and nonuniform priors. The vehicles avoid the foliage when possible, but spontaneously fly over it when needed to cover a different region.

Figure 1.4 also shows the ROC (Receiver Operating Characteristic) curves associated with the SMS decision algorithm [J13]. For a given sensor quality (i.e., for given parameters s_{d} and s_{fa}) these curves show graphically the effect of taking repeated measurements in an area and help determine how much coverage is needed before the global constraints $P_{\mathrm{d,global}}$ and $P_{\mathrm{fa,global}}$ are satisfied.

Dynamic Greedy Search, or DyGS (appendix A.2 and [J4]), is made of two parts: a grid-free decision algorithm and a trajectory planner. The trajectory planner is based on a specially-developed

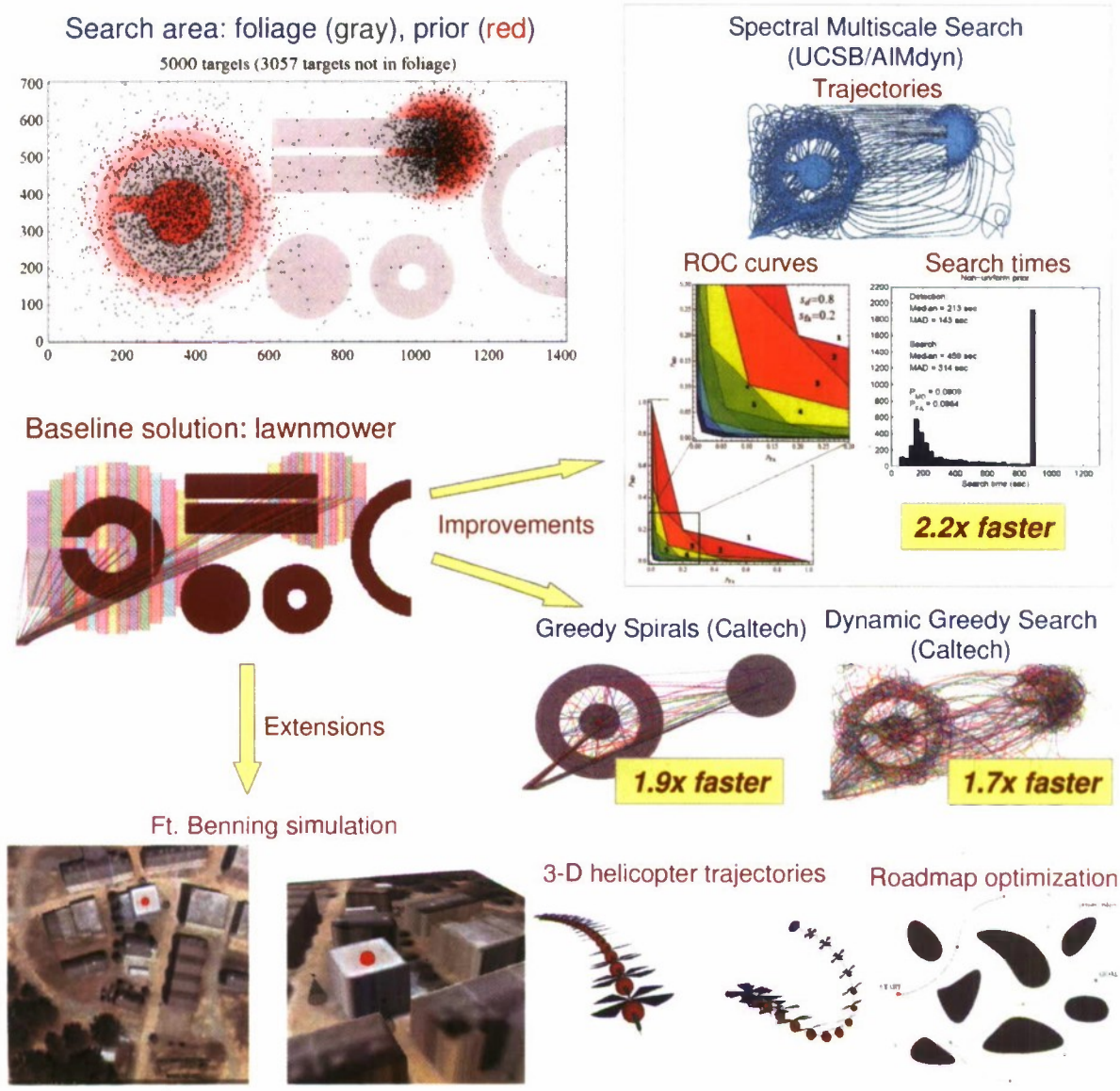


Figure 1.4: Top left: search area used for the simulations. The gray zones mark the foliage and the shades of red mark the prior distribution for the target. The dots mark 5000 target positions sampled from the prior. Also shown is the lawnmower baseline and the improved algorithms SMS, Greedy Spirals, and DyGS (see text). The bottom shows an approach combining helicopter model trajectory segments and roadmap planning for near-real time control of UAVs over Fort Benning.

method for optimizing dynamically the path to take in a computationally tractable manner while still producing realizable trajectories. It is based on the use of a library of elementary trajectory segments that individually satisfy the vehicle dynamics and can be interlocked to produce large scale roadmaps (see also section 2.3.1).

Even though the challenge problems focused on the detection of a single target, the team identified that the need to detect multiple targets would naturally arise in the future, and the necessary mathematics to treat this problem in a computationally-efficient manner had to be developed. Appendix A.4 (see also [C5]) develops such tools. The approach is novel in that it makes efficient a computation that in its original form is computationally intractable.

1.3 Organization of the report

The summary of accomplishments presented above focuses on the convergence of the different tools selected and developed as applied to the solution of the Phase I and Phase II challenge problems. In contrast, chapter 2 summarizes other tools that were mostly preparation for Phase III and were, for the most part, not used directly in the solution of the Phase I and Phase II challenge problems.

For published results we include the reference to the appropriate journal or conference proceedings, while for results that are submitted or in preparation we include the full drafts or internal reports as appendices. The appendices are mostly self-contained, and as such the citations in the appendices refer to their own bibliographies, and not to the report's main bibliography.

Chapter 2

Other tools developed

This project deals with the quantification and robust management of uncertainty in complex systems. In view of the goal of a real-time demonstration, many tools were developed with the goal of supporting fast, decentralized analysis of the situation, as well as efficient design of vehicle trajectories compatible with complex dynamics.

In this chapter we describe some of the tools developed. As these focused on improving different aspects of the problem, the tools are very heterogeneous, and fall broadly in the following three categories:

- Graph theoretic methods for system analysis and uncertainty quantification
- Decentralized estimation
- Design of dynamics

2.1 Graph theoretic methods for system analysis and uncertainty quantification

In order to make a system of many interacting components tractable, the system must be analyzed and divided into weakly connected components such that each individual component is of reasonable size. However the automatic detection of such components is a difficult task. Appendices B.5 and B.6 [J14, J15] focus on the analysis of such systems using graph-theoretic techniques to obtain both weak connections as well as causal chains. The latter are important for predicting the flow of information, and therefore uncertainty propagation, through such a system.

Similar methods were applied in reference [C3], where waveform relaxation was used to simulate a system with weakly connected components. This method was further extended in [J24] (see appendix B.1) into a unified, scalable approach to uncertainty quantification. Further uncertainty quantification techniques are compared in appendix B.2

Another use of graph-theoretic techniques is the application of diffusion maps to detect slow variables in a system [J22]. Once such slow variables have been found, an equation-free approach can be used to accelerate the simulation of the system (see [J1, J8] and [1, 13, 15]).

Finally, reference [J23] uses the connectivity of a network to do global filtering of noisy measurements.

2.2 Decentralized estimation

In anticipation of future challenges where many vehicles are collaborating in an environment with limited communication (or, equivalently, to extend life in power-limited wireless networks), several tools were developed to deal with decentralized estimation.

- For large networks and, in particular, for networks where the connectivity is changing dynamically), *stochastic* multiscale consensus was developed [C2]. Here each node decides at random whether to simply pass along a measurement received from a different node or to do a computation on it. The act of passing information along produces long scale connections that accelerate the convergence of consensus algorithms.
- For the problem of searching for a target, appendix B.3 deals with the issue of several different noisy sensors having to make a decision as to whether the target has been detected based on the limited information they have shared up to that point.

2.3 Design of dynamics tools

The area of *design of dynamics* encompasses both the design of systems that will spontaneously behave in a desired way as well as simplifying computations on complex systems so the problem of assigning tasks becomes tractable.

Examples of design of systems that spontaneously behave as desired are given in appendix C, where the self-assembly tools are described in detail. A related problem is that of *targeted activation* [C1], where the dynamics of the system are exploited to minimize the required energy input to obtain global reconfigurations.

For simplifying computations in systems with many independent actuators, a good example is the fast reconstruction of wavefronts for telescopes with adaptive optics (see [8] and [7] for experimental validation). Also, see appendix B.4, where a connection to the problem of self-localization is made.

2.3.1 Control optimization of vehicles with obstacle avoidance

With the goal of developing efficient methods to control vehicles with complex dynamics in environments with obstacles, a general framework was developed for integrating the dynamics and optimizing the motions of mechanical systems. The resulting algorithms are superior to standard

methods in numerical robustness and effciency, and can be applied to many types of vehicles such as simple helicopters and hovercraft.

The general approach is based on a combination of standard optimal control techniques and classical search and dynamic programming methods. These methods stand on top of a robust numerical representation of the underlying vehicle dynamics derived using the theory of discrete mechanics. The main results can be summarized as:

- structure-respecting geometric discretization of mechanical systems with symmetries, internal actuated shape, and nonholonomic constraints
- discrete optimal control formulation that respects the geometric structure
- combining the derived local optimal control techniques with global search methods in order to guarantee near-globally optimal solutions
- extending the basic motion planning framework to handle more specific tasks such as time-varying goal state, maximizing sensor coverage, deploying multiple vehicles to maximize information about a goal with uncertain dynamics multiple vehicles

Chapter 3

Personnel supported

UTRC Personnel: Marco Arienti, Andrzej Banaszuk, Emrah Biyik (intern from Rensselaer Polytechnic Institute), Sergei Burlatsky, Chaohong Cai, Konda Reddy Chevva, Sorin Costiner, Razvan Florea, Thomas Frewen, Jong-Han Kim (intern from Stanford University), Robert LaBarre, George Mathew, José Miguel Pasini, Tuhin Sahai, Sergey Shishkin, Troy Smith, Amit Surana.

Sikorsky personnel: Mark Lutian.

Academic consultants: Michael Dellnitz (Paderborn), George Karniadakis (Brown/MIT), Ioannis Kevrekidis (Princeton), Sean Meyn (UIUC), Gleb Oshanin (Université Pierre et Marie Curie).

University of California, Santa Barbara: Igor Mezić, Marko Budisić Bryan Eisenhower, Symeon Grivopoulos, Alice Hubenko, Yueheng Lan, Ryan Mohr, Gunjan Thakur.

AIMdyn: Caroline Cardonne, Vladimir Fonoberov, Sophie Loire.

California Institute of Technology: Jerrold Marsden, Houman Owhadi, Nawaf Bou-Rabee, Philip Du Toit, Katalin Grubits, Marin Kobilarov, Sujit Nair.

Stanford University: Sanjay Lall, Matthew West, Jong-Han Kim, Sunhwan Lee, Laurent Lessard, Tzu-Chen Liang.

Princeton University: Ioannis Kevrekidis, Thomas Frewen.

Yale University: Ronald Coifman, Yoel Shkolnisky, Amit Singer.

Plain Sight Systems: Fred Wagner.

Chapter 4

Publications

4.1 Journal papers

[J1] M. A. Amat, M. Arienti, V. A. Fonoberov, I. G. Kevrekidis, and D. Maroudas. Coarse molecular-dynamics analysis of an order-to-disorder transformation of a krypton monolayer on graphite. *J. Chem. Phys.*, 129:184106, 2008.

[J2] N. Bou-Rabee and H. Owhadi. Geometric Langevin algorithm. Submitted. Preprint available at `http://arxiv.org/abs/0712.4123`, 2009.

[J3] P. Du Toit, K. Grubits, S. Costiner, and J. Marsden. Fast generation of potentials for self-assembly of lattices. Submitted to Physical Review E, 2009.

[J4] P. Du Toit, M. Kobilarov, and J. Marsden. Search with under-actuated vehicles and uncertain sensors. In preparation, 2009.

[J5] P. Du Toit, I. Mezić, and J. E. Marsden. Coupled oscillator models with no scale separation. *Physica D*, 238:490–501, 2009.

[J6] V. A. Fonoberov, G. Mathew, A. Hubenko, and I. Mezić. A uniform coverage search strategy. In preparation, 2009.

[J7] V. A. Fonoberov, I. Mezić, and A. Banaszuk. Spatial and orientational ordering of interacting agents on corrugated substrates: Order-N calculation of phase diagrams for submonolayers of Kr, ^{4}He, and CO on graphite. In preparation, 2009.

[J8] W. Gear, D. Givon, and I. G. Kevrekidis. Constrained dynamics lifting. In preparation, 2009.

[J9] S. Grivopoulos. No crystallization to honeycomb or Kagomé in free space. *J. Phys. A: Math. Theor.*, 42:115212, 2009.

[J10] S. Grivopoulos. Some extensions of the Cucker-Smale flocking model. In preparation, 2009.

[J11] S. Grivopoulos, G. Matthew, G. Thakur, M. Budisić, and I. Mezić. Tools for design of potentials for particle self-assembly. Submitted to SIAM J. Appl. Math., 2009.

[J12] K. Grubits. Lattice quality assessment tools and their applications. In preparation, 2009.

[J13] A. Hubenko, V. A. Fonoberov, G. Mathew, and I. Mezić. Spectral multi-scale search. In preparation, 2009.

[J14] A. Hubenko and I. Mezić. Graph decomposition for biological networks. In preparation, 2008.

[J15] Y. Lan and I. Mezić. Unfolding cell regulation network anatomy through graph decomposition. In preparation, 2008.

[J16] Y. Lan and I. Mezić. Linearization at large of nonlinear systems. In preparation, 2009.

[J17] S. P. Meyn and G. Mathew. Learning macroscopic dynamics for optimal prediction. In preparation, 2009.

[J18] R. M. Mohr and I. Mezić. Designing search dynamics robust under sensor uncertainty: robust ergodic search algorithms. In preparation. UCSB confidential preprint, 2008.

[J19] S. Nair and J. E. Marsden. Collision avoidance and surveillance with autonomous vehicles. In preparation, 2009.

[J20] S. Nair, S. Ober-Blöbaum, and J. E. Marsden. The Jacobi-Maupertuis principle in variational integrators. In preparation, 2009.

[J21] T. Sahai, V. A. Fonoberov, and S. Loire. Uncertainty as stabilizer of the head-tail ordered phase in carbon monoxide monolayers on graphite. To appear in Phys. Rev. B, 2009.

[J22] A. Singer, R. Erban, I. G. Kevrekidis, and R. R. Coifman. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. Submitted to Proc. Nat. Acad. Sci., 2009.

[J23] A. Singer, Y. Shkolnisky, and B. Nadler. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. *SIAM J. Imaging Sci.*, 2:118–139, 2009.

[J24] A. Surana and A. Banaszuk. Scalable uncertainty quantification in complex dynamic networks. In preparation, 2009.

4.2 Conference papers

[C1] B. Eisenhower and I. Mezić. Actuation requirements in high-dimensional oscillator systems. In *Proceedings of the American Control Conference*, Seattle, 2008.

[C2] J.-H. Kim, M. West, S. Lall, E. Scholte, and A. Banaszuk. Stochastic multiscale approaches to consensus problems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5551–5557, Cancun, Mexico, 2008.

[C3] G. Mathew, S. P. Meyn, and A. Banaszuk. Waveform relaxation and graph decomposition. In *Proceedings of the 18th International Symposium on Mathematical Theory of Networks and Systems*, Blacksburg, Virginia, 2008.

[C4] S. P. Meyn and G. Mathew. Shannon meets Bellman: Feature based Markovian models for detection and optimization. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5558–5564, Cancun, Mexico, 2008.

[C5] S. Nair, K. Reddy, H. Owhadi, and J. Marsden. Multitarget detection using Bayesian learning. Submitted to the 48th IEEE Conference on Decision and Control, 2009.

4.3 Invited sessions

The following invited sessions and minisymposia were organized with AFOSR support and contain AFOSR-funded papers:

- SIAM Conference on Applications of Dynamical Systems 2007. Minisymposia: Uncertainty quantification in large-scale dynamical systems, Parts I and II.
- Mathematical Theory of Networks and Systems 2008. Special session on robust uncertainty management.
- Allerton Conference on Communication, Control, and Computing 2008. Session: Optimization and learning.
- 47th IEEE Conference on Decision and Control 2008. Session: Complex systems: multiplayer models.
- SIAM Conference on Applications of Dynamical Systems 2009. Minisymposia: Uncertainty quantification of high-dimensional random dynamical systems, Parts I and II.
- SIAM Conference on Applications of Dynamical Systems 2009. Minisymposium: Uncertainty management and trajectory plannning in dynamic multi-agent surveillance systems.
- SIAM Conference on Applications of Dynamical Systems 2009. Minisymposium: Graph-theoretic methods for analysis of complex networks.
- 48th IEEE Conference on Decision and Control 2009. Sessions: Learning and control, Parts I and II.

4.4 Plenary, Keynote and Invited talks

- 2007 SIAM conference of Control Theory, San Francisco CA. Plenary talk, A. Banaszuk.
- 2008 Stanford Structured Integrator Workshop. H. Owhadi.
- Dynamics Days 2008, Knoxville, Tennessee. Invited talk, A. Banaszuk.
- Lecture series on networks and complex systems. Lunteren Conference on the Mathematics of Operations Research, January 13-15, 2009. S. P. Meyn.
- SIAM Conference on Applications of Dynamical Systems 2009, Snowbird UT. Plenary talk, Igor Mezić.

4.5 Bibliography

[1] M. A. Amat, I. G. Kevrekidis, and D. Maroudas. Coarse molecular-dynamics determination of the onset of structural transitions: Melting of crystalline solids. *Phys. Rev. B*, 74:132201, 2006.

[2] D. K. Fairobent, W. F. Saam, and L. M. Sander. Density-functional theory of submonolayer phases of rare gases on graphite. *Phys. Rev. B*, 26:179–183, 1982.

[3] J. Foo, X. Wan, and G. E. Karniadakis. The multi-element probabilistic collocation method (ME-PCM): Error analysis and simulation. *J. Comput. Phys.*, 227:9572–9595, 2008.

[4] D. Frenkel and B. Smit. *Understanding Molecular Simulation.* Academic Press, second edition, 2002.

[5] L. D. Gelb and E. A. Müller. Location of phase equilibria by temperature-quench molecular dynamics simulations. *Fluid Phase Equilibria*, 203:1–14, 2002.

[6] M. Kobilarov, J. E. Marsden, and G. S. Sukhatme. Geometric discretization of nonholonomic systems with symmetries. To appear in AIMS Journal of Discrete and Continuous Dynamical Systems, 2009.

[7] L. Lessard, D. MacMynowski, M. West, A. Bouchez, and S. Lall. Experimental validation of single-iteration multigrid wavefront reconstruction at the Palomar Observatory. *Optics Letters*, 33:2047–2049, 2008.

[8] L. Lessard, M. West, D. MacMynowski, and S. Lall. Warm-started wavefront reconstruction for adaptive optics. *J. Optical Soc. Amer.*, 25:1147–1155, 2008.

[9] M. C. Rechtsman, F. H. Stillinger, and S. Torquato. Optimized interactions for targeted self-assembly: application to a honeycomb lattice. *Phys. Rev. Lett.*, 95:228301, 2005.

[10] M. C. Rechtsman, F. H. Stillinger, and S. Torquato. Designed interaction potentials via inverse methods for self-assembly. *Phys. Rev. E*, 73:011406, 2006.

[11] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* V. H. Winston and Sons, 1977.

[12] H. Wiechert and K.-D. Kortmann. Dipolar order and disorder phenomena in pure CO and dilute $(CO)_{1-x}(Ar)_x$ mixtures physisorbed on graphite. *Surf. Sci.*, 441:65–83, 1999.

[13] D. Xiu, I. G. Kevrekidis, and R. Ghanem. An equation-free, multiscale approach to uncertainty quantification. *Computing in Science and Engineering*, 7(3):16–23, May/June 2005.

[14] R. A. Young. Theory of quantum-mechanical effects at the liquid-gas critical point. *Phys. Rev. Lett.*, 45:638–641, 1980.

[15] Y. Zou, I. Kevrekidis, and R. Ghanem. Equation-free dynamic renormalization: self-similarity in multidimensional particle system dynamics. *Phys. Rev. E*, 72:046702, 2005.

Appendix A

Design of search trajectories

A.1 Spectral multi-scale search

Spectral Multi-Scale Search

Alice Hubenko*, Vladimir Fonoberov†, George Matthew‡, and Igor Mezić§

January 22, 2009

Abstract

We present a search algorithm for single or multiple searchers that finds a stationary target in presence of uncertainty in sensor radius. The considered uncertainty condition simulate the influence of the changing environment that occur in practical applications. Uncertainty in sensor radius sets this problem apart from the usual search and surveillance problem setting. Given P_D and P_{FA}, the algorithm minimizes search time to find the target with probability of detection at least P_D and probability of false alarm at most P_{FA}. We prove that the algorithm discovers the target with the desired efficiency. Computer simulations show that our algorithm has excellent performance when compared with Billiard search which is a type of random search. Form the design of the algorithm, it follows that the search time is inversely proportional to the number of searchers participating.

1 Introduction

Study of search problems as formalized mathematical models started more than 60 years ago, for a survey see [1]. During World War II mathematical theory was applied for the first time to locate German submarine threats in the Atlantic. Since its first applications search theory developed somewhat detached from practical applications. Our theory stands out from this trend because it uses realistic dynamics to model movement of searchers and in addition it is, apparently, the first model in literature that incorporates uncertainty in sensor radius that is a significant factor that affects search missions in real life. An extensively studied setting that is similar to ours, see [2], [6], is when a target is located

*Department of Mechanical Engineering, University of California, Santa Barbara, CA
†AIMdyn, Inc., Santa Barbara, CA. E-mail: vfonoberov@aimdyn.com
‡Department of Mechanical Engineering, University of California, Santa Barbara, CA
§Department of Mechanical Engineering, University of California, Santa Barbara, CA

somewhere in a region that is partitioned into a number of cells. The probability distribution for the targets position (i.e., the probability that the target is in any particular cell), and the detection function of our sensor (i.e., the probability of detection versus effort spent searching a cell, given that the target resides in that cell) are given. The goal is to maximize the probability of detection of the target, assumed that amount of total effort available for the search is fixed. A major drawback of this problem is its discrete setup, that would require perfectly functioning sensors. Besides, the theoretical solutions given to this problem assume that the search effort is infinitely divisible between cells and result in trajectories that would be physically hard to follow. Recently, several application oriented algorithms have been developed for similar problems. [8] presents a receding-horizon cooperative search algorithm that jointly optimizes routes and sensor orientations for a team of autonomous agents searching for a mobile target. The algorithm in [8] reduces the continuous search problem to an optimization on a finite graph. In [11] a framework for cooperative search using UAV swarms is described. The algorithm in [11] sweeps the area with UAVs flying side-by-side in straight lines. Unfortunately, both algorithms of [8] and [11] do not take into account changes in the environment that may occur. The changing environment (such as wind or fog) may alter the effective radius of the sensor. This would lead to leaving parts of the area completely uncovered and would reduce the performance of the search algorithm. We consider the search problem where a stationary target is placed in an area A that contains foliage F that the sensors can not penetrate. We consider the a priori distribution of the location of the target known (if it is not given we assume it to be the uniform distribution). The searchers move through the area A in continuous motion and use a circular sensor to scan the area. Our goal is to minimize search time in the presence of uncertainty in sensor radius while keeping the probability of detection of our algorithm above threshold P_D and probability of false alarm of the algorithm below threshold P_{FA}. In our Spectral Multi Scale (SMS) algorithm we utilize the Neyman-Pearson lemma, that is central in binary hypothesis testing theory, to design the decision making rule, that allows the searchers to quickly locate target suspects as they cover the area. The algorithm puts some of the searchers in rechecking mode to take some additional measurements at target suspects positions. This strategy ensures that the probability of false alarm is within the required threshold. We use the H^{-1} coverage strategy described in Section 3 to cover the area A. We tested the SMS algorithm with 50 searchers for different a-priori target distributions, each time making 5000 independent simulations. Our computer simulations show that besides demonstrating superior robustness in presence of uncertainty the SMS search vastly outperforms Billiard search when searchers start out in random directions and move in straight lines, reflecting when they reach the border. The median absolute deviation of SMS search time is 1.5 times smaller than that of Billiard

search; median search time of SMS search is 1.6 times smaller than that of Billiard search; median detection time of SMS search 1.7 times smaller than that of Billiard search. Another important advantage of the SMS algorithm its effective use of assets: the search time is inversely proportional to the number of searchers. So, for example, if we have two searchers instead of one, the expected search time is half of what we would expect with one searcher.

2 The decision making strategy

Let us consider the problem where N searchers are moving inside a search area A in $\mathbb{R}^2$ with the objective to detect a point-like target. We assume that each searcher has a circular sensor with radius at most δ. We will consider various scenarios for uncertainty in sensor radius. The target can be either in the search area A or in the foliage F where the searcher can not detect it. We assume that with probability α the target is in F and with probability $1-\alpha$ the target is in $S = A \setminus F$. We assume that the probability distribution of location of the target is known. The probability of detection for a single measurement, s_d, is the probability of getting a reading 1 on our sensor, assuming that the target is within the sensing area. The probability of false alarm for a single measurement, s_{fa}, is the probability of getting a reading 1, assuming that the target is not within the sensing area. Note, that for any sensor $s_d > s_{fa}$. The studies on real-life sensors indicate that as s_d increases, so does s_{fa}.

We denote by P_{MD} the probability of declaring that the target is in foliage, assuming that the target is in S. We denote by P_{FA} the probability of detecting the target in S, assuming that the target is not in that location. In the simulation setting it translates to the following, as seen in [5]. Denote the number of realizations of the whole search scenario N_R, the number of times the algorithm declared finding target and the target was not there N_{FA}, the number of times target was detectable (in S) N_D, and the number of times the target was detectable but the algorithm declared that it is in the foliage N_{MD}.

$$P_{FA} = \lim_{N_R \to \infty} \frac{N_{FA}}{N_R}$$

$$P_{MD} = \lim_{N_R \to \infty} \frac{N_{MD}}{N_D}$$

During the course of the algorithm the searcher moves in S, taking measurements with frequency f. For easier description of our decision making procedure, let us first assume that the searcher moves around S in steps. In each step the searcher is allowed to make several independent measurements with his sensor. Assume that at each step the searcher takes n_0 independent measurements, and declares

detection of target if at least $\gamma_0 + 1$ of the measurements are 1s. The Neyman-Pearson criterion (see [4]) allows us to find n_0 and γ_0 that maximize the probability of detection while the probability of false alarm stays under some prescribed bound (P_{FA}). The Neyman-Pearson lemma (see [4]) implies that the optimal n_0 and γ_0 are the solutions to the following optimization problem.

$$P_{FA} = P[k > \gamma_0] + \rho P[k = \gamma_0] = \sum_{k=\gamma_0+1}^{n_0} \binom{n_0}{k} s_{fa}^k (1 - s_{fa})^{n_0 - k} + \rho \binom{n_0}{\gamma_0} s_{fa}^{\gamma_0} (1 - s_{fa})^{n_0 - \gamma_0} \quad (1)$$

$$1 - P_{MD} = P[k > \gamma_0] + \rho P[k = \gamma_0] = \sum_{k=\gamma_0+1}^{n_0} \binom{n_0}{k} s_d^k (1 - s_d)^{n_0 - k} + \rho \binom{n_0}{\gamma_0} s_d^{\gamma_0} (1 - s_d)^{n_0 - \gamma_0} \quad (2)$$

We first find minimal γ_0 satisfying (1) when $\rho = 0$. Because at this point n_0 is unknown, $\gamma_0 = \gamma_0(n_0)$ is a function of n_0. Next, from the equation (1) we find $\rho = \rho(n_0)$. Finally, we substitute $\gamma_0(n_0)$ and $\rho(n_0)$ into (2) and find the minimal n_0 for which the equation still holds. Taking n_0 measurements at each location guarantees that probability of missed detection of the algorithm will be less or equal than P_{MD} it does not guarantee however that the probability of false alarm of the algorithm is less or equal than P_{FA}. Taking n_0 measurements will be a preliminary criteria in our decision making algorithm: if at least $\gamma_0 + 1$ readings are 1s the searcher will assume that there is a target suspect at that location. To achieve probability of false alarm less or equal than P_{FA} the searchers will take additional measurements.

We denote by T_{stop} the stopping time of the algorithm. The probability of false alarm for one step is the probability of detecting the target in S when the target is not in that location. Denoting the total number of steps taken by N, we can express the upper bound p_{fa} for probability of false alarm for each step as follows.

$$(1 - p_{fa})^N = 1 - P_{FA} \quad (3)$$

From equation (3) we get

$$p_{fa} = 1 - (1 - P_{FA})^{\frac{1.14 n_0}{f T_{stop}}} \quad (4)$$

p_{fa} provides an upper bound for the probability of false alarm for each step needed for the algorithm to achieve probability of false alarm at most P_{FA}. The probability of missed detection for one step is the probability of declaring that the target is in foliage when the target is S. Denoting p_{md} upper bound for probability of missed detection for each step, we get

$$p_{md} = 1 - (1 - P_{MD})^{\frac{1.14 n_0}{f T_{stop}}} \tag{5}$$

p_{md} provides an upper bound for the probability of missed detection for each step. Using the Neyman-Pearson criterion again, we obtain the constants n_1 and γ_1, that will be used by the algorithm in making the final decision. n_1 will be the upper bound on the number of measurements that the searcher may take at one step. From Neyman-Pearson lemma (see [4]) we have

$$p_{fa} = P[k > \gamma_1] + \rho P[k = \gamma_1] = \sum_{k=\gamma_1+1}^{n_1} \binom{n_1}{k} s_{fa}^k (1 - s_{fa})^{n_1 - k} + \rho \binom{n_1}{\gamma_1} s_{fa}^{\gamma_1} (1 - s_{fa})^{n_1 - \gamma_1} \tag{6}$$

$$1 - p_{md} = P[k > \gamma_1] + \rho P[k = \gamma_1] = \sum_{k=\gamma_1+1}^{n_1} \binom{n_1}{k} s_d^k (1 - s_d)^{n_1 - k} + \rho \binom{n_1}{\gamma_1} s_d^{\gamma_1} (1 - s_d)^{n_1 - \gamma_1} \tag{7}$$

Summary of parameters and variables

N	number of searchers
s_d	probability of detection for a single measurement
s_{fa}	probability of false alarm for a single measurement
p_{md}	probability of missed detection for one step
p_{fa}	probability of false alarm for one step
P_{MD}	probability of missed detection of the algorithm
P_{FA}	probability of false alarm of the algorithm
T_{stop}	stopping time of the algorithm
f	frequency of the sensor measurements
α	probability for target to be in foliage ($0 \leq \alpha < 1$)
δ	upper bound of the radius of the sensor
A	the search area
F	foliage

To find n_1, we find minimal $\gamma_1 = \gamma(n_1)$ satisfying (6) when $\rho = 0$. Next, from the equation (7) we find $\rho = \rho(n_1)$. Finally, we substitute γ_1 and $\rho(n_1)$ into (7) and find the minimal n_1 for which the inequality still holds.

The decision making algorithm

5

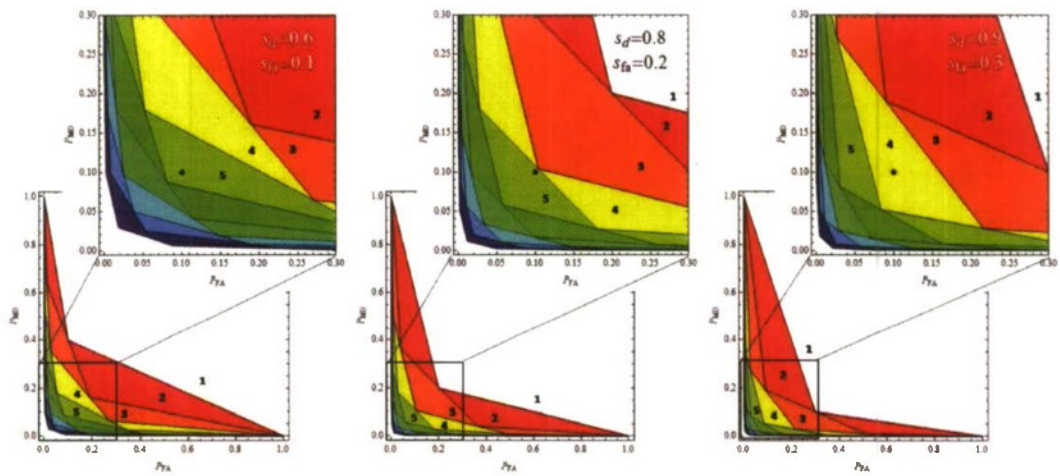


Figure 1: ROC curves

1. At each step the searcher takes n_0 measurements. If the number of 1s is less or equal than γ_0, the searcher decides that the target is not within the sensing area, and makes another step.

2. If the number of 1s is greater than γ_0, the searcher starts making additional measurements, stopping after at most n_1 measurements. After each additional measurement the searcher checks whether or not the ratio of 1s is smaller than $(\gamma_1 + 1)/n_1$. If yes, the searcher decides that the target is not within the sensing area and makes the next step. If no, the searcher makes an additional measurement. If the searcher has made n_1 measurements and the ratio of 1s is greater than $(\gamma_1 + 1)/n_1$, he declares that the target is detected.

3. If no detection occurs until time T_{stop}, the searcher stops and declares that the target is in the foliage.

In figure 1 we present performance plots of our decision making algorithm. Fixing constants n_0, s_d and s_{fa} we can compute all corresponding pairs P_{FA}, P_{MD} using equations (1) and (2). For fixed s_d and s_{fa} each color represents a constant n_0, shown on the picture. The pairs s_d , s_{fa} that we use are characteristics of real-life sensors computed in [7].

An estimate for the stopping time of the algorithm can be obtained as follows.

$$T_{stop} = \frac{1 - P_{FA}^2}{f\delta^2 N}|S| \tag{8}$$

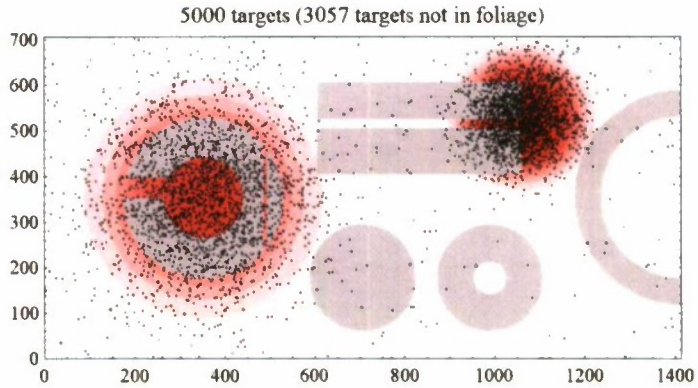


Figure 2: Search area S with foliage (grey), prior (red) and 5000 random targets

3 H^{-1} coverage

In SMS search we use design of motion described in [3] for n searchers to achieve optimal coverage of the prescribed domain. The heart of the coverage method is a Lyapunov-based control design utilizing an H^{-1} Sobolev space cost function. The method in [3] allows to design optimal dynamics for searchers with the goal to cover any part of a given area S. For instance, if the location of the target is described by a probability distribution $\mathcal{P}$, the part of S where $\mathcal{P} > 0$ has to be covered. We tested the motion design, included in SMS search, for 50 searchers and area S shown on figure 2, with foliage is shown in grey. We tested the problem in case when the prior probability distribution of the target $\mathcal{P}$ is uniform and in case when the location of the target corresponds to prior distribution shown in red (higher probability corresponds to darker shade in figure 2). As illustrated in figure 3 and 5, the H^{-1} coverage motion design of SMS search guarantees superior coverage according to prior of the target in both uniform and non-uniform cases. Note that searchers move with realistic second order dynamics. As seen in figure 5 the motion of the searchers depends on the probability of the target being in a certain part of the area: the high-probability regions are always covered better. By covering S according to a probability distribution $\mathcal{P}$ H^{-1} coverage saves time by not going to regions where the probability distribution of the target is 0. There is a potential drawback in H^{-1} coverage if $\mathcal{P}$ contains several high peaks. In that case the searchers will cover the regions close to peaks much more than needed to guarantee the desired precision threshold for SMS search. A clever way to avoid over-covering the area is to use $\log(\mathcal{P})$ instead of $\mathcal{P}$. Figure 6 illustrates the reduction of peaks using logarithm by showing two different cross-sections of $\mathcal{P}$ and $\log(\mathcal{P})$. Figure 7 shows the area that has

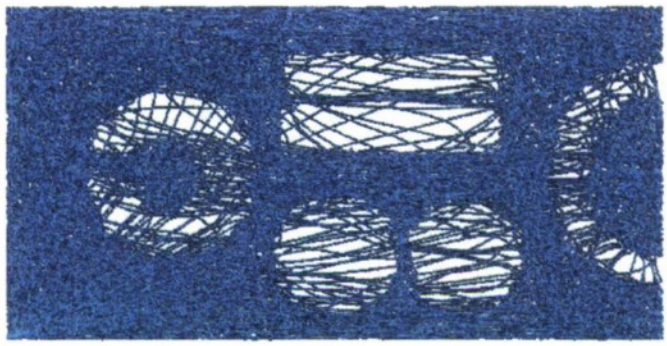


Figure 3: The area S covered by searchers shown in blue (uniform prior)

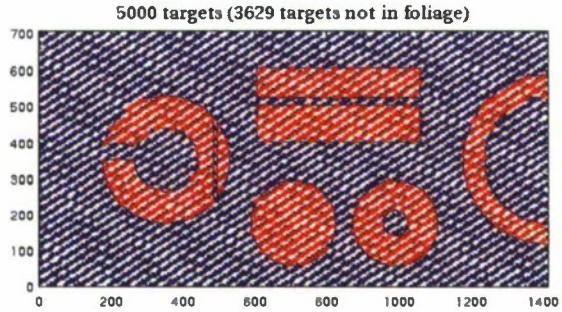


Figure 4: Typical coverage of $\log(\mathcal{P}) > 0$ by 50 searchers

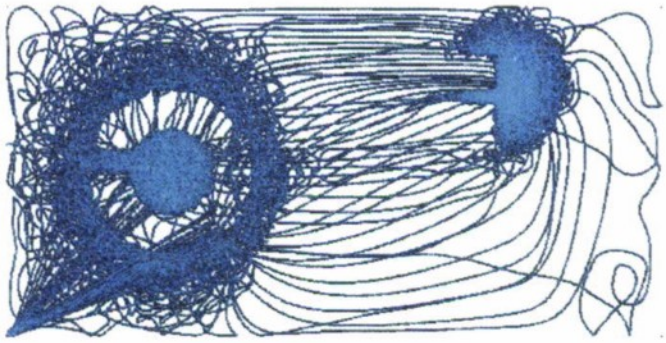


Figure 5: Typical coverage of $\mathcal{P} > 0$ by 50 searchers

8

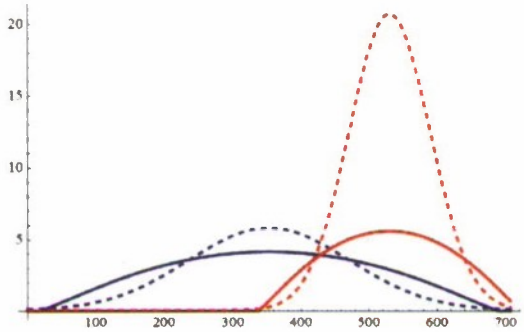


Figure 6: Reduction of high peaks in $\mathcal{P}$ (dashed), by replacing it with $\log(\mathcal{P})$ (solid line)

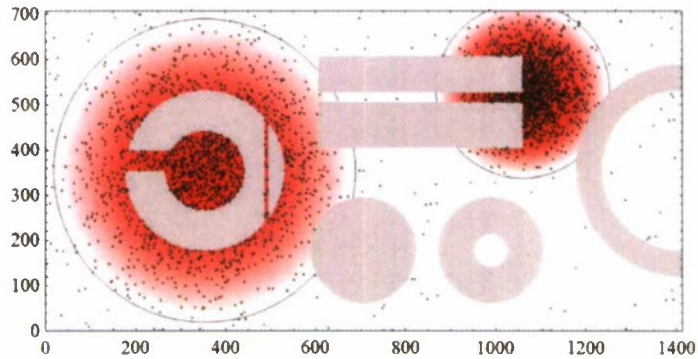


Figure 7: Search area S with of $\log(\mathcal{P}) > 0$ shown in red

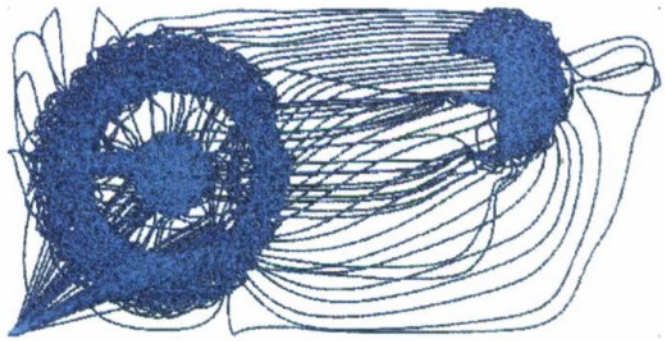


Figure 8: Typical coverage of $\log(\mathcal{P}) > 0$ by 50 searchers

to be covered after taking logarithm of the prior. As illustrated in figure 8, after reducing high peaks we avoid visiting certain areas too many times and the coverage of the desired parts of S becomes more even.

4 Spectral Multi Scale Search (SMS search)

Assume that we have n searchers and an area A where a single target is located. A may contain foliage F which the sensors can not penetrate. If the target is in the foliage it is undetectable for the searchers. We use a decision making strategy based on the Neyman-Pearson lemma, that was described in Section 2, and compute constants n_0, γ_0, n_1 and γ_1 for given P_{FA} , P_{MD}, s_{fa} and s_d. We use a measurement history map to keep track of target suspects: we divide the area S into small enough cells, and keep record of sensor measurements for each cell. At the beginning of the algorithm the belief map and the list of target suspects have no records. There will be two main modes for each searcher: explore and recheck. After deployment, all searchers start out in explore mode.

1. In explore mode the searchers cover the search area using H^{-1} coverage dynamics and update the measurement history map. When the number of measurements at a location becomes n_0 we check if the number of detections at the location exceeds γ_0: if yes, the location is added to the list of target suspects. Starting from the most likely targets (locations that have the highest ratio of positive measurements), each target suspect is assigned to an available neighboring searcher in explore mode. The searcher that has been assigned a target suspect changes his mode to rechecking, and moves to the location of the target on a straight line with maximum speed.

2. In recheck mode, the searcher has to perform n_0 measurements flying above a target suspect position. After finishing the measurements, the searcher switches to explore mode. After rechecking, if the ratio of detections to the number of measurements exceeds $\frac{\gamma_1}{n_1}$, we keep the location in the list of target suspects, otherwise we remove it from the list.

3. When the number of measurements at a location becomes n_1, we check if the number of detections at the location exceeds γ_1: if yes, we declare that the target is found and stop the search.

4. If the algorithm reaches stopping time T_{stop}, without declaring a detection, the algorithm declares that the target is in the foliage.

We tested the SMS search algorithm for 50 searchers on a rectangular area A shown in figure 2. Foliage is shown in dark grey and the prior distribution of the target is shown in red. Each searcher

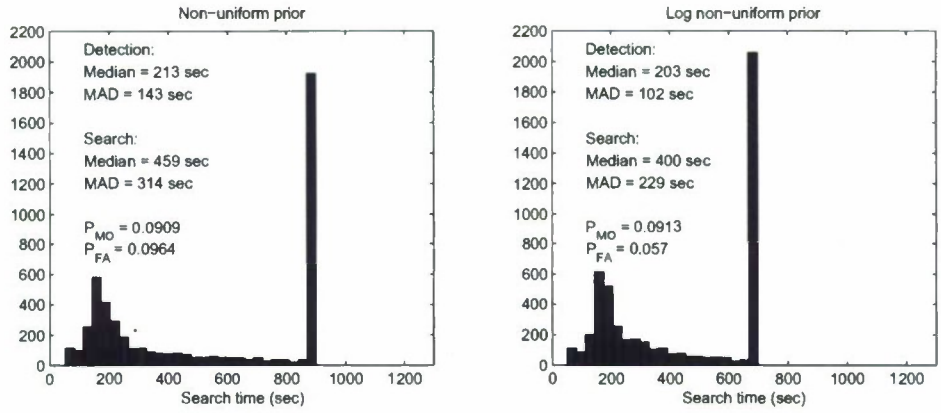


Figure 9: Histograms of SMS search for non-uniform prior and log-non-uniform prior

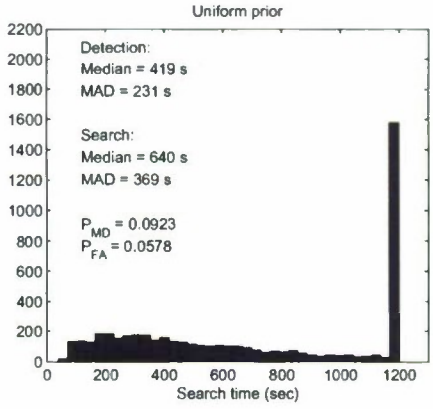


Figure 10: Histogram of SMS search for uniform prior

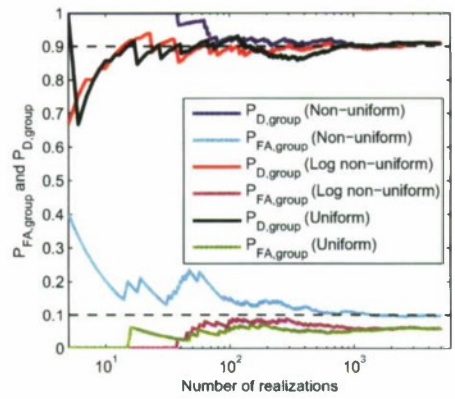


Figure 11: Proof that the algorithm stays within the desired probability thresholds

has a fixed uncertain sensor range that changes periodically with a fixed period. The probabilities of detection and false alarm for a single sensor are $s_d = 0.8$ and $s_{fa} = 0.2$ respectively. The goal is to minimize search time while satisfying requirements $P_{D,group} > 0.9$ and $P_{FA,group} < 0.1$. In figure 11 we show that $P_{D,group}$ and $P_{FA,group}$ converges above and below the required limits, respectively, as the number of realizations of SMS search increases. We tested the SMS algorithm in 5000 experiments with randomly generated targets for uniform prior, shown in figure 4, and in case of non-uniform prior, shown in figure 2. In figure 9 we compare histograms of 5000 experiments of SMS search performed on S with non-uniform prior shown in figure 2 and with log-non-uniform prior shown in figure 7. The search and detection time statistics, also presented in figure 9, show that the median search time of SMS applied to non-uniform prior is 15% bigger than in case of log-non-uniform prior and the median absolute deviation of SMS detection time applied to non-uniform prior is 40% bigger than in case of log-non-uniform prior. The histogram of SMS search performed on S with uniform prior is in figure 10. In figure 12 the median of the search time is shown as a function of the number of realizations of SMS search. Using H^{-1} coverage on non-uniform prior (see figures 2, 5) results in a median search time as compared with H^{-1} coverage for uniform prior (see figure 3). Taking logarithm of the non-uniform prior (see figures 8, 7, 6) helps to reduce the median search time even further.

The resulting median detection time was 203 sec, median search time was 400 sec, median absolute deviation (MAD) was 229 sec.

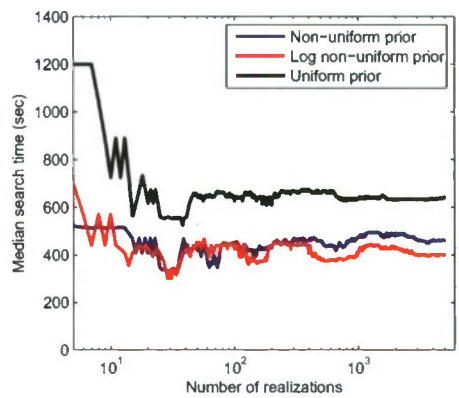


Figure 12: Median search time of SMS algorithm

5 Conclusions

In this paper we explore performance of search algorithms in presence of uncertainty in sensor radius. The introduced SMS search algorithm demonstrates excellent performance in presence of uncertainty as shown in the table below. We designed the SMS algorithm maximizing its effective use of assets: the search time is inversely proportional to the number of searchers.

Comparison of SMS search under different conditions

Algorithm	Median Detection Time	Median Search Time	Median Absolute Deviation
SMS (no uncertainty)	169 sec	303 sec	226 sec
SMS	203 sec	400 sec	229 sec
Billiard search	587 sec	1190 sec	534 sec

In the above table we compare statistics for SMS search without uncertainty, SMS search with periodically changing uncertain sensor radius, and Billiard search when searchers start out in random directions and move in straight lines, reflecting when they reach the border. We ran computer simulations of SMS search conducting 5000 independent experiments for each scenario. Computer simulations show that Median Detection Time, Median Search Time and Median Absolute Deviation of SMS search in presence of uncertainty are very close to the corresponding SMS search data without uncertainty in sensor radius.

The median absolute deviation of SMS search time is 1.5 times smaller than that of Billiard search; median search time of SMS search is 1.6 times smaller than that of Billiard search; median detection time of SMS search 1.7 times smaller than that of Billiard search.

Acknowledgements

This work was supported in part by DARPA DSO under AFOSR contract FA9550-07-C-0024.

References

[1] J.R. Frost, L.D. Stone, Review of Search Theory: Advances and Applications to Search and Rescue Decision Support, (CG-D-15-01), US Coast Guard Research and Development Center, 2001.

[2] B. O. Koopman, The Theory of Search II, Target Detection. Operations Research 4, 5 (October 1956), 503-531.

[3] G. Matthew, I. Mezić, Lyapunov-based feedback design of mobile sensor networks for uniform coverage, *manuscript.*

[4] J. Neyman, E.S. Pearson, On the Problem of the most Efficient Tests of Statistical Hypotheses, Journal Philosophical Transactions of the Royal Society of London. Series A, 1933, 231, 289-337.

[5] David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

[6] L.D. Stone, Theory of Optimal Search. Academic Press, New York, 1975.

[7] J.L. Solka, D.J. Marchette, B.C. Wallet, V.L. Irwin, and G.W. Rogers ,Identification of Man-Made Regions in Unmanned Aerial Vehicle Imagery and Videos ,IEEE transactions on pattern analysis and machine intelligence,1998, vol. 20, no.8, 852-857.

[8] J. Riehl, Gaemus E. Collins, J. Hespanha, Cooperative Graph-Based Model Predictive Search, 2007, In Proc. of the 46th Conf. on Decision and Contr.

[9] B. DasGupta, J. Hespanha, J. Riehl, E. Sontag, Honey-pot Constrained Searching with Local Sensory Information. Nonlinear Analysis: Hybrid Systems and Applications, 2006, 65(9):17731793, Nov.

[10] S.-P. Hong, S.-J. Choa, M.-J. Park A, pseudo-polynomial heuristic for path-constrained discrete-time Markovian-target search, European journal of operational research, 2009, vol. 193, no. 2 , 351-364.

[11] P. Vincent, I. Rubin, A framework and analysis for cooperative search using UAV swarms, Proceedings of the 2004 ACM symposium on Applied computing, 2004 79-86.

A.2 Design of search trajectories for vehicles with uncertain sensors

Design of Search Trajectories for Vehicles with Uncertain Sensors

Philip Du Toit
Caltech

January 21, 2009

Contents

1 Overview

Achieving the Surveillance Milestone requires the design of trajectories for aerial vehicles with uncertain sensors moving over a large complex terrain so that the time to locate a stationary target on the ground is significantly less than the time required by straightforward lawnmower trajectories. Because the individual sensor measurements include non-zero probabilities for both missed detection and false alarm, and the sensor footprint size is uncertain; the final statement provided by the search algorithm as to the location of the target cannot be provided with absolute certainty. Rather, the final declaration of the target location must meet confidence thresholds specified *a priori* for the probability that the stated target location is correct. In this report, we present the Dynamic Greedy Search (DyGS) algorithm for dynamically generating vehicle trajectories that achieves this Surveillance Milestone while ensuring that the method is robust to sensor failure, and is easily implemented for vehicles with complex dynamics and for regions with arbitrary foliage distributions. The DyGS algorithm achieves a 1.9 times speed-up over lawnmower methods.

Two essential components of the DyGS search strategy presented here are a *dynamic trajectory generation algorithm* and a precise *sensor decision algorithm.*

The trajectory generation algorithm provides dynamically updated trajectories that seek to maximize the probability of finding the target while ensuring consistency with the modeled dynamics and limited control authority of the vehicles. A hallmark of the DyGS search strategy is that

it requires no modification to handle vehicles with under-actuated or complex dynamics. Furthermore, the algorithm can be applied to real vehicles in which an accurate model of the dynamics is not known, and only recorded maneuvers of the real flight dynamics are provided.

The sensor decision algorithm takes as its input the history of measurements obtained during the search up to the current time. Given these raw inputs, the decision algorithm must decide when the search may be terminated and a declaration of the target location can be provided that satisfies the specified thresholds of ceorrectness. A classical approach to dealing with measurement data is to sample the incoming measurements onto a spatial grid, and then to calculate the probability for the target to be located at each cell in the grid. Here, we use a grid-free approach that instead stores the measurements as raw atomistic data so that inferences of the target location are made using the raw measurements in their most precise form. This approach has several advantages in that it removes approximation errors incurred by gridding, maximizes the use of information received in an information-theoretic sense, and also forgoes the need to impose a grid and a consequent scaling on the search domain that is not intrinsic to the problem. In practice, imposition of a grid can lead to ambiguity and ill-conditioned interpretation of the probabilities in each cell.

2 The Surveillance Milestone

We now provide a precise description of the Surveillance Milestone problem. In order to assess the efficiency of various search algorithms we have developed a specific test problem against which all the methods may be benchmarked. However, an important point is that the DyGS search algorithm is generally and easily applicable to arbitrary search domains and foliage distributions, and the specific domain used here is simply an example case. In particular, the search domain need not be a simply connected region.

The Search Domain:

1. The search domain, D, is a golden rectangle with an area of 1 square kilometer.
2. Regions of the search domain are designated as foliage as indicated in Figure 1.
3. A probability density f with support on D provides the probability density for finding the target at each location in the search domain. Since there is exactly one target to be found, $\int_D f\, dA = 1$. Target locations are sampled from this probability density function.

The Vehicle Model:

1. Fifty vehicles, each equipped with a single sensor, are available to search the domain.
2. The vehicles are initially deployed at rest on a 5 by 10 grid with 5m spacing in the lower left corner of the domain.
3. The dynamics of the search vehicles are modeled as a simple double integrator:

$$\ddot{x} = u \tag{1}$$

where x is the location of the vehicle in the plane, and u is a control force that must be chosen subject to the constraints that the maximum allowed magnitude of the velocity vector is 10 m/s, and the maximum allowed magnitude of the acceleration vector is 5 m/s^2.

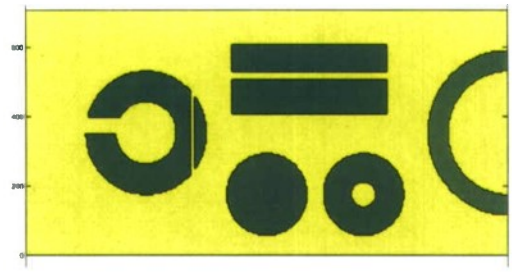


Figure 1: The search domain shown here is used to benchmark search algorithms in the Surveillance Milestone. Green areas represent foliage in which the target can hide without being seen by the sensors.

The Sensor Model:

1. Each sensor takes a measurement every 0.5 seconds.
2. During a measurement, the area scanned by the sensor is circular and lies directly below the vehicle. The radius of the area scanned by the sensor changes every 50 seconds and is chosen uniformly from the interval 5m to 10m.
3. If the target is outside the foliage and inside the small circular area scanned by the sensor, then the sensor reports with probability 0.8 that the target has been seen. In other words, the sensor probability of detection, p_{d}, is 0.8.
4. If the scanned area includes regions without foliage and the target is not in these open regions, then with probability 0.2 the sensor reports seeing the target, i.e. the sensor probability of false alarm, p_{fa}, is 0.2.
5. For all other cases, the sensor reports that the target was not seen.
6. When the sensor reports seeing the target, the exact location of the target is provided. If the sensor provides a false alarm, a location inside the open region inside the scanned area is randomly generated and returned. This location is stored by the sensor measurement generator and returned during future false alarms for cases when the scanned area includes this location.
7. With each measurement, the sensor also provides the exact location of the vehicle.

Information Available to the the Search Algorithm:

1. The search algorithm is provided with the exact geometry of the search domain and foliage, the probability distribution of the target locations, and the lower and upper bounds for the radius of the circular area scanned by the sensor.
2. At each timestep (every 0.5 seconds), the algorithm is provided with measurements from each of the sensors. The measurement data includes the precise location of the sensor, and one of the following statements:
   (a) No target was seen,

(b) The target was seen at the location (x_m, y_m).

Information Provided by the Search Algorithm:

1. At each time step, the search algorithm must provide trajectories along which the sensors must move. The trajectories must be consistent with the specified vehicle dynamics.

2. The algorithm must terminate the search and provide a final declaration of the target location. This declaration must be one of the following:

   (a) The target is in the foliage,

   (b) The target is in the open area and is located at the position (x_d, y_d).

Benchmarking Search Algorithms:

1. NT represents the number of target locations that are sampled from the target probability density. We choose NT = 5000.

2. For each target location, the search algorithm is run until the search is terminated and a declaration of the target location is made. The time taken to perform the search is recorded.

3. After all NT searches have been conducted, the correctness of the declarations is checked. Let

   ND := (Number of detectable targets) The number of target locations in the open region,
   NFA := (Number of false alarms) The number of declarations stating that the target was at a specific point in the open region that were not correct,
   NMD := (Number of missed detections) The number of declarations stating that the target was in the foliage when the target was in fact in the open region.

   Then compute

   $$P_{\mathrm{FA}} := \frac{\mathrm{NFA}}{NT} \quad \text{(Algorithm probabilty of false alarm)},$$

   $$P_{\mathrm{MD}} := \frac{\mathrm{NMD}}{\mathrm{ND}} \quad \text{(Algorithm probabilty of missed detection)}.$$

   The algorithm is considered *sufficiently correct* if P_{MD} and P_{FA} are both less than 0.1.

4. If the algorithm is sufficiently correct, then the median search time of all the NT search times is computed.

5. Sufficiently correct search algorithms are ranked by comparing their median search times.

The Surveillance Milestone Problem Statement:
Design a search algorithm so that the median search time computed using the methodology described above is less than the median search time computed for a search algorithm that uses systematic lawnmower type trajectories.

3 The Dynamic Greedy Search Algorithm

As mentioned in the overview, the Dynamic Greedy Search (DyGS) strategy comprises two algorithms: a grid-free decision algorithm for storing measurement data and deciding when a final declaration as to the target location can be made, and a trajectory generation algorithm for dynamically generating trajectories that are consistent with the vehicle dynamics and maximize the probability of finding the target.

3.1 The Grid-Free Decision Algorithm

The grid-free approach provides a method for storing and interpreting the measurement data in a manner that is computationally efficient while maximizing the utility of the raw measurements. The central data structure maintained by this algorithm is a list of suspect locations. Whenever a sensor reports that the target has been seen at a particular location, and this location has not previously been identified, this new location is added to the list of suspect locations. Associated with each suspect location in the list is a repository for all measurements that have ever been taken at locations that are less than the maximum possible sensor radius away from the suspect target location. Hence, along with the suspect location, we store every measurement that could possibly have included this location in its scanning area. With all the locally relevant measurements in hand, we can conveniently compute the local probability that the target is located at each suspect location in the list. For the case when the algorithm is told *a priori* that there is only one target, all measurements are globally dependent – a null measurement in one location increases the probability for positive measurements elsewhere in the domain. For this case, it is conceivable that we could use all the global measurement data to compute the probability that the target is located at each suspect location exactly; however, this procedure would require computationally expensive integrations and summations to compute the conditional probabilities. The approach presented here computes only a local probability but in a manner that uses the data precisely. Furthermore, the probabilities become precise for the case when the number of targets is not known and measurements separated by more than twice the maximum sensor radius are independent.

The probability that the target is located at a suspect location is updated whenever a local measurement is received using a Bayesian update scheme. Let $\{s_j\}_{j=1}^{\mathrm{NS}}$ denote the current set of NS suspect locations. Let $\{m_k^j\}_{k=1}^{\mathrm{NM}}$ denote all the NM measurements local to s_j that have been received up to the current time. Finally, let T_j denote the event that the target is in fact located at s_j. When a new suspect location is identified and added to the suspect list, the probability that the target is at this location is initialized to the value of the probability density function for the target distribution at this point. Then, the probability is updated using all measurements previously recorded. Since we are using the local approximation (that distant measurements are independent) we need only use nearby measurements to perform this calculation. As the search proceeds and more local measurements are received, the probability is further updated. The formula for updating the probability associated with suspect location s_j given a new measurement m_{k+1}^j is:

$$\mathrm{P}(\ T_j \mid m_1^j\ ) := \frac{\mathrm{P}(m_1^j \mid T_j\ ) \cdot f(s_j)}{\mathrm{P}(m_{k+1}^j \mid T_j\ ) \cdot f(s_j) + \mathrm{P}(m_1^j \mid \sim T_j\ ) \cdot (1 - f(s_j))} \qquad k = 0,$$

$$\mathrm{P}(\ T_j \mid m_{k+1}^j\ ) := \frac{\mathrm{P}(m_{k+1}^j \mid T_j\ ) \cdot \mathrm{P}(\ T_j \mid m_k^j)}{\mathrm{P}(m_{k+1}^j \mid T_j\ ) \cdot \mathrm{P}(\ T_j \mid m_k^j) + \mathrm{P}(m_{k+1}^j \mid \sim T_j\ ) \cdot \mathrm{P}(\sim T_j \mid m_k^j)} \qquad k = 1, \cdots, \mathrm{NM} - 1\,.$$

Remark 1:
The uncertainties introduced by the sensor probabilities for false alarm and missed detections, as well as the uncertainty with respect to the sensor radius are accounted for in the right hand side in the following way:
If the measurement m_{k+1}^j did not observe the target, then

$$\mathrm{P}(\ m_{k+1}^j \mid T_j) = (1 - p_{\mathrm{d}}) \cdot \mathrm{P}(\text{vehicle was in range of } s_j) + (1 - p_{\mathrm{fa}}) \cdot \mathrm{P}(\text{vehicle was out of range of } s_j).$$

If the measurement m_{k+1}^j did observe the target, then

$$\mathrm{P}(\ m_{k+1}^j \mid T_j) = p_{\mathrm{d}} \cdot \mathrm{P}(\text{vehicle was in range of } s_j) + p_{\mathrm{fa}} \cdot \mathrm{P}(\text{vehicle was out of range of } s_j).$$

The probability that the vehicle was in or out of range of the suspect location is determined by the specified uncertainty in the sensor radius. For the specified range of values for the sensor radius, and denoting the vehicle position during measurement m_k^j as x_k^j, we have $r_k^j := ||x_k^j - s_j||$,

$$\mathrm{P}(\text{vehicle was in range of } s_j) = \begin{cases} 1 & \text{if } r_k^j < 5 \\ 2 - 0.2 r_k^j & \text{if } 5 \le r_k^j \le 10 \\ 0 & \text{if } r_k^j > 10 \end{cases}, \tag{2}$$

and

$$\mathrm{P}(\text{vehicle was out of range of } s_j) = 1 - \mathrm{P}(\text{vehicle was in range of } s_j)\,. \tag{3}$$

Making the assumption that measurements separated by more than twice the maximum sensor radius are uncorrelated (which is equivalent to assuming the number of targets is unknown) implies that *we only need to consider measurements that are nearby the suspect location when performing the probability update.*

Remark 2:
Notice that a separate probability space is assigned locally to each suspect location, and that each probability space is partitioned into only two possible events: the target is at s_j (denoted T_j) or the target is not at s_j (denoted $\sim T_j$). Since all the probabilities are computed locally, there is no requirement that the sum of all the probabilities at all the suspect locations is unity. In other words, we should *not* expect $\sum_{j=1}^{\mathrm{NS}} \mathrm{P}(T_j \mid m_k^j) = 1$. What is guaranteed by construction is that

$$\mathrm{P}(\ T_j \mid m_k^j\ ) + \mathrm{P}(\sim T_j \mid m_k^j\ ) = 1\,.$$

These local probabilities provide a sense for the most likely target locations based on local measurements only.

At every timestep, the data structure described above provides a ranked list of the most likely locations where the target is hiding given the measurement history. What remains, is for the algorithm to determine when a target is located at a specific suspect location with sufficiently high probability that the search can be terminated and a declaration of the target location can be made. The algorithm must also be able to determine with sufficient confidence of correctness that the target is in the foliage. To effect these actions, we introduce three parameters: MinimumNumberOfMeasurementsToCheck, MinimumNumberOfMeasurementsToDeclare, and MaximumSearchTime that we will now proceed to describe.

Suspect Checking:
The default behavior of the search vehicles is to browse the search domain and to gather measurements in regions where the target is most likely to be found. (More on how these search trajectories are generated is provided in Section 3.2). As the vehicles browse the domain, a list of suspect locations and the local probability that the target is in fact at these locations is generated. If the local probability that the target is located at a particular suspect location rises above 0.5, and the number of measurements used to compute this probability is greater than or equal to MinimumNumberOfMeasurementsToCheck, then the algorithm will command the most nearby search vehicle to fly directly to the suspect location to gather more measurements to check the status of the suspect. During this checking process, a count of the number of *certain measurements* (measurements obtained while the sensor is less than 5m away from the suspect location) is kept. This requirement effectively removes uncertainty due to the variable sensor radius. When the number of certain measurements reaches MinimumNumberOfMeasurementsToDeclare, the algorithm has sufficient data to determine the status of the suspect location. If more than half of the certain measurements indicate that the target is present, then the algorithm can terminate the search and declare that the target is located at the suspect location. Otherwise, the suspect location is flagged as checked, the search vehicle resumes browsing, and the search continues.

Search Termination:
The search algorithm proceeds with sensors simultaneously browsing and checking suspect locations. The longer the search continues without uncovering the location of the target, the greater the probability that the target is located in the foliage. When the search time reaches MaximumSearchTime, the search is terminated and a declaration is made that the target is in the foliage. Choosing larger values of MaximumSearchTime will provide greater confidence in the correctness of this declaration.

Choosing Values for the Parameters:
In short, the values of MinimumNumberOfMeasurementsToCheck, MinimumNumberOfMeasurementsToDeclare, and MaximumSearchTime are chosen so as to minimize the median search time. In this way, freedom in the choice of the parameter values allows the search algorithm to be optimized for the specific search problem at hand.

In practice, MinimumNumberOfMeasurementsToCheck is chosen from the nominal set of values $\{2, 3, 4, 5\}$, and then a simple search is performed to find the values of the remaining two parameters

so that the search time is minimized.

It should be noted that for a given sensor p_d and p_{fa}, the Neyman-Pearson Lemma provides a minimum number of measurements that are required to test the hypothesis that the suspect is indeed the target, and the number of positive measurements that are required in order to declare that the suspect is indeed the target with sufficient correctness. These theoretical results are helpful in guiding the choice of the parameters, however, in practice these choices tend to be too conservative. The freedom to choose and adjust the parameters allows the algorithm to exploit structure in the given problem. Inhomogeneous structure arises, for example, from the uneven distribution of foliage, the shape of the target probability distribution, the geometry of the domain, and the concentrated deployment location of the vehicles. All these factors introduce non-trivial effects that are not accounted for in a straightforward application of the Neyman-Pearson lemma that assumes a homogeneous structure and distribution of events.

Finding optimal values of the remaining two parameters, MinimumNumberOfMeasurementsToDeclare and MaximumSearchTime, proceeds very quickly because of the relatively simple functional dependence of the search time on the parameters. We choose the parameters as low as possible to decrease the search time, subject to the following constraints:

C1. Decreasing the stopping time increases P_{MD} (The algorithm is too hasty to declare that the target is in the foliage).

C2. Decreasing MinimumNumMeasurements increases P_{FA} (The algorithm is too hasty to declare that the target is located at a suspect location).

In practice, we sequentially decrease these two parameters as much as possible without violating the specified P_{FA} and P_{MD}. This is a simple bisection search that requires perhaps 5 to 10 runs on each parameter. Each run (which includes 5000 searches) takes 2 to 4 minutes on a cluster at Caltech.

The main features of the sensor decision algorithm are summarized as follows:

1. No grid is required. The original problem has no inherent grid and we do not impose one.
2. We do not need to assign measurements to a grid. This process inherently introduces uncertainty that is not a part of the original problem.
3. The final declaration of the target position is exact. It is either at that exact location or it is not.
4. An optimized Neyman-Pearson criterion is used in the local binary hypothesis.
5. Time is optimized while ensuring the constraints on P_{FA} and P_{MD} are satisfied.

3.2 Dynamic Greedy Trajectory Generation

We now describe the algorithm for dynamically generating trajectories that are consistent with the vehicle dynamics. In essence, the trajectories for each vehicle are chosen from pre-computed trajectory segments so as to maximize passage through regions of the domain in which the target is most likely to be found and that have not previously been visited.

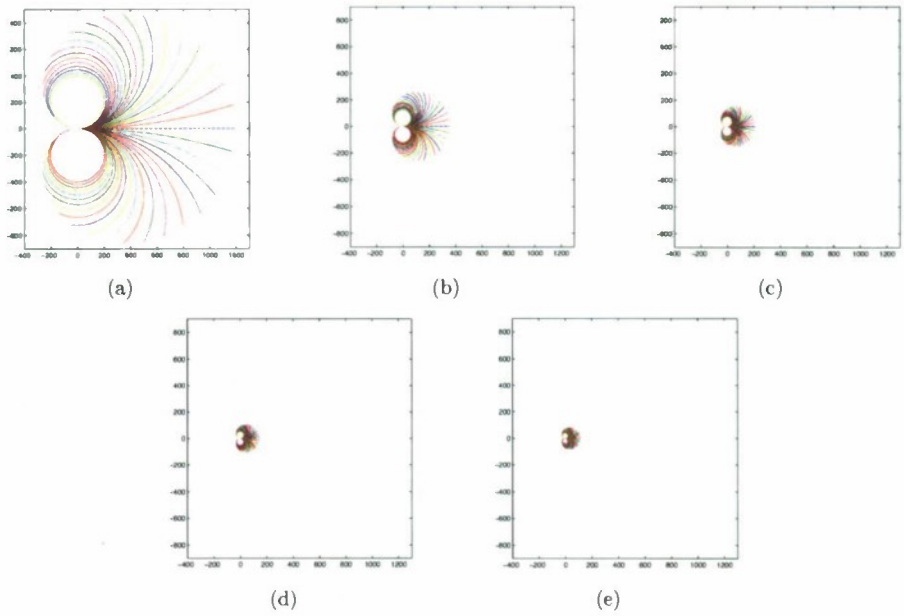


Figure 2: The trajectory library consists of motion primitives that can be pieced together to generate a vehicle trajectory that is consistent with the dynamics and limited control authority of the vehicles. Elements in the trajectory library are shown in figures (a) through (e) in order to emphasize how the library allows for a range of motion that covers the multiple scales of the search domain.

Library of Trajectories:
Each sensor has a library of trajectories stored in memory. A trajectory consists of a list of waypoints where each waypoint is a triple (x, y, θ) describing the position and heading angle of the vehicle along the trajectory at time intervals commensurate with the frame rate of the camera. The library is pre-computed and can be stored in memory at the factory where the vehicles are manufactured.

The library currently implemented in the DyGS algorithm has 205 trajectories. As shown in Figure 2, the library includes trajectories ranging from sharp left to sharp right turns and range in length from 12 seconds to 120 seconds. This library can be easily enriched with more trajectories or replaced with new trajectories without requiring any changes to the search algorithm. A guideline for choosing trajectories to include in the library, is that they should cover all scales of the search domain.

Vehicle Constraints:
By construction, the trajectories in the library satisfy the dynamic constraints of the vehicles. For the case when the dynamics is modeled with a double integrator, the trajectories are designed so that the vehicles always move at the maximum allowed velocity, and accelerations are only applied perpendicular to the direction of motion.

Choosing Trajectories from the Library:

At each time step, each sensor is moved to the next waypoint along its current trajectory. If the sensor is at the end of its current trajectory, then the sensor must be assigned a new trajectory from the library. The sensor performs a scan through the trajectories in the library to determine which trajectory will maximize the time-averaged probability of the prior visited during the trajectory. Visits to locations in the foliage and outside the search domain are considered to have zero probability. Also, any location along the trajectory that has been previously visited more than MaxNumVisits times, is considered to have zero probability. MaxNumVisits is a parameter that is chosen to adjust the "thickness" of the coverage.[1]

Computational Effort:
The computational overhead required to implement this trajectory generation scheme is minimal since at each time step, the majority of sensors are simply moved to the next waypoint along their current trajectory. Computation is only required when a sensor reaches the end of its assigned trajectory. A search conducted by fifty sensors can be simulated 100 times faster than real time on a single modest CPU. The computational effort of the search is dominated by the decision algorithm (updating suspect lists etc.) and not the trajectory optimization.

Extensions:
The dynamic trajectory generation algorithm can be easily extended to more complicated dynamics by simply replacing the library; no changes to the code are necessary. For example, we have successfully implemented search with vehicles whose dynamics is described by a simple three-dimensional model for under-actuated helicopter dynamics. The helicopters are modeled as rigid bodies with controls for forward and roll pitch of the main blades, as well as a yaw control from the tail blades. Conceivably, the trajectory generation algorithm can be applied to a vehicle in which an accurate model of the dynamics is not known; the trajectory library can be generated by recording the motion of the vehicle as it is driven through various maneuvers by an actual pilot.

Robustness:
Since the trajectories are dynamically generated, they easily and quickly adapt to sudden changes or updates in the prescribed search domain, foliage distribution, or initial target probability distribution. In contrast, the trajectories used in a systematic search strategy such as the lawnmower approach, are pre-computed and difficult to alter dynamically if new information about the search domain is provided or it becomes apparent that one of the sensors has failed. Moreover, lawnmower styled trajectories become increasingly difficult to design for search domains with complex geometry and foliage distributions. The ability of the DyGS algorithm to dynamically update the trajectories and the resulting chaotic nature of the these trajectories affords them more robustness when faced with search domains that have complex geometries.

4 Results and Conclusion

A search method that uses systematic lawnmower styled trajectories was implemented for purposes of comparison with the DyGS algorithm. When applied to the test domain described in Section 2,

[1] MaxNumVisits can be adjusted continuously through the positive reals via use of a coin. For example, to realize MaxNumVisits = 3.67 we ask if a random number in the unit interval is greater than 0.67. If yes, then choose MaxNumVisits = 3, otherwise choose MaxNumVisits= 4.

the lawnmower method achieves a median search time of 887.5 seconds. The median search time obtained for the DyGS algorithm is 458.5 seconds. We conclude that *the* DyGS *algorithm achieves the Surveillance Milestone by producing a 1.9 times speed up over the systematic lawnmower search strategy.* What is more, the DyGS algorithm provides robustness to sensor failure and to changes in the search domain geometry and foliage distribution. The DyGS algorithm is computationally fast, and can handle vehicles with complex, under-actuated, or even unknown vehicle dynamics.

A.3 Collision avoidance and surveillance with autonomous vehicles

Collision Avoidance and Surveillance With Autonomous Vehicles *

Sujit Nair and Jerry Marsden
(nair@cds.caltech.edu, marsden@cds.caltech.edu)
Control and Dynamical Systems
California Institute of Technology 107-81
Pasadena, CA 91125

9 July 2009

Abstract

The main aim of this paper is two fold. First is to present a new technique for collision avoidance of mechanical systems and second is to demonstrate the role of collision avoidance to enhance surveillance. The traditional techniques for collision avoidance are based on shaping the potential energy [18] or by introducing gyroscopic forces into the system [6]. In this paper, we close the story by introducing kinetic shaping for collision avoidance in the spirit of the Method of Controlled Lagrangians [5, 4]. We also provide analytic guarantee for no collision under some mild conditions depending only upon the energy and momentum of the system as it enters into a collision avoidance mode. The corresponding control effort is compared with known techniques. For an example vehicle model, it is shown that potential based collision avoidance methods are the most control efficient. We then briefly discuss the dependence of collision avoidance control cost on vehicle models. In particular, for systems with efficient steering, we expect gyroscopic collision avoidance to be the most efficient. We then show how collision avoidance can be used to randomize surveillance to give efficient chaotic search algorithms. The results are illustrated using multiple underactuated hovercraft. Perhaps most interesting, using a mix-norm, the area surveillance by multiple hovercrafts is shown to approach optimality quickly compared to the time taken to survey 90% of the region.

1 Introduction

1.1 Overview

The main goal of this paper is to introduce a new tool for collision avoidance and to demonstrate the role of collision avoidance in randomizing surveillance using multiple vehicles. We will also discuss quantities which can be used to evaluate quality of

*This work was in part supported by DARPA DSO under AFOSR contract FA9550-07-C-0024. Approved for public release, distribution unlimited.

surveillance heuristics. In particular, we introduce a new kinetic shaping based collision avoidance controller and surveillance with arguably chaotic properties, similar to the billiard problem [15]. Collision avoidance is critical when managing multiple vehicles in an environment with obstacles. They have also been used recently [6] to demonstrate their role in simulating flocking behaviour in addition to their intended task of avoiding collisions. The traditional potential based methods [18] and the recent gyroscopic force based methods [6] can be thought of as shaping the potential and the linear part of kinetic energy of the Lagrangian. In this paper, we *complete the story* of energy shaping based collision avoidance schemes by designing a collision avoidance controller by shaping the kinetic energy of the system. Using an example vehicle model and a scenario, we compare how the new technique fares with the previous two methods by evaluating the the L^2 and L^∞ norm of the control effort. The former norm quantifies the fuel consumed and the latter norm quantifies the maximum acceleration the vehicle experiences in avoiding collision. We also note that kinetic shaping collision avoidance gives analytic guarantee for no collision under a mild assumption depending only upon the energy and momentum of the system as it enters into a collision avoidance mode.

Surveillance using multiple vehicles is an important engineering problem with widespread applications. Traditionally, this task has been categorized into the static surveillance problem and dynamic surveillance problems. In the former, one needs to find the most optimal placement for sensors or cameras in an environment to maximize the area coverage [2, 8]. In the latter problem, one has sensors on dynamic objects like vehicles and maybe moving targets [3, 9]. The problem now is to design surveillance heuristics to guarantee target capture and tracking, i.e., to search and secure. The main research areas that come under dynamic surveillance are searcher coordination, flocking/formation control, role of communication topology and line of sight maintenance. Please see [11] for a detailed overview on the existing literature on these topics. In searcher coordination, the problem is to make sure no part of the region of interest is left out. The problem of searching for missing people trapped in mines falls in this category. In flocking and formation control, the goal is to make a group of vehicles move in a group to achieve a particular task like adaptive ocean sampling [7]. Whenever one has multiple vehicles, ensuring that the controller is robust to communication link losses is of upmost important. These problems fall under the communication topology category. The main aim is to demonstrate that whatever strategy one comes up with, the surveillance is not compromised because of noise or communication link breakages etc. Finally, when one has radio signals for intervehicle communication, it becomes important to make sure line of sight is maintained. See [11] for more references on this problem. In this paper, our focus will be on surveillance, i.e., we will not consider communication issues. Our main goal is to make sure the region is surveyed and that too in a "chaotic" or randomized manner. Such randomized searches are important in problems where one has mobile or "intelligent" targets and we do not want the targets to predict the searchers path for evasion purposes [16]. In our case, we want our search effort to be distributed with respect to some probability density depending upon the apriori

belief for target distribution and at the same time appear "chaotic" enough to keep the targets from predicting the next location of the sensors. We will present our surveillance algorithm and demonstrate this property by comparing the mix-norm of our vehicle trajectories with those of ideal zero inertia vehicles used as a benchmark.

1.2 Organization

In §2, we introduce kinetic shaping collision avoidance controller and give an explicit formula for the case when the vehicle is a double integrator. In §2.2, we compare the controller cost for the kinetic shaping controller introduced in §2 with the traditional potential based controller and with gyroscopic forcing based controller and interpret the results. In §3, we show how collision avoidance enhances surveillance and quantify its performance. In the same section, we discuss ideal vehicles and the mix-norm. In §4, we present some simulation results for the case of multiple underactuated hovercraft surveying a circular region. We demonstrate that using collision avoidance, the hovercraft trajectories approach a uniform distribution quickly compared to the time taken to survey 90% of the region. We conclude this paper with a summary of the main results and a discussion in §5.

2 Kinetic Shaping for Collision Avoidance

Intervehicle collision avoidance and obstacle avoidance is one of the most important issues in multivehicle tasks. The vast amount of literature focussing on this particular task bears testimony to this claim. Traditional methods based on potential design can be traced to the original works of Rimon and Koditschek [18]. The basic idea is to assign repelling potentials to obstacles and other vehicles and shape the potential energy of the moving vehicle accordingly. To get rid of some of the drawbacks of potential based collision avoidance, viz, global knowledge of obstacles etc, the authors in [6] design gyroscopic forcing based collision avoidance schemes. One of the main advantages of this new approach is that it tends to avoid gridlocking. Moreover, it conserves the total energy of the system as the steering forces act perpendicular to the instantaneous velocity. This in turn implies that we can use the readymade energy function of the system as a Lyapunov function for stability analysis purposes. Moreover, the gyroscopic forces do not interfere with the global potential function designed for a particular control task. The gyroscopic force based collision avoidance is also completely decentralized. Each vehicle has a sensing radius and goes into a collision avoidance mode only when it senses other vehicles within its sensing radius.

Gyroscopic based collision avoidance discussed in [6] can be thought of shaping the Lagrangian by introducing a term linear in velocity. For example, if $(q_i) = \mathbf{q}$ are generalized coordinates, then we can introduce gyroscopic forces into the system by adding the term $G = \sum_{i=1}^{n} A_i(\mathbf{q})\dot{q}^i$ to the Lagrangian[1]. The gyroscopic forcing term responsible for collision avoidance acts in a direction "perpendicular"

[1]Intrinsically, we are adding the action of the one form $\lambda = A_i dq^i$ on the tangent bundle to the Lagrangian.

to the instantaneous velocity. What this means is that the Euler-Lagrange term corresponding to G adds the term $\left(\frac{\partial A_k}{\partial q^i} - \frac{\partial A_i}{\partial q^k}\right)\dot{q}^k$ to the right hand side of the Euler-Lagrange equation for the original Lagrangian in the ith direction. It can be verified that the dot product of this term with $\dot{\mathbf{q}}$ is zero, i.e., the gyroscopic forcing term is "perpendicular" to the instantaneous velocity. Note also that if the one form $A_i dq^i$ is closed, the gyroscopic forcing term is zero. Potential based collision avoidance discussed in [14] also shapes the potential energy part of the Lagrangian. But now, the resulting collision avoidance forcing term can be made to act either along the center lines of the vehicles as in [18], or in a direction perpendicular to the centerline or a combination of these two directions. In the same paper, it was observed in simulation examples that when the forcing term acts in a direction perpendicular to the centerline, it takes a much longer time for the vehicles to get out of collision avoidance mode. After studying the above two examples, one could ask if its possible to derive collision avoidance schemes by shaping the kinetic energy part of the Lagrangian. We provide an answer in positive in this section.

The model we will be using in this section is going to be

$$\ddot{\mathbf{x}} = \mathbf{u}_T + \mathbf{u}_C + \mathbf{u}_D \tag{2.1}$$

where $\mathbf{u}_T$ is the controller which takes the vehicles to a particular target location or achieve a particular task, $\mathbf{u}_C$ is the collision avoidance controller and $\mathbf{u}_D$ is the dissipation controller linear in velocity. Each vehicle has a sensing radius r_s and a collision radius r_c, i.e., a vehicle senses another vehicles which is within a distance r_s from it. If the distance between two vehicles is less than r_c, its considered a collision. So our goal is to derive a collision avoidance scheme which guarantees that the distance between two vehicles is always greater than $2r_c$. The collision avoidance controller $\mathbf{u}_C$ is nonzero only when the vehicle senses an obstacle or another vehicle inside its sensing radius. When the vehicle is in its collision avoidance mode, $\mathbf{u}_T$ and $\mathbf{u}_D$ are both set to zero.

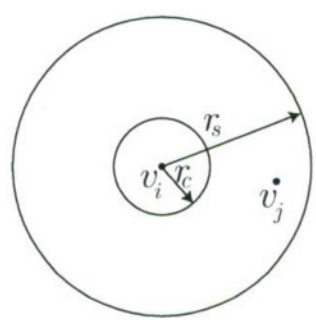


Figure 2.1: Vehicle j within the radius of sensing r_s of vehicle i.

Consider two vehicles in $\mathbb{R}^n$ whose controlled equations of motion are

$$\ddot{\mathbf{x}}_1 = \mathbf{u}_{T1} + \mathbf{u}_{C1} + \mathbf{u}_{D1} \tag{2.2}$$

$$\ddot{\mathbf{x}}_2 = \mathbf{u}_{T2} + \mathbf{u}_{C2} + \mathbf{u}_{D2} \tag{2.3}$$

The Lagrangian for each of these vehicles are given by their respective kinetic energies $L_i = \frac{1}{2}\dot{\mathbf{x}}_i{}^T\dot{\mathbf{x}}_i$. As stated before, when vehicles i senses another vehicle within its tolerance radius, it goes into a collision avoidance mode where

$$\mathbf{u}_{Ti} = 0; \quad \mathbf{u}_{Di} = 0 \tag{2.4}$$

In this collision avoidance mode, we would now like to choose the collision avoidance controller $\mathbf{u}_{Ci}$ such that the resulting closed loop system consisting of the two vehicles is also a Lagrangian system with a Lagrangian which looks like a kinetic energy term. Consider the Lagrangian L given by

$$L = \frac{1}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2(\dot{\mathbf{x}}_1^2 + \dot{\mathbf{x}}_2^2) \tag{2.5}$$

This Lagrangian is regular as long as $\mathbf{x}_1 \neq \mathbf{x}_2$. The energy E corresponding to this Lagrangian is the Lagrangian itself and the momentum corresponding to translation symmetry is $\mathbf{p} = \|\mathbf{x}_1 - \mathbf{x}_2\|^2(\dot{\mathbf{x}}_1^2 + \dot{\mathbf{x}}_2^2)$. We will show that for two vehicles following the Euler-Lagrange equations for this Lagrangian, the vehicles do not collide as long as their initial velocities are not equal and opposite to each other. More precisely, we have the following result.

Theorem Consider two vehicles with configuration variables given by $\mathbf{x_1}, \mathbf{x_2}$ and following the equations of motion corresponding to the Lagrangian given by (2.5). If at $t = 0$, we have $\mathbf{x_1}(0) \neq \mathbf{x_2}(0)$ and $\dot{\mathbf{x_1}}(0) + \dot{\mathbf{x_2}}(0) \neq \mathbf{0}$, then the two vehicles never collide at any instant of time, i.e., $\mathbf{x_1}(t) \neq \mathbf{x_2}(t)$ for all $t > 0$. Moreover, if $E_0 > 0$ and $\mathbf{p}_0 \neq \mathbf{0}$ are the nonzero energy and momentum at $t = 0$ respectively, then $\|\mathbf{x}_1 - \mathbf{x}_2\| \geq \sqrt{\frac{\|\mathbf{p}_0\|^2}{4E_0}}$ for all time.

PROOF For the Lagrangian given by (2.5), the momentum $\mathbf{p}$ corresponding to translation symmetry and the energy E are conserved quantities. Assume that $\|\mathbf{x}_1 - \mathbf{x}_2\| > 0$. Then we have

$$\begin{aligned} E_0 &= \frac{1}{4}\|\mathbf{x}_1 - \mathbf{x}_2\|^2\left[\|\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2\|^2 + \|\dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_2\|^2\right] \\ &\geq \frac{1}{4}\|\mathbf{x}_1 - \mathbf{x}_2\|^2\|\dot{\mathbf{x}}_1 + \dot{\mathbf{x}}_2\|^2 \\ &= \frac{1}{4}\frac{\|\mathbf{p}_0\|^2}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} \qquad (\text{ Assuming } \|\mathbf{x}_1 - \mathbf{x}_2\| > 0) \end{aligned} \tag{2.6}$$

Therefore, we get that if $\|\mathbf{x}_1 - \mathbf{x}_2\| > 0$, then $\|\mathbf{x}_1 - \mathbf{x}_2\| \geq \sqrt{\frac{\|\mathbf{p}_0\|^2}{4E_0}} > 0$. Thus, using continuity and the fact that $\mathbf{x_1}(0) \neq \mathbf{x_2}(0)$, we get that the minimum distance of

approach for the vehicles is bounded below by a quantity depending only on the values of energy and momentum at $t = 0$. ■

If we now use the equations of motion given by the Euler Lagrange equation corresponding to the Lagrangian (2.5), we have designed a collision avoidance controller which guarantees no collision as long as $\sqrt{\frac{\|\mathbf{p}_s\|^2}{16E_s}} > r_c$. Here, subscript s denotes the instance when the vehicles are within each others sensing radius and have sensed each other. At this instant, the only controller acting on the vehicles are the individual collision avoidance contoller derived from the Lagrangin (2.5). The final controller for the individual vehicle turns out to be

$$\begin{aligned} \mathbf{u}_{C1} &= \frac{1}{(\mathbf{x}_1 - \mathbf{x}_2)^2}\left(-2(\mathbf{x}_1 - \mathbf{x}_2)^T(\dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_2)\dot{\mathbf{x}}_1 + (\dot{\mathbf{x}}_1^2 + \dot{\mathbf{x}}_2^2)(\mathbf{x}_1 - \mathbf{x}_2)\right) \\ \mathbf{u}_{C2} &= \frac{1}{(\mathbf{x}_1 - \mathbf{x}_2)^2}\left(-2(\mathbf{x}_1 - \mathbf{x}_2)^T(\dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_2)\dot{\mathbf{x}}_2 + (\dot{\mathbf{x}}_1^2 + \dot{\mathbf{x}}_2^2)(\mathbf{x}_2 - \mathbf{x}_1)\right) \end{aligned} \tag{2.7}$$

Figure (2.2) illustrates trajectories in the (x_1, x_2) plane for two vehicles in $\mathbb{R}$ with coordinates x_1, x_2. The initial position is $(0.5, 0)$ and various trajectories correspond to various initial velocities starting from this particular initial position. As shown, the trajectories never "collide" with the line $x_1 = x_2$ unless its initial momentum is zero in which case they collide in finite time. We have the following result.

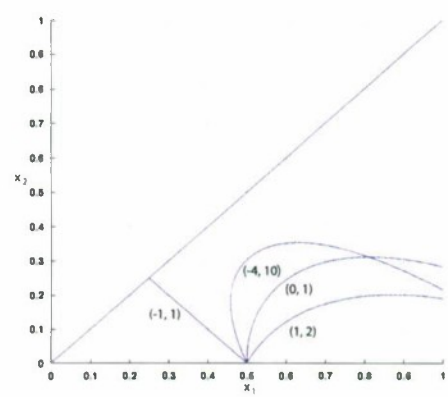


Figure 2.2: Trajectories in (x_1, x_2) plane for two vehicles in $\mathbb{R}$ with positions given by x_1, x_2 and corresponding to the Lagrangian given by (2.5). The initial position is $(0.5, 0)$ and the initial velocities are shown in the figure.

Result Consider two vehicles in $\mathbb{R}$ with positions given by x_1, x_2 and following the equations of motion given by the Lagrangian (2.5). If $\dot{x}_1(0) + \dot{x}_2(0) = 0$ and

$x_1(0) \neq x_2(0)$, then

$$z_1(t) = z_1(0)\sqrt{1 - \frac{2\dot{z}_1(0)}{z_1(0)}t} \tag{2.8}$$

where $z_1(t) = x_1(t) - x_2(t)$ and $z_1(0) \neq 0$ by assumption. Therefore, we have collision in finite time.

PROOF Proof follows by writing down the equations of motion for the Lagrangian (2.5) in $\mathbb{R}$ using the variables

$$\begin{aligned} z_1 &= x_1 - x_2 \\ z_2 &= x_1 + x_2 \end{aligned} \tag{2.9}$$

and verifying (2.8). ■

We also have the following time to minimum approach result.

Result Consider two vehicles in $\mathbb{R}$ with positions given by x_1, x_2 and following the equations of motion given by the Lagrangian (2.5). If $E = 2L(0) > 0$ and $p = (x_1(0) - x_2(0))^2 (\dot{x}_1(0) + \dot{x}_2(0)) \neq 0$, the the vehicles approach their minimum distance at time $T_{min} = -\frac{c_2}{c_1}$ where

$$\begin{aligned} c_2 &= \frac{1}{4}\left(2\sqrt{a}(z_2(0) - k) + \sinh(2\sqrt{a}(z_2(0) - k))\right) \\ c_1 &= \frac{1}{4}2\sqrt{a}\dot{z}_2(0)\left(1 + \cosh(2\sqrt{a}(z_2(0) - k)\right) \end{aligned}$$

where z_1, z_2 are as defined above in (2.9) and $k = \frac{\cosh^{-1}(\sqrt{a}z_1(0))}{\sqrt{a}} - z_2(0)$.

PROOF We have,

$$L = \frac{1}{4}z_1^2(\dot{z}_1^2 + \dot{z}_2^2) \tag{2.10}$$

$$p = \frac{z_1^2\dot{z}_2}{2} \tag{2.11}$$

If $E > 0$ and $p \neq 0$, we have $\frac{dz_1}{dz_2} = \sqrt{\frac{2Ez_1^2}{p^2} - 1}$. This gives, $z_1 = \frac{1}{\sqrt{a}}\cosh(\sqrt{a}(z_2 - k))$. Therefore, $\dot{z}_1 = \sinh(\sqrt{a}(z_2 - k))\dot{z}_2$ and $\ddot{z}_2 = -2\sqrt{a}\tanh(\sqrt{a}(z_2 - k))\dot{z}_2^2$. Therefore,

$$z_2(t) = k + \frac{1}{2\sqrt{a}}\text{Root}(\sinh(z) + z - 4(tc_1 + c_2) = 0) \tag{2.12}$$

where

$$c_2 = \frac{1}{4}\left(2\sqrt{a}(z_2(0) - k) + \sinh(2\sqrt{a}(z_2(0) - k))\right) \tag{2.13}$$

$$c_1 = \frac{1}{4}2\sqrt{a}\dot{z}_2(0)\left(1 + \cosh(2\sqrt{a}(z_2(0) - k)\right) \tag{2.14}$$

Now, z_1 is minimum when $z_2 = k$ at $t = T_{min}$. This implies, $T_{min} = -\frac{c_2}{c_1}$. ■

2.1 Comparison with potential and gyroscopic forcing based controllers

In this section, we will compare the performance of kinetic shaping collision avoidance controller with the traditional potential shaping based and the recent gyroscopic shaping based controller. Figure 2.3 compares qualitatively the difference in trajectories for the three different collision avoidance schemes. In this figure, we have two double integrator vehicles in $\mathbb{R}^2$, initially located at $(30, 0)$ and $(-30, 0)$. Starting with zero initial velocities, their task is to swap their positions avoiding collisions. The controllers will be stated precisely in (2.17). The plot illustrates the trajectories as the vehicles swap their positions. We see that qualitatively, the trajectories for collision avoidance based on gyroscopic forcing and kinetic shaping are comparable. Whereas in the traditional potential forcing based controller, we see the typical spring-mass like bouncing back phenomena. We also see that gyroscopic based collision avoidance starts steering the vehicles away from each other earlier on and performs a less aggressive maneuver compared to kinetic shaping based controller.

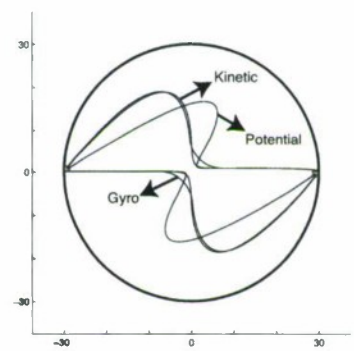


Figure 2.3: Two vehicles swapping their positions. The three trajectories correspond to three different collision avoidance schemes. As can be seen, the trajectories for kinetic shaping and gyroscopic forcing based collision avoidance are qualitatively comparable.

Collision avoidance when $n > 2$ When we have more than two vehicles, there are atleast two strategies to handle this situation by essentially reducing it to a two vehicle problem. In the first case, each vehicle detects all the other vehicles within its sensor radius and treats the average state of its neighbours as another vehicle, thus reducing the multiple neighbour problem to a two body problem. This strategy is not new and seems to work in simulations. See [6] for example and references therein. The second alternative is for each vehicle to avoid collision with its nearest neighbour amongst. We will use the latter in all the simulations in this paper. It also appears that kinetic shaping based controller can be extended to $n > 2$ case by

considering a Lagrangian of the form

$$L = \frac{1}{2} \sum_{1 \leq i < j \leq n} \left(\|\mathbf{x}_i - \mathbf{x}_j\|^2 (\dot{\mathbf{x}}_i^2 + \dot{\mathbf{x}}_j^2) \right) \tag{2.15}$$

without resorting to any of the heuristics discussed above. We will not be considering this scenario in this paper and rather focus on avoiding collision with the nearest neighbor and compare the performances between kinetic, potential and gyroscopic controllers.

2.2 Cost comparison with potential and gyroscopic forcing based controllers

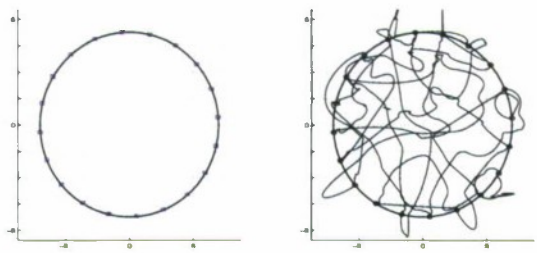


Figure 2.4: Trajectories of 20 vehicles, each starting on the circumference and reaching its assigned target location with collision avoidance based on kinetic shaping controller.

We now use a specific task to compare how our kinetic shaping based colllision avoidance compares with the potential and gyroscopic based controller. We choose the following scenario and use two norms to do the comparison. Consider n vehicles, each with a sensor radius of 2m, initial distibuted at an equal spacing on the circumference of a circle with radius 7m. We vary n from 10 to 20. At $t = 0$, these vehicles are assigned a random target position, again lying on the circumference. Figure 2.4 shows a sample trajectory. The plot on the left shows the initial configuration and the plot on the right shows the projection of the trajectories in time onto the plane after the vehicles have reached their respective target locations.

We choose the three different kinds of collision avoidance controller as discussed above to evaluate the performances of each. Specifically, we compare the cost functions given by $\int_0^T \mathbf{u}^2$ and $\|\mathbf{u}\|_\infty$ (L^2 and L^∞ norm of $\mathbf{u}$) over 20 runs for each value of n and each collision avoidance scheme. Here, $\mathbf{u}$ is the sum $\mathbf{u_T} + \mathbf{u_C} + \mathbf{u_D}$. The former norm quantifies the amount of fuel consumed and the latter norm quantifies the maximum acceleration experienced by the vehicles. We illustrate the results in Figures 2.5, 2.6 and 2.7.

For the cost comparison, we choose the following controllers.

$$\begin{aligned} \mathbf{u}_{Ti} &= -k_T(\mathbf{x}_i - \mathbf{x_{Ti}}) \\ \mathbf{u}_{Di} &= -k_D \dot{\mathbf{x}}_i \end{aligned} \tag{2.16}$$

Here, $\mathbf{x_{Ti}}$ is the target point assigned to the ith vehicle, k_T is the potential gain which drives the ith vehicle to its target and k_D is the dissipation gain. The only controller that varies is the collision avoidance controller $\mathbf{u}_{Ci}$. Let us denote the state of the nearest neighbour of the ith vehicle by $\mathbf{x}_{Ni}, \dot{\mathbf{x}}_{Ni}$. Then, we have the following expression for the collision avoidance controller for the ith vehicle.

$$\mathbf{x}_{Ci} = \frac{\left(-2(\mathbf{x}_i - \mathbf{x}_{Ni})^T(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_{Ni})\dot{\mathbf{x}}_i + (\dot{\mathbf{x}}_i^2 + \dot{\mathbf{x}}_{Ni}^2)(\mathbf{x}_i - \mathbf{x}_{Ni})\right)}{(\mathbf{x}_i - \mathbf{x}_{Ni})^2} \tag{2.17a}$$

OR

$$= k_{pc}\exp\left(\frac{1}{\|\mathbf{x}_i - \mathbf{x_{Ni}}\|}\right)(\mathbf{x}_i - \mathbf{x_{Ni}}) \tag{2.17b}$$

OR

$$= k_{gc}\exp\left(\frac{1}{\|\mathbf{x}_i - \mathbf{x_{Ni}}\|}\right)\hat{K}\dot{\mathbf{x}}_i \tag{2.17c}$$

Here, (2.17a),(2.17b) and (2.17c) correspond to kinetic shaping, potential shaping and gyroscopic shaping based collision avoidance controllers respectively. k_{pc}, k_{gc} are controller gains and $\hat{K}$ is a constant skew symmetric matrix. Please see [14] for more details on the potential and gyroscropic controllers in (2.17). The parameters k_T, k_D in (2.16) are fixed and k_{pc}, k_{gc} in (2.17) are chosen so as to keep the average minimum distances over 20 simulations to be around 2% of the sensor radius. This is illustrated in Figure 2.5. In our case of planar vehicles, the sensor radius is 2m and $\hat{K}$ is chosen to be

$$\hat{K} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{2.18}$$

As can be seen in Figure 2.6, for the gains corresponding to the minimum distances in Figure 2.5, the collision avoidance based on potentials seems to have the least average control effort. Both, the kinetic based and gyroscopic based collision avoidance are comparable to each other and have a similar pattern. Even when one looks at the maximum acceleration norm $\|\mathbf{u}\|_\infty$, as Figure 2.7 illustrates, collision avoidance based on potentials has the least maximum acceleration. From the above discussion, it appears that atleast for the particular task we have considered and the particular double integrator model for the vehicles, potential based collision avoidance seems to be the least expensive. Ofcourse, for different vehicle models, one needs to do a similar study and make conclusions. In our double integrator vehicle model, thrusting and steering were both equally expensive. For vehicles which have a cheaper way of gyroscopic steering as compared to ours, we expect gyroscopic based collision avoidance to perform better. One also needs to be a bit more careful when making these comparison. For example, it is known that using potential based techniques, it is observed that vehicles do get stuck in local extremums [6].

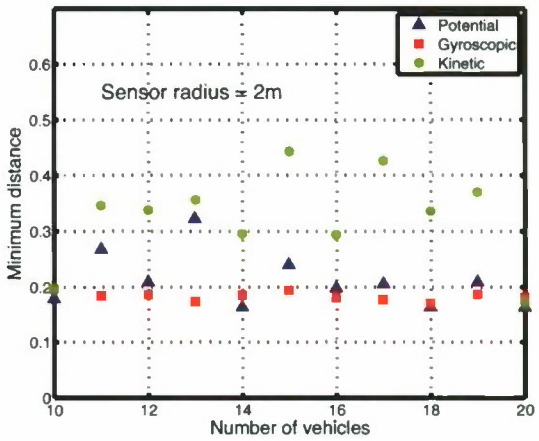


Figure 2.5: The parameters in (2.16) and (2.17) are chosen such that the average minimum distance over 20 runs which any two vehicles approach are comparable for all the three strategies based on kinetic, potential and gyroscopic collision avoidance. The sensor radius for the vehicles in the numerical simulation are all 2m.

3 Randomized Area Surveillance Using Collision Avoidance

In this section, we will demonstrate how collision avoidance can be used to enhance area surveillance by "randomizing" them. We also make sure this is done in a decentralized and scalable manner and is independent of the search domain and its topology. The main motivation for such search strategies is to make sure that an adversary target is not able to predict the state of search vehicles in order to dodge them. This is in stark contrast to the conventional "lawnmover" techniques in which case a target can avoid detection by just following the lawnmovers. Some typical applications of randomized search strategies or policies are for example police patrolling and airport security systems [16].

The mix-norm We now make precise what we mean by "randomizing" searches. Our main interest is two fold. One is to make sure the whole region is swept without any missed spots. Second, we need to do this such that the vehicle trajectories appear random to the targets. For the first case, the problem is more interesting in the case when the ratio of sensing radius to the area of the region is arbitrarily small. In this case, in order to quantify randomization, we use the mix-norm introduced in [13] in the context of fluid mixing. In the same paper, a scalar field is said to be well-mixed if its averages over arbitrary open sets are uniform. For the vehicle trajectory case, the basic idea is the following. We say that the trajectory of a vehicle is uniformly distributed in a region if the time average of a L_2 function over the trajectory asymptotically approaches its space average with respect to the

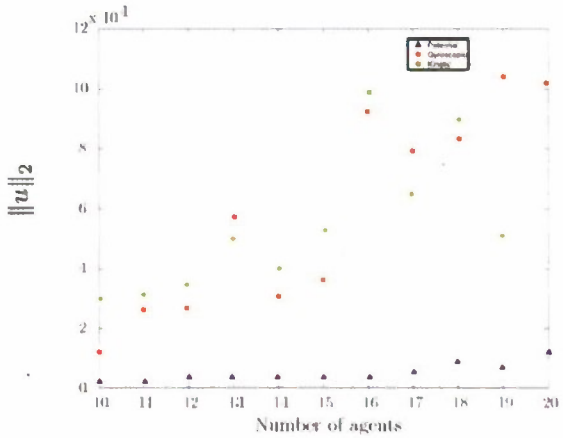


Figure 2.6: Plot comparing the total control cost to achieve a task using three kinds of collision avoidance, kinetic, potential and gyroscopic based. As can be seen, collision avoidance based on potentials seems to be having the least control effort.

uniform distribution μ. It is known that this is a necessary and sufficient condition for ergodicity [17]. If W_t is the distribution which takes a function and evaluates its mean over the trajectory upto time t, then we demonstrate that once collision avoidance is introduced into the system, the distribution W_t approaces μ when t for t much smaller than the time taken for the vehicles to survey 90% of the region. This is the sense in which our search is optimally randomized. Once the trajectories start approximating a uniform distribution, we expect the targets not being able to predict the future behavior of the vehicles.

As an illustration, consider the plots in Figure 3.1 and Figure 3.2. The first figure depicts Hilbert curves for indices n from 1 to 4 which are space filling curves in the limit $n \to \infty$. And Figure (3.2) is a plot of the decay of mix-norm over these curves. We see that as the index increases, i.e., as the Hilbert curves gets closer to a space filling curve, the mix-norm decays at a faster rate.

Benchmark vehicles Another way to see randomness is the following. The best surveillance one can achieve is using ideal vehicles with zero inertia hoping around instantaneously in the region such that their discrete trajectory points come from a uniform distribution. We are essentially comparing our vehicles with these ideal ones. When we say W_t approximates a uniform distribution, we mean that mean of samples of a function along the trajectory approaches the mean of samples of the same function along the same number of points chosen from a uniform distribution.

Note that if we have apriori knowledge of target locations represented by a distribution, we can exploit this information further by combining collision avoidance with random way point assignment which are chosen from the apriori. In this

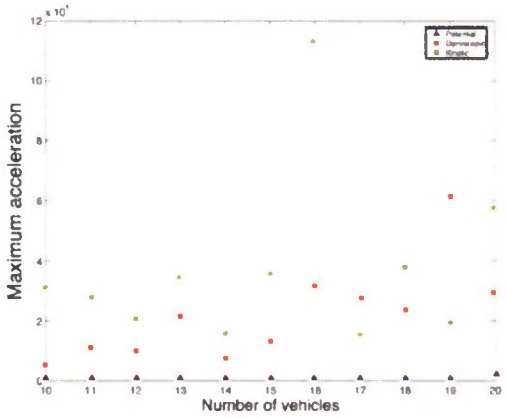


Figure 2.7: Plot of the maximum acceleration a vehicle experiences versus the total number of vehicles in our task. Again, one can see that collision avoidance based on potentials has the least maximum acceleration.

setting, we have a decentralized collision avoidance and a centralized "manager", which keeps updating the waypoints for inidividual vehicles. The vehicles are free to decide how they approach the waypoints and what kind of collision avoidance they want to adopt. This is reminiscent of the inner and outer loop philosophy in [10] for example. We briefly discuss this setting below.

Problem Setting With Apriori Knowledge We have a region A, with maybe obstacles, which needs to be surveyed. The region A also has a probability distribution for events, i.e., there could be areas within A where events are more likely to happen. We also have a finite number of vehicles with a finite sensor radius. Consider a lattice L with l points which reflects the probability distribution for the events. See the figure below which illustrates a lattice for uniform and Gaussian probability distributions respectively for the case when the region A is circular. The lattice spacing is a function of the individual sensor radius.

Consider Algorithm 1 below. The vehicles, labelled 1 to n are initially in some arbitrary locations. We assume that each vehicle has a set point controller which asymptotically takes it to a particular target location.

Algorithm 1 Surveillance Algorithm A

1: Assign n random points from L as target positions to the vehicles
2: Once the n vehicles reach their individual point, select new points from L and assign them as target point to the vehicles
3: Repeat this procedure until L is exhausted

Throughout the steps above, we incorporate collision avoidance into the setting,

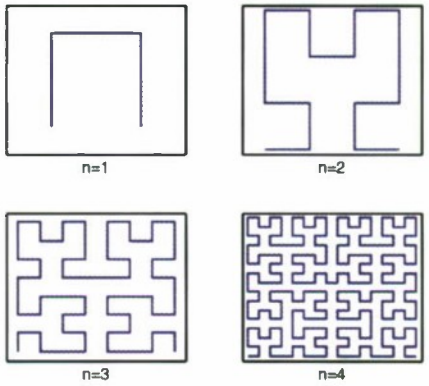


Figure 3.1: Illustration of Hilbert curves for indexes running from 1 to 4

i.e., if in the process of reaching their respective target point two or more vehicles come close to each other, they go into a collision avoidance mode. This strategy has some advantages over the traditional lawnmover strategies. The manager only assigns target positions to the individual vehicles as opposed to *complete individual trajectory*. Thus, we have a target assignment problem instead of a trajectory generation problem which works for a much broader class of regions. This is because generating lawnmover trajectories for arbitrary regions is nontrivial [1]. Our method also has advantages when considering underactuated vehicles, for which we only need to design a set point controller as opposed to making them follow a particular trajectory. The individual vehicles are free to choose how to reach their target positions providing the freedom to chose optimal controllers for example.

It can be numerically verified that with this strategy the trajectories start approximating a uniform distribution quickly compared to the time taken to survey 90% of the region. Note that in this case, it is not very clear where this randomization is coming from. We have a combination of random waypoint assignments as well as collision avoidance. To resolve this and to make precise the role of collision avoidance, we instead consider the following scenario where there is no apriori knowledge.

Problem Setting Without Apriori Knowledge We have a region A, with maybe obstacles, which needs to be surveyed and consider two cases. In the first case, one starts with random positions for the n vehicles and assigns them random initial velocities. This essentially determines the future of all the vehicles as the survey the area and we compute the mix norm of the trajectories. In the second case, one incorporates collision avoidance into the first case and again evaluates the mix norm of the trajectories. It turns out that once collision avoidance is incorporated, the mix norm decays at a much faster rate thus validating our claim

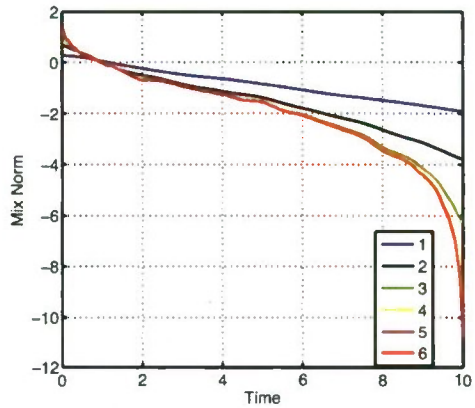


Figure 3.2: Plot of decay of mix-norm over the hilbert curves for the indices in Figure 3.1.

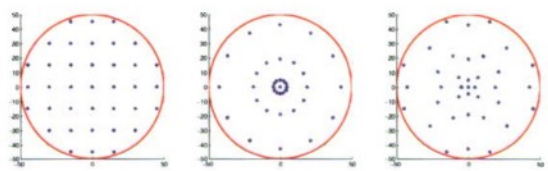


Figure 3.3: Illustration of a region A with a grid reflecting uniform and Gaussian probability distribution for target locations.

that collision avoidance enhances the surveillance by randomizing it.

Cumulative Area Coverage Computation To calculate the area covered, we distribute points from a uniform distribution in A and label them red. When a vehicle moves over it, we turn it green. Then the cumulative area coverage (CAC) is the ratio of green points to the total number of points. This way, we can take care of a broad class of surveillance regions and sensor sensing profiles. For example, calculating the exact area occupied by a finite number of circles with varying radii involves a nontrivial implementation of the inclusion exclusion principle.

4 Simulation

We now apply our surveillance strategy using vehicles which are simplified models for a hovercraft. The concrete situation is the following. Consider a circular region A with radius 50 m and 8 underactuated identical hovercraft as illustrated in Figure 4.1 with the following Lagrangian.

$$L = \frac{1}{2}(m\dot{x}^2 + m\dot{y}^2 + J\dot{\theta}^2). \tag{4.1}$$

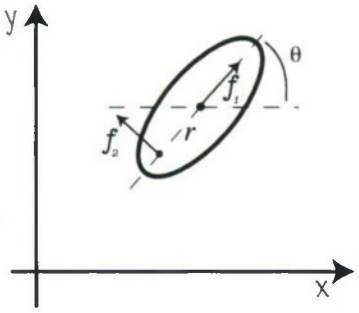


Figure 4.1: Model for an underactuated hovercraft.

The control force f_1 acts on the center of mass and the control torque f_2 acts on the body at a distance r from the center of mass as shown in Figure 4.1. The equations of motion can be derived to be

$$\begin{pmatrix} m\ddot{x} \\ m\ddot{y} \\ J\ddot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta)f_1 - \sin(\theta)f_2 \\ \sin(\theta)f_1 + \cos(\theta)f_2 \\ -rf_2 \end{pmatrix}. \tag{4.2}$$

In Appendix A, we show how to choose f_1, f_2 to design set point controller, dissipative controller and gyroscopic force based collision avoidance controller for this particular hovercraft system. Using these controllers, we implement our surveillance strategy with collision avoidance.

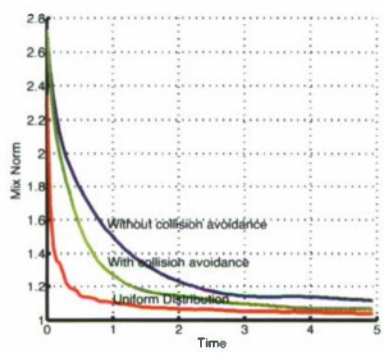


Figure 4.2: Plot illustrating surveillance enhancement with collision avoidance.

As shown in Figures 4.4 and 4.3, for our algorithm, the vehicles take about 30 seconds to survey 90% of the circular region and it takes only about 2 seconds for the trajectories to approximate a uniform distribution. Please see Appendix B for details on how we compute the mix-norm for the hovercraft trajectories.

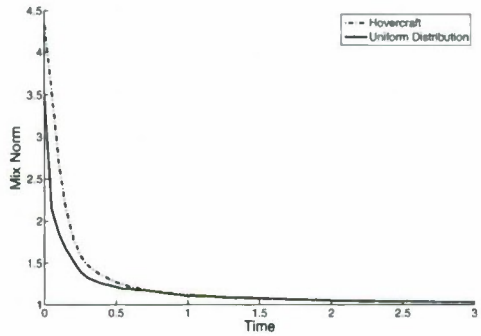


Figure 4.3: Plot of minx-norm of the vehicle trajectories and those of a uniform distribution. As can be seen, the two norms match pretty well after about 1 sec even though the time taken to survey 90% of the area is around 30 sec.

5 Conclusions

We now conclude this paper with a summary of the main results. We introduced a kinetic shaping based collision avoidance and compared its performance in the L^2 and L^∞ norms with that of potential and gyroscopic based collision avoidance schemes. For the particular vehicle model we considered, it turns out that potential based collision avoidance were the most efficient. For other vehicles with more complicated dynamics, it could very well be that potential based collision avoidance is not the most efficient. For examples, vehicles in which thrusters are more expensive compared to steering, it is expected that gyroscopic forcing based collision avoidance will turn out to be cheaper.

We also demonstrated the role of collision avoidance in efficiently randomizing surveillance similar to the billiard problem. The randomization was quantified using the mix-norm from fluid mixing literature. In our simulation, we showed that for our case of underactuated hovercraft, the trajectories approximate a uniform distribution much earlier than the time taken to survey 90% of the region. We hope to analytically prove the decay of mix-norm for the hovercraft system in a future publication. We also demonstrated randomization in essentially decentralized manner. This is in contrast with ?] where they achieve uniformization by optimizing a global cost function in a centralized manner. Merging the work in ?] with ours will be an interesting future direction.

References

[1] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, 17(1-2):25–50, 2000.

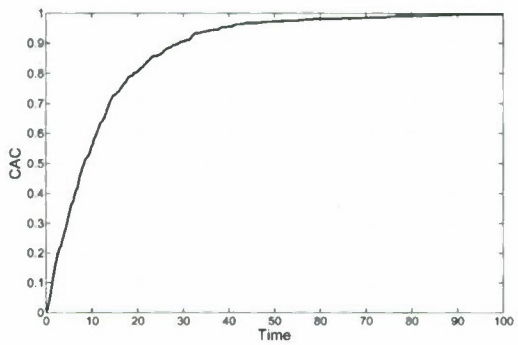


Figure 4.4: Plot of the cumulative area coverage versus time. As can be seen, it takes around 30 sec for the hovercraft to survey 90% of the area.

pages

[2] M. Batalin and G. S. Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 373–382, Fukuoka, Japan, Jun 2002. pages

[3] M. Batalin and G. S. Sukhatme. Multi-robot dynamic coverage of a planar bounded environment. Technical report, CRES-03-011, 2003. pages

[4] A. M. Bloch, D. Chang, N. E. Leonard, and J. E. Marsden. Controlled Lagrangians and the stabilization of mechanical systems II: Potential shaping. *IEEE Trans. Aut. Cont.*, 46(10):1556–1571, 2001. pages

[5] A. M. Bloch, N. E. Leonard, and J. E. Marsden. Controlled Lagrangians and the stabilization of mechanical systems I: The first matching theorem. *IEEE Trans. Aut. Cont.*, 45(12):2253–2270, 2000. pages

[6] D. Chang, S. Shadden, J. Marsden, and R. Olfati-Saber. Collision avoidance for multiple agent systems. In *Proc. IEEE Conf. Decision and Control*, Maui, Hawaii, December 2003. pages

[7] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. In *Proc. IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations (AUV04)*, Sebasco, ME, June 2004. pages

[8] A. Howard, M. J. Matarić, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 299–308, 2002. pages

[9] M. M. Jr and K. H. Johansson. Robust area coverage using hybrid control. In *TELEC'04*, 2004. pages

[10] O. Junge, J. Marsden, and S. Ober-blobaum. Optimal reconfiguration of formation flying spacecraft – adecentralized approach. pages 5210–5215, Dec. 2006. pages

[11] M. Lindhe. Survey of search-and-secure algorithms for surveillance ugvs. In *Scientific Report*, number FOI-R–2267–SE in ISSN-1650-1942. FOI, Swedish Defence Research Agency, April 2007. pages

[12] G. Mathew and I. Mezić. Spectral multiscale search: A uniform coverage algorithm for mobile sensor networks. *Preprint*, 2009. pages

[13] G. Mathew, I. Mezić, and L. Petzold. A multiscale measure for mixing. *Physica D*, 211:23–46, Nov. 2005. pages

[14] S. Nair and E. Kanso. Configuration control of non-colliding agents. In *Proc. of* 46st *IEEE Conf. Decision and Control*, pages 2534–2539, December 2007. pages

[15] C. Nikolai and R. Markarian. *Chaotic Billiards*, volume 127 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2006. pages

[16] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 273–280, New York, NY, USA, 2006. ACM. pages

[17] K. E. Petersen. *Ergodic theory*, volume 2. Cambridge University Press, Cambridge, 1989. pages

[18] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992. pages

[19] B. Yamauchi. A frontier-based approach for autonomous exploration. In *CIRA '97: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, page 146, Washington, DC, USA, 1997. IEEE Computer Society. pages

APPENDIX A

6 Controllers for the underactuated hovercraft

In this section, we will design potential, dissipative and gyroscopic force based controller for the hovercraft system.

Design of gyroscopic controller Let $\hat{S}$ be a skew symmetric matrix given by

$$\hat{S} = \begin{pmatrix} 0 & s_1 & s_2 \\ -s_1 & 0 & s_3 \\ -s_2 & -s_3 & 0 \end{pmatrix}. \tag{6.1}$$

For the right hand side of (4.2) to be a gyroscopic force, the following equation must hold true.

$$\begin{pmatrix} \cos(\theta)f_1 - \sin(\theta)f_2 \\ \sin(\theta)f_1 + \cos(\theta)f_2 \\ -rf_2 \end{pmatrix} = \hat{S}\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}. \tag{6.2}$$

Substituting for $\hat{S}$ from (6.1) in (6.2) and comparing the coefficients of $\dot{x}, \dot{y}, \dot{\theta}$, we get the following conditions on entries of the matrix $\hat{S}$.

$$\begin{aligned} s_2 &= -s_1 r\cos(\theta) \\ s_3 &= -s_1 r\sin(\theta). \end{aligned}$$

Therefor, the only parameter we have for tuning the gyroscopic force is s_1. We now solve for f_1, f_2 in terms of s_1 to get the final equations of motion with gyroscopic forcing as

$$\begin{pmatrix} m\ddot{x} \\ m\ddot{y} \\ J\ddot{\theta} \end{pmatrix} = s_1 \begin{pmatrix} \dot{y} - r\cos(\theta)\dot{\theta} \\ -\dot{x} - r\sin(\theta)\dot{\theta} \\ r\cos(\theta)\dot{x} + r\sin(\theta)\dot{y} \end{pmatrix}. \tag{6.3}$$

One can readily check that the dot product of the right hand side of above equation with velocity vector is indeed zero. Equation (6.3) represents the most general gyroscopic forcing for the hovercraft. For our simulations in §4, we choose the parameter s_1 to be a constant.

Design of potential controller We design potential based setpoint controller for the hovercraft. Because of underactuation, we cannot make the hovercraft flow along an arbitrary potential field. Instead, we derive conditions on the possible potentials which one can use for a hovercraft. In (4.2), assume the right hand side is given by a poteltial $V(x, y, \theta)$, i.e.,

$$\begin{pmatrix} \cos(\theta)f_1 - \sin(\theta)f_2 \\ \sin(\theta)f_1 + \cos(\theta)f_2 \\ -rf_2 \end{pmatrix} = -\begin{pmatrix} \frac{\partial V}{\partial x} \\ \frac{\partial V}{\partial y} \\ \frac{\partial V}{\partial \theta} \end{pmatrix}. \tag{6.4}$$

We can derive the following condition on V.

$$\frac{\partial V}{\partial \theta} = r\sin(\theta)\frac{\partial V}{\partial x} - r\cos(\theta)\frac{\partial V}{\partial y}. \tag{6.5}$$

Using method of characteristics, we can show that the most general solution for the potential V in (6.5) can be found out to be

$$V = V(x - r\cos(\theta), y - r\sin(\theta)) \triangleq V(z_1, z_2). \tag{6.6}$$

Using V as in (6.6), we can design controller which will asymptotically take the hovercraft to the desired set $(z_1, z_2) = (z_{1d}, z_{2d})$. This in particular implies that we can only guarantee that the hovercraft reach a neighbourhood of a desired target point (x_d, y_d) in the plane without control over its final orientation given by θ.

Design of dissipation controller In this case, we want to choose f_1, f_2 such that

$$\begin{pmatrix} \cos(\theta)f_1 - \sin(\theta)f_2 \\ \sin(\theta)f_1 + \cos(\theta)f_2 \\ -rf_2 \end{pmatrix} = -k \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}. \tag{6.7}$$

One can easily check that for the following choices for f_1, f_2, (6.7) is satisfied.

$$f_1 = -k(\cos(\theta)\dot{x} + \sin(\theta)\dot{y}) \tag{6.8}$$

$$f_2 = \frac{k}{r}\dot{\theta} \tag{6.9}$$

APPENDIX B

Computing H^{-1} norm Let $f(t) \in \mathbb{R}^2$, $t \in [0, T]$ be the trajectory of the system upto time T. We need to compute how "close" this trajectory is to a uniform distribution. To compute this, we take N samples with N sufficiently large, uniformly in time given by $f(t_i)$ where $i \in \{0, \cdots, N-1\}$ and $t_i = \frac{i}{N-1}T$. Denote this collection by $S_N = \{f(t_i)\}$. Pick another set of N points chosen from a uniform distribution in the domain and denote this by $U_N = \{u(t_i)\}$ for $i \in \{0, \cdots, N-1\}$. Then we have

$$\|S_N\|^2_{H^{-1}} = \sum_{k_1, k_2 \in \mathbb{Z}} \frac{\| \sum_{i=1}^{N} \exp\left(-i\left(k_1 f_1(t_i) + k_2 f_2(t_i)\right)\right) \|^2}{1 + k_1^2 + k_2^2} \tag{6.10}$$

Here, $f_1(t_i)$ and $f_2(t_i)$ are the x and y components of $f(t_i)$ respectively. The mix-norm for U_N is similarly computed. Note that $\sum_{i=1}^{N} \exp\left(-i\left(k_1 f_1(t_i) + k_2 f_2(t_i)\right)\right)$ is just the Fourier component of the δ distribution given by the samples of the trajectory $f_s(x, y) = \sum_{i=1}^{N} \delta(x - f_1(t_i))\delta(y - f_2(t_i))$.

A.4 Multiple target detection using Bayesian learning

Multiple Target Detection using Bayesian Learning

Sujit Nair, Konda Reddy Chevva, Houman Owhadi and Jerrold Marsden

Abstract

The need to develop fast, robust multiple target search algorithms has generated a lot of interest recently among scientists and mathematicians. In this paper, we develop a computationally efficient multiple target search strategy given a fixed number of search vehicles and fixed number of stationary targets in a region. Two different cases depending on whether the number of targets is known or unknown are considered. The search area is divided into cells. The belief map is updated using Bayes' theorem and an optimal reassignment (teleporting) of vehicles based on the values of the current belief map is adopted. Exact computation of the belief map update is expensive and often an approximation is needed. In this paper, we show that the Bayesian update can be exactly computed in an efficient manner by using the detection history in each cell and results from the theory of symmetric polynomials.

I. INTRODUCTION

Multi-target detection and tracking [1], [2], [3], [4] are important elements of a surveillance system. In multiple target detection and tracking problems, one is interested in determining the number as well as the dynamics of targets. Radar and sonar based tracking of objects for air traffic control and navigation are some of the applications of multi-target detection and tracking. In order to successfully detect and track targets, one needs to effectively extract useful information about the target's state from observations.

In this paper, we restrict out attention to detection of an unknown number of stationary targets using measurements from a fixed number of search vehicles. Though we mainly focus on multi-target detection, tracking and detection are closely related areas with significant overlaps. We, therefore, begin with a brief review of the techniques for multi-target tracking and detection. One of the earliest techniques for multi-target tracking is the multiple hypothesis tracking [1], [2]. In multiple hypothesis tracking, one associates a set of detections of the target position with existing tracks. This association if successful leads to new tracks. Otherwise, the set of detections are deemed false alarms. Typically, Kalman filter type algorithms are used to update the existing tracks after association. Multiple hypothesis tracking suffers from some problems. For instance, some information is lost when the detections are generated from raw sensor returns. So if the targets rarely produce returns above the detection threshold, then the tracking algorithm fails to accurately track the targets. A more robust solution to multi-target tracking is called the track-before-detect [5] where raw sensor returns are available to the tracker.

Sujit Nair, Houman Owhadi and Jerry Marsden are with Control and Dynamical Systems, Caltech, Pasadena, CA 91125, USA {nair,owhadi,marsden}@cds.caltech.edu

Konda Reddy is with Control Systems, United Technologies Research Center, E. Hartford, CT, 06108, USA ChevvaKR@utrc.utc.com

The above approaches suffer from two drawbacks. First, the above approaches are not recursive in nature. Secondly, if the target motion is complicated, then the above approaches have difficulty in modeling the target motion. Multi-target detection and tracking using a Bayesian perspective is a more promising and robust approach. This is the approach followed in this paper. Stone and coworkers [3] developed a mathematical theory of multi-target detection and tracking from a Bayesian point of view. Some early work in this area was done by Miller and coworkers[6], Kastella [7] and Mahler [8]. In the Bayesian approach, the multi-target state is a markov process. The Bayesian approach can handle complicated target dynamics. It also provides a solution that is recursive in nature.

The problem of simultaneous detection and tracking of multiple targets can be formulated in the Bayesian framework. The problem can be solved, in principle, by exact computation of the Joint Multi-target Probability Density (JMPD) [7], [9] that accounts for the uncertainty about the number of targets and their states. The JMPD is a high-dimensional quantity and its exact computation is almost impossible. A big challenge in multi-target detection and tracking research is to develop sophisticated numerical techniques to approximate the JMPD. Particle filter methods that provide a stochastic grid approximation to the exact solution of Bayesian state estimation have been proposed to approximate the JMPD for cases involving a large number of targets moving in two-dimensions [10], [11]. More recently, a unified approach to multi-target detection and tracking based on recursive approximation of the JMPD was presented in [12] where an efficient particle filtering scheme was proposed.

Clearly, an important research theme in multi-target detection and tracking is the development of efficient computational techniques to approximate the exact solution of the Bayesian state estimate. In this paper, we consider the problem of detecting an unknown number of stationary targets. Since we only deal with detection, association is not a concern. Though the detection problem is simpler in nature than the tracking problem, we would like to emphasize that the exact computation of the Bayesian update becomes increasing challenging as the number of unknown targets and the grid size increases. The main motivation for this work is to develop fast, computationally efficient techniques for exact computation of the Bayesian update.

The main contributions of the paper are as follows. We have demonstrated optimal Bayesian updates for detecting unknown number of targets in a given search region. The belief states of the system grow exponentially and usually cannot be solved exactly. However, we show that we can indeed solve the problem in a computationally efficient way by using the detection history in each cell. We also show that using Newton's identities from the theory of symmetric polynomials [13] helps us to exactly update the belief map. Newton's identities relate two different ways of describing the roots of a polynomial. They have applications in invariant theory and combinatorics and have connections to algebraic geometry. To the best of our knowledge, this is the first time that Newton's identities have been used in multiple target detection problems.

The paper is organized as follows. In Sec. II we review the well known Bayes' theorem for calculating

conditional probabilities. In Sec. III, we formulate the search problem given a fixed number of search vehicles and targets. In Sec. IV, we propose a new way to calculate the belief map in a computationally efficient manner. We also present a novel application of Newton's identities from the theory of symmetric polynomials to exactly calculate the belief map. Simulation results are presented in Sec.V.

II. Conditional Probability and Bayes' Theorem

Conditional probability is the probability of some event given the occurrence of some other event. Let (Ω, F, P) be a probability space where Ω, F and P have their usual meaning. Let $A, B \in F$ be two events with $P(B) > 0$. The *conditional probability* of A given B is defined as:

$$P(A|B) = \frac{(P(A \cap B)}{P(B)} \tag{1}$$

where $P(A \cap B)$ is the *joint* probability. Similarly, the conditional probability of B given A is given as

$$P(B|A) = \frac{(P(A \cap B)}{P(A)}. \tag{2}$$

Bayes' theorem for conditional probabilities follows from equations (1) and (2),

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{3}$$

Probabilities $P(A|B)$, $P(A)$, $P(B|A)$ and $P(B)$ are usually referred to as *posterior, prior, likelihood* and *marginal* respectively. More generally, if $\{A_i\}$ is a *partition* of A, then

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \tag{4}$$

for any A_i in the partition.

III. Problem formulation

The main problem we are interested in is the following: Given a search area with a fixed number of stationary targets and search vehicles, design computationally efficient strategies for detecting all the targets. More precisely, consider a search area that is divided into n cells as shown schematically in Fig. 1. Let V be the total number of search vehicles. We will consider two cases. In one case, the number of targets is known *apriori*. In the second case, the number of targets is not known. However, we assume that there is an upper bound $\bar{T}$ for the number of targets in the region. We assume that the targets are stationary. We adopt a simple model for the vehicle dynamics. At each time step, the vehicles can jump to any other cell or remain in their current cell. Each cell can only be occupied by a single search vehicle and/or a single target. The vehicles detect a target with probability p_d. We will also consider false alarms with probability p_f. At each time step, the vehicles send their detection data, i.e., detect or no detect, to a central manager, that updates the belief map based on the vehicle measurements. The belief map is a vector of numbers, $P(T_j)$, $j = 1, \ldots, n$, where $P(T_j)$ denotes the probability that the target is in cell j. The vehicles are then reassigned to new cells based on the updated belief map. There are several ways to reassign the vehicles to new cells. In this paper, we adopt an ideal policy where the vehicles are reassigned or *teleported* to cells that correspond to the maximum values of the belief map. Though this reassignment scheme is not entirely realistic, it provides lower bounds for the detection times.

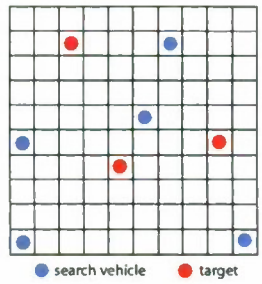


Fig. 1. A schematic of the search grid with fixed number of search vehicles and targets

In order to illustrate the application of Bayes' theorem for the search problem, consider the simple case of a single target and a single search vehicle in a grid of size n. The essential idea is to update the belief map at each time step based on the measurements made by the vehicles. We assume that the initial target distribution is uniform, that is, $P(T_j)^0 = 1/n$ for all j. Suppose that the vehicle is in cell i and that d_i is the detection data. That is, $d_i = 1$ if the vehicle detects and $d_i = 0$ if the vehicle fails to detect. Then, given the detection data d_i, Bayes' theorem can be used to update the belief map:

$$P(T_j)^1 = P(T_j|d_i) = \frac{P(d_i|T_j)P(T_j)^0}{P(d_i)} \tag{5}$$

$$= \frac{P(d_i|T_j)P(T_j)^0}{\sum_{k=1}^{n} P(d_i|T_k)P(T_k)^0} . \tag{6}$$

Note that

$$P(d_i|T_j) = p_d \quad \text{if} \quad d_i = 1 \quad \& \quad i = j \tag{7}$$

$$= p_f \quad \text{if} \quad d_i = 1 \quad \& \quad i \neq j \tag{8}$$

$$= 1 - p_d \quad \text{if} \quad d_i = 0 \quad \& \quad i = j \tag{9}$$

$$= 1 - p_f \quad \text{if} \quad d_i = 0 \quad \& \quad i \neq j . \tag{10}$$

The above formula can be easily extended to the case of multiple vehicles. Before we do that, we introduce some notation. $P(A|B)$ stands for the usual conditional probability of A given B. $P(T_{i_1},\ldots,T_{i_j})$ denotes the probability of the event $T_{i_1} \cap \ldots \cap T_{i_j}$. Here it is assumed that the indices $i_1,\ldots,i_j$ are distinct. By $D \setminus D_i$, we mean the detection data consisting of all cells except the ith cell. By $\sum_{\binom{n}{k}} P(A_{i_1},\ldots,A_{i_k}|B)$, we mean the sum over all $P(A_{i_1},\ldots,A_{i_k}|B)$ for $\binom{n}{k}$ ways of choosing indices $i_1,\ldots,i_k$ from the set $\{1,\ldots,n\}$.

One observes that if there are k targets, then the belief map consists of $\binom{n}{k}$ numbers given by $P(T_{i_1},\ldots,T_{i_k})$. The number of computations required to update the belief map blows up exponentially as the number of cells and targets increases. For example, for a 50×50 grid with 5 targets, the belief map is of the order of 10^{14}. This is a very important consideration if a particular search strategy needs to be implemented in real time. In

the next section, we describe a simple, computationally efficient and exact method to update the belief map. This method is particularly attractive when the number of targets and grid size is large.

IV. Computationally Efficient Belief Map Update

In this section, we propose an alternative formulation where instead of having a belief map of size $\binom{n}{k}$, we construct a belief map of size n by considering the history of detections/no detections in each cell. At each instant of time, vehicles make measurements about targets in their respective cells and report back to a central manager whether they have detected or not detected a target in their respective cells. For each cell, the total number of *detections* and total number of *no detections* is stored. Using this information, at each time step, the belief map is constructed as follows: If d_i and m_i are the total number of detections and no detections respectively in cell i up to that time, the detection data can be written as the following $2 \times n$ matrix

$$D = \begin{bmatrix} d_1 & \dots & d_n \\ m_1 & \dots & m_n \end{bmatrix}. \tag{11}$$

A. Known Number of Target Case

We now consider the case when the number of targets is known *apriori* and is assumed to be k. For the detection data D, we can construct the belief map as follows:

$$\begin{aligned} & P(T_i|D, U=k) \\ &= \frac{P(D|T_i, U=k)P(T_i|U=k)}{P(D|U=k)} \\ &= \frac{\sum_{\binom{n}{k-1}} P(D|T_i, T_{j_1}, \dots, T_{j_{k-1}}, U=k)}{P(D|U=k)} \\ &\quad \times \sum_{\binom{n}{k-1}} P(T_{j_1}, \dots, T_{j_{k-1}}|T_i, U=k)P(T_i|U=k) \end{aligned}$$

where U is the number of targets. The denominator can be written as

$$\begin{aligned} & P(D|U=k) \\ &= \sum_{\binom{n}{k}} P(D|T_{j_1}, \dots, T_{j_k}, U=k)P(T_{j_1}, \dots, T_{j_k}|U=k) \ . \end{aligned}$$

Assuming the initial prior is uniform, we get

$$
\begin{aligned}
& P(T_i|D,U=k) \\
&= \frac{P(T_{j_1},\ldots,T_{j_{k-1}}|T_i,U=k)P(T_i|U=k)}{P(T_{j_1},\ldots,T_{j_k}|U=k)} \\
&\quad \times \frac{\sum_{\binom{n}{k-1}} P(D|T_i,T_{j_1},\ldots,T_{j_{k-1}},U=k)}{\sum_{\binom{n}{k}} P(D|T_{j_1},\ldots,T_{j_k},U=k)} \\
&= \frac{\frac{1}{\binom{n-1}{k-1}}\frac{\binom{n-1}{k-1}}{\binom{n}{k}}}{\frac{1}{\binom{n}{k}}} \frac{\sum_{\binom{n}{k-1}} P(D|T_i,T_{j_1},\ldots,T_{j_{k-1}},U=k)}{\sum_{\binom{n}{k}} P(D|T_{j_1},\ldots,T_{j_k},U=k)} \\
&= \frac{\sum_{\binom{n}{k-1}} P(D|T_i,T_{j_1},\ldots,T_{j_{k-1}},U=k)}{\sum_{\binom{n}{k}} P(D|T_{j_1},\ldots,T_{j_k},U=k)} .
\end{aligned}
$$

Therefore, assuming that the detections are independent of each other, we can write the above equation as

$$
\begin{aligned}
& P(T_i|D,U=k) \\
&= P(D_i|T_i)\frac{\sum_{\binom{n}{k-1}} P(D\setminus D_i|T_{j_1},\ldots,T_{j_{k-1}},U=k)}{\sum_{\binom{n}{k}} P(D|T_{j_1},\ldots,T_{j_k},U=k)}
\end{aligned}
$$

The main advantage of this formulation is that we have a very significant reduction in memory requirement as we are only storing $2n$ numbers at each stage instead of $\binom{n}{k}$. We will later show how to *exactly* compute the numerator and denominator in the above equation using Newton's identities from the theory of symmetric polynomials that further leads to computational savings.

B. Unknown number of Target Case

In this case, we consider that the number of targets is unknown *apriori*. However, we assume that the maximum number of targets possible is $\bar{T}$. This case is considerably difficult than the previous case where the number of targets were known *apriori*. The belief map update proceeds in the following manner:

$$
\begin{aligned}
& P(T_i|D) \\
&= \sum_{j=1}^{\bar{T}} P(T_i|D,U=j)P(U=j|D) \\
&= \sum_{j=1}^{\bar{T}} \frac{P(D|T_i,U=j)P(T_i|U=j)P(U=j|D)}{P(D|U=j)} \\
&= \sum_{j=1}^{\bar{T}} \frac{P(D|T_i,U=j)P(T_i|U=j)P(U=j)}{P(D)} \\
&= \sum_{j=1}^{\bar{T}} \frac{P(D|T_i,U=j)P(T_i|U=j)P(U=j)}{\sum_{l=1}^{\bar{T}} P(D|U=l)P(U=l)}
\end{aligned}
$$

Assuming uniform prior for $P(U=j)$, we get

$$
\begin{aligned}
& P(T_i|D) \\
&= \sum_{j=1}^{\hat{T}} \frac{P(D|T_i, U=j)P(T_i|U=j)}{\sum_{l=1}^{\hat{T}} P(D|U=l)} \\
&= \frac{\sum_{j=1}^{\hat{T}} P(D|T_i, U=j)P(T_i|U=j)}{\sum_{l=1}^{\hat{T}} P(D|U=l)} .
\end{aligned}
$$

After a few manipulations, we get

$$
\begin{aligned}
& P(T_i|D) \\
&= \frac{\sum_{j=1}^{\hat{T}} \frac{P(D_i|T_i)}{\binom{n}{j}} \sum_{\binom{n}{j-1}} P(D \setminus D_i|T_{l_1}, \ldots, T_{l_{j-1}}, U=j)}{\sum_{l=1}^{\hat{T}} \frac{1}{\binom{n}{l}} \sum_{\binom{n}{l}} P(D|T_{l_1}, \ldots, T_{l_T}, U=l)} .
\end{aligned}
$$

Now, for both the single and multiple target case, we need to compute

$$P(D|T_{i_1}, \ldots, T_{i_j}, U=j) = \prod_{l=1}^{n} P(D_l|T_{i_1}, \ldots, T_{i_j}, U=j)$$

where D_l is the detection data corresponding to cell l. Define

$$a_l = P(D_l|T_{i_1}, \ldots, T_{i_j}, U=j) = \binom{d_l+m_l}{d_l} p_d^{d_l}(1-p_d)^{m_l}$$

if $l \in \{i, i_1, \ldots, i_j\}$ and

$$b_l = P(D_l|T_{i_1}, \ldots, T_{i_j}, U=j) = \binom{d_l+m_l}{m_l} p_f^{d_l}(1-p_f)^{m_l}$$

otherwise. Therefore, we need an algorithm to efficiently compute terms of the form

$$
\begin{aligned}
& \sum_{\binom{n}{j}} P(D|T_{i_1}, \ldots, T_{i_j}, U=j) \\
&= \sum_{\binom{n}{j}} \prod_{l=1}^{n} P(D_l|T_{i_1}, \ldots, T_{i_j}, U=j) && (12) \\
&= \sum_{\binom{n}{j}} a_{i_1}, \ldots, a_{i_j} b_{i_{j+1}} \ldots b_n && (13)
\end{aligned}
$$

This can be written as

$$\sum_{\binom{n}{j}} P(D|T_{i_1}, \ldots, T_{i_j}, U=j) = P_B \sum_{1 \le i_1 < \ldots < i_j \le n} c_{i_1}, \ldots, c_{i_j} \qquad (14)$$

where $P_B = b_1 \ldots b_n$ and $c_i = \frac{a_i}{b_i}$.

As one can see, computing $P(T_i|D)$ involves a larger number of operations and can considerably slow down the belief map update. We are faced with the following problem. Given n numbers $c_1, \ldots c_n$ and an integer k,

compute the symmetric polynomials

$$J_k = \sum_{1 \le i_1 < \ldots < i_k \le n} c_{i_1} \ldots c_{i_k} \tag{15}$$

in an efficient manner? In the next section, we briefly review the theory of symmetric polynomials and show how the symmetric polynomials can be expressed in terms of the power sums using *Newton's identities*. This novel application of Newton's identities considerably reduces the number of computations needed to update the belief map.

C. Symmetric Polynomials and Newton's Identities

A symmetric polynomial on n variables $x_1, \ldots, x_n$ is a function that is invariant to any permutation of its variables. That is, the symmetric polynomials satisfy

$$f(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_n), \tag{16}$$

where $y_i = x_{\pi i}$ and π being an arbitrary permutation of the indices $1, 2, \ldots, n$. The elementary symmetric polynomials $J_k(x_1, x_2, \ldots, x_n)$ are given by

$$\begin{aligned}
J_1(x_1, x_2, \ldots, x_n) &= \sum_{1 \le i \le n} x_i \\
J_2(x_1, x_2, \ldots, x_n) &= \sum_{1 \le i < j \le n} x_i x_j \\
J_3(x_1, x_2, \ldots, x_n) &= \sum_{1 \le i < j < k \le n} x_i x_j x_k \\
&\vdots \\
J_n(x_1, x_2, \ldots, x_n) &= \prod_{1 \le i \le n} x_i
\end{aligned}$$

The power sum $S_p(x_1, x_2, \ldots, x_n)$ is defined as

$$S_p(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{n} x_k^p \tag{17}$$

The relation between the symmetric polynomials J_k and the power sums S_p is given by *Newton's* identities. The first few identities are

$$J_1 = S_1 \tag{18}$$

$$J_2 = \frac{1}{2}\left(S_1^2 - S_2\right) \tag{19}$$

$$J_3 = \frac{1}{6}\left(S_1^3 - 3S_1S_2 + 2S_3\right) \tag{20}$$

$$J_4 = \frac{1}{24}\left(S_1^4 - 6S_1^2S_2 + 3S_2^2 + 8S_1S_3 - 6S_4\right) \tag{21}$$

In general, the symmetric polynomials can be computed using the following determinant

$$J_k = \begin{vmatrix} S_1 & 1 & 0 & 0 & \cdots & 0 \\ \frac{1}{2}S_2 & \frac{1}{2}S_1 & 1 & 0 & \cdots & 0 \\ \frac{1}{3}S_3 & \frac{1}{3}S_2 & \frac{1}{3}S_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 1 \\ \frac{1}{k}S_k & \frac{1}{k}S_{k-1} & \frac{1}{k}S_{k-2} & \frac{1}{k}S_{k-3} & \cdots & \frac{1}{k}S_1 \end{vmatrix} \tag{22}$$

All of these identities can be computed very easily using any commercial symbolic package. The main advantage of writing it in this form is that computing S_i is much cheaper as it involves vector processing. For example, if $n = 100$ and $k = 4$, the brute force implementation using four **for** loops takes around 25 seconds on a MacBook Pro laptop whereas using the Newton's identities, it takes only about 10^{-4} seconds! This clearly shows that one can get computational speed-up of orders of magnitude by using Newton's identities.

V. Simulation Results

In this section, we provide simulation results for a grid of size 15×15. Note that the state space for the system is $\binom{225}{5}$ for 5 targets. We choose 5 target at locations $43, 77, 99, 155, 216$ at $t = 0$. The initial *prior* is assumed to be uniform. The values of p_d and p_f are chosen to be 0.9 and 0.1 respectively. As shown in the Figure (2) , the belief map converges to the expected value in about 80 iterations. The *optimal teleporting* scheme gives lower bounds on the detection times and serves as a baseline against which other strategies can be compared. An immediate extension of this strategy is *local optimal teleporting* where each vehicle is moved to a neighboring cell with the maximum belief map value. This is ongoing work. Preliminary results suggest that the local optimal reassignment strategy performs satisfactorily. The vehicle dynamics are more realistic in this case.

We have also examined the effect of the number of search vehicles and p_d on the detection times. As expected, as the number of search vehicles increases, the detection time decreases with the number of search vehicles. One such simulation result is shown in Fig. 3. However, it is interesting to note that beyond a critical value of the number of vehicles, the detection time does not change much with the number of vehicles. Such an observation can provide guidelines in choosing the number of search vehicles for a given search mission. Figure 4 shows the variation of the detection times with p_d. Once again, as expected the detection times decreases almost linearly with p_d.

VI. Conclusions

In this paper, we have demonstrated optimal Bayesian updates for detecting unknown number of targets in a region that is divided into cells. The belief states of the system grow exponentially and cannot be solved exactly. We develop a different formulation and show that we can indeed solve the problem exactly using results from the theory of symmetric polynomials and using the detection history. As future work, we will provide theoretical estimates of the detection times when the vehicles are optimally teleported. We will also compare local optimal vehicle reassignment with teleporting.

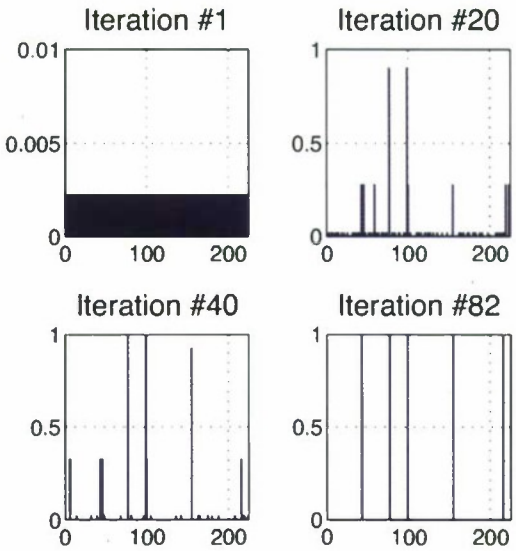


Fig. 2. Evolution of belief map for a 15×15 grid. The cell number is plotted on the x-axis and the belief map values are plotted on the y-axis. After the 82nd iteration, targets are detected at location 43,77,99,155,216.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge DARPA for funding this research. This research was partially supported by the DARPA Dynarum program through AFOSR contract FA9550-07-C-0024. This material is approved for public release; distribution is unlimited.

REFERENCES

[1] S. Blackman, *Multiple-Target Tracking with Radar Applications*. Norwood, MA: Artech House, 1986.

[2] Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances*. Norwood, MA: Artech House, 2000.

[3] C. A. B. L. D. Stone and T. Corwin, *Bayesian Multiple Target Tracking*. Norwood, MA: Artech House, 1999.

[4] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1993.

[5] M. A. B. Ristic and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA: Artech House, 2004.

[6] A. S. M. I. Miller and U. Grenander, "Conditional mean estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Trans. Signal Process.*, vol. 48, no. 11, pp. 2678–2690, 1995.

[7] K. Kastella, "Event averaged maximum likelihood estimation and mean-field theory in multitarget tracking," *IEEE Trans. Autom. Control*, vol. 50, no. 6, pp. 1070–1073, 1995.

[8] R. Mahler, "Statistics 101 for multisensor, multitarget fusion," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 53–64, 2004.

[9] K. Kastella, "A maximum likelihood estimator for report-to-track association," in *Proc. SPIE*, vol. 1954, 1993, pp. 386–393.

[10] K. K. C. M. Kreucher and A. O. Hero, "Tracking multiple targets using a particle filter representation of the joint mulitarget probability density," in *Proc. SPIE Conf. Signal Data Processing of Small Targets*, 2003, pp. 258–269.

[11] N. G. M. Arulampalam, S. Maskell and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process*, vol. 50, no. 2, pp. 174–188, 2002.

[12] C. K. M. Morelande and K. Kastella, "A bayesian approach to multiple target detection and tracking," *EEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1589–1604, 2007.

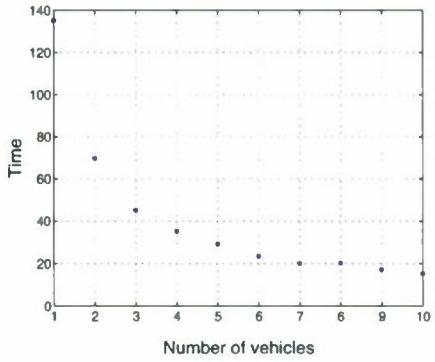


Fig. 3. Expected time versus the number of search vehicles

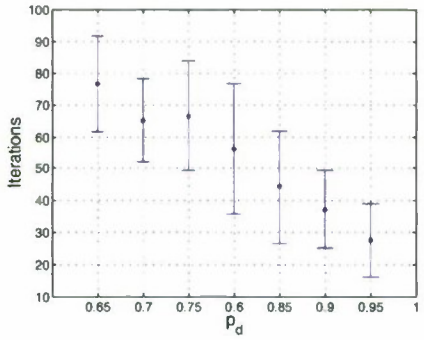


Fig. 4. Expected time (number of iterations to detect) versus p_d

[13] I. G. Macdonald, *Symmetric Functions and Hall Polynomials*. Oxford Clarendon Press, 1995.

Appendix B

Novel numerical methods and system analysis tools

B.1 Scalable uncertainty quantification in complex dynamical networks

Scalable uncertainty quantification in complex dynamic networks

Amit Surana & Andrzej Banaszuk

January 12, 2009

1 Abstract

Many large scale systems of interest (e.g. power systems, biological networks) are often composed of weakly interacting subsystems. We propose an iterative scheme that exploits such weak interconnections to overcome dimensionality curse associated with traditional uncertainty quantification methods and radically accelerate uncertainty propagation in systems with large number of uncertain parameters. This approach relies on integrating graph theoretic methods and waveform relaxation with uncertainty quantification techniques like probabilistic collocation and polynomial chaos. We analyze convergence properties of this scheme and illustrate it on a power network.

2 Introduction

The issue of management of uncertainty for robust system operation is of interest in a large family of complex networked systems. Such systems typically involve a large number of heterogeneous, connected components, whose dynamics is affected by possibly an equally large number of parameters. Uncertainty Quantification (UQ) methods provide means of calculating probability distribution of system outputs, given probability distribution of input parameters. One of the most commonly used methods is Monte Carlo sampling (MCS), or one of its variants. Although MCS is straightforward to apply as it only requires repetitive executions of deterministic simulations of the system, typically a large number of such executions are needed as the solution statistics converge relatively slowly, e.g., the mean value typically converges as $\mathcal{O}(1/\sqrt{N})$ where N is the number of realizations [13]. Quasi Monte Carlo (QMC) methods on the other hand offer better convergence for moderate number of random parameters, the rate being $\mathcal{O}(\frac{(\log N)^p}{N})$, where p is the number of random parameters in the system. However, while MCS suffer from poor coverage of the space being sampled, QMC methods often lead to incorrect density of the sampled points. Recently, Mezic [11] developed a new scheme known as *DSample*, which exploits ergodic dynamics to generate samples which do not suffer form such limitation: the sample points have proper coverage and correct density in the sample space. Most importantly the convergence of this method follows the fast $\mathcal{O}(1/N)$ scaling, independent of p the number of random parameters.

Generalized polynomial chaos (gPC) is another recently developed technique which belongs to the class of non-sampling UQ methods. In gPC, stochastic quantities are expressed as orthogonal polynomials of the input random parameters, and different types of orthogonal polynomials can be chosen to achieve better convergence (under certain circumstances, exponential convergence can also be attained) [12]. When applied to differential equations with random inputs, the gPC expansion is typically combined with Galerkin projection, such that the resulting set of equations for the expansion coefficients are deterministic and can be solved via conventional numerical techniques. However, stochastic Galerkin (SG) procedure can be challenging when the governing stochastic equations take a complicated form. To this end, high-order probabilistic collocation method (PCM)

has been developed [14]. PCM combines the advantages of both Monte Carlo sampling and gPC-Galerkin method. The implementation of a PCM algorithm is similar to that of MCS, i.e., only repetitive realizations of a deterministic solver is required; and by choosing a set of sampling points based on the theory of multivariate polynomial interpolations, it retains the high accuracy and fast convergence of gPC expansion, similar to SG. In higher dimensions, however, the use of standard tensor products of one-dimensional quadrature points as sampling points leads to an exponential growth of the number of points. The work of [14], is a first systematical attempt to avoid using tensor product constructions. Instead it employs the so-called *sparse grid* [5], to tackle problems with large number of random variables more efficiently. In addition, to deal with UQ in PDE systems, mutli element formulation of gPC and PCM have also been recently developed [7, 2].

As described above, while, there have been various efforts to overcome dimensionality curse associated with UQ methods, none of such extension exploits the underlying structure and dynamics of the networked systems. In fact, many networks of interest (e.g. power systems, biological networks), are often composed of weakly interacting subsystems. As a result, it is plausible to simplify and accelerate the simulation, analysis and uncertainty propagation in such systems by suitably decomposing them. For instance, authors in control theory studied large-scale interconnected dynamical systems using graph theoretic decomposition of the system. These studies, however, concentrated mostly on questions of stability and robustness (see e.g. [15, 16]). In contrast Mezic et al. [10], introduced a framework for studying more general asymptotic behavior and uncertainty propagation in such multicomponent nonlinear systems. They showed that, the use of graph decomposition in conjunction with Perron Frobenius operator theory can greatly simplify the invariant measure structure and uncertainty quantification, for a particular class of networks. While this approach exploits the underlying structure of the system, it does not take advantage of the weakly coupled dynamics of the subsystems.

In this paper, we propose an iterative UQ approach that exploits the weak interactions among subsystems in a networked system to overcome the dimensionality curse associated with traditional UQ methods, and radically accelerate uncertainty propagation. This approach relies on integrating graph decomposition techniques and waveform relaxation scheme, with probablistic collocation and generalized polynomial chaos. Graph decomposition can be realized by spectral graph theoretic techniques to identify weakly interacting subsystems. Waveform relaxation, a parallelizable iterative method, on the other hand, exploits this decomposition and evolves each subsystem forward in time independently but coupled with the other subsystems through their solutions from the previous iteration. At each waveform relaxation iteration we propose to apply PC at subsystem level and use gPC to propagate the uncertainty among the subsystems. Since UQ methods are applied to relatively simpler subsystems which typically involve a few parameters, this renders a scalable iterative approach to UQ in complex networks. In an alternative approach the random differential equations can be transformed to a deterministic system, by employing a stochastic Galerkin projection. Subsequently, graph decomposition and waveform relaxation can be applied to accelerate simulation of this deterministic system, leading to uncertainty quantification in the original system.

This paper is organized in six sections. In section 3 we set up the mathematical framework and state precisely the problem of uncertainty quantification. Section 4 deals with spectral graph decomposition while waveform relaxation is described in section 5. We give an overview of gPC and PCM methods, in the section 3. These techniques form the basic ingredients of the scalable approaches to UQ, which are discussed in section 7. We propose two such iterative approaches: the first one requires access to equations which describe the underlying dynamics of the system, while the other one treats the system as a black box. In section 8 we illustrate this iterative procedure on a simplified power system network and numerically analyze its convergence properties. Finally, in section 9 we summarize the main results of this paper and lay down some future research questions.

3 Uncertainty Quantification in Networked Systems

Consider a nonlinear system described by as system of random differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \xi, t), \tag{1}$$

where, $\mathbf{f} = (f_1, f_2, \cdots, f_n) \in \mathbb{R}^n$ is a smooth vector field, $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$ are state variables and $\xi = (\xi_1, \xi_2, \cdots, \xi_p) \in \mathbb{R}^p$ is p dimensional vector of uncertain parameters of interests. The solution to initial value problem $\mathbf{x}(t_0) = \mathbf{x}_0$ will be denoted by $\mathbf{x}(t;\xi)$, where for brevity we have suppressed the dependence of solution on initial time t_0 and initial condition $\mathbf{x}_0$. The Jacobian J associated with system (1) is given by

$$J(\mathbf{x}, \xi, t) = [\frac{\partial f_i(\mathbf{x}(t;\xi), \xi, t)}{\partial x_j}], \tag{2}$$

and describes the linearized dynamics of the system about the solution trajectory $\mathbf{x}(t;\xi)$. The average value of Jacobian along the solution for nominal value of parameters ξ_m, will be denoted by

$$\overline{J} = [\frac{1}{T}\int_{t_0}^{t_0+T} J_{ij}(\mathbf{x}(t;\xi_m), \xi_m, t)dt]. \tag{3}$$

Let us also define a set of quantities

$$\mathbf{z} = (z_1, z_2, \cdots, z_d) = G(\mathbf{x}) = (g_1(\mathbf{x}, \cdots, g_d(\mathbf{x}))), \tag{4}$$

as observables or quantities of interests. The goal is to numerically establish the effect of input uncertainty of ξ on output observables $\mathbf{z}$.

In what follows we will adopt a probabilistic framework and model $\xi = (\xi_1, \xi_2, \cdots, \xi_p)$ as a $p-$variate random vector with independent components in the probability space $(\Omega, \mathcal{A}, \mathcal{P})$, whose event space is Ω and is equipped with $\sigma-$algebra $\mathcal{A}$ and probability measure $\mathcal{P}$. Without loss of generality, we would assume that these parameters $(\xi_1, \xi_2, \cdots, \xi_p)$ are mutually independent of each other. Let $w_i : \Gamma_i \to \mathbb{R}^+$ be the probability density of the random variable $\xi_i(\omega)$, with $\Gamma_i = \xi_i(\Omega) \subset \mathbb{R}$ being its image. Then,

$$w(\xi) = \prod_{i=1}^{p} w_i(\xi_i), \quad \forall \xi \in \Gamma \tag{5}$$

is the joint probability density of the random vector $\xi = (\xi_1, \cdots, \xi_p)$ with the support,

$$\Gamma = \prod_{i=1}^{p} \Gamma_i \subset \mathbb{R}^p. \tag{6}$$

Without loss of generality we would assume that $\Gamma_i = [-1 \quad 1], i = 1, \cdots, p$. Naturally, the solution for system (1) and the observables (4) are functions of same set of of random variables ξ, i.e

$$\mathbf{x} = \mathbf{x}(t;\xi), \qquad \mathbf{z} = \mathbf{z}(t,\xi) = G(\mathbf{x}). \tag{7}$$

As described in the introduction,, this problem of uncertainty quantification in large systems ($n \gg 1$) with large number of uncertain parameters ($p \gg 1$) is computationally intensive. Recently developed UQ methods like Dsample [11] and sparse-grid based probabilistic methods [14] can handle this complexity to some extent. These methods however do not make use of the underlying structure and the dynamics of the system, which can be often be taken to an advantage. The goal of this paper is to develop scalable uncertainty quantification approaches which exploits this structure and dynamics. The key methodologies for accomplishing this are the spectral graph decomposition, waveform relaxation, and gPC and PCM, which are discussed in the subsequent sections.

4 Graph Decomposition

The problem of partitioning the system of equations (1) into subsystems based on how they interact or are coupled to each other, can be formulated as a graph decomposition problem. Given the set of states $x_1, \cdots, x_n$ and some notion of dependence $w_{ij} \geq 0, i = 1, \cdots, n, j = 1, \cdots, n$ between pairs of states, a graph $G = (V, E)$ can be constructed. The vertices v_i in this graph represent the states x_i and two vertices are connected with an edge of weight w_{ij}, if w_{ij} between the corresponding states x_i and x_j is positive (or larger than a certain threshold).

Different choices of the weight matrix $W = [w_{ij}]$, would lead to different decompositions of the graph. For instance, Mezic et al. [10], proposed a horizontal-vertical decomposition (HVD) of the graph G based on the structural properties encoded in the Jacobian 2, associated with the system 1. Specifically, in this case

$$w_{ij} = \mathcal{I}(|J_{ij}|), \tag{8}$$

where, $\mathcal{I}$ is an indicator function. Note $w_{ij} = 1$ if state i affects state j, and $w_{ij} = 0$ otherwise. The HVD of the graph is obtained by recursively identifying transient and recurrent non-null sets of the Markov chain corresponding to the weight matrix W, with weights given by (8). Vertically, the system is decomposed into a linear series of subsystems, where the subsystems above is influenced by the subsystems below but not vice versa. So, the input signal propagates unidirectionally from the bottom to the top. Horizontally, each subsystem is decomposed into independent groups with no edges connecting different groups. So, each group has its own input and output and are functioning independently. Each group in HVD is a connected component of the graph and HVD creates a partial ordering on the set of connected components.

Another plausible partition of G is to assign the nodes into different components such that nodes in the same components are strongly coupled and nodes in different components are weakly coupled to each other. This requires a notion of coupling strength between nodes or states, which would depend on the nature of the problem; for our case, we propose to use

$$w_{ij} = \frac{1}{2}[|\overline{J}_{ij}| + |\overline{J}_{ji}|], \tag{9}$$

which measures the interdependence of states x_i and x_j on each other, corresponding to the linearized dynamics. The problem of decomposition can now be formulated as follows: we want to find a partition of the graph G with edge weights (9) such that the edges between different components have a very low weight and the edges within a components have high weight. The main tools for accomplishing this decomposition are graph Laplacian matrices. There exists a whole field dedicated to the study of those matrices, called spectral graph theory [19]. Note that in the literature, there is no unique convention as to which matrix is exactly called the graph Laplacian and how the different matrices are denoted. In the following we always assume that G is an undirected, weighted graph with weight matrix W, where $w_{ij} = w_{ji} \geq 0$. The unnormalized graph Laplacian matrix is defined as

$$L = D - W, \tag{10}$$

where, D is the diagonal degree matrix whose ith diagonal entry $d_i = \sum_j w_{ij}$. There are two other matrices which are called normalized graph Laplacians in the literature. Both matrices are closely related to each other and are defined as

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}, \tag{11}$$

$$L_{rw} = D^{-1} L = I - D^{-1} W. \tag{12}$$

The procedure to decompose the graph using graph Laplacian, is summarized below, details can be found in [19]. Given m, the number of components or subsystems,

1 Construct the graph $G = (V, E)$ by the procedure described above. Let W be its weighted adjacency matrix.

2 Compute the graph Laplacian L.

3 Compute the first m eigenvectors $v_1, \cdots, v_m$ of L.

4 Let $V \in R^{n \times m}$ be the matrix containing the vectors $v_1, \cdots, v_m$ as columns.

5 For $i = 1, \cdots n$ let $u_i \in R^m$ be the vector corresponding to the i-th row of V.

6 Cluster the points $u_i, i = 1, \cdots, n$ in R^m with the k-means algorithm into clusters $C_1, \cdots, C_m$.

7 Output: Clusters $A_1, \cdots, A_m$ with $A_i = \{j | u_j \in C_i\}$.

Typically, m the number of weakly coupled subsystems (or components) in the system, is not known apriori. The spectral gap in the spectrum of the Laplacian can be used to identify m. For discussion on other methods, see [19]. We shall denote by

$$\mathcal{N}_i = \{j : \exists k \in A_i \quad \& \quad \exists l \in A_j, \text{ st } s_{kl} > 0\}, \quad i = 1, \cdots, m, \tag{13}$$

the set of indices of the components (or subsystems) to which the $i-$th component (or subsystem) is weakly connected.

In practise HVD and spectral decomposition can be combined: the connected components identified in horizontal layers after HVD can be further decomposed into weakly interacting subsystems by using graph Laplacians, as described above. In summary the graph decomposition partitions the system into appropriate subsystems, allowing the application of waveform relaxation, an iterative scheme to accelerate simulation of a system composed of weakly coupled subsystems.

5 Waveform Relaxation

In this section we describe the basic mathematical concept of the Waveform Relaxation (WR) method. We consider dynamical systems which are described by a system of differential equations of the form (1). For purpose of discussion here, we would assume that the parameter values ξ in the system (1) are fixed. The general structure of a WR algorithm for analyzing system (1) over a given time interval $[0 \quad T]$ consists of two major processes, namely the *assignment partitioning process* and the *relaxation process* [17, 18].

In the assignment-partitioning process, the system is partitioned into m disjoint subsystems of equations. This partition can be accomplished, for instance by the graph decomposition procedure, as described in the previous section. Without loss of generality, we can rewrite Eq. 1 after the assignment-partitioning process as:

$$\begin{aligned} \dot{\mathbf{y}}_1 &= \mathbf{F}_1(\mathbf{y}_1, \mathbf{d}_1(t), \Lambda_1, t) \\ \dot{\mathbf{y}}_2 &= \mathbf{F}_2(\mathbf{y}_2, \mathbf{d}_2(t), \Lambda_2, t) \\ &\vdots \\ \dot{\mathbf{y}}_m &= \mathbf{F}_m(\mathbf{y}_m, \mathbf{d}_m(t), \Lambda_m, t) \end{aligned} \tag{14}$$

where, for each $i = 1, \cdots, m$,

$$\mathbf{F}_i \equiv \{f_j : j \in A_i\}, \tag{15}$$

$\mathbf{y}_i \in \mathbb{R}^{p_i}$ is the subvector of $\mathbf{x}$ assigned to the $i-$th partitioned subsystem, i.e.

$$\mathbf{y}_i(t) \equiv \{x_j(t) : j \in A_i\}, \tag{16}$$

$$\Lambda_i(t) \equiv \{\xi_j : j \in A_i\}, \tag{17}$$

and

$$\mathbf{d}_i(t) \equiv \{\mathbf{y}_{j_i}(t) : j_i \in \mathcal{N}_i\}, \tag{18}$$

are decoupling vectors. It is clear that if the vectors $\mathbf{d}_i(t)$ are treated as the input variables of the system described by Eq. 14, then the system can be easily solved by solving m independent subsystems associated with $\mathbf{F}_1, \cdots, \mathbf{F}_m$ respectively.

The relaxation process is an iterative process, which starts with an initial guess of the waveform solution of the original dynamical equations (14) in order to initialize the approximate waveforms of the decoupling vectors. During each iteration, each decomposed subsystem is solved for its assigned variables for $t \in [0 \quad T]$ by using the approximate waveform of its decoupling vector. Two most commonly used types of relaxation are: Gauss-Seidel(GS) relaxation and the Gauss-Jacobi (GJ) relaxation. For the GS relaxation, the waveform solution obtained by solving one decomposed subsystem is immediately used to update the approximate waveforms of the decoupling vectors of the other subsystems. For the GJ relaxation, all waveforms of the decoupling vectors are updated at the beginning of the next iteration. The relaxation process is carried out repeatedly until satisfactory convergence is achieved.

Let the superscript index I denote the WR iteration count. Then the general structure of a WR algorithm can be formally described as follows:

- Step 0 (Assignment-partitioning process): Partition (1) into m subsystems of equations as given by (14).
- Step 1: (Initialization of the relaxation process): Set $I = 1$ and guess an initial waveform $\{\mathbf{y}_i^0(t) : t \in [0 \quad T]\}$ such that $\mathbf{y}_i^0(0) = \mathbf{y}_{0i}$.
- Step 2 (Analyzing the decomposed system at the I-th WR iteration): For each $i = 1, \cdots, m$, set
$$\mathbf{d}_i^I(t) = \{\mathbf{y}_{j_i}^{I-1}(t) : j_i \in \mathcal{N}_i\} \tag{19}$$
for the GJ relaxation, and solve for $\{\mathbf{y}^I(t) : t \in [0 \quad T]\}$ from
$$\dot{\mathbf{y}}_i^I = \mathbf{F}_i(\mathbf{y}_i^I, \mathbf{d}_i^I(t), \Lambda_i, t), \tag{20}$$
with initial condition $\mathbf{y}_i^I(0) = \mathbf{y}_{i0}$.
- Step 3 (Iteration) Set $I = I + 1$ and go to step 2.

Remarks

1 A simple guess for $\{\mathbf{y}_i^0(t)\}$ is $\mathbf{y}_i^0(t) = \mathbf{x}_i(0)$ for all $t \in [0 \quad T]$.

2 In the actual implementation, the relaxation iteration will stop when the difference between the waveforms $\{\mathbf{y}^I(t) = (\mathbf{y}_1^I(t), \cdots, \mathbf{y}_m^I(t)) : t \in [0, T]\}$ and $\{\mathbf{y}^{I-1}(t) = (\mathbf{y}_1^{I-1}(t), \cdots, \mathbf{y}_m^{I-1}(t)) : t \in [0 \quad T]\}$,i .e. $\sup_{t\in[0 \quad T]} ||\mathbf{y}_I(t) - \mathbf{y}_{I-l}(t)||$, is sufficiently small.

3 In analogy to the classical relaxation methods for solving linear or nonlinear algebraic equations, it is possible to modify a WR algorithm by using a relaxation parameter $\omega \in (0, 2)$. By introducing ω, the iteration equation (20) gets modified to yield
$$\dot{\tilde{\mathbf{y}}}_i^I = \mathbf{F}_i(\tilde{\mathbf{y}}_i^I, \mathbf{d}_i^I(t), \Lambda_i, t), \tag{21}$$
where,
$$\mathbf{y}_i^I = \mathbf{y}_i^{I-1} + \omega(\tilde{\mathbf{y}}_i^I - \mathbf{y}_i^{I-1}). \tag{22}$$

Note the following two important characteristics of the WR Algorithm:

1 The analysis of the original system is decomposed into the independent analysis of m subsystems.

2 The relaxation process is carried out on the entire waveforms, i.e. during each iteration each subsystem is individually analyzed for the entire given time interval $[0, T]$.

The conditions that guarantee the convergence of WR method have been analyzed in detail in [17].

6 Uncertainty Quantification Methods

As discussed in introduction, UQ methods can be classified as sampling and non-sampling based. In this section, we describe two such interrelated approaches: generalized polynomial chaos (gPC) and probablistic collocation method (PCM).

6.1 Generalized Polynomial Chaos

In the finite dimensional random space Γ defined in (6), the gPC expansion seeks to approximate a random process via orthogonal polynomials of random variables. Let us define one-dimensional orthogonal polynomial spaces

$$W^{k,d_k} \equiv \{v : \Gamma_k \to \mathbb{R} : v \in \text{span}\{\psi_i(\xi_k)\}_{i=0}^{d_k}\}, \qquad k = 1, \cdots, p, \tag{23}$$

where, $\{\psi_i(\xi_k)\}$ denotes the polynomial basis from the so called Wiener-Askey polynomial chaos [12]. According to the Cameron-Martin theorem [21], the Wiener-Askey polynomial chaos expansion can approximate and describe all stochastic processes with finite second-order moment, which is satisfied for most physical systems. The Askey scheme of polynomials contains various classes of orthogonal polynomials and with their associated weighting functions which coincide with probability density function of different distributions. For example, uniform distributions are associated with Legendre polynomials, and Gaussian distributions are associated with Hermite polynomials. An important property of the Wiener-Askey polynomial chaos is orthogonality:

$$\int_{\Gamma_i} \psi_i(\xi_k)\psi_j(\xi_k)w_k(\xi_k)d\xi_k = \delta_{ij}, \tag{24}$$

where, δ_{ij} is the Kronecker delta.

The corresponding P-variate orthogonal polynomial space in Γ is defined as

$$W_p^P \equiv \bigotimes_{|\mathbf{d}| \le P} W^{i,d_i} \tag{25}$$

where the tensor product is over all possible combinations of the multi-index $\mathbf{d} = (d_1, d_2, \cdots, d_p) \in \mathbb{N}^p$ satisfying $|\mathbf{d}| = \sum_{i=1}^p d_i \le P$. Thus, W_p^P is the space of N-variate orthonormal polynomials of total degree at most P, and its basis functions satisfy

$$\int_\Gamma \Psi_i(\xi)\Psi_j(\xi)\mathbf{w}(\xi)d\xi = \mathcal{E}(\Psi_i(\xi)\Psi_j(\xi)) = \delta_{ij}, \tag{26}$$

for all $1 \le i, j \le \dim(W_p^P) = M = \frac{(P+p)!}{P!p!}$, where $\mathcal{E}$ is the expectation operator.

The major advantage of applying the gPC is that a random differential equation can be transformed into a system of deterministic equations. A typical approach is to employ a stochastic Galerkin projection, in which all the state variables $x_1, x_2, \cdots, x_n$ are expanded in polynomial chaos basis with corresponding modal coefficients $(a_i^k(t))$, as

$$x_k^P(t,\xi) = \sum_{i=1}^M a_i^k(t)\Psi_i(\xi), \quad k = 1, \cdots, n. \tag{27}$$

Substituting, these expansions in Eq. (1), and using the orthogonality property of polynomial chaos (25), we obtain

$$\dot{a}_j^k = \int_\Gamma f_k(x_1^P(t,\xi), \cdots, x_n^P(t,\xi), \xi, t)\Psi_j(\xi)\mathbf{w}(\xi)d\xi, \quad k = 1, \cdots, n, \quad j = 1, \cdots, M, \tag{28}$$

a set of deterministic modal ODEs, with initial conditions

$$a_j^k(0) = \int_\Gamma x_k(0,\xi)\Psi_j(\xi)\mathbf{w}(\xi)d\xi, \quad k = 1,\cdots,n, \quad j = 1,\cdots,M. \tag{29}$$

This system can be solved with any numerical method dealing with initial-value problems, e.g., the Runge-Kutta method. Similarly, the observable can be expanded in gPC basis, as

$$z_k^P(t,\xi) = \sum_{i=1}^{M} b_i^k(t)\Psi_i(\xi), \quad k = 1,\cdots,d, \tag{30}$$

where,

$$b_j^k = \int_\Gamma z_k(\xi)\Psi_j(\xi)\mathbf{w}(\xi)d\xi = \int_\Gamma g_k(\mathbf{x}(t,\xi))\Psi_j(\xi)\mathbf{w}(\xi)d\xi, \quad k = 1,\cdots,d, \quad j = 1,\cdots,M. \tag{31}$$

Hence, once the solution to the system (28) has been obtained, the coefficients b_j^k can be approximated as

$$b_j^k \approx \int_\Gamma g_k(x_1^P(t,\xi),\cdots,x_n^P(t,\xi))\Psi_j(\xi)\mathbf{w}(\xi)d\xi, \quad k = 1,\cdots,d, \quad j = 1,\cdots,M. \tag{32}$$

Such a Galerkin procedure has been used extensively in the literature. However, when (1) takes a complicated form, the derivation of Galerkin projection in (28), and subsequently the gPC approximation of the observable in (32), can become highly non-trivial, if not impossible. To circumvent this difficulty, probablistic collocation method has been developed.

6.2 Probabilistic Collocation Method

The collocation method is an alternative approach to solve stochastic random processes with the gPC. Instead of projecting each state variable onto the polynomial chaos basis, the collocation approach evaluates the integrals of form (31) by evaluating integrand at the roots of the appropriate basis polynomials. Two underlying concepts for the PCM are the orthogonal polynomial and their associated quadrature rule. Given a probability density function $w(\xi)$ (let $p = 1, d = 1$ for now), the PCM based on Gauss quadrature rule, approximates an integral of a function g with respect to density $w(\xi)$, as follows

$$\int_{-1}^{1} g(\xi)w(\xi)d\xi \approx \sum_{r_i \in C_1^p} g(r_i)W_i, \tag{33}$$

where,

$$C_q^1 = \{r_k : \psi_{p+1}(r_k) = 0, k = 1,\cdots,q+1\}, \tag{34}$$

is the set of Gauss collocation points with ψ_i being the orthogonal polynomials corresponding to the probability distribution $w(\xi)$, as described in previous section. The weights W_i are given by

$$W_i = \int_{-1}^{1} \frac{\psi_p(y)}{(y-\xi_i)\psi_p'(x_i)} w(y)dy. \tag{35}$$

The Gauss quadrature formula, which is a well-known numerical integration technique, yields an exact integration value for any function in a polynomial form of order less than or equal to $2q+1$. Other quadrature rules can be also be used, some of which have nested quadrature points. One such rule known as Clenshaw Curtis (CC) quadrature is described in the Appendix.

For higher dimensional discussion (i.e. for $p > 1$), we would denote 1D quadrature rule (Gauss or CC) along each random dimension as

$$\mathcal{U}_{l_j}^1[g] = \sum_{k=1}^{m_{l_j}} W_{l_j k} g(r_{l_j k}), \quad j = 1,\cdots,p, \tag{36}$$

where, l_j is the accuracy level of quadrature formula, and m_{l_j} is the number of quadrature points corresponding to that accuracy level. Building on the 1D quadrature formula, the full grid PCM leads to following cubature rule,

$$\int_{-1}^{1}\int_{-1}^{1}\cdots\int_{-1}^{1} g(\xi_1,\cdots,\xi_p)\mathbf{w}(\xi)d\xi \tag{37}$$

$$\approx I(l_1,\cdots,l_p,p)[g] = (\mathcal{U}_{l_1}^1 \otimes \mathcal{U}_{l_2}^1 \cdots \mathcal{U}_{l_p}^1)[g] \tag{38}$$

$$= \sum_{j_1=1}^{m_{l_1}}\sum_{j_2=1}^{m_{l_2}}\cdots\sum_{j_p=1}^{m_{l_p}}(W_{l_1j_1}W_{l_2j_2}\cdots W_{l_nj_n})g(r_{l_1j_1},\cdots,r_{l_pj_p}) \tag{39}$$

To compute $\mathcal{I}(l_1,\cdots,l_p,p)$ we need to evaluate the function on the full collocation grid $C(\mathbf{l},p)$ (where, $\mathbf{l}=(l_1,\cdots,l_p)$) which is given by tensor product of 1D grids

$$\mathcal{C}(\mathbf{l},p) = C_{l_1}^1 \times \cdots \times C_{l_p}^1, \tag{40}$$

with a total number of collocation points being $Q=\prod_{j=1}^{p} l_j$. In this framework, therefore, for any t, the approximations to the model coefficients a_j^k (see Eq. 27) and b_j^k (see Eq. 30) can be obtained as

$$a_j^k(t) = \int_\Gamma x_k(t,\xi)\Psi_j(\xi)\mathbf{w}(\xi)d\xi \approx \sum_{j_1=1}^{m_{l_1}}\sum_{j_2=1}^{m_{l_2}}\cdots\sum_{j_p=1}^{m_{l_p}}(W_{l_1j_1}W_{l_2j_2}\cdots W_{l_nj_n})x_k(t,r_{l_1j_1},\cdots,r_{l_pj_p}), \tag{41}$$

and

$$b_j^k(t) = \int_\Gamma g_k(\mathbf{x}(t,\xi))\Psi_j(\xi)\mathbf{w}(\xi)d\xi \approx \sum_{j_1=1}^{m_{l_1}}\sum_{j_2=1}^{m_{l_2}}\cdots\sum_{j_p=1}^{m_{l_p}}(W_{l_1j_1}W_{l_2j_2}\cdots W_{l_nj_n})g_k(\mathbf{x}(t,r_{l_1j_1},\cdots,r_{l_pj_p})). \tag{42}$$

Note to compute summations arising in above equations (41,42), the solution $\mathbf{x}(t,r_{l_1j_1},\cdots,r_{l_pj_p})$ of the system (1) is required for each sampling point $(r_{l_1j_1},\cdots,r_{l_pj_p})$ in the full collocation grid $C(\mathbf{l},p)$. Thus, simplicity of collocation framework only requires repeated runs of deterministic solvers (without explicitly requiem access to the system equations), resulting in a faster algorithm than gPC.

If we choose the same order of sampling points in each dimension, i.e. $l_1 = l_2 ... \cdots = l_p \equiv l$, the total number of points is $Q = l^p$. Hence, the computational cost increases rather steeply with the number of uncertain parameters p. This problem can be circumvented by using different cubature rule. One such example is Smolyak rule which requires significantly smaller number of points while maintaining the same accuracy as full grid PCM. Smolyak's algorithm is a method first developed to handle high dimensional quadrature [4] and later extended to accomplish high dimensional interpolation [6]. Its basic idea is to use the solution to several low-dimensional problems to span the space and then linearly combine these to yield the solution to higher dimensional problem [5]. Grid generated using this rule is known as *sparse grid*; for further discussion, the reader is referred to the Appendix.

7 Scalable Uncertainty Quantification Approach

In this section we discuss how generalized polynomial chaos and probabilistic collocation method can be integrated with graph decomposition and waveform relaxation scheme, leading to scalable UQ approaches. We describe two such iterative approaches: the first one requires access to the equations which describe the underlying dynamics of the system, while the other one treats the system as a black box.

7.1 Equation Based Approach

The procedure of transforming a random differential equation to a deterministic system, by employing gPC and stochastic Galerkin projection, was outlined in section 6.1. The deterministic set of ODE's so obtained describe the evolution of modal coefficients (see section 6.1 for details) and can be compactly written as

$$\dot{\mathbf{a}} = \mathbf{H}(\mathbf{a}, t), \tag{43}$$

with the initial condition $\mathbf{a}(0)$ (see Eq. 29 as well), where,

$$\mathbf{a} = (a_1^1, a_2^1, \cdots, a_M^1, a_1^2, \cdots, a_M^2, \cdots, a_1^n, \cdots, a_M^n), \tag{44}$$

$$\mathbf{H} = (h_1^1, h_2^1, \cdots, h_M^1, h_1^2, \cdots, h_M^2, \cdots, h_1^n, \cdots, h_M^n), \tag{45}$$

and

$$h_j^k(\mathbf{a}, t) = \int_\Gamma f_k(\sum_{i=1}^M a_i^1(t)\Psi_i(\xi), \cdots, \sum_{i=1}^M a_i^p(t)\Psi_i(\xi), \xi, t)\Psi_j(\xi)\mathbf{w}(\xi)d\xi, \quad k = 1, \cdots, n, \quad j = 1, \cdots, M. \tag{46}$$

Note that order of the deterministic system (43) is nM, where $M = \frac{(P+p)!}{p!P!}$ and is significantly greater than the order n of the original random system (1). However, by applying graph decomposition and waveform relaxation to the deterministic system (43), we can accelerate the computation of solution to the initial value problem stated above. The main assumption here is that if the system (1 is composed of weakly interacting subsystems, so would be the transformed system (43). In summary, the three step approach to UQ is:

- Step I: Apply gPC and stochastic Galerkin projection to system (1) to obtain a deterministic system (43).
- Step II: Apply graph decomposition (see section 4 for details) to identify weakly interacting subsystems in the system (43).
- Step III: Apply waveform relaxation (see section 5 for details) to the decomposed system obtained in Step II.

After these steps, one can evaluate the effect of uncertainty on observables as described in section 6.1. One of the major limitation of the above approach is that, in order to accomplish step 1 we need access to the equations (1), which may not be readily available for large multicomponent systems. Moreover, even when equations are available, the Galerkin projection step may be very tedious, if not impossible. We describe an alternative approach in the next section, which treats the system more or less as a black box.

7.2 Probabilistic Waveform Relaxation Based Approach

In section 5 we described a deterministic waveform relaxation, in which the decoupling vectors are deterministic function of times. By incorporating UQ methods, we would extend WR to the probabilistic setting we are dealing with. Specifically, at each waveform relaxation iteration we propose to apply PCM at subsystem level and use gPC to propagate the uncertainty among the subsystems. Since UQ methods are applied to relatively simpler subsystems which typically involve a few parameters, this renders a scalable iterative approach to UQ in large networks. Moreover, like in PCM this method does not require access to the equations describing the system dynamics.

Consider the $i-$th subsystem of the system (14), written as system of 1st order ODE's

$$\begin{aligned} \dot{y}_1^i &= F_1^i(\mathbf{y}_i, \mathbf{d}_i(t), \Lambda_i, t) \\ \dot{y}_2^i &= F_2^i(\mathbf{y}_i, \mathbf{d}_i(t), \Lambda_i, t) \\ &\vdots \\ \dot{y}_s^i &= F_s^i(\mathbf{y}_i, \mathbf{d}_i(t), \Lambda_i, t) \end{aligned} \tag{47}$$

where, $\mathbf{y}_i = (y_1^i, y_2^i, \cdots, y_{N_s}^i)$ with $N_s = |A_i|$ (where, $|\cdot|$ denotes the cardinality of the set) and $\mathbf{d}_i(t)$ is the decoupling vector (18). Note that the $i-$th subsystem is directly affected by the parameters Λ_i and indirectly by other parameters through the decoupling vector. We shall denote by $\Sigma^i \subseteq \{\xi_1, \xi_2, \cdots, \xi_p\}$ the set of all parameters which directly or indirectly affect the $i-$th subsystem. Clearly, $\Sigma^i \supset \Lambda_i$ and

$$\Sigma_c^i = \Sigma^i \setminus \Lambda_i \tag{48}$$

will be the set of parameters which indirectly affect the $i-$th subsystem. Also we shall denote by

$$\Lambda_c^i = \bigcup_{j \in \mathcal{N}_i} \Lambda_j. \tag{49}$$

Analogous to the P-variate orthogonal polynomial space introduced in section 6.1, we consider a P-variate space formed over the random parameter set $\Xi \subseteq \{\xi_1, \xi_2, \cdots, \xi_p\}$

$$W_{\Xi}^P \equiv \bigotimes_{|\mathbf{d}| \leq P} W^{k,d_k}, \tag{50}$$

such that $|\mathbf{d}| = d_1 + d_2 + \cdots d_N \leq P$, where $N = |\Xi|$. Following this notation for the $i-$th subsystem, we shall denote the polynomials belonging to $W_{\Sigma_i}^P$, by $\Psi_j^i, j = 1, \cdots M_i$, where $M_i = \frac{(P+N_i)!}{P!N_i!}$ and $N_i = |\Sigma_i|$. In this polynomial space we seek polynomial chaos expansions of the form,

$$y_k^{iP}(t, \Sigma^i) = \sum_{j=1}^{M_i} a_{jk}^i(t) \Psi_j^i(\Sigma^i), \tag{51}$$

where, the superscript i in the above variables, denote that the all variables correspond to the $i-$th system. We shall denote by $\mathbf{y}_i^P(t, \Sigma^i) = (y_1^{iP}, y_2^{iP}, \cdots, y_{N_s}^{iP})$ the vector of $P-$variate expansion for $i-$th subsystem. With such an expansion, we can rewrite each equation in the subsystem (47) as

$$\dot{y}_k^i \quad = \quad F_k^i(\mathbf{y}_i, \mathbf{y}_{ic}(t, \Sigma_c^i), \Lambda_i, t), \qquad k = 1, \cdots, N_s. \tag{52}$$

where, we have expressed the decoupling vector as $\mathbf{y}_{ic}(t, \Sigma_c^i) = (\mathbf{y}_{j_1}^P(t, \Sigma^{j_1}), \cdots, \mathbf{y}_{j_{N_n}}^P(t, \Sigma^{j_{N_n}}))$, with $j_k \in \mathcal{N}_i$ and $N_n = |\mathcal{N}_i|$.

The full collocation grid $C(\mathbf{l}, n_i)$ for the $i-$th subsystem will be represented as

$$\mathcal{C}(\mathbf{l}, n_i + n_i^c) = \mathcal{C}(\mathbf{o}, n_i) \times \mathcal{C}(\mathbf{m}, n_i^c), \tag{53}$$

where, $\mathbf{l} = (\mathbf{o}, \mathbf{m})$,

$$\mathcal{C}(\mathbf{o}, n_i) = C_{o_1}^1 \times \cdots \times C_{o_{n_i}}^1, \tag{54}$$

is the collocation grid corresponding to parameters Λ_i, with $n_i = |\Lambda_i|$, $\mathbf{o} = (o_1, \cdots, o_{n_i})$, and

$$\mathcal{C}(\mathbf{m}, n_i^c) = C_{m_1}^1 \times \cdots \times C_{m_{n_i^c}}^1, \tag{55}$$

is the collocation grid corresponding to parameters Σ_i^c, with $n_i^c = |\Sigma_c^i|$ and $\mathbf{m} = (m_1, \cdots, m_{n_i^c})$. Since, the behavior of $i-$th subsystem weakly depends on the parameters Σ_i^c, we can take,

$$\max_{i=1,\cdots,n_i^c} m_i \ll \max_{k=1,\cdots,n_i} o_k. \tag{56}$$

With this framework, we are ready to outline the second UQ approach:

- Step I: Apply graph decomposition (see section 4 for details) to identify weakly interacting subsystems in the system (1).
- Step II: Apply probabilistic waveform relaxation, which involves following sub steps.

- Step 0 (Assignment-partitioning process): Partition (1) into m subsystems (obtained in Step I) leading to system of equations given by (14). Obtain, Λ_i, Λ_i^c Σ^i and Σ_c^i for each subsystem, $i = 1, \cdots, m$.
- Step 1: (Initialization of the relaxation process with no coupling effect incorporated): Set $I = 1$ and for each $i = 1, \cdots, m$, guess an initial waveform $\{\mathbf{y}_i^0(t) : t \in [0 \quad T]\}$ such that $\mathbf{y}_i^0(0) = \mathbf{y}_{0i}$, so that, the decoupling vector becomes

$$\mathbf{y}_{ic}^1(t) = (\mathbf{y}_{j_1}(0), \cdots, \mathbf{y}_{j_{N_n}}(0)), \qquad j_k \in \mathcal{N}_i, \quad N_n = |\mathcal{N}_i|, \tag{57}$$

and solve for $\{\mathbf{y}_i^1(t, \Lambda^i) : t \in [0 \quad T]\}$ from

$$\dot{\mathbf{y}}_i^1 = \mathbf{F}^i(\mathbf{y}_i^1, \mathbf{y}_{ic}^1(t), \Lambda_i, t), \tag{58}$$

with an initial condition $\mathbf{y}_i^1(0) = \mathbf{y}_i^0(0)$ on a collocation grid $C(\mathbf{o}, n_i)$. Compute the gPC expansion over $P-$variate polynomial space $W_{\Lambda_i}^P$, leading to

$$y_k^{iP1}(t, \Lambda^i) = \sum_{j=1}^{M_i} a_{jk}^i(t) \Psi_j^i(\Lambda^i), \tag{59}$$

for $k = 1, \cdots, M_i$. From now on we shall denote the solution vector of the $i-$the subsystem at $I-$th iteration by $\mathbf{y}_i^{PI} = (y_1^{iPI}, \cdots, y_{N_n}^{iPI})$.

- Step 2: (Initialization of the relaxation process, incorporating first level of coupling effect): Set $I = 2$ and for each $i = 1, \cdots, m$, set

$$\mathbf{y}_{ic}^2(t, \Lambda_c^i) = (\mathbf{y}_{j_1}^{P1}(t, \Lambda^{j_1}), \cdots, \mathbf{y}_{j_{N_n}}^{P1}(t, \Lambda^{j_{N_n}})), \qquad j_k \in \mathcal{N}_i, \quad N_n = |\mathcal{N}_i|, \tag{60}$$

for the GJ relaxation, and solve for $\{\mathbf{y}_i^2(t, \Sigma^i) : t \in [0 \quad T]\}$ from

$$\dot{\mathbf{y}}_i^2 = \mathbf{F}^i(\mathbf{y}_i^2, \mathbf{y}_{ic}^2(t, \Lambda_c^i), \Lambda_i, t), \tag{61}$$

with an initial condition $\mathbf{y}_i^2(0) = \mathbf{y}_i^0(0)$, over a collocation grid $C(\mathbf{1}, n_i + n_i^c)$. From this obtain the $P-$variate expansions over the polynomial space $W_{\Sigma_i}^P$, so that

$$y_k^{iP2}(t, \Sigma^i) = \sum_{j=1}^{M_i} a_{jk}^i(t) \Psi_j^i(\Sigma^i), \quad k = 1, \cdots, M_i. \tag{62}$$

- Step 3 (Analyzing the decomposed system at the I-th WR iteration): For each $i = 1, \cdots, m$, set

$$\mathbf{y}_{ic}^I(t, \Sigma_c^i) = (\mathbf{y}_{j_1}^{P(I-1)}(t, \Sigma^{j_1}), \cdots, \mathbf{y}_{j_{N_n}}^{P(I-1)}(t, \Sigma^{j_{N_n}})), \qquad j_k \in \mathcal{N}_i, \quad N_n = |\mathcal{N}_i|, \tag{63}$$

for the GJ relaxation, and solve for $\{\mathbf{y}^k(t, \Sigma^i) : t \in [0 \quad T]\}$ from

$$\dot{\mathbf{y}}_i^I = \mathbf{F}^i(\mathbf{y}_i^I, \mathbf{y}_{ic}^I(t, \Sigma_c^i), \Lambda_i, t), \tag{64}$$

with initial condition $\mathbf{y}_i^I(0) = \mathbf{y}_i^0(0)$ over a collocation grid $C(\mathbf{1}, n_i + n_i^c)$. Obtain the expansions,

$$y_k^{iPI}(t, \Lambda^i) = \sum_{j=1}^{M_i} a_{jk}^i(t) \Psi_j^i(\Lambda^i), \quad k = 1, \cdots, M_i. \tag{65}$$

- Step 4 (Iteration) Set $I = I + 1$ and go to step 3.

Note that in above approach, PCM is applied at subsystem level with the collocation grid $C(\mathbf{l}, n_i + n_i^c)$ (where, $(\mathbf{l} = (\mathbf{o}, \mathbf{m}))$with $\mathbf{o} = (o_1, \cdots, o_{n_i})$ and $\mathbf{m} = (m_1, \cdots, m_{n_i^c}))$ being sparse for the parameters which affect that subsystem indirectly (see condition 56). The table below, shows how much computational savings (see last column) can be obtained by using above framework, instead of full collocation over the entire parameter space. In the table, m be the number of subsystem, p_i will be number of parameters occurring in the $i-$th subsystem and l be the order of accuracy of collocation along each dimension for the collocation grid over the entire parameter space with $p = \sum_{i=1}^{m} p_i$ parameters. Lets assume for simplicity of analysis, that $o_i = l_s, i = 1, \cdots, n_i$ and $m_i = l_c, i = 1, \cdots, n_i^c$ and $l_c < l_s$. $I_{\max}$ denote the maximum number of waveform relaxation iterations.

Subsystems & parameters	Collocation parameters	Full Collocation ($\mathcal{R}_F$)	Iterative ($\mathcal{R}_I$)	$\frac{\mathcal{R}_F}{\mathcal{R}_I}$
$m = 2, p_i = 5, i = 1, 2$	$l = 5,\ l_s = 5,\ l_m = 2$	9,765,625	2,006,251	5
$m = 3, p_i = 5, i = 1, , 2, 3$	$l = 5,\ l_s = 5,\ l_m = 2$	3.0518e+10	96,009,376	300

In the table above entries in third column are computed using formula $\mathcal{R}_F = l^p$ which denotes the number of deterministic runs of the complete system (1). Similarly, the fourth column is obtained using

$$\mathcal{R}_I = 1 + \sum_{i=1}^{m} l_s^{p_i} + I_{\max}(\sum_{i=1}^{m} l_s^{p_i} \bigotimes_{j \neq i} l_c^{p_j}), \tag{66}$$

assuming $I_{\max} = 10$, and measures the total number of deterministic runs of the subsystems involved. Clearly, advantage of the iterative approach becomes evident as the number of subsystems and parameters in the system increase. Also note that this approach is parallelizable, and hence highly scalable.

7.3 Some Remarks on two algorithms:

1 There are number of parameters that need to properly identified to

2 Analytical conditions under which these algorithms will converge, is not known and need to be established. These conditions would provide an We perform some experiments to numerically study the convergence behavior of probabilistic WR.

3 Note that in both algorithm graph decomposition is applied to identify weakly interacting system, before waveform relaxation is initiated. It is assumed that this decomposition remains valid as the system evolves during relaxation process. However, the

4 The full grid collocation used in above algorithm can be replaced by sparse grid collocation in a straightforward manner. With this, additional computational gain can be attained as sparse grid methods are computationally efficient compared to the full grid (see section ?? and Appendix A for further details).

8 Example Problems

In this section we illustrate the iterative procedure developed in previous section on a simplified power system network and numerically analyze its convergence properties.

8.1 Stability Problem

In order to illustrate the iterative algorithm proposed in section 7.2 and study its convergence behavior, we first consider a simple system, with two states (x_1, x_2),

$$\dot{x}_1 = ax_1^2 + cx_2^2 - v_1, \tag{67}$$
$$\dot{x}_2 = cx_1^2 + bx_2^2 - v_2, \tag{68}$$

where, $a, b, \mathrm{c}, v_1, v_2$ are the parameters. Here the parameter c determines the coupling strength between two subsystems described by the two equations. It would be assumed that c, v_1, v_2 are deterministic parameters, while a, b are uncertain with Gaussian distribution. The objective here is to determine the uncertainty in the stability of system, which can be quantified by looking at the distribution of λ_{max}, the maximum eigenvalue of the Jacobian,

$$J(a, b, \mathrm{c}) = \begin{pmatrix} 2ax_{10} & 2cx_{20} \\ 2cx_{10} & 2bx_{20} \end{pmatrix}, \tag{69}$$

where, x_{10}, x_{20} is the equilibrium satisfying

$$ax_{10}^2 + cx_{20}^2 - v_1 = 0, \tag{70}$$
$$cx_{10}^2 + bx_{20}^2 - v_2 = 0. \tag{71}$$

Figures below show result of probabilistic waveform relaxation for different values of coupling parameter c. For all cases, the ground truth is computed based on collocation on the parameter space (a, b) with $l = 10$, while $l_s = 5$, $l_m = 3$ and $P = 5$. In all cases considered, the iterative approach converges to the ground truth, as shown by the histogram of λ_{max} (see figures ??), and its mean and variance (see figures ??). As the coupling strength increases, the number of iterations required for the convergence increases.

9 Conclusion and Future Work

In this paper we have proposed uncertainty quantification approaches which exploit the underlying dynamics and structure of the system. In specific we considered a class of networked system, whose subsystems are weakly coupled to each other. We showed how these weak interactions can be exploited to overcome the dimensionality curse associated with traditional UQ methods, and radically accelerate uncertainty propagation in large systems. By integrating graph decomposition and waveform relaxation with generalized polynomial chaos and probabilistic collocation framework, we proposed two scalable iterative UQ approaches: *equation based* which requires access to the equations describing the underlying dynamics of the system, while the other one, which we called *probabilistic waveform relaxation*, treats the system more or less as a black box. The second approach is more practical as for most complex networked systems, it may be non trivial to obtain system equations, if not impossible. We illustrated the probabilistic waveform relaxation approach on a simple system with promising results.

Many questions further need to be investigated. First of all, analytical conditions under which the two iterative schemes proposed in this paper converge, need to be established. The choice of collocation parameter l_m (see section 7.2) plays a critical role in how much computational gain can be obtained in probabilistic waveform relaxation; a systematic procedure for selecting this parameter is therefore crucial. Finally, this algorithm need to be tested on a larger system, to establish its true potential of being scalable.

A Sparse Grid Methods

A.1 Clenshaw Curtis Quadrature

ClenshawCurtis (CC) quadrature employs a change of variables $\xi = \cos\theta$ and uses a discrete cosine transform (DCT) approximation for the cosine series, in order to compute the integral in Eq. 33. More precisely, the cosine series expansion

$$g(\cos\theta) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\theta), \tag{72}$$

leads to

$$\begin{aligned} \int_{-1}^{1} g(\xi)w(\xi)d\xi &= \int_0^{\pi} g(\cos(\theta))w(\cos(\theta))\sin(\theta)d\theta \\ &= \frac{a_0}{2}W_0 + \sum_{k=1}^{\infty} a_k W_k \\ &\approx \frac{a_0}{2}W_0 + \sum_{k=1}^{p} a_k W_k \end{aligned} \tag{73}$$

where, for $k = 0, \cdots,$

$$a_k = \frac{2}{\pi}\int_0^{\pi} g(\cos\theta)\cos(k\theta)d\theta, \tag{74}$$

and

$$W_k = \int_0^{\pi} w(\cos\theta)\cos(k\theta)\sin\theta d\theta. \tag{75}$$

Unlike computation of arbitrary integrals, Fourier-series integrations for periodic functions (like $f(\cos\theta)$)) in Eq (74), up to the Nyquist frequency $k = p$, are accurately computed by the p equally-weighted points

$$C_q^1 = \{r_k : r_k = \cos(\frac{\pi(k-1)}{q}), k = 1, \cdots, q+1\}, \tag{76}$$

except the endpoints, which are weighted by 1/2, to avoid double-counting. With this the integral (74) can be approximated as

$$a_k \approx \frac{2}{q}\left[\frac{g(1)}{2} + \frac{g(-1)}{2}(-1)^k + \sum_{i=2}^{q} g(x_i)\cos(\frac{\pi(i-1)k}{q})\right]. \tag{77}$$

For most $w(\xi)$, the integral (75) cannot be computed analytically. Since the same weight function is generally used for many integrands $g(\xi)$, however, one can afford to compute these W_k numerically to high accuracy beforehand, like in Gauss quadrature. Note the following features of CC quadrature:

1. Since, by definition the Chebyshev polynomials $T_k(\xi)$, satisfy $T_k(\cos\theta) = \cos(k\theta)$ CC quadrature can be thought of as employing the expansion of the integrand (see Eq. 72) in terms of Chebyshev polynomials to compute the integral.

2. In CC quadrature, the integrand is always evaluated at the same set of points, given by C_p (Eq. 76) regardless of probability density function. On other hand, in Gaussian quadrature, different density functions lead to different orthogonal polynomials, and thus different roots where the integrand is evaluated.

3. For $q = 2^i, i \geq 1$, $C_i^1 \subset C_{i+1}^1$ (this would be the notation we would use for CC 1D grid from now on, i.e. $C_i^1 \equiv C_{q(i)}^1$), the CC quadrature points become nested. Gaussian quadrature points lack this property.

4. The CC formula is less accurate, the $(q+1)$ point CC rule can provide an accurate result for integrating polynomial functions of order up to q, compared to $2q+1$ for Gauss. For practical purposes, however both method lead to comparable accuracy [3]. This is possible because most numeric integrands are not polynomials, and approximation of many functions in terms of Chebyshev polynomials converges rapidly.

In summary, besides having fast-converging accuracy comparable to Gaussian quadrature rules, CC quadrature naturally leads to nested quadrature rules, which is important for both adaptive quadrature and multidimensional quadrature (cubature). There are other quadrature rules with nested properties, details can be found in [5].

A.2 Smolyak Quadrature

Smolyak's algorithm is a method first developed to handle high dimensional quadrature [4] and later extended to accomplish high dimensional interpolation [6]. Its basic idea is to use the solution to several low-dimensional problems to span the space and then linearly combine these to yield the solution to higher dimensional problem [5]. Let

$$\Delta_k^1[g] = (\mathcal{U}_k^1 - \mathcal{U}_{k-1}^1)[g] \tag{78}$$

be the difference quadrature formula with, $\mathcal{U}_0^1 \equiv 0$. In general, the difference are therefore quadrature formulas on the union of grids $C_k^1 \cup C_{k-1}^1$, which is just C_k^1 in nested case. Based on these difference formulas, Smolyak construction approximates the integral in Eq. (4) by

$$S(l,p)[g] = \sum_{|\mathbf{i}|\le l+p-1} (\Delta_{i_1}^1 \otimes \Delta_{i_2}^1 \cdots \otimes \Delta_{i_p}^1)[g], \tag{79}$$

where, $|\mathbf{i}| = i_1 + i_2 \cdot + i_p$. This can be expressed in terms of $\mathcal{U}_1^{i_1}$, as

$$S(l,p)[g] = \sum_{l\le|\mathbf{i}|\le l+p-1} (-1)^{l+p-|\mathbf{i}|-1} \binom{p-1}{|\mathbf{i}|-l} (\mathcal{U}_{i_1}^1 \otimes \mathcal{U}_{i_2}^1 \cdots \otimes \mathcal{U}_{i_p}^1)[g]. \tag{80}$$

where,

$$\binom{p-1}{|\mathbf{i}|-l} = \frac{(p-1)!}{(|\mathbf{i}|-l)!(p-|\mathbf{i}|+l)!}, \tag{81}$$

is the factorial. From above equation we see that like full tensor-quadrature, Smolyak quadrature formulas are special tensor-product rule which are constructed from tensor products of one-dimensional quadrature formulas, but these are combined so that in only some dimensions quadrature formulas of high order are used while formulas of lower order are used in the other dimensions. One could also write above formula in recursive fashion, like

$$S(l,p)[g] = \sum_{k=1}^{l-1} (\Delta_k^1 \otimes S(l-k,p-1))[g], \tag{82}$$

and

$$S(l+1,p+1)[g] = \sum_{|\mathbf{i}|\le l+p-1} (\Delta_{i_1}^1 \otimes \Delta_{i_2}^1 \cdots \otimes \Delta_{i_p}^1 \otimes \mathcal{U}_{l+p-|\mathbf{i}|}^1)[g]. \tag{83}$$

Note that to compute (79), we only need to evaluate function g over so called *sparse grid*, which is given by union over the pariwise disjoint grids $S_{i_1} \times \cdots \times S_{i_p}$,

$$\mathcal{S}(l,p) = \bigcup_{|\mathbf{i}|\le l+p-1} \mathcal{S}_{i_1} \times \cdots \times \mathcal{S}_{i_p}, \tag{84}$$

where, $\times$ denotes the usual Cartesian product. For non nested case

$$\mathcal{S}_k = C_k^1, \tag{85}$$

while, for nested case

$$\mathcal{S}_k = C_k^1 \backslash C_{k-1}^1, \tag{86}$$

with $C_0^1 \equiv \varnothing$, where recall C_k^1 is a set of 1D quadrature points corresponding to level k. Thus, unlike full grid (Eq. 40), the sparse grid is a union of several tensor products. Based on Eq. (80), an appropriate choice of vector $\mathbf{i} = (i_1, i_2, \cdots, i_p)$ gives level of accuracy in each dimension, from which the sparse grids are obtained by usual product i.e. $\mathcal{S}_{i_1} \times \mathcal{S}_{i_2} \cdots \mathcal{S}_{i_p}$. In case the univariate formulas are nested, the sparse grids are also nested, i.e.

$$\mathcal{S}(l,p) \subset \mathcal{S}(l+1,p). \tag{87}$$

The number of points Q in sparse grids $\mathcal{S}(l,p)$ are given by

$$Q = \sum_{|\mathbf{i}| \leq l+p-1} p_{i_1} \cdots p_{i_p}. \tag{88}$$

where,

$$p_k = \mathrm{Card}(S_k), \tag{89}$$

is the cardinality i.e. number of points in the set S_k. If $m_l = \mathcal{O}(2^l)$ the order of Q is

$$Q = \mathcal{O}(2^l l^{p-1}) \tag{90}$$

This shows that the dependence on dimension is much weaker on n the number of uncertain parameters, compared to $\mathcal{O}(2^{lp})$ for full grid. Table 1 gives a comparison of the number of grid points in different schemes.

1 For $n \geq 5$ the sparse grid methods prove significantly more advantageous than full grid.

2 Nested quadrature rules lead to sparser grids compared to non-nested quadrature rule.

3 The asymptotic accuracy of sparse grid method is comparable to that of full grid.

More precise results on accuracy and convergence properties of Smolyak grids with different quadrature rules can be found in [5]; for multielement formulation of sparse grid see [2].

In summary, the Smolyak formula (Eq. 80) can be expressed as

$$S(l,p)[g] = \sum_{|\mathbf{i}| \leq l+p-1} \sum_{j_1=1}^{p_{i_1}} \cdots \sum_{j_n=1}^{p_{i_p}} W_{\mathbf{ij}} g(\mathbf{r_{ij}}), \tag{91}$$

with, $\mathbf{i} = (i_1, i_2, \cdots, i_p)$,$\mathbf{j} = (j_1, j_2, \cdots, j_p)$, and $\mathbf{r_{ij}} = (r_{i_1 j_1}, \cdots, r_{i_n j_n})$. Again note that $\mathbf{i} = (i_1, i_2, \cdots, i_p)$ is vector specifying levels of quadrature formula in each dimension. For each i_k, p_{i_k} (see Eq. 89) denotes the actual number of points in the 1D grid (S_k, see Eq. 85 or 86) in k-th dimension; hence j_k goes till p_{i_k} to cover all points in the grid in each dimension.

For non nested case, weights are combined as follows

$$W_{\mathbf{ij}} = W_{i_1 j_1} \cdots W_{i_p j_p}, \tag{92}$$

while in nested case,

$$W_{\mathbf{ij}} = \sum_{|\mathbf{i}+\mathbf{q}| \leq l+2d-1} v_{(i_1+q_1)j_1} \cdots v_{(i_n+q_p)j_n}, \tag{93}$$

Random Dimension (n)	Level (l)	FG	SG CC	SG Gauss
3	2	8	7	10
	3	27	25	52
	4	64	69	195
	5	125	177	609
	6	215	441	1710
	7	343	1,073	4502
5	2	32	11	16
	3	243	61	131
	4	1,024	241	746
	5	3,125	801	3376
	6	7,776	2,433	13,083
	7	16,807	6,993	45,458
	8	32,768	19,313	145,873
	9	59,049	51,713	440,953
	10	100,000	135,073	1,272,848
10	2	1,024	21	31
	3	59,049	221	486
	4	1,048,576	1,581	5166
	5	9,765,625	8,801	42,101
	6	60,466,176	41,265	281,867
20	2	1,048,576	41	61
	3	3,486,784,401	841	1871
	4	1,099,511,627,776	11,561	38,531
	5	95,367,431,640,625	120,401	600,226

Table 1: Comparison of full gird and sparse grid PCM using different quadrature rules. In order to generate above table, $m_l^1 = l + 1$ for Gauss, while $m_l^1 = 2^{l-1} + 1, l > 1$ (and $m_1^1 = 1, l = 1$) for nested CC univariate quadrature formula (see Eq. 36). This table has been partially taken from [9], with last column generated using ME-gPC code (see next section).

with $\mathbf{q} \in \mathbb{N}^n$ and

$$v_{(k+q)j} = \begin{cases} W_{kj} & \text{if} \quad q = 1 \\ W_{(k+q-1)r} - W_{(k+q-2)s} & \text{if} \quad q > 1, \end{cases} \tag{94}$$

where, s, r are such that $r_{kj} = r_{(k+q-1)r} = r_{(k+q-2)s}$.

The weights can be precomputed in both cases, so that there is no practical difference concerning the overall cost of the quadrature formula. Note that Smolyak formulas can contain negative weights even if the underlying univariate quadrature formula have positive weights. Convergence is guaranteed, because values of the weights remain relatively small. For more details on stable numerical implementation of Smolyak quadrature, the reader is referred to [5].

References

[1] Xiu, D., Efficient Collocational Approach for Parametric Uncertanity Analysis, Communications in computational physics (2007), Vol 2(2), 293.

[2] Foo, J. Wan, X. and Karniadakis, G, The Multi-element Probablistic Collocation Method (ME-PCM): Error Analysis and Applications, Journal Of Computational Physics, accepted 2008.

[3] Lloyd N. Trefethen, Is Gauss quadrature better than Clenshaw-Curtis?, SIAM Review (2008), Vol 50 (1), 67.

[4] Smolyak S., Quadrature and interpolation formulas for tensor products of certain classes of functions, Soviet Mathematics, Doklady (1963), Vol 4, 240.

[5] Gerstner, T. and Griebel, M., Numerical integration using sparse grids, Numerical Algorithms (1998), 18(3-4), 200.

[6] Novak E. Barthelmann V. and Ritter K., High dimensional polynomial interpolation on sparse grids, Advances in Computational Mathematics (2000), Vol 12, 273.

[7] Wan, X. and Karniadakis, G., Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM J. Sci. Compt.(2006), Vol 28, 901.

[8] Gautschi, W., Algorithm 726: ORTHPOL - A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules, ACM Trans. Math. Software (1994), Vol 20 , 21.

[9] Prempraneerach, P., Uncertainty Analysis in a Shipboard Integrated Power System using Multi-Element Polynomial Chaos (2007, MIT Thesis.

[10] Mezic, I., Coupled Nonlinear Dynamical Systems: Asymptotic Behavior and Uncertainty Propagation Igor Mezic 43rd IEEE Conference on Decision and Control December 14-17, 2004, Atlantis, Paradise Island, Bahamas.

[11] Mezic, I., Personal communications.

[12] Xiu, D. and Karniadakis, G., The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput., 24 (2002), 619-644

[13] Fishman, G., Monte Carlo: Concepts, Algorithms, and Applications, Springer-Verlag, New York, 1996.

[14] Xiu, D. and Hesthaven, J., High-order collocation methods for differential equations with random inputs, SIAMJ. Sci. Comput., 27 (2005), 1118-1139.

[15] M. Callier, W. S. Chan, and C. A. Desoer, Input-output stability theory of interconnected systems using decomposition techniques, IEEE Transactions on Circuits and Systems, vol. 23, no. 12, pp. 714 729, 1976.

[16] T. T. Georgiou and M. C. Smith, Linear systems and robustness - a graph point of view, Lecture Notes in Control and Information Sciences, vol. 183, pp. 114121, 1992.

[17] Lelarasmee, E., The Waveform Relaxation Method for time domain analysis of large scale integrated circuits: Theory and Applications, Memorandum No. UCB/ERL M82/40, 19 May 1982.

[18] White, J., Odeh, F., Vincentelli, A. S., and Ruehli, A., Waveform Relaxation: Theory and Practice, Memorandum No. UCB/ERL M85/65, April 1985.

[19] Luxburg, U. V., A Tutorial on Spectral Clustering, Technical Report No. TR-149, Max Planck Institute for Biological Cybernetics, Aug 2006,

[20] Chung, F. (1997). Spectral graph theory. Washington: Conference Board of the Mathematical Sciences.

[21] Cameron, R.H., and Martin, W.T., The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals, Annals of Mathematics, 48, pp. 385-392, 1947.

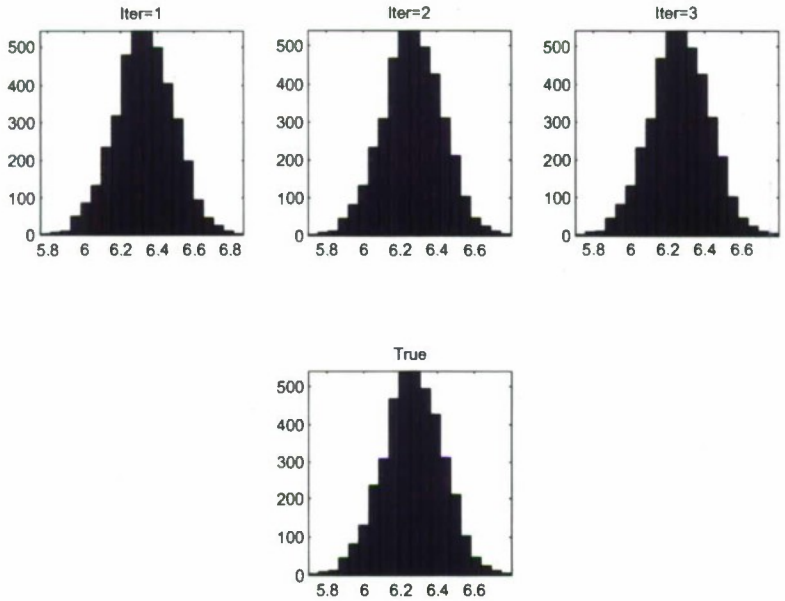


Figure 1: Convergence of distribution, $c = 0.1$

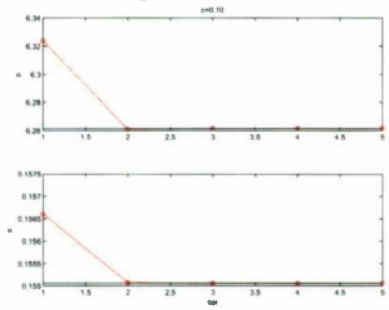
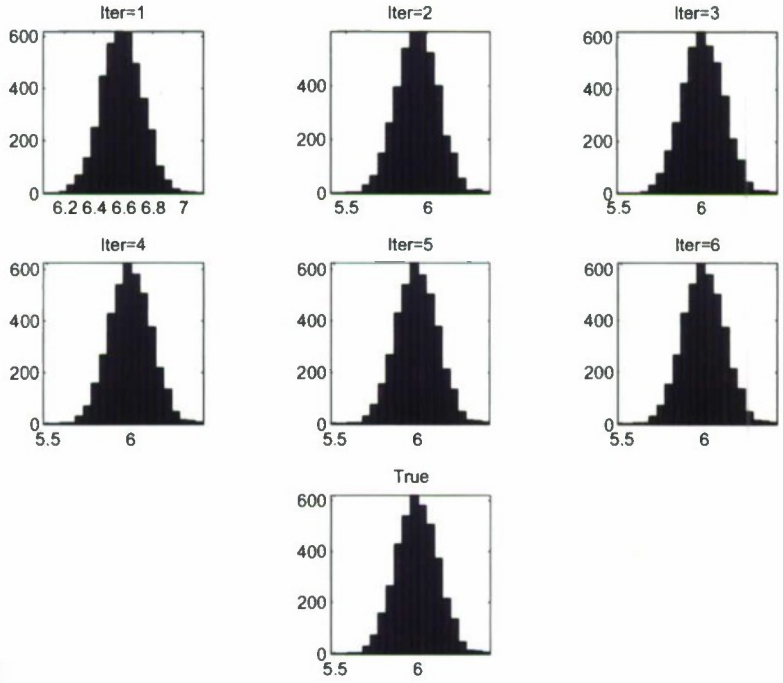


Figure 2: Convergence of mean and variance



[hbt]

Figure 3: Convergence of distribution, $c = 1.0$

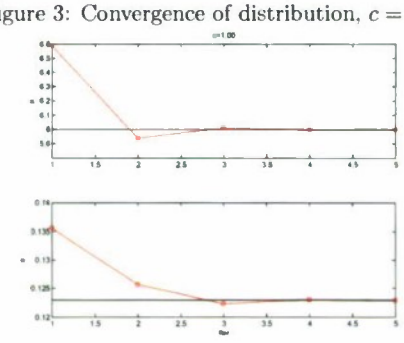


Figure 4: Convergence of mean and variance

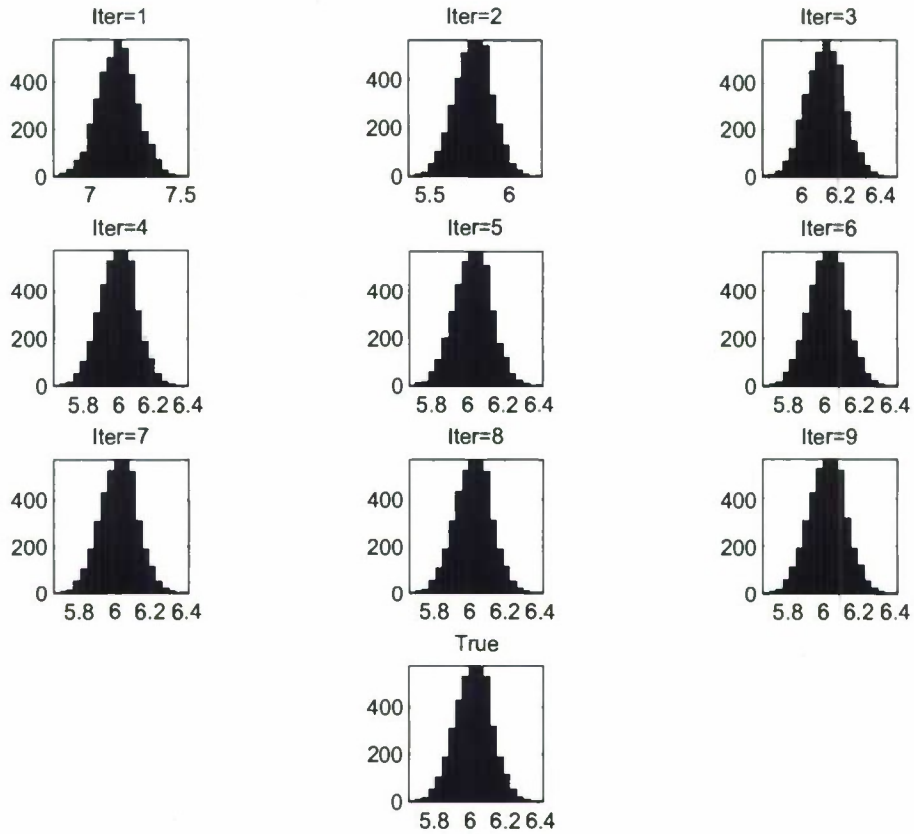


Figure 5: Convergence of distribution, $c = 2.0$

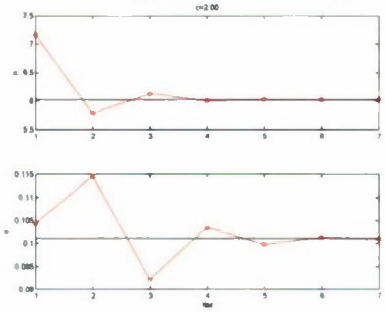


Figure 6: Convergence of mean and variance

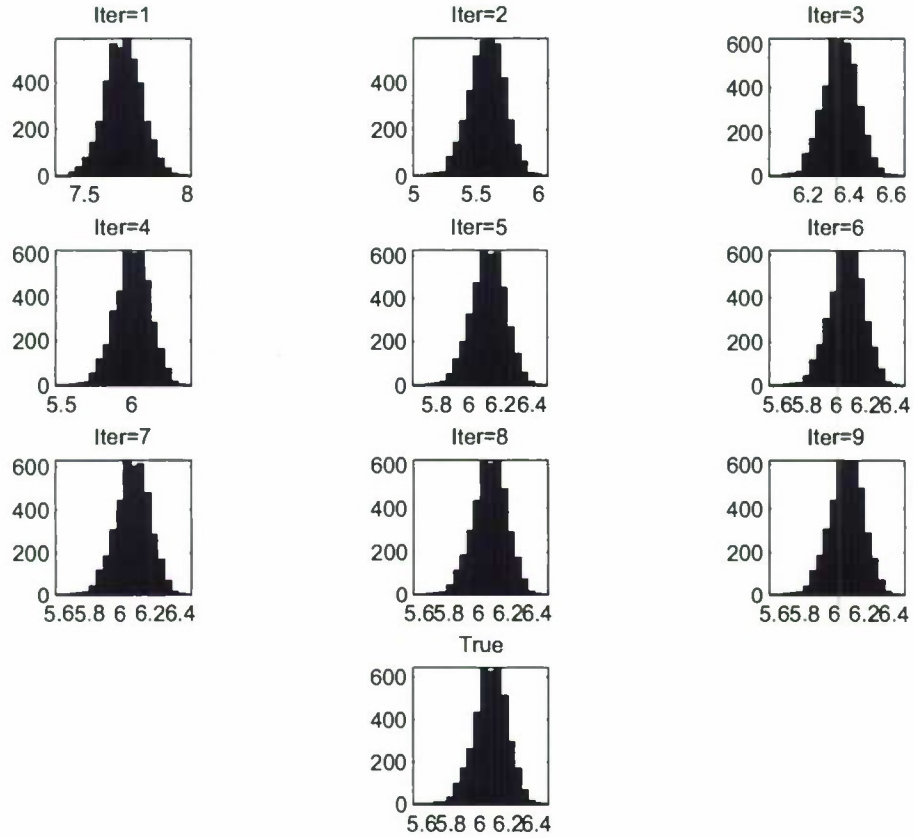


Figure 7: Convergence of distribution, $c = 2.8$

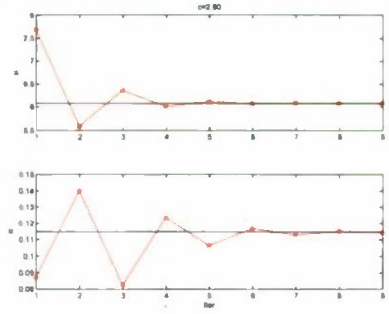


Figure 8: Convergence of mean and variance

B.2 Uncertainty propagation by various methods

Uncertainty Propagation by Response Surfaces, Domain Exploration by Hierarchical Stochastic Covers, Multigrid Approach for Sparse Data in High Dimensions, Uncertainty Propagation for the New England Power Grid

Ronald Coifman, Igor Mezic, Vladimir Fonoberov,
Andrzej Banaszuk, Jose Miguel Pasini, Marco Arienti, Tuhin Sahai, Amit Surana,
George Karniadakis, Sean Meyn, Mihai Putinar, Sorin Costiner
(Yale, UCSB, MIT/Brown, UIUC, UTRC)

May 28 2009

Contents

Summary

The main results presented in this work are summarized next.

PDF Fitting for Uncertainty Propagation, Long Tail input PDFs, High dimensions, Unknown Unknowns. The approximation of PDFs of outputs of systems that depend on stochastic inputs can be used to estimate statistical measures of the outputs. These PDFs can be used to propagate further the uncertainty in complex systems. Computations of intermediate PDFs and merging of PDFs can provide large computational savings in uncertainty propagation (UP) computations for complex systems. Formulas that evaluate these computational savings are provided. Methods of fitting PDFs are illustrated on several UP problems as: a) Computation of phase change temperatures depending on 4 input parameters with long tail PDFs; b) In high dimension examples, PDFs with up to 2000 input parameters are effectively estimated; c) The effectiveness of estimating PDFs is illustrated for high dimension cases that also include unknown unknowns(inputs) with unknown PDFs.

Optimized Interpolation for parameters and number of eigenvectors using probability weighted objectives is an interpolation procedure that combines a) a Nystrom kernel based interpolation approach with b) an optimization of the parameters of the kernel such that to minimize the difference between statistical measures of the interpolated function and of the data. This method was used in the response surface (RS) UP approach.

Domain Exploration approaches, uniform domain covers, hierarchical stochastic covers for Optimization and UP. Finding positions of points that provide quasi uniform domain covers are important for designs of experiments, optimization, UP, interpolation, domain discretization, solving PDEs etc. Domain exploration approaches are more and more often used in industrial applications where an understanding of the space of possible designs is thought for, e.g., for regions of robust solutions, or for optimal solutions with respect to multiple objectives. Domain exploration approaches usually offer more information about the design space than optimization approaches. Finding good sets of sampling points is important for the efficiency of domain exploration approaches and for UP. It is desired to obtain the maximum amount of information about the design space using a smallest number of samples. Simple and robust hierarchical stochastic cover techniques are presented. A repelling particle method is illustrated where n points in a domain that act as repelling particles pushed by potentials that aim to enforce given properties such as uniform spacing. The particles may be denser in regions where given PDFs are higher (the PDFs are treated as separate potentials). Treatment of boundaries and constraints in searches and optimization is performed by three techniques: direct enforcing of the

constraint, using of boundary particles and using of boundary potentials that repel the particles towards the interior. We discuss a Hierarchical Architectural Optimization approach using these ideas. Results for Domain Exploration, Global Optimization and finding Multiple Local Minima by Adaptive Hierarchical Repelling Particle Techniques are demonstrated. In addition, it is demonstrated that the optimization will track the found local minima in a case when the objective varies continuously in time (or depending on parameters), i.e., the problem has a dynamic objective.

A Generic Unstructured Multigrid (MG) Approach in High Dimensions is proposed that combines the presented domain covering approaches, with the interpolation approaches, and with known multigrid formulations. These generalize known MG techniques to sparse data in high dimensions and combine the efficiency of hierarchical structures with local iterations and with approximation of solutions by reduced models.

A Hierarchical Richardson Approach for Model Extrapolation is suggested. Coefficients of a sequence of models are identified by system identification and extrapolated as in a Richardson procedure. Models may be decomposed into deterministic sub-models (e.g., trends) and stochastic sub-models (e.g., noise). Model extrapolation can be used for both deterministic and stochastic models, hence for UP. Large computational savings may be obtained by extrapolating a sequence of coarse model results to approximate fine model results.

A **Response Surface (RS) Uncertainty Quantification Approach** is compared with MC, a Polynomial Chaos Collocation approach, and with DSAMPLE (an effective quasi MC technique) on the **Kr Milestone Problem.** Advantages and disadvantages of the RS approach are discussed.

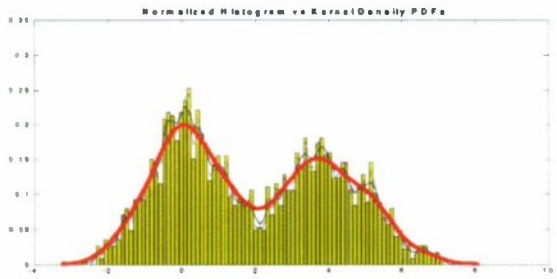
A Sum of Gaussians State Uncertainty Propagation method was demonstrated for a **New England Power Grid** model and the computational time savings using the proposed merging of PDFs approach is discussed.

1. PDF Fitting for Uncertainty Propagation, Long Tail input PDFs, High dimensions, Unknown Unknowns

The approximation of PDFs of outputs of systems that depend on stochastic inputs can be used to estimate statistical measures of the outputs. These PDFs can be used to propagate further the uncertainty in complex systems. It is shown that appropriate computations of intermediate PDFs can significantly reduce the complexity of uncertainty propagation in (UP) computations. Methods of fitting PDFs are illustrated on several UQ problems as:
a) Computation of phase change temperatures depending on 4 input parameters with long

tail PDFs; b) In high dimension examples, PDFs with up to 2000 input parameters are effectively estimated; c) The effectiveness of estimating PDFs is illustrated for high dimension cases that also include unknown unknowns(inputs) with unknown PDFs.

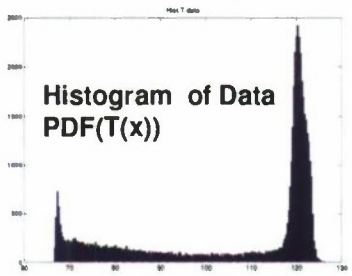
The main idea is to approximate the PDF(f) by fitting a histogram of f(x) as shown in the figure below.



Multiple techniques have been applied such as sums of Gaussians and kernel density functions. For example, the PDF(f) may be approximated a by a sum of Gaussians or of other basis functions (such as rectangles), or a spline fitting a histogram etc.

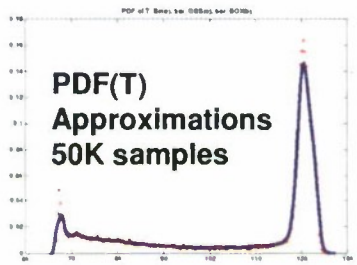
If the PDF(f) is smooth, then a small number of points may be sufficient to approximate it well. The approximation of the PDF(f) is an 1D problem, although f(x) may depend on the variables x in high dimensions. Moreover, the fitting does not use the PDF(x). Even more, the PDF(f) may depend on many unknown unknowns with unknown PDFs.

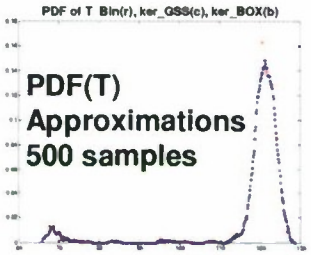
PDF(T) Approximations by Kernel Density, Long Tail 4D
The parameters and their PDFs are not used
Works in high dimensions for any number of parameters and with unknown distributions; Works for unknown unknowns
Model: T(x) = Gauss1 * Reg1 + Gauss2 * Reg2 fitting MD data, x in 4D LT



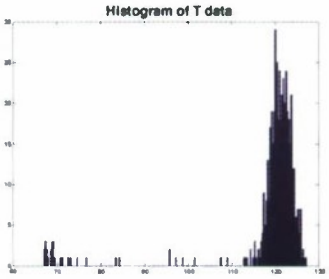
Convergence of Mean and Var

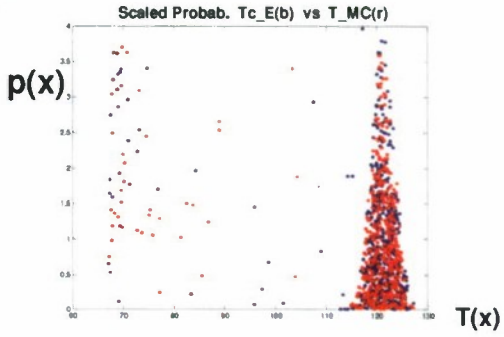
50K samples:
mean_T_reg MC = 106.65
var_T_reg MC = 400.02
mean_T_bin Hist = 106.65
var_T_bin Hist = 400.02
mean_TE_ker GSS = 106.65
var_TE_ker GSS = 400.31

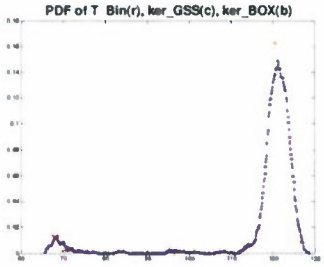




PDF(T) for 500 Samples





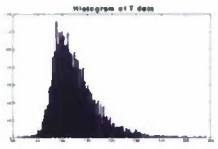


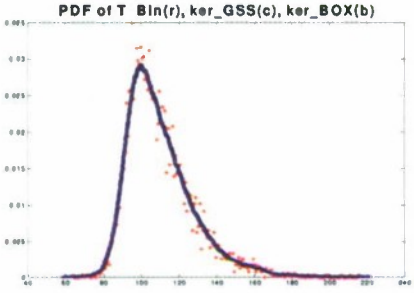
Probability of p(x) for different x plotted for T(x)
!!! T(x)=T(y) may accumulate 2 different probabliities :

$$p(x) \neq p(y) \qquad for \qquad x \neq y$$

PDF(T) ~ Sum_x g(T(x), T) / N
Sum of Boxes, Gaussians, Kernels

DARPA Kr Milestone 4 Gaussian PDFs (5000 samples)
The PDF statistical measures reflect the Measures of the Samples
The quality of the sampling is reflected in the quality of the PDF and they should have similar convergence rates,
e.g., DS + PDF should converge faster than MC + PDF
!! The time savings come when Uncertainty is Propagated using PDFs





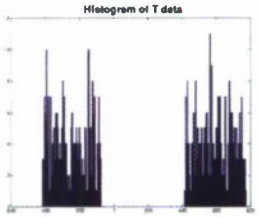
mean MC = 110.13
var MC = 309.04

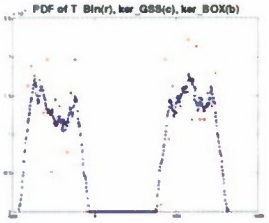
mean PDF = 110.13
var PDF = 317.45

Analytic results Vladimir:
mean T An: 110.30
var T An: 315.61

The Kernel PDF (blue) is smoother than the Histogram (red)

High Dimensions, Unknown Unknowns
2000 parameters (Normal PDFs)
"unknown" parameters random (uniform) drive the mode switch
500 samples, 10^(-3) relative error MC vs PDF
5000 samples, 10^(-4) relative error MC vs PDF





500 samples
mean MC = 186.53
mean GSS PDF = 186.54

var MC = 1.9002e+005
var GSS PDF = 1.9021e+005

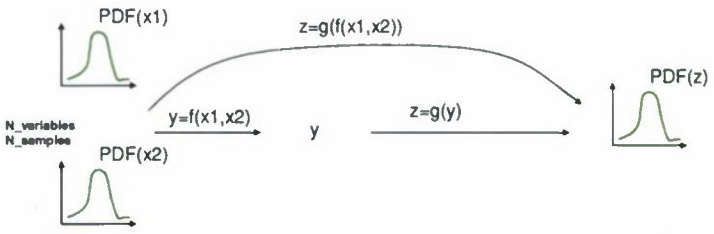
5000 samples
mean_MC = 183.35711
mean_PDF = 183.35707

var_MC = 185579.5
var_PDF = 185542.8

Propagation of PDFs in complex Systems
Complexity reduction by merging PDFs

The propagation of PDFs in complex systems by computations of intermediate PDFs and merging of PDFs can significantly reduce the complexity of uncertainty propagation in (UP) computations. The following schemes provide the computational time saving using PDF merging.

High Complexity of approximating PDFs by MC/Quasi – MC, ...



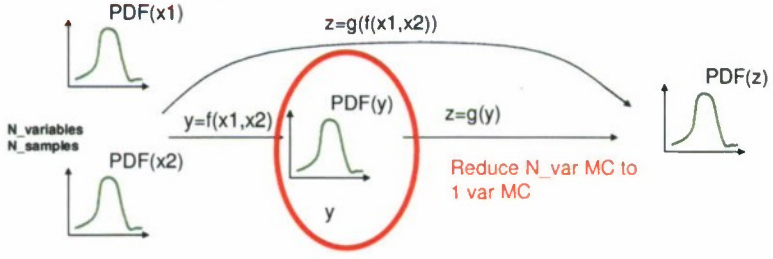
T1 = Time to estimate PDF(z) directly from x1, x2 by MC / QMC, ...

T1 = (N_variables * N_samples) * (Tf + Tg) ; Tf, Tg = time to evaluate f, g

23

Time Saving by Merging PDFs

Besides computing y also compute PDF(y):
Merge N_variable PDFs into 1 PDF



T1 = Time to estimate PDF(z) directly from x1, x2 by MC / QMC, ...

T1 = (N_variables * N_samples) * (Tf + Tg) ; Tf, Tg = time to evaluate f, g

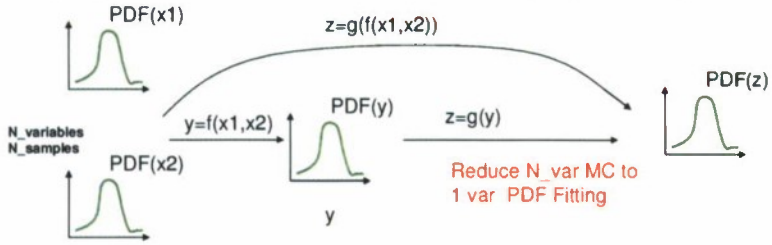
T2 = Time to estimate PDF(z) from x1, x2 computing intermediate PDF(y):

T2 = (N_variables * N_samples) * Tf + N_samples * Tg,

Time Savings in approximating PDF(z) = T1 – T2 = (N_variables -1) * N_samples * Tg 24

Time Saving by Merging PDFs + Reducing MC to PDF Fitting

Besides computing y also compute PDF(y):
Merge N_variable PDFs into 1 PDF + Replace MC for y by PDF Fitting



T1 = Time to estimate PDF(z) directly from x1, x2 by MC / QMC, ...

T1 = (N_variables * N_samples) * (Tf + Tg) ; Tf, Tg = time to evaluate f, g

T2 = Time to estimate PDF(z) from x1, x2 computing intermediate PDF(y):

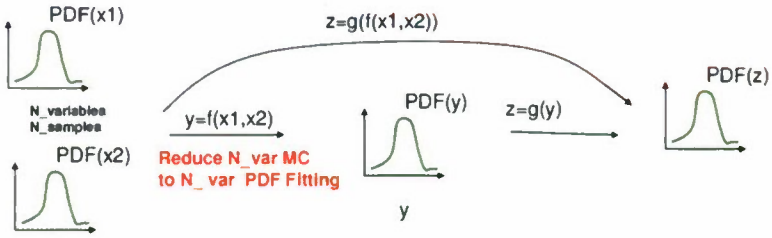
T2 = (N_variables * N_samples) * Tf + N_PDFfit * Tg.

Time Savings in approximating PDF(z) = T1 – T2 = Tg * (N_variables * N_samples - N_PDFfit)

25

Time Saving by Merging PDFs + Reducing MC to PDF Fitting

Besides computing y also compute PDF(y):
Merge N_variable PDFs into 1 PDF + Replace MC for y by PDF Fitting



T1 = Time to estimate PDF(z) directly from x1, x2 by MC / QMC, ...

T1 = (N_variables * N_samples) * (Tf + Tg) ; Tf, Tg = time to evaluate f, g

T2 = Time to estimate PDF(z) from x1, x2 computing intermediate PDF(y):

T2 = (N_variable_PDFfit) * Tf + N_PDFfit * Tg.

Time Savings in approximating PDF(z) = T1 – T2 =
Tf* (N_variables * N_samples - N_variable_PDFfit) + Tg * (N_variables * N_samples - N_PDFfit)

26

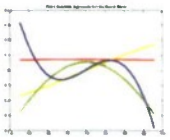
2. Optimal Interpolation: Interpolation optimized for parameters and number of eigenvectors using probability-weighted objectives.

The presented interpolation procedure that combines a) a Nystrom kernel based interpolation approach with b) an optimization of the parameters of the kernel such that to minimize the difference between statistical measures of the interpolated function and of the data. This method was used in the response surface (RS) UP approach.

Response Surfaces: Nystrom Interpolation with PDF weighted Optimization

Fit functions using Eigenvectors of a Kernel as basis functions

Given the values $f(x_i)$ of f at the points $\{x_1,\ldots,x_n\}$, approximate $f(x)$ and derivatives of f at x.



1) $Kv = \lambda v$, K is a kernel, e.g., $K(x,y) = \exp(-\|x-y\|^2/\sigma^2)$

2) $v(x_i) = \frac{1}{\lambda}\sum_l K(x_i,x_l)v(x_l)$ eigenvectors of K

3) $v(x) = \frac{1}{\lambda}\sum_l K(x,x_l)v(x_l)$ extend the eigenvectors using K

4) $f(x_i) \approx \sum_j a_j v_j(x_i)$ fit f by a PDF weighted Optimization (see next) using a subset of eigenvectors

5) $f(x) \approx \sum_j a_j v_j(x)$ extend f using extension of eigenvectors 3)

Optimization Added to Nystrom

6) $f(x) = \sum_{j=1}^{d} \frac{a_j}{\lambda_j}\left(\sum_{l=1}^{N} K(x,x_l)v_j(x_l)\right)$

7) $\partial_x f(x) = \sum_{j=1}^{d} \frac{a_j}{\lambda_j}\left(\sum_{l=1}^{N} \partial_x K(x,x_l)v_j(x_l)\right)$ get derivatives of f from derivatives of K

Optimization of the RS to fit the data, the mean and var of the data
The samples x are fixed, Find the RS, R(x) that best fits the data f(x) by varying the number of eigenvectors (M), kernel parameters (sigma), and coefficients (a) of the eigenvector expansion

Nystrom: Given the N values $f(x_i)$ of f at the N points $\{x_1,\ldots,x_N\}$, approximate $f(x)$ at x.

1) $Kv=\lambda v$, K is a kernel, e.g., $K(x,y)=\exp(-\|x-y\|^2/\sigma^2)$ $N\times N$

2) $f(x_i)\approx R(x_i)=\sum_j^M a_j v_j(x_i)$ approximate f using a subset of M eigenvectors and an Optimization as:

3) $O_1=\min_{M,\sigma}(\min_a \|f-R\|/\|f\|)$ for M, σ, a, gives good mean and var approximations

4) $O_2=\min_{M,\sigma}\{\min_a|\sum|f(x_i)-R(x_i)|\,p(x_i)/(\sum p(x_i)+\mathrm{w}\,|\,\mathrm{var(f)-var(R)}\,|\,\}\}$

minimizez mean and variance errors using known PDFs

5) $O_3=\min_{M,\sigma}\{\min_a[norm,\ mean,\ \mathrm{var},\ PDF(RS)]\}$

insures R converges to f in norm(and...PDF), where $PDF(RS)$ is an approximation of the histogram(f)

6) Bootstarp approaches: compare N different RS's, each for another subset of N-1 samples

Several potential advantages of RS for uncertainty quantification

1) The RS approach may work in large dimensions (10^2 - 10^4 ?) for RS a small number of "good" eigenvectors may be sufficient. The eigenvectors may represent well the reduced dimensionality of the data.

2) The RS approach may be easily applied on general domains and on manifolds

3) The RS approach may use a relatively small number of evaluation points (e.g., quasi uniform in high dimensions)

4) The eigenvectors used in RS may depend on the problem – an advantage in approximating features of the surface

5) The eigenvectors may have a local character (e.g., sums of Gaussians) hence may represent local features of surfaces

6) Error and convergence estimates based on eigenvectors and eigenvalues may be provided

7) The RS approach is general: any probability distributions of parameters may be used for the RS

8) The RS approach may also be used when unknown unknowns (with unknown distributions) are present (as shown in our approach, where about 10^4 random initial positions and initial velocities are present), (see the extrapolation approach that estimates models of noise).

9) Extension/Extrapolation using eigenfunctions may have improved reliability for bounded functions.

10) RS provide multiple alternatives/freedom in: selecting the kernels, evaluation points, handling convergence, handling high dimensions, handling general domains etc.

Disadvantages of Eigenvector based RS Approaches

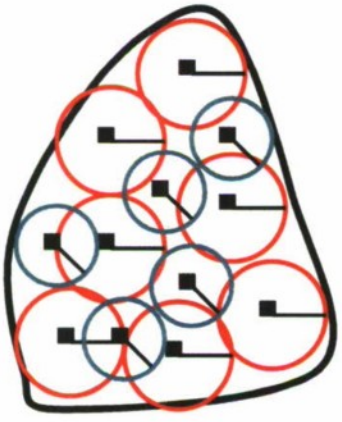
An expensive diagonalization of a large matrix is required. The matrix may be sparse but it is not banded usually. Eigenvalue computing time and accuracy are major concern issues. Numerical instabilities may happen due to small eigenvalues, especially in schemes that involve division by the small eigenvalues (that often have large relative errors). This difficulty may be handled by the hierarchical RS approaches. The selection of the kernels and of parameters of kernels are not obvious. It is not easy to define local parameters, although the structure of the problem may require this, e.g., a Gaussian sigma should depend on the density of the data.

3. Domain Exploration, Hierarchical Uniform Domain Covers for UQ. Hierarchical Architectural Optimization.

Finding positions of points that provide quasi uniform domain covers are important for designs of experiments, optimization, UP, interpolation, domain discretization, solving PDEs etc. Domain exploration approaches are more and more often used in industrial applications where an understanding of the space of possible designs is thought for, e.g., for regions of robust solutions, or for optimal solutions with respect to multiple objectives. Domain exploration approaches usually offer more information about the

design space than optimization approaches. Finding good sets of sampling points is important for the efficiency of domain exploration approaches and for UP. It is desired to obtain the maximum amount of information about the design space using a smallest number of samples. Simple and robust hierarchical stochastic cover techniques are presented. A repelling particle method is illustrated where n points in a domain that act as repelling particles pushed by potentials that aim to enforce given properties such as uniform spacing. Treatment of boundaries and constraints in searches and optimization is performed by three techniques: direct enforcing of the constraint, using of boundary particles and using of boundary potentials that repel the particles towards the interior. The approach is used for Hierarchical Architectural Optimization.

In the hierarchical stochastic covering approach, a domain is covered stochastically by non-intersecting balls of same radius. A sequence of covers of decreasing ball radii is used as shown below. Stochastic stopping criteria are proposed.
Similar, using a particle repelling approach to cover a domain by repelling particles, for example repelled by Gaussian potentials of given width, a hierarchical cover is obtained by selecting Gaussians of different widths.



Applications:

1. UQ for selecting sampling points
2. RS, Interpolation, Fitting
3. Design of Experiments (DOE), Domain Exploration
4. Global Optimization using hierarchical approaches

Advantages:

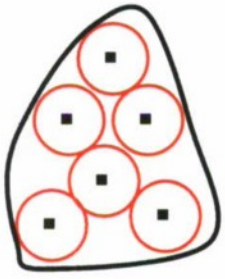
1. Works in high dimensions
2. Uniform full domain cover.
3. Avoids meshing - griding
4. Works on unstructured domains, manifolds
5. Multiscale covering
6. Handling of Boundaries, Constraints

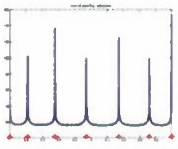
Domain Exploration-DOE using Repelling Potentials and Particle Dynamics

Particles driven by a repelling potential cover uniformly an unstructured domain

Ronald Coifman (Yale), Mihai Putinar (UCSB)

X(t+1) = X(t) – dt * grad(P(X(t))
direct: min(P(X(t))





Riesz Potential based coverage

E.B.Saff, D.P. Hardin; Discretizing Manifolds via Minimum Energy Points; Notices of the American Mathematics Society, November 2004, pp. 1186-1194

Mihai Putinar; A renormalized Riesz potential and applications; Advances in Constructive Approximation, 2004

Design of Experiments (DOE) Problem:
Position K points uniformly inside a general domain.

Approach: set K initial particles randomly inside the domain and drive them by a dynamics based on a repelling potential. The steady state provides a "uniform" covering solution. Handle boundaries and constraints by potentials.

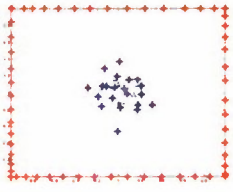
Advantages compared to other DOE techniques:

1. **Any number K of points may be distributed, even for very small K >= 2, in large dimensions.**
2. **The distribution of points is uniform**
3. **The domain may be unstructured (e.g., not a box) or a manifold**
4. **Constraints may be handled by potentials.**
5. **More efficient than Latin Hypercube on general domains and manifolds**
6. **Additional structure may be imposed using interior potentials representing probabilities or regions of interest.**

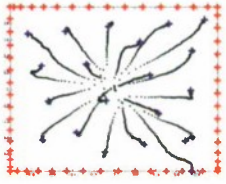
5

Quasi Uniform Distribution of Points by Repelling Potentials
The Boundary Points are kept fixed

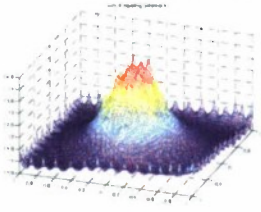
Iniatial Particle Positions



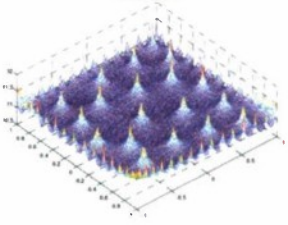
Final Particle Positions



Iniatial Repelling Potential



Final Repelling Potential



Exploration Approaches; **Hierarchical Architectural Optimization**

Hierarchical Architectural Optimization finds best Configurations satisfying a set of Objectives and Constraints

Goal: Build a Dictionary of Good Architectures/Designs/Configurations, e.g., for different costs, performances, weights, design options.

How:
Build a Hierarchy of Clusters: Poor Clusters are eliminated starting with Coarse Large Clusters

Good Clusters are refined Hierarchically

Elimination Criteria are added as new Constraints during the Optimization by Analysis of Solutions

Continuous Optimization is performed only close to the set of best solutions

Main techniques used:

1. **Hierarchical Optimization**
2. **Elimination Criteria added adaptively**
3. **Global Searches for Multiple Objectives**
4. **Local Continuous Optimization**

Highlights of Results
•Orders of magnitude savings in experimental work and time
•Reliable Global optima for exhaustive hierarchical searches (10000 times reduction in number of configurations)
•Very high confidence that best solutions have been found
•Can search all possible discrete combinations in finite unstructured domains, (with no deterministic combinatorial algorithm)

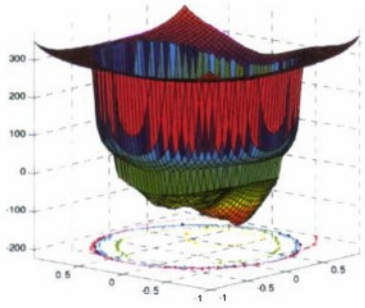
Domain Exploration, Global Optimization and finding Multiple Local Minima by Adaptive Hierarchical Repelling Particle Techniques.

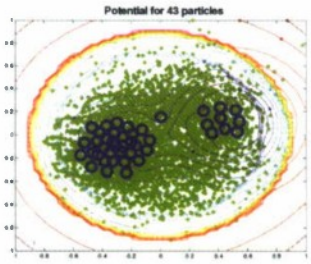
The figure below illustrates a domain exploration and global optimization approach. The objective is changing in time (it rotates in this example) and presents two large local minima to be found and tracked in time. A hierarchy of optimizations is performed by a hierarchy of potentials. The potentials generate broad large step searches at start, in a global domain exploration phase, and then small step local searches close to local minima. Boundary constraints are implemented by boundary potentials, in this case, the boundary potential keeps the particles inside the circular domain.

Domain Exploration, Hierarchical Global Optimization and finding Multiple Local Minima for Time Dependent Objectives

Two large local minima vary in time.
The Hierarchy of Optimizations is performed by a Hierarchy of Potentials that generate broad large step searches at start (Domain Exploration phase) and then small step local searches close to local minima.
The objective is time dependent (it rotates in this example)
Boundary Constraints are implemented by Boundary Potentials.

Potential for 43 particles





26

4. An Unstructured Multigrid Approach in High Dimension

A Generic Unstructured Multigrid (MG) Approach in High Dimensions is proposed that combines the presented domain covering approaches, with the interpolation approaches, and with known multigrid formulations. These generalize known MG techniques to sparse data in high dimensions and combine the efficiency of hierarchical structures with local iterations and with approximation of solutions by reduced models.

Multilevel approaches try to accelerate the solution of a problem by other problems on different levels, for example by coarser representations of the initial problem.

The multilevel approaches comprise two main procedures:

1) solvers of each problem on its level;

2) inter-level transfers of data, variables, and operators.

Problem formulation :

$L(x,u) = b$, x: points in R^n, u unknown variables at x

b known data at x, L: a general operator

Inter – level Transfers

Let X^I and X^J be two sets of points

The transfer $I^{I,J}$ of u^I, b^I from X^I to X^J is performed by Interpolation

$$u^J = I^{I,J} u^I, \quad b^J = I^{I,J} b^I$$

Coarse Level Operator

Given the operator L^I define the operator L^J by an approximation of $L^J = I^{I,J} L^I I^{J,I}$

Coarse Level Problem (Full Approximation Scheme (FAS))

Given the level I approximate solution u^I of $L^I(X^I, u^I) = b^I$

Define the level J term b^J and level J problem formulation by

$$L^J(X^J, u^J) = b^J = L^J(X^J, I^{I,J} u^I) + I^{I,J}(b^I - L^I(X^I, u^I))$$

Update of Fine Level Solution

The (FAS) correction of level I solution u^I by level J solution u^J is:

$$u^I = u^I_{old} + I^{J,I}(u^J - I^{I,J} u^I_{old})$$

Single Level Solvers, Single Level Relaxation

Iteratively update subsets of variables e.g., by local optimization or Newton approaches as follows.

Problem formulation on a given level: $L(x,u) = b$,

x : points in R^n indexed by I, X^I,

u unknown variables at x, b known data at x, L : a general operator

Local update of solutions

Let X^J be a subset of the points X^I

The associated equations

(1) $L^J(X^J, u^J) = b^J$ are solved approximately for u^J by :

an optimization or solver procedure (e.g., Newton)

(2) The equation (1) is solved for different subsets of indices J of I

until a convergence criterion is satisfied (Block Newton Gauss - Seidel)

If $J = I$ then a direct solver may be used, e.g., on coarse levels.

This procedure is a block type Gauss - Seidel iterative relaxation procedure.

A Local Interpolation Approach using a Partition of Unit

Let X be a finite set of points and $f{:}X \to R$ with known values on X

Let X^l be subsets of X indexed by the indices I

Let g^l approximate or interpolate f(x) on X^l, e.g., be local interpolations/fits of f

Let p^l be functions equal to 0 outside a set containing X^l, $\sum_l p^l(X^l) = 1$

The interpolation/fitting has a local form :

$$f(x) \approx g(x) = \sum p^l g^l$$

g^l may be computed for example by a local regression or by local optimization

g^l may be based on given basis functions

Multigrid Schemes as particular cases of the suggested Multilevel Schemes

Classical Multigrid (MG) FAS Schemes in low dimensions (up to 3 usually) on regular meshes are obtained as particular cases of the suggested multilevel schemes.

For example, for uniform grids, an operator L can be a finite difference approximation of a differential operator; Relaxations may be Gauss-Seidel or block Gauss-Seidel or Newton-Gauss_Seidel relaxations, etc.
The interpolations are based on local grid interpolations as in MG.
Local refinement reduces to performing a local multilevel scheme.
FAS schemes reduce to MG FAS schemes.

Observation

Suggested Unstructured MG Approach (that might work in High Dimensions)

Use Sparse Sets of Points and Local Expansions for Interpolation and Derivatives

P1

P2

Derivatives and Interpolaton Use an Expansion(e.g.,by Eigenfuncions)

$v^1 \longleftrightarrow Continuous\ \ v \longleftrightarrow v^2$

$$u^1(x_j) = \sum_{i=1}^{k} a_i v_i^1(x_j)$$

$$\partial_x u^1(x_j) = \sum_{i=1}^{k} a_i \partial_x v_i^1(x_j)$$

The MG FAS (Full Approximation Scheme)

1) $P^1: \ F^1(u^1) = t^1$

2) $P^2: \ F^2(u^2) = t^2 = F^2(I_2^1(u^1)) + I_2^1(t^1 - F^1(u^1))$

3) $u_{new}^1 = u_{old}^1 + I_2^1(u^2 - I_2^1(u_{old}^1))$

Classic Low Dimension MG	Unstructured MG
Structured Meshes of Points	Unstructured sparse sets of Points
Interpolation/Derivatives Use Meshes	Interpolation/Derivatives Use (Local) Expansions
Defining Coarse Problems	Same
Fine level exact solutions transferred to coarse levels are solutions of the coarse level problems.	Same
Coarse level solutions do not change exact fine level solutions	Same
Fourier Components	Basis Functions/ Eigenfunctions

8

5. A Hierarchical Richardson Approach for Model Extrapolation

A Hierarchical Richardson Approach for Model Extrapolation is suggested. Coefficients of a sequence of models are identified by system identification and extrapolated as in a Richardson procedure. Models may be decomposed into deterministic sub-models (e.g., trends) and stochastic sub-models (e.g., noise). Model extrapolation can be used for both deterministic and stochastic models, hence for UP. Large computational savings may be obtained by extrapolating a sequence of coarse model results to approximate fine model results.

Multilevel Extrapolation Builds and Extrapolates a Sequence of Models

Richardson Type Extrapolation:
Use a Sequence of Reduced Order Models for Speedup

Algorithm:

1) Find coefficients of a sequence of Models by System Identification
2) Build models for the sequences of corresponding coefficients and do a Sys ID
3) Extrapolate the models of the coefficients

Build $F_n(x,C(n))$: System Identification Model, for n particles, $C(n)$ are coefficients

E.g., $F_n(x,C(n))=\sum_i C_i(n)g_i(x)$

Build models for $C(n)$ e.g., $C_i(n)=\sum_k D_{i,k}(\frac{1}{n})^k$

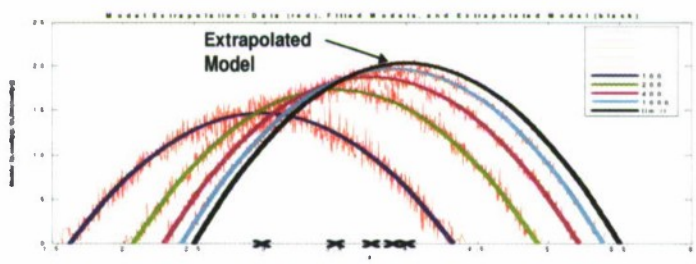
Build $C=\lim_{1/n\to 0} C(n)$, e.g., $C=D_{i,0}$

Build $F(x,C)$ the limit/extrapolated model for large n

E.g., $F(x,C)=\sum_i C_{i,0}(n)g_i(x)$

Uncertainty Management Apply the above procedure for Noise Models, e.g.,

$NoiseModel=Data-F$



Noise Models

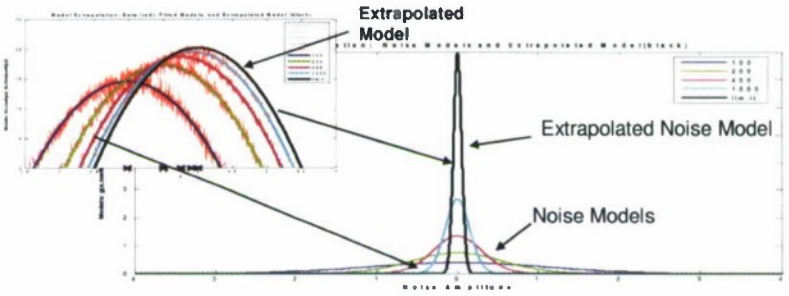
Build and Extrapolate a Sequence of Noise Models by Multilevel Extrapolation

Apply the Multilevel Model Extrapolation for Noise Models:

Find the models F by System Identification, Define Noise as:

Noise = *Data - F*

Build Noise Models by System ID using for the Noise data



37

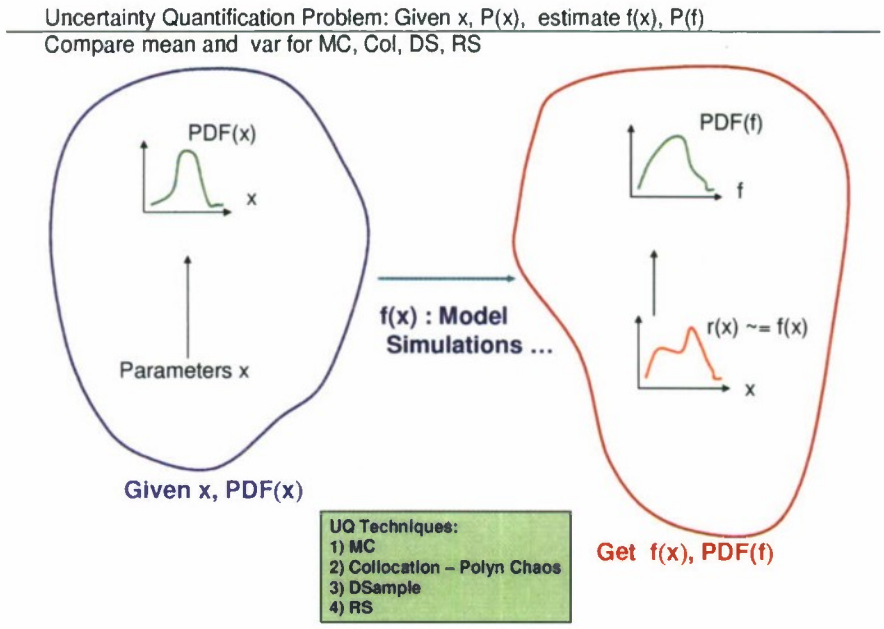
6. Comparison of Collocation, RS, DS, MC

Applications to the Kr Phase Diagram Milestone

A Response Surface (RS) Uncertainty Quantification Approach is compared with MC, a Polynomial Chaos Collocation approach, and with DSAMPLE (an effective quasi MC technique) on the Kr Milestone Problem. Advantages and disadvantages of the RS approach are discussed.

Response Surfaces for Uncertainty Quantification

The Response Surface (RS) approach of uncertainty quantification (UQ) builds response surfaces (models) r(x) as approximations of desired outputs (responses) f(x) as functions of parameters x with known PDFs, p(x). In addition one may be interested in approximating the PDF of f, see the diagram below.



Different uncertainty measures of f(x) are approximated by corresponding measures of r(x), for example the mean or variance of f are approximated by the mean and variance of r. The RS approach may be effective when the evaluation of f(x) is expensive and a when good approximations r ~= f of f can be obtained by a small number of evaluations of f(x). From the point of view of UQ one is interested to approximate for example mean(r) ~= mean(f) or var(r) ~= var(f). The main tasks in the RS approach are: 1) finding a relatively small number of sample points {x_i } and 2) finding a good RS/interpolation procedure that will use the samples {x_i } to build r such that r~= f and statistical measures of r will provide good estimates of the desired statistical measures of f. The following sections will discuss an RS approach, in our case a kernel based interpolation procedure that was used in the numerical experiments; domain exploration approaches for building sample points {x_i }; techniques for fitting the pdf of the output p(f); and a comparison of the RS approach with Monte Carlo (MC), a Polynomial Chaos - Collocation approach, and the DSAMPLE.

The 4 Uncertainty Quantification Approaches that have been compared, MC, DSAMPLE, RS, and Collocation, can be summarized as:

1) Monte Carlo: perform number N of MC evaluations of T(E) for E generated by p(E)

mean ~= sum(T(Ei)) / N

2) DSample (Quasi MC)

mean ~= sum(T(Ei)) / N

3) Response Surface/Surrogate Build a Surrogate of T(E), e.g., using a basis of functions, (eigenfunctions),

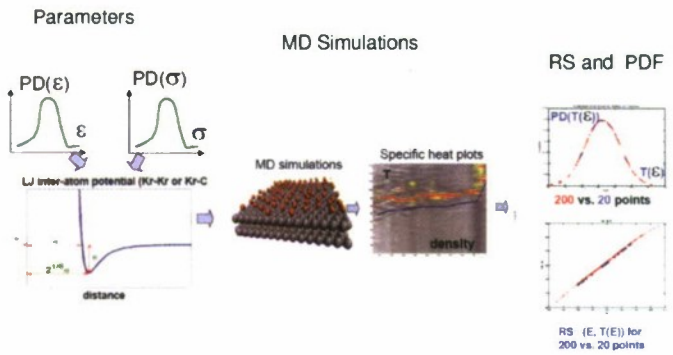
mean ~= sum(T(Ei) * p(Ei)) / sum(p(Ei)) (use RS with Importance sampling)

4) Collocation – Polynomial Chaos Evaluate T(E) at collocation points Ei, with given weights wi

mean ~= sum(T(Ei) * wi) / sum(wi)

The DSAMPLE approach is a new quasi MC approach developed by Igor Mezic (UCSB), presented in another chapter.

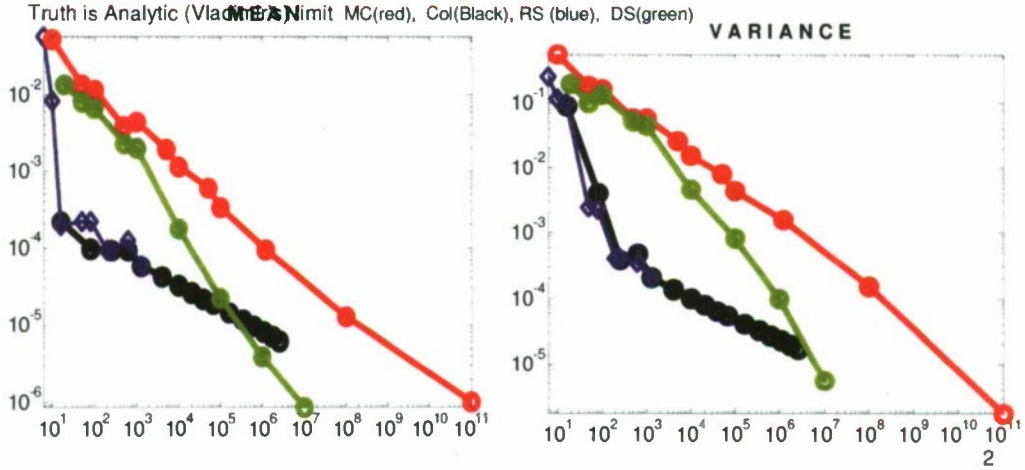
Uncertainty Quantification for the Krypton Milestone



RS with Collocation points reproduces PC efficiency (up to 1300 samples)
RS works with any number of samples, and may be effective in high dimensions too

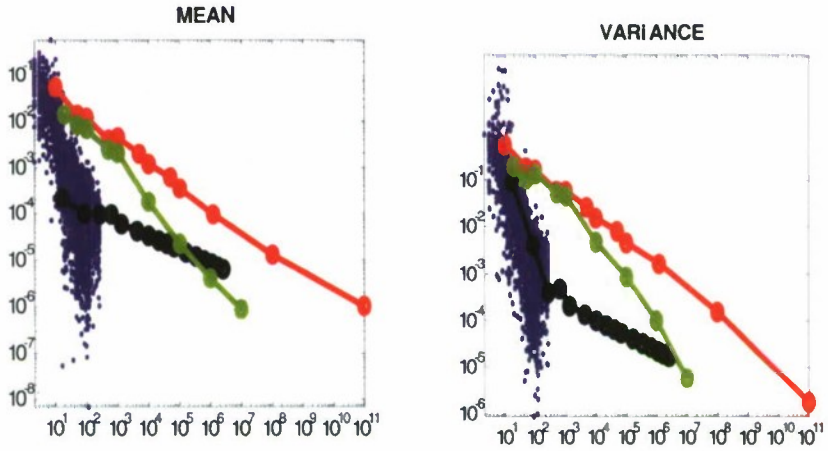
Nystrom with probabilities included in the objective;
RS Optimization (by # of eigenvectors and sigma, Optimization is not used in this example)
The RS results are close to the collocation results,
RS advantage: any number of sampling points can be used (subset of the collocation points)
RS disadvantage: numerical instability and difficulties related to eigenvalue computations (time and accuracy for diagonalization of large matrices).

Truth is Analytic (Vladimir) Limit MC(red), Col(Black), RS (blue), DS(green)



RS is effective for a small number of samples, selected by DS
The RS is Optimized (by N eigenvectors, sigma, a)
RS results show a good upper trend and many "very good solutions" for tens of samples

100 x 120 optimal solutions are generated (sets of samples from 2 to 120 are generated 100 times, for each fixed set of samples the RS is optimized for N of eigenvectors, sigma, a)
The mean and variance of the optimal RS are selected (and a blue point of the solution is shown,)



New England Power Grid – State Uncertainty Propagation

The propagation of state PDFs in Dynamical Systems by sums of Gaussians can be performed as in Extended Kalman Filter approaches (see for example: Uncertainty Propagation for Nonlinear Dynamical Systems using Gaussian Mixture Models Gabriel Terejanu , Puneet Singlat , Tarunraj Singht , Peter D. Scott1 AIAA Guidance, Navigation and Control Conference and Exhibit AIAA 2008-7472 ; 18 -21 August 2008, Honolulu, Hawaii) by the following algorithm:

- **The initial state x(0) of a dynamical system x(t+1)=f(x(t)) has given PDF(x(0))**
- **Propagate PDF(x(t)) as an approximation of a Sum of Gaussians:**

Gi(x(t)) = wi * fi * exp (- (x-mi)' Si^(-1) (x-mi) / 2) ; mi: mean, wi: amplitude;

Si: covariance matrix, fi = 1 / (det(Si)^(1/2) * (2pi)^n/2)

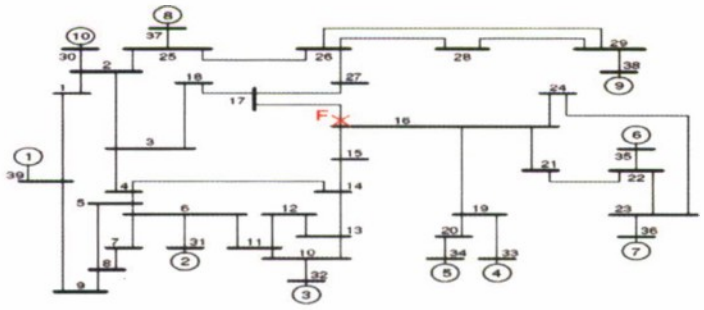
- **Propagate means mi by the dynamical system x_mean(t+1) = f(x_mean(t))**

- **Propagate covariance (width) using Jacobians (at means):**
Si(t+1) = (Df/Dx) Si(t) (Df/Dx)'

- **Keep the weights wi constant or approximate them by an optimization**

The approach was applied to the New England Power Grid model (as in the paper: Global Swing Instability of Multimachine Power Systems; Yoshihiko Susuki, Igor Mezi´c, Takashi Hikihara; Proceedings of the 47th IEEE Conference on Decision and Control Cancun, Mexico, Dec. 9-11, 2008) as shown in the figures below.

New England Power Grid



$$\left.\begin{aligned} \dot{\delta}_i &= \omega_i, \qquad i = 2, \ldots, 10 \\ \frac{H_i}{\pi f_s}\dot{\omega}_i &= -D_i\omega_i + P_{\mathrm{mi}} - G_{ii}E_i^2 - \sum_{j=1, j\neq i}^{10} E_i E_j \cdot \\ &\quad \cdot \{G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)\}. \end{aligned}\right\}$$

Inertia — Damping Coefficient — Mechanical Input Power — Internal Voltage — $G_{ij} + jB_{ij}$ Transfer impedance between generators

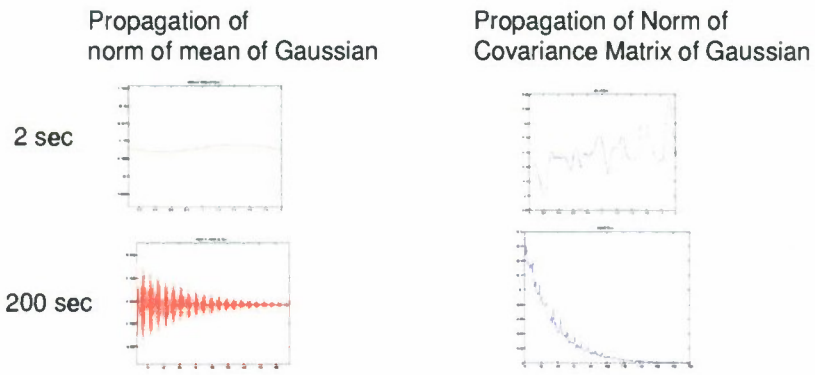
Global Swing Instability of Multimachine Power Systems
Yoshihiko Susuki, Igor Mezi´c, Takashi Hikihara
Proceedings of the 47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

3

New England Power Grid – State Uncertainty Propagation

Propagate the PDF of the State of the system, PDF(x) = Sum of Gaussians

x: 18 states; propagation of 1 Gaussian with initial mean and Covariance
Implicit ODE simulator ODE15s,



The System tends to behave more and more deterministic (due to stability)

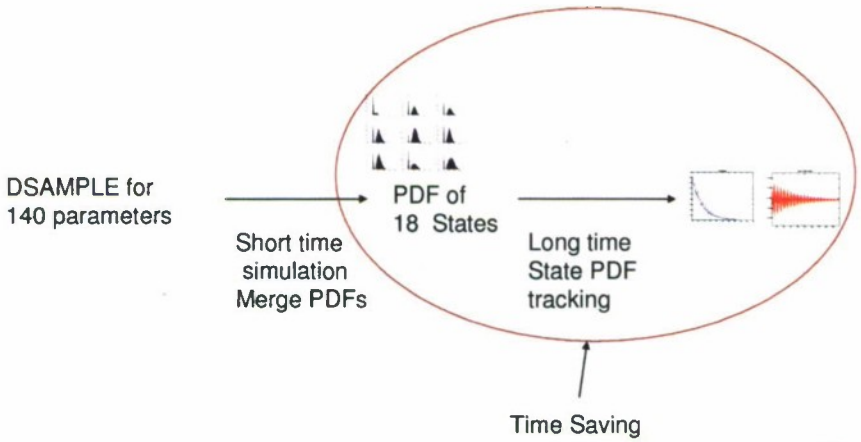
4

PDF merging for the New England Power Grid

A combination of the proposed PDF merging approach with the propagation of sums of Gaussians approach is suggested in the next scheme. The large number of input PDFs may come from the loads of the grid, for example PDFs of grid loads of different cities. In this case, large computational savings can be obtained by an early merging of the many input PDFs into state PDFs (there are only 18 states), that are further approximated and propagated by sums of Gaussians as illustrated above. The first half of the scheme, the simulation for 140 input PDFs and generation of the 18 state PDFs was performed using DSample. The second half, although not computed, can be performed with the sum of Gaussians code for the NE Grid as illustrated above. The computational time savings can be estimated by the formulas provided in section 1. For example, assuming that about 10^n samples are needed for the initial simulations using the input PDFs to be merged into state PDFs, and that about 10^3 simulations are needed for the propagation of state PDFs, and that the simulation time used for merging the PDFs is short compared to the total timulation time, then the time savings would be of order 10^(n-3).

PDF merging for the New England Power Grid

Combination of DS with State PDF Propagation



7

B.3 Approximate solutions for decentralized detection/estimation problems

Approximate solutions for decentralized detection/estimation problems

DyNARUM report

Jong-Han Kim

1 Introduction

We consider a class of decentralized detection/estimation problems, where multiple agents seek for the optimal decision rule or estimator for the global objective function, *i.e.*, the objective function depends on every agent's performance. Each agent uses its own measurements, and we especially consider cases where the communication between agents is impossible. In this specific setting, the optimization problem is formulated and solved *a priori* by the mission center, then the optimal rule for each agent is implemented on the on-board softwares before the search mission begins. As mentioned, no inter-agent communication during the mission is allowed, therefore the only information available to an agent is the measurements taken by itself.

2 Decentralized detection

We describe the 2-agent decentralized detection problem as follows.

$$\begin{aligned} \text{minimize} \quad & \sum_{y}^{L\times L} \sum_{u}^{M\times M} C_{yu} K_{yu} \\ \text{subject to} \quad & K = K^{(1)} \otimes K^{(2)} \\ & K^{(n)}\mathbf{1} = \mathbf{1} \\ & K^{(n)} \in \mathbb{B}^{L\times M} \qquad \text{for } n = 1, 2 \end{aligned}$$

where C describes some cost definition, and $K^{(n)}$ represents the nth agent's decision matrix. The operator $\otimes$ represents the *Kronecker* product of two matrices.

Note that solving this problem is hard because of the nonconvexity due to the bilinear function and the Boolean constraints. So we explore a relaxation technique that solves the relevant problem easily, providing an acceptable bounds for the optimal cost.

Defining the new variables $x = \begin{bmatrix} k_1^T & \cdots k_M^T & g_1^T & \cdots & g_M^T & 1 \end{bmatrix}^T$ and $Z = xx^T$, where $K^{(1)} = \begin{bmatrix} k_1 & \cdots & k_M \end{bmatrix}$ and $K^{(2)} = \begin{bmatrix} g_1 & \cdots & g_M \end{bmatrix}$, the problem is transformed to a semidefinite programming (SDP) with the rank -1 nonconvex constraint. Dropping the rank -1 constraint yields the following SDP relaxation.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{y}^{L\times L}\sum_{u}^{M\times M} C_{yu} \begin{bmatrix} \tilde{Z}_{11} & \tilde{Z}_{12} & \cdots & \tilde{Z}_{1M} \\ \tilde{Z}_{21} & \tilde{Z}_{22} & \cdots & \tilde{Z}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Z}_{L1} & \tilde{Z}_{L2} & \cdots & \tilde{Z}_{LM} \end{bmatrix}_{yu} \\
\text{subject to} \quad & AZb = \mathbf{1},\ AZ = \mathbf{1}b^T Z,\ b^T Zb = 1 \\
& \operatorname{diag}(Z) = Zb,\ 0 \preceq \operatorname{diag}(Z) \preceq 1 \\
& Z_{ij} \geq 0,\ Z_{ij} \leq Z_{ii},\ Z_{ij} \leq Z_{jj} \\
& Z_{ii} + Z_{jj} - Z_{ij} \leq 1 \qquad\qquad \text{for all } i,\ j,\ i \neq j \\
& Z \succeq 0 \\
\text{where} \quad & A = \begin{bmatrix} I_{L\times L} & \cdots & I_{L\times L} & \mathbf{0}_{L\times L} & \cdots & \mathbf{0}_{L\times L} & 0 \\ \mathbf{0}_{L\times L} & \cdots & \mathbf{0}_{L\times L} & I_{L\times L} & \cdots & I_{L\times L} & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} \mathbf{0}_{2LM\times 1} \\ 1 \end{bmatrix}, \\
& \tilde{Z}_{ij} = \begin{bmatrix} Z_{LM+1,(j-1)L+i} & Z_{L(M+1)+1,(j-1)L+i} & \cdots & Z_{L(2M-1)+1,(j-1)L+i} \\ Z_{LM+2,(j-1)L+i} & Z_{L(M+1)+2,(j-1)L+i} & \cdots & Z_{L(2M-1)+2,(j-1)L+i} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{LM+L,(j-1)L+i} & Z_{L(M+1)+L,(j-1)L+i} & \cdots & Z_{L(2M-1)+L,(j-1)L+i} \end{bmatrix} \\
& \text{for } i \in \{1,\ldots,L\},\ j \in \{1,\ldots,M\}
\end{aligned}
$$

Note that the SDP relaxation provides a lower bound.

$$J^*_{SDP} \leq J^*$$

If, fortunately, the relaxed problem finds the optimum, Z^*_{SDP}, such that $\operatorname{rank}(Z^*_{SDP}) = 1$, it suffices to prove that the above bound is tight and the relaxed optimum is equal to the true optimum. *i.e.*, $J^*_{SDP} = J^*$.

Though the condition or the proof for the zero optimality gap is not known yet, it turns out that the relaxed problem successfully finds the true optimum in the major number of random numerical simulations. However it sometimes fails to find the rank -1 solution, resulting in a solution infeasible to the original problem. The following heuristic approach will help to find a suboptimal rank -1 solution for such cases.

Consider the following SDP.

$$
\begin{aligned}
\text{minimize} \quad & \operatorname{trace}(Z^T W) \\
\text{subject to} \quad & 0 \preceq W \preceq I \\
& \operatorname{trace}(W) = 2LM
\end{aligned}
$$

where the optimal value is equal to the sum of all the eigenvalues except the largest one.

Note that W^* is obtained explicitly as $W^* = UU^T$, where $U = \begin{bmatrix} u_2 & u_3 & \cdots & u_{2LM} \end{bmatrix}$. u_k represents the kth singular vector of Z. Here, we are assuming that Z is known (from the previous iteration). Since Z orthogonal to such fixed W attains the minimum, augmenting this to the relaxation problem tends to *pull* the optimum to the set of rank -1 matrices.

Now augmenting the relaxed problem with this heuristic, we get the *augmented SDP*

relaxation.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{y}^{L\times L}\sum_{u}^{M\times M} C_{yu} \begin{bmatrix} \tilde{Z}_{11} & \tilde{Z}_{12} & \cdots & \tilde{Z}_{1M} \\ \tilde{Z}_{21} & \tilde{Z}_{22} & \cdots & \tilde{Z}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Z}_{L1} & \tilde{Z}_{L2} & \cdots & \tilde{Z}_{LM} \end{bmatrix}_{yu} + \text{trace}(Z^T W) \\ \text{subject to} & Z \in \mathcal{F} \\ & W \text{ is obtained from,} \\ \text{minimize} & \text{trace}(Z_{prev}^{*T} W) \\ \text{subject to} & 0 \preceq W \preceq I \\ & \text{trace}(W) = 2LM \end{array}$$

where $\mathcal{F}$ is the convex feasible set of the SDP relaxation problem.

Solving the augmented relaxation iteratively will give a rank-1 solution, hopefully finding the true optimum of the original problem, or at least providing a reasonable upper bound of the optimal cost.

$$J^*_{SDP} \leq J^* \leq J^*_{AUG}$$

A large number of numerical simulations are conducted in order to investigate the performance of the SDP relaxation and the augmented SDP relaxation. The random cost matrices with $L = 4$, and $M = 4$ are constructed by several different rules. Although the series of random simulations does not represent all the posibilities or can not prove anything, it provides an idea of how well the proposed schemes is going to perform in average sense. In summary, it was observed that in approximately more than 98% of the total proper trials, the SDP relaxation / augmented relaxation successfully found the true optimum. Even for the rest cases where it failed, the provided gap was fairly acceptable (about several % of the optimum value).

3 Decentralized static estimation

Decentralized static estimation problem described here is a natural counterpart of the decentralized detection case. Recall that the decision function of the detection problem maps a discrete measurement to a discrete decision variable. *i.e.*, $\gamma_n(y_n) : \{1, \ldots, L\} \to \{1, \ldots, M\}$.

In the decentralized static estimation problem, each agent takes a noisy measurement vector $y_n \in \mathbb{R}^{m_n}$ and the estimator function $\gamma_n(y_n)$ returns the optimal estimate $u_n \in \mathbb{R}^d$ minimizing some global objective function.

We consider the 2-agent decentralized static estimation problem. For a static state $x \in \mathbb{R}^d$ with its *a priori* statistics known as $x \sim \mathcal{N}(\mu, P)$, the mesurements $y_1 \in \mathbb{R}^{m_1}$ and $y_2 \in \mathbb{R}^{m_2}$, corrupted by the independent noise $w_i \sim \mathcal{N}(0, N_i)$, are given to each agent respectively, *i.e.*,

$$\begin{aligned} y_1 &= A_1 x + w_1 \\ y_2 &= A_2 x + w_2 \end{aligned}$$

Then the optimal estimate is found by,

$$\text{minimize} \quad \mathbf{E}c(u_1, u_2, x)$$

where $u_n = \gamma_n(y_n)$ is the estimator of nth agent.

If the function $c(u_1, u_2, x)$ is convex quadrataic, it is known that there exists the optimal γ_n which is linear. With the reasonable choice of the cost function $c(u_1, u_2, x) = ||u_1 - x||^2 + ||u_2 - x||^2 + \delta^2||u_1 - u_2||^2$, the optimal estimator $u_n^* = \gamma_n^*(y_n) = K_n^* y_n + z_n^*$ are found from the following relations.

$$\begin{bmatrix} (1+\delta^2)(A_1PA_1^T + N_1) & -\delta^2 A_1 P A_2^T & 0 & 0 \\ -\delta^2 A_2 P A_1^T & (1+\delta^2)(A_2PA_2^T + N_2) & 0 & 0 \\ (1+\delta^2)\mu^T A_1^T & -\delta^2 \mu^T A_2^T & 1+\delta^2 & -\delta^2 \\ -\delta^2 \mu^T A_1^T & (1+\delta^2)\mu^T A_2^T & -\delta^2 & 1+\delta^2 \end{bmatrix} \begin{bmatrix} K_1^T \\ K_2^T \\ z_1^T \\ z_2^T \end{bmatrix} = \begin{bmatrix} A_1 P \\ A_2 P \\ \mu^T \\ \mu^T \end{bmatrix}$$

For the simplest case with $\delta = 0$, the problem is decomposed into two independent problems. Then the solution corresponds to the well-known update formula.

$$\begin{aligned} K_n &= PA_n^T(A_nPA_n^T + N_n)^{-1} \\ z_n &= (A_n^T N_n^{-1} A_n + P^{-1})^{-1} P^{-1}\mu \\ \text{or} \quad u_n &= \mu + PA_n^T(A_nPA_n^T + N_n)^{-1}(y_n - A_n\mu) \end{aligned}$$

The optimal solution is also found from the following.

$$\begin{aligned} \text{minimize} \quad & \left\|(\hat{C} + DK\hat{B})\right\|_F^2 \\ \text{subject to} \quad & K = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}, \qquad (C + DKB)\begin{bmatrix} \mu \\ 0 \end{bmatrix} + Dz = 0 \end{aligned}$$

where $z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$, $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, $N = \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix}$, $B = \begin{bmatrix} A_1 & I & 0 \\ A_2 & 0 & I \end{bmatrix}$,
$C = \begin{bmatrix} -I & 0 \\ -I & 0 \\ 0 & 0 \end{bmatrix}$, $D = \begin{bmatrix} I & 0 \\ 0 & I \\ \delta I & -\delta I \end{bmatrix}$, $\hat{B} = B\begin{bmatrix} P^{1/2} & 0 \\ 0 & N^{1/2} \end{bmatrix}$, and $\hat{C} = C\begin{bmatrix} P^{1/2} & 0 \\ 0 & N^{1/2} \end{bmatrix}$.

This is a simple convex problem, whose optimal solution should be identical to the explicit result in the previous section.

4 Next directions

The relaxed decentralized detection problem, which provides a lower bound of the optimal cost, was studied. Then the relaxed problem is augmented and iterated to find a sub-optimal primal feasible solution. A series of random simulations demonstrated that the combination finds the true optimum in most cases, or at least provide a satisfactorily tight gap. We are interested in discovering the conditions under which the relaxed problem or the augmented iteration finds the exact optimum with zero bounds, and developing the proofs for such conditions.

As a natural extension of the decentralized detection problem, we formulated basic frameworks for the decentralized static estimation problem. Under the convex quadratic cost and the affine estimator assumptions, we showed that the optimum is easily found. An interesting and realistic extension is the decentralized dynamic estimation problem (with limited inter-agent communication assumption), which we are currently putting our efforts to.

B.4 Reduced order representations for efficient computation

Reduced Order Representations for Efficient Computation

Laurent Lessard
Stanford University

Motivation

Consider a large distributed estimation problem with N agents that must estimate each other's positions. The optimal estimate can depend on global information structures, such as a densely populated $N \times N$ covariance matrix.

Taking a simple example: we would like to estimate $x_1, x_2, \ldots, x_N$, and each agent takes a single measurement $y_k = x_k + v_k$, where v is white noise with covariance σ^2, and the vector x has a normally distributed prior with covariance matrix C.

In this case, the optimal estimate is the vector: $x_{\text{opt}} = C\,(C+\sigma^2 I)^{-1}\,y$. Our goal is to compute such quantities efficiently, in $O(N)$. This means that the total computation cost is proportional to the number of agents; adding more agents does not change the computational burden of each agent. Note that if we can also distribute the computation evenly amongst the agents and execute it in parallel, the estimation requires a constant $O(1)$ cost.

In order to achieve this, we perform certain basic computations. Namely, if w is an arbitrary vector of length N, we should know how to compute matrix multiplications and inverses: computing Cw or $C^{-1}w$ in $O(N)$. We consider problems in which C has a high degree of symmetry. Namely, it is shift-invariant (Toeplitz or circulant). We will show how to factorize such systems in a way that yields the desired computational speedup.

Sparse Matrices

An $N \times N$ sparse matrix is a matrix that contains $O(N)$ nonzero elements. If P is sparse, Pw can be computed in $O(N)$. Also, for certain classes of P, $P^{-1}w$ can be computed in $O(N)$ as well. This is typically done with a multigrid method using a small fixed number of iterations. Such algorithms can be implemented in a distributed fashion. Note that P^{-1} is not sparse in general, and that iterative solvers never compute P^{-1} explicitly.

Circulant Factorization

Circulant matrices are constant along each diagonal, and the diagonals wrap around. In other words, each column of a circulant matrix can be obtained by taking the previous column and applying a circular shift of one. All circulant matrices are diagonalized by the discrete Fourier transform (DFT) matrix.

If the eigenvalues λ_k are generated by a rational function:

$$\lambda_k = \frac{p_0 + p_1 z_k + \cdots + p_m z_k^m}{q_0 + q_1 z_k + \cdots + q_n z_k^n}$$

where $z_k = \exp(2\pi ik/N)$ are evenly distributed on the unit circle, we can factor C as:

$$C = \underbrace{\begin{bmatrix} p_0 & & & & p_1 \\ \vdots & p_0 & & & \\ p_m & & \ddots & & \\ & \ddots & & p_0 & \\ & & p_m & \cdots & p_0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} q_0 & & & & q_1 \\ \vdots & q_0 & & & \\ q_n & & \ddots & & \\ & \ddots & & q_0 & \\ & & q_n & \cdots & q_0 \end{bmatrix}}_{Q}^{-1}$$

where P and Q are sparse banded circulant matrices. The factorization $C=PQ^{-1}$ is very beneficial, because it allows us to compute Cw by first solving $w=Qz$, and then computing $y=Pz$. The total cost is $O(N)$. We can compute $C^{-1}w$ in $O(N)$ similarly. Note that without this factorization, the cost would be $O(N\log N)$, using an FFT method.

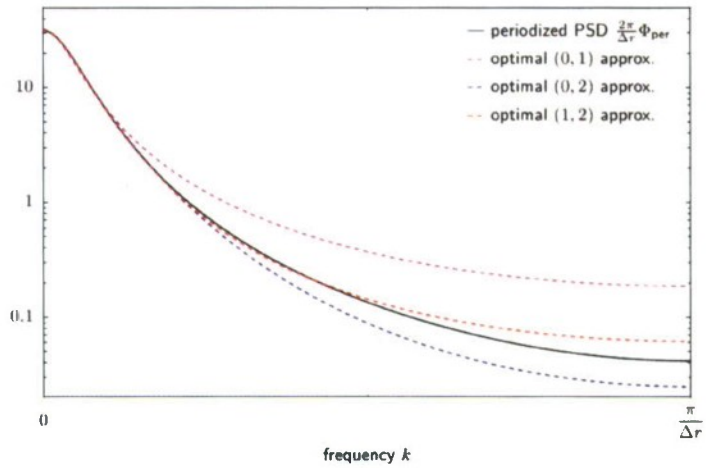
This factorization is only exact when we are dealing with a rational eigenvalue distribution, as explained above. In the general case, we will seek an approximate factorization $C\approx PQ^{-1}$. For some classes of problems, we can find an optimal approximate factorization for C in the sense that we can trade off sparsity (width of the bands in P and Q) with solution accuracy (the error $|Cw - PQ^{-1}w|$).

Example: wavefront sensing

In wavefront sensing for adaptive optics, one encounters the von Kármán power spectral density (PSD) for atmospheric turbulence in 1-D:

$$\Phi(k) = \frac{1}{(k^2+1)^{4/3}}$$

The correlation between two points separated by a distance r can be computed by integrating and sampling the PSD. If we have an array of evenly spaced detectors with spacing Δr, the pairwise coefficients can be assembled in a Toeplitz matrix C. We would like to compute Cw efficiently for any w, but C's PSD is irrational so the eigenvalues distribution is as well. Using a Principal Axis optimization routine, we computed some low-order approximations for the eigenvalue distribution that minimize the mean-squared approximation error under white-noise inputs. In other words, we minimized $E|Cw - PQ^{-1}w|^2$. See below for a figure comparing the PSD error in various rational approximations:



It turns out that in the limit of large N, this is equivalent to minimizing the L_2 error in the PSD's approximation. For example, the factorization we found in the (1,2) case using Δr=0.2 is:

$$\hat{C} = \begin{bmatrix} p_0 & p_1 & & & p_1 \\ p_1 & p_0 & & & \\ & & \ddots & & \\ & & & & p_1 \\ p_1 & & & p_1 & p_0 \end{bmatrix} \begin{bmatrix} q_0 & q_1 & q_2 & & q_2 & q_1 \\ q_1 & q_0 & q_1 & & & q_2 \\ q_2 & q_1 & & & \ddots & \\ & & & \ddots & & q_2 \\ q_2 & & \ddots & & q_0 & q_1 \\ q_1 & q_2 & & q_2 & q_1 & q_0 \end{bmatrix}^{-1}$$

where p = [0.100126 –0.0301038] and q = [1.0000000 –0.650885 0.151518]. This case leads to an average error of about 2.5% in each component of the product Cw. This error does not depend on N in the limit, and in fact, the same factorization can be used for larger N (keep the same p and q, and extend the matrices).

Using such a factorization is always $O(N)$. If we wanted to do better than 2.5%, we could find a higher-order approximation. This would increase the amount of computation each agent has to do, but the computation required would always be independent of the number of agents, N.

Conclusion

We showed that a shift-invariant matrix can be factored as $C=PQ^{-1}$ using sparse matrices P and Q if the eigenvalues of C are rational functions on the unit circle. Such a factorization leads to fast distributed computation without any performance penalty.

If the eigenvalues are not rationally distributed, we can look for a rational approximation. We showed an example of a distributed estimation problem where we could find optimal rational approximations of a given order. By varying the chosen order, we obtain a family of $O(N)$ solvers that trade off computational accuracy and mean-squared error performance.

B.5 Graph decomposition for biological networks

Graph Decomposition for Biological Networks

Alice Hubenko and Igor Mezić
Department of Mechanical Engineering
University of California, Santa Barbara
Santa Barbara, CA

November 14, 2008

Abstract

Contemporary systems biology presents science and engineering with many new challenges. Finding effective ways to study large networks of interacting components is one of the key tasks of this growing discipline. Because of their numerous applications in drug target design and bioreactor technology, metabolic networks are of special interest. These networks are usually modeled as dynamical systems on graphs. This approach allows to accurately simulate the network behavior in different conditions by solving large systems of differential equations with an even larger number of parameters. However as we attempt to analyze larger networks, the growing number of equations complicates numerical analysis. In the present paper we employ a horizontal-vertical decomposition (HVD) method that helps to analyze dynamical systems on graphs that have very large dimensions. We illustrate the HVD decomposition on metabolic networks of several single cell organisms. Using HVD decomposition we identify modular structure and polarity in the network, which in turn allows us to reduce the dimension of the problem.

1 Introduction

In modern biology and medicine there is a great need of understanding biological processes integrated in their system environment. The study of complex interactions in biological networks has grown into an independent discipline of systems biology important parts of which are mathematical modeling and dynamic simulation [18].

Even though presently the amount of experimental data is growing fast, quantitative measurements of many cellular components are still unavailable. Shortage of data in combination with biological complexity makes it very challenging to apply standard engineering methods for modeling, simulation and

mathematical analysis of biochemical networks [12]. The existing biochemical data measurements usually come from different experimental settings and are described on different levels of information quality, besides for most processes, quantitative information on reaction rates and molecular concentrations is not directly accessible in vivo. [12]. In cases where the mechanistic details are unclear, it is necessary to fill in the gaps by suggesting simple mechanisms without having any kinetic parameters available [10]. If the dynamic behavior of a system is highly dependent on the value of (some of) the parameters, then accurate and reliable quantification of the parameters is essential for the development of predictive models [19]. Therefore being able to predict how values of certain variables of the biological system affect other variables is very important.

In [11] Mezić introduced a horizontal - vertical decomposition method (HVD) to analyze asymptotic behavior and uncertainty propagation in nonlinear dynamical systems on graphs. We illustrate the HVD decomposition on metabolic networks of several single cell organisms and assume that the dynamics of the system is described by a system of non-linear ordinary differential equation (ODE). In case of metabolic networks, the variables may represent, for instance, concentrations of the metabolites, and the equations describe how changes in concentrations depend on each other. Modelling metabolic networks as dynamical systems has many advantages. The dynamical system approach is very convenient to describe features like feedforward and feedback loops, and to study concepts of stability, robustness, homeostasis and adaptation. Homeostasis is one of the most typical properties of highly complex open systems. In dynamical system setting it means that when a disturbance occurs, interdependent regulation mechanisms make the system converge to a dynamic equilibrium. HVD allows us to extract valuable information on asymptotic dynamics of metabolic networks using its structure without precise estimates of the differential equations.

Large genome sequencing projects made possible the reconstruction of complete metabolic and signaling networks for many organisms. In [14] Barabasi and his coworkers performed quantitative analysis of the topological properties of complete metabolic networks of 42 organisms, 25 of which were fully sequenced. These metabolic networks were reconstructed based on data deposited in WIT database, see [13]. We analyze qualitative features of metabolic networks from [14], using essentially the same model with the only difference that we do not represent the temporary substrate-enzyme complexes in the network.

2 Horizontal-vertical decomposition

Metabolic networks are usually modelled as nonlinear dynamical systems on graphs. In our model vertices of the graph represent the metabolites (substances produced or used during metabolism) participating in reactions. Each reaction is catalyzed by a unique enzyme which we do not represent by a vertex. For each chemical reaction that takes place in the metabolism, the vertices representing reactants are connected to the products of the reaction. For example, if we have a reaction $V_1 + V_2 = V_3 + V_4$ it will be represented by the graph shown in figure 2. The connecting edges are directed and always point to the product of the reaction. We assume that there are no multiple edges in the graph. It means that there will be one edge going from vertex A to B even if there is more than one reaction involving reactant A and producing reactant B. As an example of such model, the graph shown in figure 1 is the metabolic network model of Chlamydia pneumoniae (C. pneumoniae). This graph has 187 vertices (the number of substances in the network) and 652 edges.

We assume that to each vertex of the graph belongs a variable which is the concentration of that metabolite and an ODE that describes how the change in concentration of a given metabolite depends on other metabolite concentrations. It is clear that a system of the 187 differential equations corresponding to this graph can be very complicated. However, using HVD decomposition we can reduce the dimension of the problem by revealing modular structure and polarity of the network. In [11] in Theorem 13 it was proved that

Theorem 2.1 *A dynamical system can be decomposed into k vertical levels, such that each higher level is driven by the dynamics of levels below. Every horizontal level $i \in \{1, ..., k\}$ can be decomposed into m_i sets, which have dynamics independent of each other.*

This decomposition is called horizontal-vertical decomposition (HVD) of the system. The proof of theorem 2.1 is based on repeatedly separating sets of variables with isolated dynamics using its Jacobian (see [11]). We assume that at each step we separate the maximal number of such sets, and we will regard this unique horizontal-vertical arrangement as the HVD decomposition.

Note, that in general the number of layers k in decomposition of this type may vary. For example, we could define a separate layer for each set having isolated dynamics in which case the number of layers would be equal to $\sum_{i=1}^{k} m_i$. However, for a fixed set of parameters, the partition of the set of variables into sets with isolated dynamics is unique and does not depend on the number of layers in the decomposition.

The adjacency matrix of a directed graph G on n vertices $\{v_1, v_2, ..., v_n\}$ is a $n \times n$ matrix A with

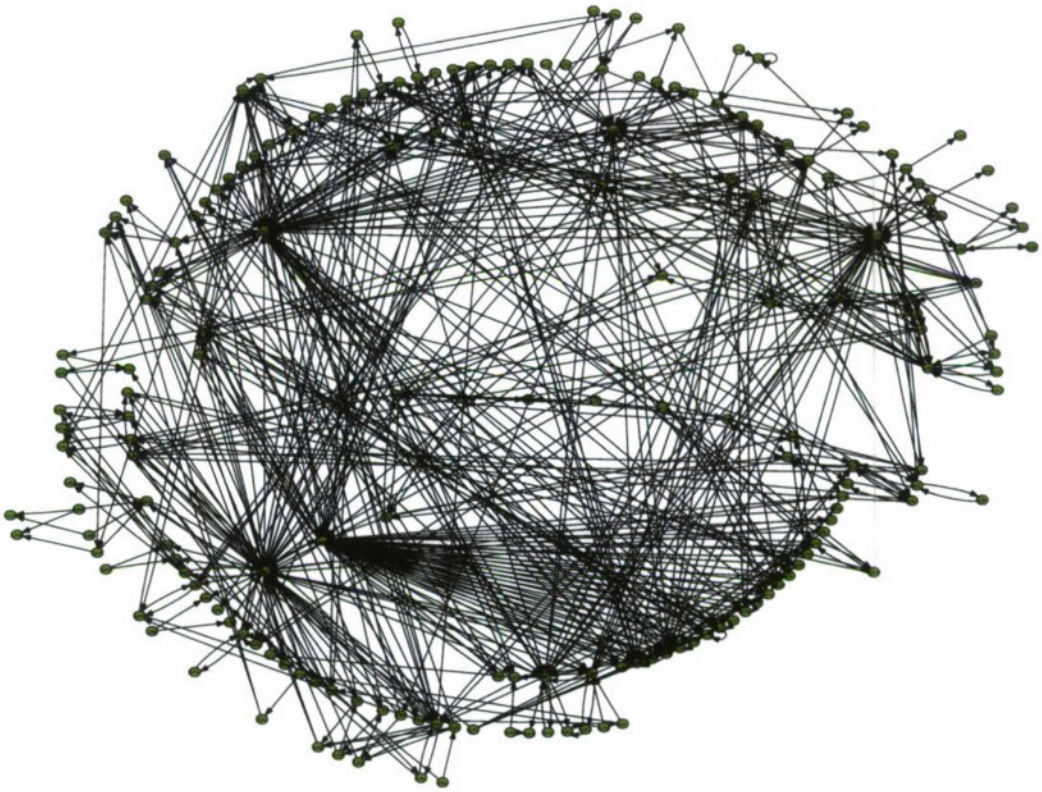


Figure 1: Metabolic network of C. pneumoniae

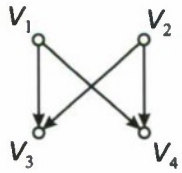


Figure 2: Graph with adjacency matrix A

entry $a_{i,j}$ equal to 1 if $v_i v_j$ is an edge and 0 otherwise. In figure 2 is an example of a directed graph that has the following adjacency matrix.

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The following two important observations for our metabolic network model follow from [11].

Observation 2.2 *Assume that a metabolic network is described by a directed graph with an $n \times n$ adjacency matrix A and a system of n differential equations with Jacobian J. An entry of the transpose of the adjacency matrix A^T is non-zero if and only if the corresponding entry of J is non-zero.*

A directed graph is called strongly connected if for every pair of vertices u and v there is a directed path from u to v and from v to u. The strongly connected components are the maximal strongly connected subgraphs of a directed graph. To shorten the description, we will use the term strong component instead of strongly connected component. We will use notation SC for a strong component. Strong components form a partition of the vertex set of the graph.

Corollary 2.3 *Assume that a metabolic network is described by a directed graph G with an $n \times n$ adjacency matrix A and a system of n differential equations with Jacobian J. Then for every level $i \in \{1, ..., k\}$ of the HVD the subsets m_i with independent dynamics described in Theorem 2.1 correspond to strong components of G.*

The proof of the following lemma is straightforward, see [5].

Lemma 2.4 *If each strong component is contracted to a single vertex, the resulting graph is cycle–free, i.e. is a directed acyclic graph.*

Assume that we have a metabolic network described by a directed graph G. From Corollary 2.3 and Lemma 2.4 it follows that if we contract each strong component of G to a single vertex , the resulting graph $SC(G)$ is acyclic. Each vertex of $SC(G)$ corresponds to a set with isolated dynamics in the HVD decomposition. To complete the HVD decomposition of the metabolic network, we have to assign each vertex of $SC(G)$ to a level. We call a vertex sink, if it has only incoming or in-edges edges (but no out-edges). We call a vertex source if it has only out-edges edges (but no in-edges).

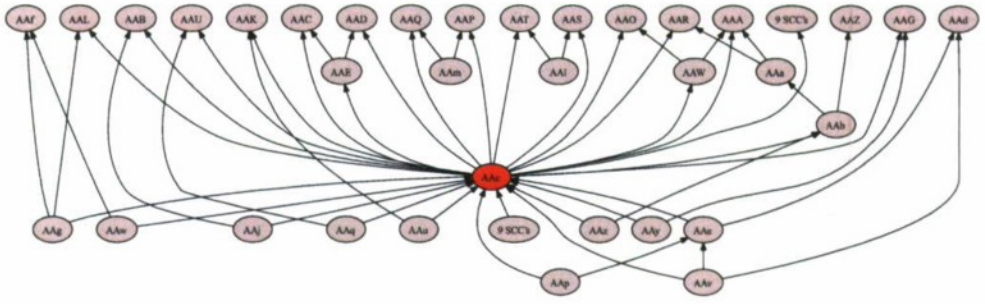


Figure 3: HVD graph of the metabolic network of C. pneumoniae

First, we find the set of all the sinks in $SC(G)$, S_1 and place them in the top level. The vertices of S_1 correspond to the sets with isolated dynamics that we separate first, according to the proof of Theorem 2.1. Next, we delete the set of vertices S_1 from $SC(G)$, and place the set of all the sinks S_2 of $SC(G) - S_1$ in the next level. Then, we delete the sets of vertices S_1 and S_2 from $SC(G)$, and place the set of all the sinks S_3 in the resulting graph $SC(G) - S_1 - S_2$ in the next level... Continuing in this manner we obtain the HVD graph G^* of the metabolic network. All the sinks of $SC(G)$ will be in the top level of G^* and all the sources of $SC(G)$ will be in the bottom level of G^*. To summarize the above procedure, finding the partition of the dynamic model of the metabolic network into sets with isolated dynamics is equivalent to finding the strong components of the corresponding graph G, which is a well known problem in graph theory. HVD decomposition G^* of a dynamical system with graph G can be found in 3 steps:

1. Find strong components of G
2. Contract each strong component to a single vertex to find $SC(G)$
3. Assign vertices of $SC(G)$ to levels, to obtain the HVD graph G^*

If c_i is a concentration of interest, we can find the trajectory of c_i considering the system of dynamical equations corresponding to all variables that are below c_i in the HVD decomposition and can reach c_i via directed path. Using the metabolic network of Chlamydia pneumoniae as an example, let us demonstrate how this approach can result in a considerable reduction in the dimension of this problem. In figure 3 there is a picture of the HVD graph of the metabolic network of Chlamydia pneumoniae. There are 52 vertices in this graph and each vertex represents a strong component. There are 9 single-vertex components that send an edge to component AAc, they are schematically

represented by a single vertex. Also, there are arc 9 single-vertex components each of that receive an edge from component AAc, they are also represented by a single vertex on the picture of the HVD graph. We will call the size of a strong component the number of vertices in it. There is one strong component of size 136 (AAc, shown in figure 3 in red), all other strong components consist of a single vertex. Assume, that the concentration c_i that is of interest to us corresponds, for instance, to the single-vertex component AAd. In this case it is sufficient to consider the system of equations corresponding to AAd, AAe, AAv and AAp to find trajectory of c_i. This means that instead of having to solve a system of 187 differential equations as defined in the original problem we reduced finding the trajectory of c_i to solving a system of 4 differential equations. However, if c_i is in the 136 vertex component AAc we still have to consider a system of 149 equations (these correspond to vertices in components AAc, AAu, AAp, AAz, AAv, the 9 single-vertex components represented by vertex 9SCCs and the component AAc itself). It is clear from our example that in order to substantially reduce the dimension we have to also decompose the giant strong component itself. In the next chapter we will propose a way to do it.

There are several algorithms that compute strong components of a graph. Tarjan's algorithm [5] is one of the most favored in practice. It finds strong components in time linear in the size of the graph.

3 The giant strongly connected component

Each of 42 metabolic networks that we tested, had one very large strong component, containing about 84% of all vertices and many single–vertex components with occasional components of size 2 or 3. We will refer to the largest strong component of the metabolic networks as the giant strongly connected component, denoted GSC [9]. Metabolic networks belong to the class of scale–free networks [14], which means that their degree sequences follow a power law, so there are a few vertices of very high degree in the network. These vertices are called hubs, they integrate the functionally independent modules of the metabolism into a robust network [15]. Most of the vertices are connected through hubs by a relatively short path [1], [16].

In figure 5 we plotted the degree sequences of metabolic networks of C. pneumoniae, Escherichia coli (E.coli) and Saccharomyces cerevisiae (S. cerevisiae). To distinguish functional modules based on the network topology alone, several hubs are usually removed. Cutting vertices of highest degrees increases the modularity of the remaining network [6] and the independent functional units become cohesive subsets of vertices that are sparsely interconnected with each other. In a preprocessed network modules can be distinguished as highly connected subsets [15]. The argument behind it is that these

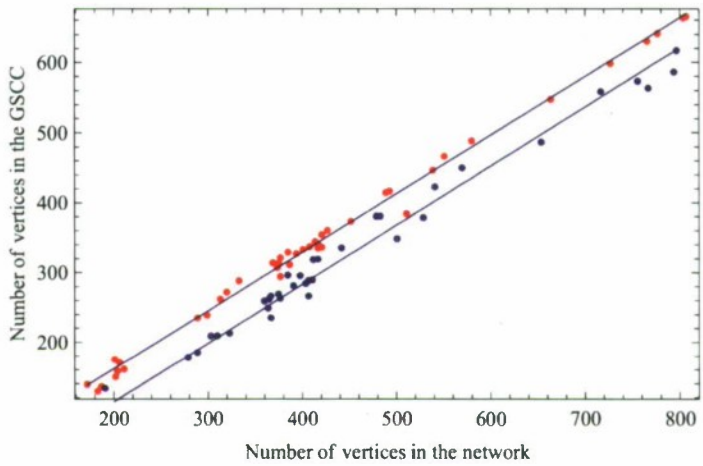


Figure 4: Size of GSC before and after cutting 10 hubs

The data was fitted with the lines $y = 0.84x - 6$ for the original networks and $y = 0.85x - 55$ for networks after cutting 10 hubs.

metabolites have secondary role in synthesis pathways within the cell [15]. Such metabolites are for example ATP, ADP, NADH, NAD+ and have primary role in energy supply to sustain the synthesis. In [8], [9] current metabolites and cofactors like ATP, ADP, NADH, NAD+, H_2O and Pi have been removed (small molecules such as H_2O, NH_3, O_2, CO_2 and phosphate are also considered as current metabolites). These metabolites function as carriers for transferring electrons and certain functional groups (phosphate group, amino group, one carbon unit, methyl group etc.). The classification of current metabolites in [8], [9] is similar to the classification of internal and external metabolites in [17], where external metabolites are removed. These are metabolites that can be regarded as externally buffered with respect to the system. Their concentrations can be considered constant in a normally functioning cell, in other words, they put virtually no constraints on network dynamics. Since current metabolites have high degrees, they are represented by hubs in studies where abundant metabolites have not been removed [6], [14]. From the above observation it follows that in a dynamically modelled metabolic network variables representing hubs can be safely removed and replaced by uncertain constants, which in turn means that the corresponding vertices (and edges attached to them) will disappear from the graph of the metabolic network. We call a degree of a vertex the sum of its in-degree and out-degree. The plot in figure 4 shows in red the size of GSC as a function of the network size for the 42 networks that we tested. Blue dots show the size of GSC after cutting the first 10 hubs from

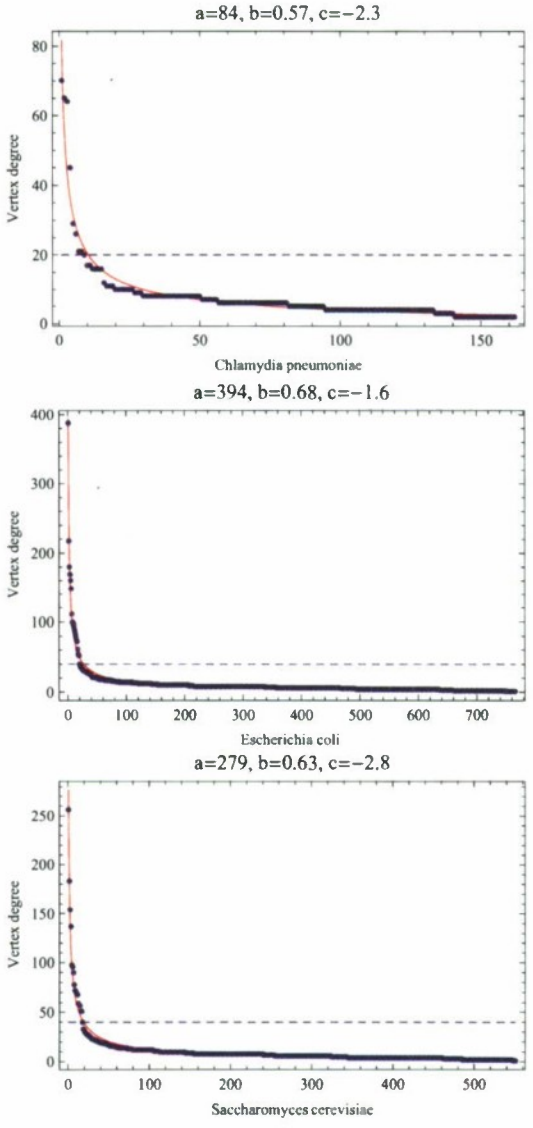


Figure 5: Degree distributions of C. pneumoniae, E.coli and S. cerevisiae

The degree sequences of each of the 42 metabolic networks that we have tested can be fitted by a function of type $y = c + a/x^b$. The vertices above dotted line (hubs) were removed from the network.

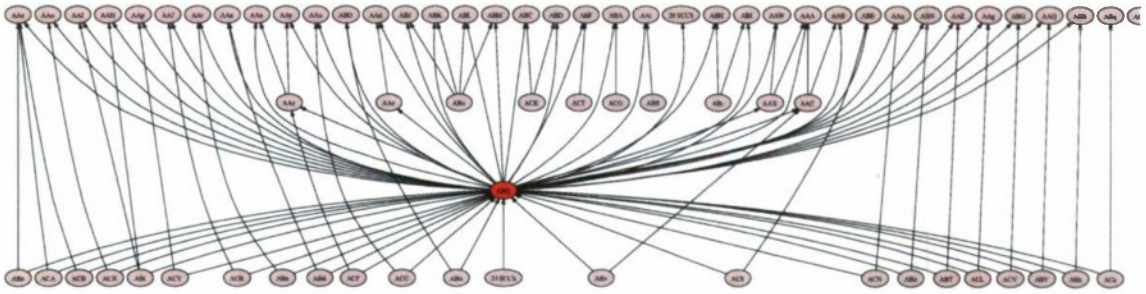


Figure 6: HVD graph of the metabolic network of E.Coli

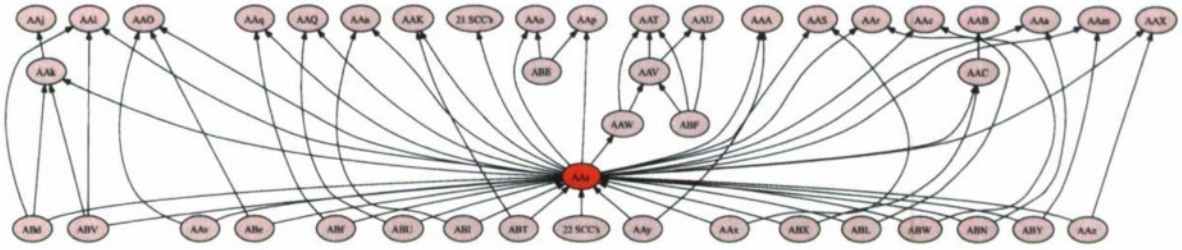


Figure 7: HVD graph of the metabolic network of S. cerevisiae

each of 42 network models. The data was fitted with linear functions. The size of GSC before cutting the hubs behaves approximately as $y = 0.84x - 6$, and the size of GSC after cutting 10 hubs behaves approximately as $y = 0.85x - 55$.

GSC is the core part of the metabolic network, that constitutes the dense and complicated central unit that processes a large number of metabolites fanning in and produces from them many metabolites that are fanning out. One can observe this characteristic on HVD graphs of metabolic networks of C. pneumoniae, E.coli and S. cerevisiae are shown in figure 3, figure 6 and figure 7 respectively (GSC is shown in red). The metabolic network model of C. pneumoniae has 187 vertices and 652 edges see fig. 3, GSC has 136 vertices and 551 edges. The metabolic network model of E.coli has 766 vertices and 3988 edges see fig. 6, GSC has 630 vertices and 3701 edges. The metabolic network model of S. cerevisiae has 551 vertices and 2806 edges see fig. 6, GSC has 466 vertices and 2631 edges. The type of structure observed above is called bow-tie architecture, it is the key design principle in biological networks that allows them to be robust yet flexible and evolvable [7].

In figure 8 we plotted the average degrees in GSC of the original and network and after cutting 10 hubs from all networks, except C. pneumoniae, Helicobacter pylori, Methanococcus jannaschii,

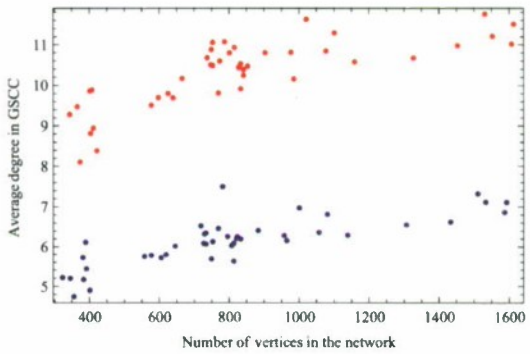


Figure 8: Average degrees in GSC of the original network and after cutting 10 hubs
Red: average degree in GSC of the original network
Blue: average degree in GSC after cutting about 10 hubs

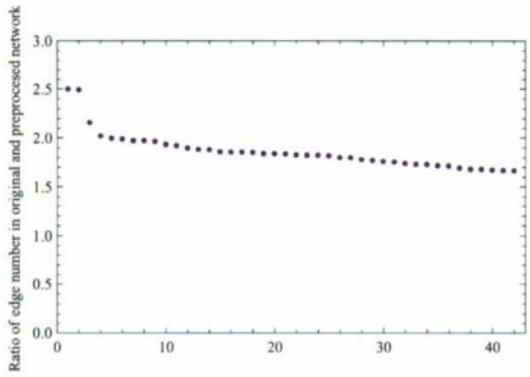


Figure 9: Ratio of number of edges in the original and after cutting about 10 hubs

Streptococcus pyogenes, Treponema pallidum where we cut 9 hubs, because the 10th and 11th hubs in these networks have the same degree.

Cutting about 10 hubs from the network reduced the number of edges almost by a factor of 2 for most vertices, and even more for smaller networks, see figure 9. This suggests that we could achieve a better reduction of number of edges in the networks, if we cut a larger number of hubs as the network size grows. In particular, the number of edges in C. pneumoniae network became 2 times smaller (323 edges); the number of edges in E. coli network became 1.7 times smaller (2386 edges); the number of edges in S. cerevisiae network became 1.7 times smaller (1624 edges). The HVD graph of C. pneumoniae after cutting the first 9 hubs is shown in figure 10. The GSC of this preprocessed network has 65 vertices and 154 edges. Which means that the GSC graph is relatively sparse, with average degree 4.7. If we cut the first 20 hubs in E. coli network, and the 18 first hubs in S. cerevisiae network, we will achieve reduction in edge number similar to that observed in C. pneumoniae after cutting 9 hubs. The number of edges in E. coli network after cutting 20 hubs is 1804, which makes a reduction ratio of 2.2. The GSC of E. coli network after cutting 20 hubs has 499 vertices and 1424 edges, which makes the average degree inside GSC 5.7. The number of edges in S.cervisiae network after cutting 18 hubs is 1259, which makes a reduction ratio of 2.2. The GSC of S. cerevisiae network after cutting 18 hubs has 381 vertices and 1030 edges, which makes the average degree inside GSC 5.4. In figure 11 is the picture of a HVD graph of S. cerevisiae after cutting the first 18 hubs. In figure 12 is the picture of a HVD graph of E.coli after cutting the first 20 hubs. After cutting several hubs, the metabolic network does not consist of one connected unit anymore, but its largest connected component still has a bow-tie structure with a giant strong component in the center of the network.

HVD decomposition introduces a polarity in the metabolic network graph. If we could come up with a decomposition method for the GSC that extends the polarity introduced by HVD inside the GSC, this would help us to further reduce dimension of the problem for variables inside GSC. Next, we propose a way to achieves this. We will illustrate our decomposition method on GSC of C. pneumoniae. First we locate vertices of GSC that have incoming edges from outside the GSC. We call these vertices sources. Next, we identify vertices that have edges pointing from them outside the GSC. We call these vertices sinks (the sinks are shown in blue in figure 13).

We put sources in the same bottom level, then we perform distance-labeling of the vertices in GSC, starting with the bottom level. In each step we construct a new level: it will include all vertices that can be reached by directed edges pointing to them from the last constructed level. In figure 13 is shown the decomposition of the GSC of C. pneumonia, where we can observe a clear pathway structure in GSC. The decomposition described above can be easily combined with the HVD decomposition of the

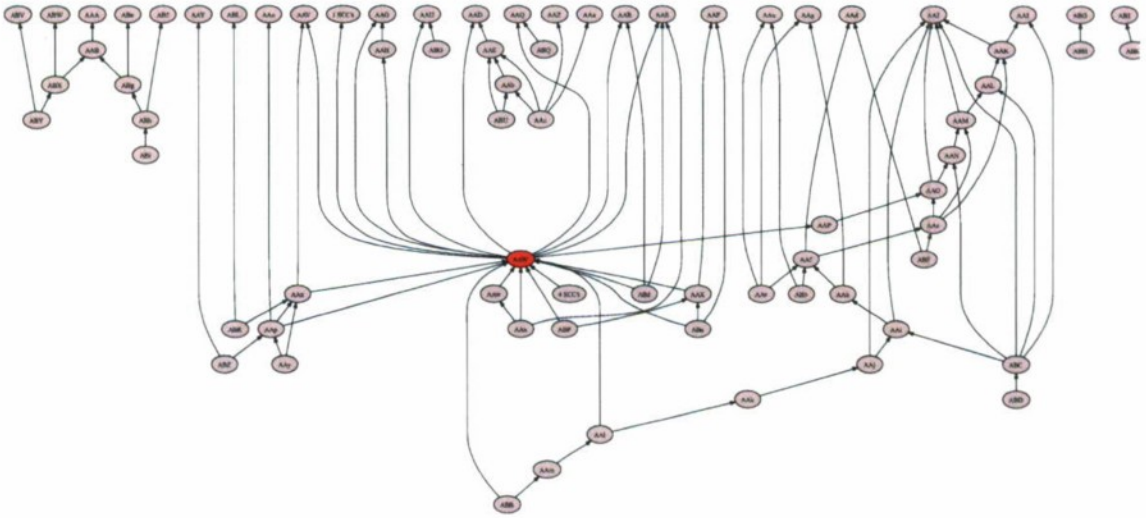


Figure 10: HVD graph of the C. pneumoniae network after cutting the first 9 hubs

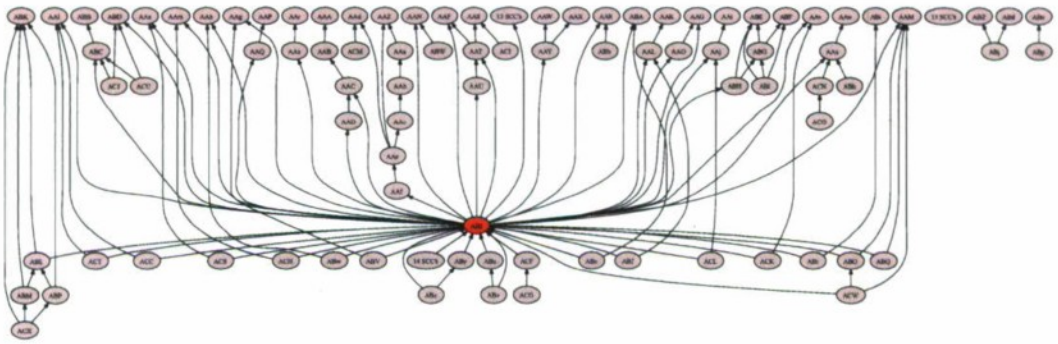


Figure 11: HVD graph of the S. cerevisiae network after cutting the first 18 hubs

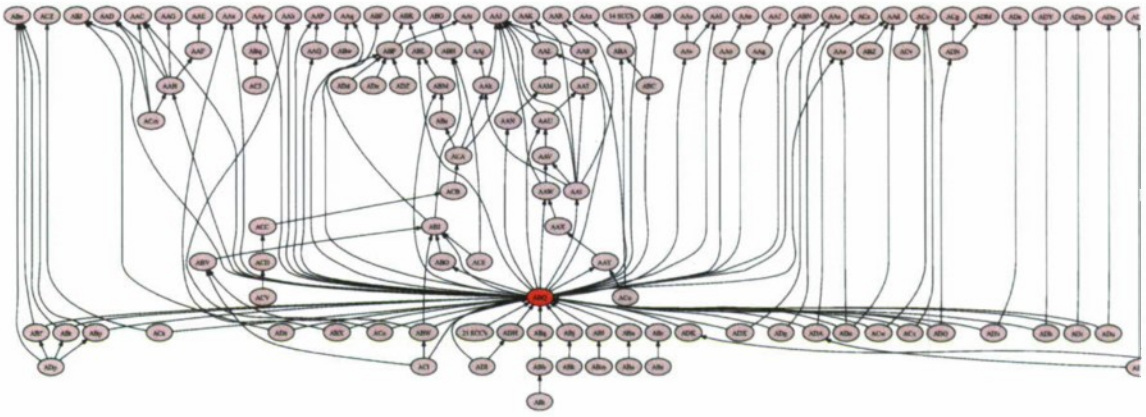


Figure 12: HVD graph of the E. coli network after cutting the first 21 hubs

entire network. Because the GSC is relatively sparse, there are few edges connecting vertices in the same level, and in the rest of the GSC we uncovered a clear pathway structure, see figure 13.

If two metabolites are connected with two edges pointing in opposite direction, one of these edges must indicate the direction in which the synthesis goes in the metabolism under normal conditions (forward edge), and the function of the other edge must be stabilizing the system in case if conditions change (feedback edge). The look at the decomposition of GSC from dynamical perspective suggests that most likely forward edges are always pointing up and feedback edges are pointing down. Of course, further tests are needed to confirm our hypothesis.

4 Conclusion

We use a novel approach, HVD decomposition, to reduce dimension of problems on large-scale biological networks. We show examples of HVD decomposition for metabolic networks of several organisms, and describe how to use it to reduce the dimension of the problem of finding the trajectory of a concentration of a given metabolite. The bow-tie structure of metabolic networks with GSC in its center is reconfirmed. We show that cutting current metabolites from the metabolic network does not affect its dynamical system model. Moreover, cutting as few as 10 hubs from the network decreases the number of edges in the network about by a factor of 2 and significantly decreases the average degree in the GSC. This allows us to further extend the HVD decomposition inside the new, much sparser, GSC. We state as hypothesis, that HVD decomposition and its extension inside GSC reveals forward

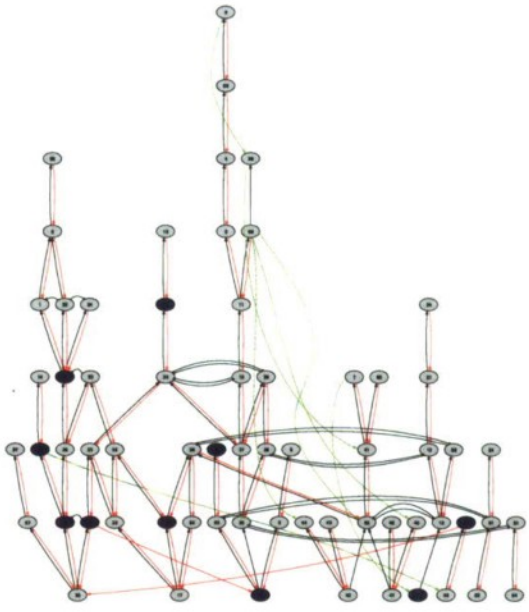


Figure 13: Giant strong component of C. Pneumoniae network after cutting 9 hubs

and feedback edges of the network based only on structural properties of the metabolic network.

Acknowledgements

This work was in part supported by DARPA DSO (Dr. Cindy Daniell PM) under AFOSR contract FA9550-07-C-0024 (Dr. Fariba Fahroo PM).

References

[1] R. Albert and A. L. Barabasi, Statistical mechanics of complex networks,*Reviews of Modern Physics*, **74**,(2002), 47-97.

[2] V. Bouchitté and M. Morvan eds., Orders, Algorithms and Applications, **Springer**, (1994).

[3] E. R. Gansner, E. Koutsofios, S. C. North, K.-P. Vo, A Technique for Drawing Directed Graphs, *IEEE transactions on software engineering*, **19**, 3, (1993), 214–230.

[4] E. R. Gansner,S. C. North, An open graph visualization system and its applications to software engineering, *Software-practice & experience*, **30**, 11, (2000), 1203–1233.

[5] J. Bang-Jensen, G. Gutin, Digraphs Theory, Algorithms and Applications, **Springer-Verlag**, (1997).

[6] M. Huss and P. Holme, Currency and commodity metabolites: Their identification and relation to the modularity of metabolic networks,*IET systems biology*, **1**, 5, (2007), 280–285.

[7] M. Csete and J. Doyle, Bow ties, metabolism and disease, *Trends in Biotechnology*, **22** no. 9, (2004), 446–450.

[8] H. Ma and A. Zeng, Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms, *Bioinformatics*, **19** no. 2, (2003), 270-277.

[9] H. Ma and A. Zeng, The connectivity structure, giant strong component and centrality of metabolic networks, *Bioinformatics*, **19** no. 11, (2003), 1423-1430.

[10] S.R.Neves and R. Iyengar, Modeling of signaling networks, *Bioessays*, **24** (2002), 11101117.

[11] I. Mezić, Coupled Nonlinear Dynamical Systems: Asymptotic Behavior and Uncertainty Propagation, *43rd IEEE Conference on Decision and Control December 14-17*,(2004).

[12] Natal A.W. van Riel, Dynamic modelling and analysis of biochemical networks: mechanism-basedmodels and model-based experiments, *Briefings in bioinformatic*,**vol 7**, no. 4,(2006), 364–374.

[13] R. Overbeek, N. Larsen, G.D. Pusch, M. DSouza, Jr E. Selkov, N. Kyrpides, M. Fonstein, N. Maltsev, and E. Selkov, WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction, *Nucleic Acids Res.*, **28**, (2000), 123125.

[14] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.-L. Barabási, The large-scale organization of metabolic networks, *Nature*, **407**, (2000), 651–654.

[15] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai and A.-L. Barabási, Hierarchical Organization of Modularity in Metabolic Networks, *Science*, **297**, (2002), 1551–1555.

[16] S. H. Strogatz, Exploring complex networks, *Nature*, **410**,(2001), 268-276.

[17] S. Schuster, T. Pfeiffer, F. Moldenhauer, I. Koch, and T. Dandekar, Exploring the pathway structure of metabolism: Decomposition into subnetworks and application to Mycoplasma pneumoniae,*Bioinformatics*,**18**, 2002, 351361.

[18] N.A.W. van Riel and E.D. Sontag, Parameter estimation in models combining signal transduction and metabolic pathways: the dependent input approach,*IEE Proc Syst Biol*, **153**, (2006), 263274.

[19] H.V. Westerhoff and B. O. Palsson, The evolution of molecular biology into systems biology, *Nat Biotech*, **22**, (2004), 12491252.

B.6 Unfolding cell regulation network anatomy through graph decomposition

Unfolding cell regulation network anatomy through graph decomposition

Yueheng Lan,[1] Igor Mezić,[1*]

[1]Department of Mechanical and Environmental Engineering, University of California, Santa Barbara, CA 93106, USA

*To whom correspondence should be addressed; E-mail: mezic@engineering.ucsb.edu.

We propose a novel graph theoretic decomposition scheme for probing the generic structure of complex biological networks. For several most common cell regulation networks, according to the chosen polarity defined by the input and output signals, we identified the "minimum production unit"(MPU) which responds quickly and robustly to external signals and the feedback controllers which adjust the output of the MPU to desired values usually at a larger time scale. Detailed illustration and discussion are made to explain the network structures and how they are tied to biological functions. The proposed scheme may be potentially applied to parameter evaluation and key regulation factor identification in a variety of biological systems.

Cellular behavior, including motility, metabolism and reproduction is controlled by complex biochemical reaction networks, many of which have been identified and studied in detail (*1*). These networks realize their regulatory role through complex molecular interactions. Contemporary high through-put experiments produce unprecedented amount of data that serve to pinpoint the players and their interactions, resulting in complex chemical reaction graphs. How

to analyze these intricate graphs and gain insight into the regulation mechanism employed by cell has become a central problem of the molecular biology.

Much progress has been made in the analysis of complex networks, both deterministically (*2, 3*) and stochastically (*4–7*). These studies concentrate on investigation of dynamics of given networks by checking their stability, parameter dependence, robustness, input-output relation. However, for large-scale networks such as those commonly found in important bilogical processes (*8, 9*), the incurred computational load often severely limits detailed analysis. More critically, with continuous experimental efforts of revealing details of a network, it has become increasingly hard to identify the underlying characteristic structures and thus gain insight into the key mechanism that shape the network function. Recently, useful concepts distilled from the study of statistical physics such as the small-world and the scale-free network (*10, 11*), begin to see their application in gene regulation network and lead to considerable success. However, this type of statistical analysis mainly aims at gross features of networks (*12*) and thus ignores detailed inhomogeneities embedded which often determine the functioning of a network in an essential way, since disparate network topologies and dynamics fit for different functional requirements. Other methods of analysis (*13–15*) often fail to highlight interactions between the key elements of a system or properly reflect the dynamics associated with their function.

Normal cell life involves physical or chemical activities at vast range of spatial and temporal scales and one central task of systems biology is to identify the characteristic structures at all scales and study their roles in relation to a particular cell function (*16–21*). These key structures are called modules, the existence of which contributes almost to every aspect of the cell regulation: robustness, sensitivity, adaptivity, evolvability. Their detection and study much simplifies the analysis of complex networks since a set of modules could be a lot simpler than a collection of entangled individual agents (*22*). The simplification may be carried out further by constructing modules of modules. Nevertheless, the determination of modular structure

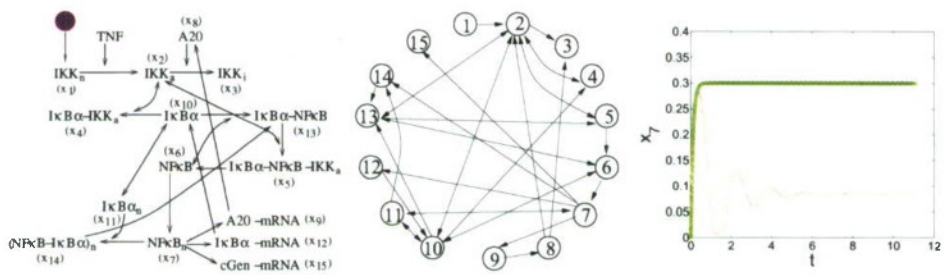
in a large network is not straightforward since one molecular species may be involved in many different pathways with very distinct external connections. Such inter-correlation is easily under appreciated and yet can have profound effects on the organism.

In this paper we propose a minimal production unit (MPU) - feedback controller theory of the biochemical networks based on the control theoretic point of view. In this theory, a network is decomposed into two motifs: one is the pipeline of linear production unit which serves to generate the output in a quick and robust way; the other is the set of feedback loops which act as controllers to the production. These two motifs are decided based on the information flow in the network. Input and output nodes can always be selected in a cell regulatory network, which defines a polarity of the network. The information is received at the input, processed and then sent to the output, which defines an overall forward direction. The units that carry on the information along the forward direction belongs to the production motif, while the remaining units direct the information in the opposite direction and are thus treated as feedback controllers. Every module in the network belongs to either of the two motifs and can be further decomposed in a similar way if necessary. In this way, a complex network can be analyzed into a hierarchy of modules with different sizes and internal dynamics. From biological evolution point of view, it is likely that this nested structure stems from a simple core and is later wrapped with complex controller in evolution. So, our theory reveals the stable generic feature of a biochemical network, which can be used to explore either the intricacies of a single structure or interdependencies of a series of systems. In the following, we show that this particular structure are accessible via graph theoretic analysis and revealing special production-controling facilities in cell regulation networks.

The dissection of large networks into functional modules greatly facilitates their analysis. The functional modules can be studied individually with well-designed boundary conditions. The properties of the whole network are gathered by piecing together the modules in an ordered

way. Henceforth, our strategy of analysis is characterized by a decomposition and recombination procedure. Below, we carry out the decomposition and show that a network can be algorithmically dissected into disparate functional units. The horizontal-vertical decomposition (HVD) developed in a recent publication (*23*) decomposes a network into strongly connected components (SCC) in an ordered way, displaying the underlying pipeline structure at the largest scale. To look into each SCC of interest, we design a new scheme which first searches for the embedded cycles and then determines the feedbacks by a selection procedure. By combined use of the HVD and feedback selection, it is possible to identify the forward edges and feedbacks at all scales. In addition, by properly cutting certain feedback edges, we obtain a skeleton network with the dominant agents and key interactions identified, as well as their ordering and underlying topological structures, which is called a minimal production unit (MPU).

In the following, we will use NFκB as an example to explain our graph theoretic analysis procedure and display the production-controler structure. The chemotaxis network of *E. coli* will be analyzed to further show this universal topological-dynamical structure. More examples will be given in the appendix.



(a) (b) (c)

Figure 1: The diagram of a model of the NF-κB signaling module.

The NFκB regulatory module concerns the switching dynamics of the nuclear factor NFκB, which regulates various genes important for pathogen or cytokine inflammation, immune re-

sponse, cell proliferation and survival (*24, 25*). In the cytoplasm of a resting cell, NFκB usually binds to IκBα and its activity is suppressed. Certain external signals activate the switch protein IKK which phosphorylates IκBα such that NFκB is released (*26*). The free NFκB then translocates into the nucleus and initiates the transcription of a large set of proteins, including protein IκBα and protein A20. Protein IκBα, once synthesized in the cytoplasm, enters the nucleus, binds to NFκB, transports it out to the cytoplasm again and thus terminates the transcription. Protein A20 deactivates IKK. Therefore, the module mainly consists of two forward proteins IKK and NFκB and two feedback proteins IκBα and A20. Also, the translocation of the proteins between the nucleus and the cytoplasm is an important biological process that achieves the spatial localization of different protein species.

The diagram of a detailed model of the NFκB regulatory network is shown in Fig. 1(a) where we use x_i's to represent the concentration of various proteins. The associated chemical kinetic model is given and explained in Appendix A. With physiological initial conditions (*27*), the concentration of the nuclear NFκB changes as a damped oscillation as shown with the thin dotted curve in Fig. 1(c). At the beginning, it shoots up to a very high value in a short time and then relaxes to a much lower steady value in an oscillatory way.

For any networked system described by certain dynamical equations, it is easy to write an interaction graph with the vertices representing the reacting agents and the edges directed from each agent to the ones influenced by it. The interaction graph for the NFκB model is shown in Figure 1(b).

For clarity, we omitted the self loops which represent self-interactions. It is straightforward to write down the adjacency matrix for the interaction graph, which marks 1 at the entries corresponding to connected edges and zero otherwise. The interaction graph and the adjacency matrix neglect details of the interactions and only maps out the network topology which holds almost everywhere in the phase space and the parameter space, except for a set of measure

zero (*23*). This great robustness confers great flexibility to characterize vastly different dynamics described by ODEs or mappings or even stochastic equations. Certain system properties, like the uniqueness of the stationary point sometimes can be deduced from pure topological consideration of network structures (*28, 29*). So, understanding of structure of interaction graphs helps unveil the essential elements in a complex system which possibly has uncertainties in the parameter values or is influenced by a noisy environment. Graph theoretic techniques will be developed here to decompose the interaction graph and disclose its generic structure used to realize its biological function.

The horizontal-vertical decomposition (HVD) of an interaction graph has been contrived and discussed in a recent paper (*23*). Vertically, the system is decomposed into a linear series of modules, where the module above is influenced by the module below but not vice versa. So, the input signal propagates unidirectionally from the bottom to the top. Horizontally, each module is decomposed into independent groups with no connection between them. So, each group has its own input and output and are functioning relatively independently. The direct application of the HVD to teh interaction graph in Fig. 1(b) results in three layers with the top and bottom layer consist of the vertex sets $\{x_1\}$ and $\{x_3, x_{15}\}$, respectively. The rest vertices are strongly connected and belong to the middle layer. This type of structure with dominant intermediate processing unit exists for most of biological networks as a result of omnipresent feedback loops and reversibility of many biochemical reactions.

Further analysis is needed to determine the forward and backward edges for the processing unit in the middle layer. First, its polarity is easy to identify. The vertex x_{15} is the output signal that is of interest while x_1 receives the external input. Therefore, in the middle layer, x_2 is the input vertex and x_7 is the output one. They should stand at the beginning and the end of the middle module. Next, we notice that in a strongly connected component (SCC), the forward and backward edges always make cycles and vice versa every cycle contains at least one forward

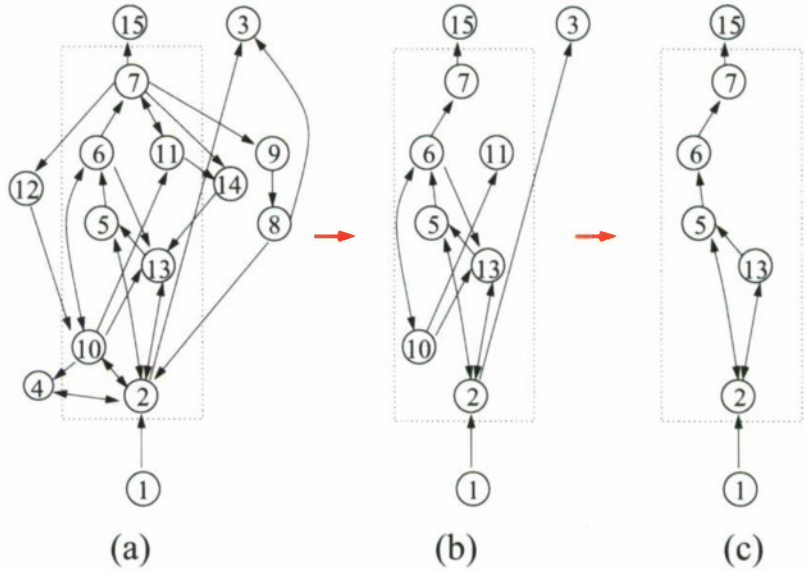


Figure 2: The structure decomposition and extraction of the MPU of the NF-κB signaling regulatory network. (a) the structured diagram derived from the graph theoretic analysis; (b) with feedbacks removed; (c) with irrelevant vertices removed.

and one feedback edge. Since cycles are obvious topological invariants of a network and easy to seek, our strategy consists of two steps: first, search for all cycles that exist in the graph; second, determine the feedbacks through a selection procedure, which depends on the information flow direction and approximately minimizes the number of feedbacks. The detailed illustration of our technique is contained in Appendix B. Here we only show the result of the computation in Fig. 2(a).

From Fig. 2(a), we see that there are 4 feedback loops:

- FB_a - the one through vertex 4: IKKα associates with free IκBα and catalyzes its decay.
- FB_b - the one through vertex 14: IκBα_n captures NFκB$_n$ to form (IκBα-NFκB)$_n$, which then moves out of the nucleus.
- FB_c - the one through vertex 12: NFκB$_n$ promotes the production of the IκBα mRNA

which translocates to the cytoplasm and initiates a burst of IκBα production.

- FB$_d$ - the one through vertices 8 and 9: NFκB$_n$ promotes the production of the A20 mRNA and thus initiates the production of A20, which catalyzes the decay of IKK$_a$.

This identification agrees very well with the usual recognition of feedback loops in the literature based on biological reasoning. In Fig.2(a), all the feedback loops $FB_{a,b,c,d}$ have at least one edge attached to the input or the output vertex, so at least 4 cuts are needed to remove these feedbacks. Also, there are 7 extra 2-cycles in the graph, which require 7 more cuts. So, for the NFκB model, at least 11 cuts are required to remove all the feedbacks. With our algorithm, we successfully identified a minimal cutting set. Those big cycles are uniquely determined while the uncertainty of the decomposition comes from the vertex set $S_m = \{x_5, x_6, x_{10}, x_{11}, x_{13}\}$ connected by those 2-cycles exclusively.

We emphasize that we acquired the network structure by an automatic procedure based on the graph decomposition. So, the method developed here can be potentially applicable to real complex networks in cell signal transduction or gene regulation. It is also understood that because of the existence of binary or dissociative reactions, the rates represented by some edges are correlated since they symbolize the same reactions. In the above consideration, we ignored this correlation and carried out our analysis from a pure graph theoretic point of view. Further refinement of the ordering needs to incorporate these detailed reaction information, as shown below.

After the structured network is constructed as in Fig. 2(a), it is very convenient to Extract the minimal production unit. In the case of signal transduction network, the minimal production unit is the minimal subgraph of the original network such that the response to external stimuli continues to be produced though its value may not be correct. It obviously depends on the initial state of the system and on the specific response under investigation.

In the NFκB network, there exists a particularly simple procedure to extract the MPU after we identified the forward and backward edges. First, the feedbacks and the associated reactions are removed as we now only consider the forward production part. As a consequence, $\{x_4, x_8, x_9, x_{12}, x_{14}\}$ are removed, which results in Fig. 2(b) where the correlation among edges have been considered. Second, all the outputs except the one we are interested in are removed. That is, $\{x_3, x_{11}\}$ are removed. From here we see that the final MPU indeed depends on what signal we are looking at. Finally, remove other irrelevant vertices. It is x_{10} here since its initial value is zero and does not receive any contribution from other vertices or the environment. The arrow from x_6 comes from a binary reaction and x_{10} can not be produced there. At the end of the day, we produce the MPU depicted in Fig. 2(c). It contains the vertex set $s_m = \{x_1, x_2, x_5, x_6, x_7, x_{13}, x_{14}\}$, while all other vertices can be regarded as functioning controllers.

To check if what we got in Fig. 2(c) is indeed the MPU, we keep only the variables in the vertex set S_m and their interactions in the evolution equation. Numerical simulation of this reduced set of equations produced an output curve depicted with the thick solid line in Fig. 1(c), which displays a fast relaxation to a large value. It is interesting to note that the saturation value and the relaxation time are very close to those of the first oscillation peak of the full equationa. The vertex set S_m constitute the MPU of the NFκB gene regulation network, which generate a quick and large response to the external signal. The rest vertieces act as controllers to bring down the initial pulse to a desired steady value in a much longer time scale. Both the short and the long time response bear important biological significances (*25*).

In Ref. (*25*), it is pointed out that a two-component system with a negative feedback exhibits the basic oscillating or saturation behavior depending on the parameter values. Here, the two components are replaced by two subnetworks residing in cytoplasm and nucleus, respectively. The forward edges and feedbacks are realized by the translocation of several different

molecules. As extra resources are needed to produce these regulatory molecules, the question why the extra feedback loops and intermediate steps exist is naturally raised. External signals could have directly acted on NFκB to control its activity. Those extra features in network structure and interaction dynamics bring about extra robustness and sensitivity to the network for fulfilling its basic function (*30*).

Lying at the core of this regulatory network is the association and disassociation of NFκB with IκBα. When binded, they stay away from the nucleus and the NFκB-initiated transcription is terminated; when they separate, NFκB tends to enter the nucleus and starts prompting transcription. The feedback FB_b mainly facilitates the step of clearing NFκB out of the nucleus.

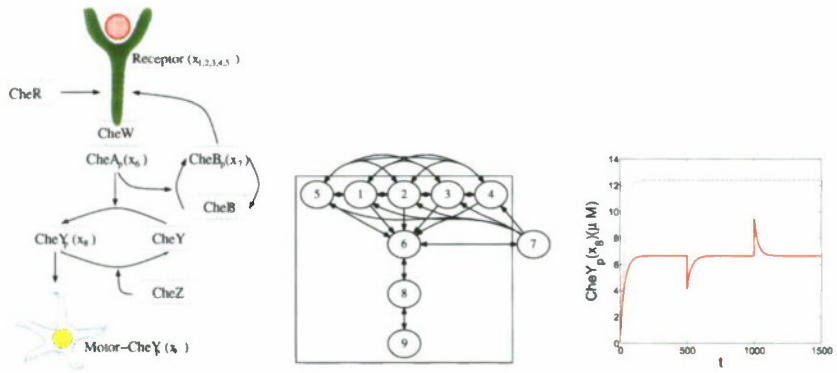
The feedbacks FB_c and FB_d are two negative feedbacks. FB_c is to restore the concentration of IκBα that has been consumed by the IKK_a-catalyzed decay. FB_d is to deactivate IKK_a by *A*20 to bring down the activation level of the whole network. Like other feedback signaling from the output (*4, 31*) it provides adaptivity and sensitivity. When a signal such as TNF just arrives, IKK_n constantly gets activated into IKK_a while the deactivation of IKK_a is minimized since its constitutive decay rate is small. So, the concentration of IKK_a will rapidly increase until A20 is produced by the feedback loop and starts the catalyzed decay of IKK_a. The forward reaction rate is thus maximized transiently and enables cell response to signals with short duration (*25*). The negative feedback will eventually bring down the IKK_a concentration to a steady level which is much lower than the transient peak. So, the network has a very sensitive and fast transient response, which is essential for certain signaling pathways (*25*).

The oscillation observed in this process is a signature of trading stability for sensitivity (*21*). It confers easy excitability to the network while brings oscillations at the same time.

The reaction rates of all the biochemical processes are to some extent influenced by the environment variables like temperature, pH value, concentration of certain ions (*32*). To work properly under different conditions, the chemical network should possess structural stability.

Here the double feedbacks FB_c and FB_d offer extra structural stability against parameter uncertainty. If the parameter changes incur an temporary increase of the concentration of $NF\kappa B_n$, then both FB_c and FB_d will act to bring it down. Even when the rate of FB_c or FB_d changes, the other feedback will try to minimize the effect. Therefore, the double feedbacks act like a double safe for keeping the system stable under parameter fluctuations.

The above procedure of searching for MPU could be easily generalized to more complex networks. The critical step lies in our capability of detecting the feedback loops. Once the feedback controllers are decided, the MPU is obtained by removing all the feedbacks and then all the irrelevant outputs and inputs. The exhibition of fast forward production and the slow feedback controller is also quite universal as demonstrated by the next example of *E. coli* chemotaxis network, as well as those in Appendix C.



(a) (b) (c)

Figure 3: (a)The chemotaxis model of *E. coli*. (b) Feedback and forward structure through graph decomposition. (c) The response of $CheY_p$ to the external cue of the full network (thick solid line) and the MPU (thin solid line), ligands added at $t = 500$ and removed at $t = 1000$.

The chemotaxis of E. coli is regulated by its chemotaxis pathway. Chemoattractants binds to and activates the transmembrane receptors, which stimulate CheA through the adaptor CheW. Activated CheA phosphorylates CheY, which binds to the flagellar motor and increases the fre-

quency of *E. coli* tumbling. The activation of the receptor complex is controled by its methylation states. Higher methylation states indicate higher probability to be activated. In the model, CheR binds only to the inactive receptors to increase methylation and phosphorylated CheB only to the active receptors to decrease methylation.

The chemotaxis model is shown in Fig. 3(a). Fig. 3(b) displays its feedback and forward structure upon application of graph decomposition. The first level constists of the vertex set $\{x_1, x_2, x_3, x_4, x_5\}$ which are different methylation states of the receptor complex. External signals propagate down through x_8, x_8 and finally reaches the flagellar protein x_9. There is one feedback vertex through x_7 (CheB$_p$). The minimal production unit (MPU) is obtained after all the reactions involving x_7 is removed and is contained in the box in Fig. **??**(b).

With the feedback through CheB$_p$ (x_7), the system has sensitive detection and robust adaptivity as shown with thick solid line in Fig. 3(c). Starting with zero value , the CheY$_p$ quickly reaches the saturation level. At $t = 500s$, an external stimulus - $10\mu M$ concentration ligand is supplied, which induced a jump of CheY$_p$ concentration followed by an exponential decay back to the saturation value. At $t = 1000s$, the ligand is removed which triggered a drop of CheY$_p$ concentration but regained its stable value exponentially fast. When the feedback is removed, the MPU retains the stable value after a quick initial rise no matter how the concentration of external ligand changes. The robustness is retained but the sensitivity is lost. So, in this example the feedback is essential for system's transient response to external stimulus and guarantees the sensitivity. As in the previous example, the forward production reacts quickly accounting for the sensitivity of the network while the controller effects at a longer time scale to realize the adaptivity. This has been observed over again in several other most commonly encountered biological networks presented in Appendix C.

We devised an automatic procedure to identify the key functional units of cell regulation networks by applying graph theoretic method and further dynamical systems analysis. We

identified the forward production pipeline and feedback controller in the NFκB regulatory network with the HVD and the feedback loop searching algorithm. They form disparate functional modules and can be analyzed further with the same program if necessary. Based on the genericity and topological nature of the anatomy of the NFκB network, it is reasonable to believe the existence of similar structures in other cell regulatory networks.

The detection of modular structures provides additional insight how a regulatory network works and thus gives clear indication of key protein species and key reactions in a cascade, which finds vast applications in the drug design and synthetic biology (*33*). The disclosure of the dominating skeleton subnetwork in a regulatory pathway also avails the determination of reaction rates of *in vivo* biochemical reaction since the distracting unimportant reaction components have been removed from the skeleton structure and the Jacobian matrix becomes more balanced (*34*).

With the help of the current program, the analysis of large-scale network possibly gains much efficiency by building the hierarchical connection among different scales. In the top-down direction, the network may be broken into functional modules at different scales by the above decomposition technique. From bottom up after the property of each module is conveniently explored, a hierarchy of modules of increasing size may be built until the whole network is covered. This is a topological generalization of multi-scale analysis (*35*) to networked systems. Future application of the decomposition and the modular analysis technique could go far beyond biological networks.

This reduction is closely related to the problem of determining rates of *in vivo* chemical reactions that sensitively depend on the cell conditions and are hard to measure directly. They are usually estimated based on the response curves of the network, with known interaction graphs. A generic system turns out to be very sensitive to some parameters while inert to others (*36*). This observation is pertinent to the structural and functional redundancy commonly found in

biological networks. In order to maintain similar biological function under possibly vastly different environment, a typical cell regulatory network contains much more extra pathways than is needed for the basic activity in normal conditions. In a particular experiment, only part of the network properties are measured, which is likely to be carried out only with part of the network so the rest seems redundant and the corresponding parameters are inert. Our current approach could be used to determine these inert parameters and thus disclose the skeleton of the network relevant to the experimental probation. The part of the network represented by the inert parameters can be viewed as adaptive gears which take effects only under stress conditions and are thus negligible in normal conditions. In the following, we analyze the dynamical system Eq. (1) to excavate the inert parameters and hence pin down the major part of the network.

APPENDIX

A Chemical kinetic equation of the NFκB network

Deterministically, the dynamics of the NFκB network is described by a set of chemical kinetics equations:

$$
\begin{aligned}
\dot{x_1} &= k_{prod} - k_{deg}x_1 - k_1 u x_1 \\
\dot{x_2} &= k_1 u x_1 - k_3 x_2 - k_2 u x_2 x_8 - k_{deg} x_2 - a_2 x_2 x_{10} + t_1 x_4 - a_3 x_2 x_{13} + t_2 x_5 \\
\dot{x_3} &= k_3 x_2 + k_2 u x_2 x_8 \\
\dot{x_4} &= a_2 x_2 x_{10} - t_1 x_4 \\
\dot{x_5} &= a_3 x_2 x_{13} - t_2 x_5 \\
\dot{x_6} &= c_{6a} x_{13} - a_1 x_6 x_{10} + t_2 x_5 - i_1 x_6 \\
\dot{x_7} &= i_1 k_v x_6 - a_1 x_7 x_{11} \\
\dot{x_8} &= c_4 x_9 - c_5 x_8 \\
\dot{x_9} &= c_2 + c_1 x_7 - c_3 x_9 \\
\dot{x_{10}} &= -a_2 x_2 x_{10} - a_1 x_6 x_{10} + c_{4a} x_{12} - c_{5a} x_{10} - i_{1a} x_{10} + e_{1a} x_{11} \\
\dot{x_{11}} &= -a_1 x_7 x_{11} + i_{1a} k_v x_{10} - e_{1a} k_v x_{11} \\
\dot{x_{12}} &= c_{2a} + c_{1a} x_7 - c_{3a} x_{12} \\
\dot{x_{13}} &= a_1 x_6 x_{10} - c_{6a} x_{13} - a_3 x_2 x_{13} + e_{2a} x_{14} \\
\dot{x_{14}} &= a_1 x_7 x_{11} - e_{2a} k_v x_{14} \\
\dot{x_{15}} &= c_{2c} + c_{1c} x_7 - c_{3c} x_{15} \,, \qquad (1)
\end{aligned}
$$

where the notation is adopted from the reference (*27*) and marked in Fig. 1(a). Protein IKK has three different forms, the neutral form IKK$_n$ (x_1), the activated form IKK$_a$ (x_2) and the deactivated form IKK$_i$ (x_3). IKK$_n$ is passive but can be activated into IKK$_a$ by external cues like TNF or IL-1. $u \in \{0, 1\}$ is a switch variable which is equal to one when the external

cue is present but equal to zero otherwise. IKK_a is deactivated by A20 (x_8) into IKK_i (x_3) which is different from IKK_n and cannot be activated. IKK_a is able to bind with IκBα (x_{10}) to form the protein complex IκBα-IKK_a (x_4) or with the complex IκBα-NFκB (x_{13}) to form a tri-molecular complex IKK_a-IκBα-NFκB (x_5). Once IKK_a binds with IκBα, in either the bimolecular or the trimolecular form, it phosphorylates and initiates proteolysis of IκBα so that NFκB (x_6) is released from the complex and restores its enzymatic capability, entering the nucleus and being denoted by NF$\kappa$$B_n$ (x_7). When NF$\kappa$$B_n$ binds to the relevant promoter regions in the DNA, a great variety of transcriptions are initiated. The transcripts (x_9) for A20 and the transcripts (x_{12}) for IκBα move to cytoplasm and start synthesis of the corresponding proteins. Other interesting transcripts (x_{15}) for certain signaling proteins may be generated as well. The newly synthesized protein IκBα will enter the nucleus, wherein denoted by IκBα_n (x_{11}), and bind with NF$\kappa$$B_n$ to form the complex IκBα-NF$\kappa$$B_n$ (x_{14}), leaving the nucleus for the cytoplasm. Thus, the transcription is temporarily terminated by the association with protein IκBα. The switch protein IKK_a itself is also constantly deactivated by A20. In the mean time, almost all the proteins and transcripts decay spontaneously, and the bindings and unbindings all have constitutive reaction rates which are much smaller than the corresponding enzymatic reaction rates.

B Identifi cation of forwad and feedback edges

In this appendix, we will design an algorithm to identify the forward and feedback edges with given polarity. First, a cycle search program is discussed which produces all the cycle generators for a strongly connected component. Then a selection procedure is discussed which generates a partial order of the vertices and enables the detection of feedbacks in a straightforward way.

B.1 Principle of minimum feedbacks

Very often, in engineering systems, multi-step processes are carried out in a well-ordered sequential way with a small number of feedback controllers modulating the behavior of the system. The cascade structure yields both robustness and evolvability to a networked system. It also has the advantage of maximizing operation efficiency and minimizing energy cost by adjusting magnitude of the feedback control. As an analogue, we propose that a minimum feedback principle is also utilized by the cell: the feedback edges should be a minimum set in a biochemical network like cell regulation networks.

How to find a minimum set of edges is an NP-hard problem in graph theory but there exist approximate algorithms which could do the job relatively fast. It is conceivable that the solution might not be unique.

In the signal transduction network, extra constraints are imposed. From a control theory point of view, the signal transduction network consists of two major components, the information forwarding part and the feedback controller. The forward part receives external signal at one end, pass and process it along different paths, produce an output at the other end. So, the associated information flow defines a direction on the network. The feedback component controls the flow by sending downstream signals back to upstream nodes. The identification of these two components is essential for understanding the function of individual parts of a network. So, the problem of searching for the minimal set of feedback arcs in this new context has to be consistent with the constraints brought up by these extra features. The problem could be put in an equivalent way: find an ordering of the vertices with the input and output vertices sitting at opposite ends, such that the number of feedback edges is minimized. To achieve this goal, in the following, the HVD and the cycle searching and selection techniques are discussed and applied to the graph in Fig. 1(b).

B.2 Cycle search

For a finite graph, all cycles can be obtained by algebraically combining a set of independent cycle generators. For an SCC, all the edges are included in the generator set C_{gen} since there always exists at least one cycle passing any edge and thus each edge is contained in at least one generator. In the following, we introduce a searching-collapsing scheme to find the generators of all the cycles of a general graph $\mathscr{G}$ with the adjacency matrix A. The main idea is to identify shortest cycles and then simplify the graph in an iterative way:

(1) Record all the self-loops of $\mathscr{G}$ which are encoded by the nonzero diagonal elements of A. After removing the corresponding edges from $\mathscr{G}$, we obtain a new graph $\mathscr{G}_1$ and a new adjacency matrix A_1.

(2) Search and record a shortest m-cycle $l_1=\overline{[a_1, ..., a_m]}$ of A_1 where a_i's represent vertices of $\mathscr{G}$ by looking for the nonzero diagonal elements of the powers of A.

(3) The induced subgraph $\mathscr{H}_1$ by the vertex set $\{a_1, ..., a_m\}$ has an adjacency matrix B_1 which is a submatrix of A_1. Each nonzero element (i, j) of B_1 represents an edge from the vertex a_j to the vertex a_i which can be made to a cycle by connecting a_i back to a_j with part of the cycle l_1, *e.g.*, by the chain of edges $[a_i, a_{i+1}, ..., a_j]$. So for each edge in $\mathscr{H}_1$ but not in l_1, we can identify and record a cycle.

(4) Collapse all the edges and vertices in the subgraph $\mathscr{H}_1$ into one point P_1, and we obtain the updated graph $\mathscr{G}_2$ for which a new adjacency matrix A_2 is written down. If $\mathscr{G}_2$ only contains P_1, the iteration is terminated. Otherwise, we go back to step (2) and repeat the procedure with the new graph $\mathscr{G}_2$ and the new adjacency matrix A_2.

It is easy to show that each cycle of $\mathscr{G}_1$ corresponds to a unique cycle either in $\mathscr{H}_1$ or in $\mathscr{G}_2$. Vice versa, each cycle l in $\mathscr{G}_2$ can be identified with a unique cycle in $\mathscr{G}_1$: if the cycle l runs through P_1, then its entering vertex and leaving vertex in $\mathscr{H}_1$ can be connected by a unique path embedded in the cycle l_1 and thus a unique cycle in $\mathscr{G}_1$ is produced by concatenating this path to

the edges contained in l; if the cycle l stands apart from P_1, it directly corresponds to one cycle in $\mathscr{G}_1$. So, after the search is done, finally, we can trace backward all the cycles contained in the original graph $\mathscr{G}$. Here, we only recorded the set C_{gen} of cycle generators. As we pointed out earlier, however, their linear combinations are able to produce all the cycles and so cover the whole transitive part of the graph. The generators derived from the above algorithm are prime in the sense that any proper subset of a generator is not a cycle. Note that the set C_{gen} is not unique since the selected cycle in step (2) is not unique, but the number of cycle generators is a constant which depends only on the graph itself. The important point here is that all the feedback edges appear at least once in C_{gen} as argued above.

To the NF-κB gene regulatory network, we apply the cycle-searching technique and find that the total number of cycle generators are 33 with 15 1-cycles and 8 2-cycles. Therefore, 10 cycle generators has length longer than 2.

B.3 Selection procedure

In order to determine the forward and feedback edges from the cycles found in the previous section, we utilize the polarity of the network that has the input and the output points. We take the middle layer of the NFκB network from the HVD result as an example. Here, the input point is x_2 as it receives signals from x_1 and the output point is x_7 as it sends signals to x_{15}. The goal is to find all the forward paths that go from x_2 to x_7 and all the feedback loops. The problem is clear conceptually but not rigorously defined mathematically since different choices may lead to different forward and feedback sets. Here, we provide one choice that seems reasonable. After adding a direct connection $(2\,,7)$ from x_7 to x_2, we apply the cycle searching technique developed above and identify a set of cycle generators.

For a graph with tree structure, it is always possible to find an ordering of the vertices, such that feedback edges do not exist. For instance, the HVD could generate such an ordering.

With cycles present, at least one feedback exists no matter how the vertices are ordered. Here, we implement the principle of minimal feedbacks to extract the minimal set of the edges the removal of which leads to generation of tree structures in the network. Therefore, by a greedy algorithm it is tempting to cut those edges that are common to most number of cycles. It is not uncommon that by removing one edge quite a few cycles get detroyed. 1-cycles (self-loops) and 2-cycles are special and need to be treated differently. 1-cycles always attach to the corresponding vertices and are not regarded as feedback loops. the 2-cycles correspond to bidirectional edges most likely representing the forward and backward reactions since many of biochemical reactions are reversible. These 2-cycles are important for keeping the chemical balance but not to be regarded as feedbacks from a signal transduction point of view. Each 2-cycle contributes exactly one feedback edge. They have to be cut one by one irrespective of the ordering of the vertices. Therefore, they have no impact on the vertex ordering regarding the search of minimal set of feedbacks. Hence, we consider only cycles of length larger than or equal to 3 when ordering the vertices. In a graph possessing polarity, for the input vertex, every out-edge is regarded as a forward edge and every in-edge a feedback. The opposite is true for the output vertex.

With the long cycles having been determined in the cycle search program, we first look for those passing the edge $(2,7)$ and thus obtain a set of forward paths that go from x_2 to x_7. Then, from the remaining long cycles, we search for cycle segments that parallel to these already obtained forward path segments so that alternative paths from x_2 to x_7 are constructed. We repeat the search until no more alternative paths can be generated from the available long cycles. Now, we have a subgraph $\mathscr{F}$ expanded by the vertices and the edges contained in these forward paths. For the NFκB regulation network, the vertex set in $\mathscr{F}$ has been computed as $V_f = \{x_2, x_5, x_6, x_7, x_{10}, x_{11}, x_{13}\}$ which are displayed inside the rectangle in Fig. 2(a); the complementary vertex set consists of $V_b = \{x_4, x_8, x_9, x_{12}, x_{14}\}$. They do not belong to the

forward pipeline from x_2 to x_7, so they must be included in the feedback motifs. Next, we partially order all the vertices and edges according to the determined forward path.

First, the HVD is applied to $\mathscr{F}$. We get 4 layers and 7 groups with one vertex in each group since there is no feedback edge in $\mathscr{F}$. So, from x_2 to x_7, we generated a partial ordering by the layered structure. The vertices in each layer are not ordered. We rearrange the order of the vertices in $\mathscr{F}$ according to the partial order and append the vertices in V_b to the end. Using this ordering for the SCC, a new adjacency matrix A_t is generated. All the subdiagonal entries stand for forward edges while the entries above the main diagonal indicate backward edges. We can further identify all the 2-cycle edges by checking the diagonal-symmetric partners of nonzero superdiagonal entries. If the subdiagonal partner is also nonzero, then together they stand for a 2-cycle. If the partner is zero, then we have a long feedback loop, the collection of which are clearly exhibited by the vertices and the edges outside the rectangle in Fig. 2(a).

C Examples of Several other cell regulation networks

C.1 B. Subtilis chemotaxis network

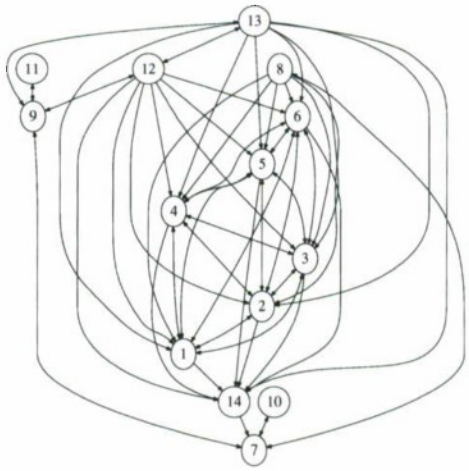


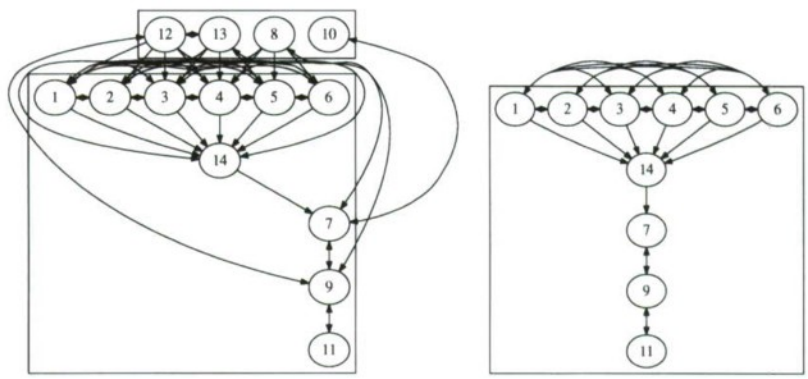
Figure 4: Network representation of the chemotaxis model of *B. Subtilis*.

x_1	T_{20}	x_6	T_{11}	x_{11}	Motor $-$ $CheY_p$
x_2	T_{10}	x_7	$CheA_p$	x_{12}	R_Y
x_3	T_{00}	x_8	$CheB_p$	x_{13}	RY_p
x_4	T_{01}	x_9	$CheY_p$	x_{14}	R^A
x_5	T_{02}	x_{10}	$CheV_p$		

Table 1: T_{ij} denotes six different methylation states of the receptor dimer where the index i and j mark the methelation states of residue 630 and 637, respectively. CheABVY denote different kinases in chemotaxis signaling. RY_p, R_Y denote receptors with and with no $ChemY_p$ binding, while R^A denotes activated recetors.

The chemotaxis signaling network of *B. subtilis* retains many features of that of *E. coli* but also has varied considerably. The model is adapted from (*37*). See Fig. 4 for the interaction graph and the notation is explained in Table 1. Here, in a similar way, the receptor (T) adopts different configurations (active, inactive, weakly active, weakly inactive) according to its methylation states and external ligand concentration. However, the activation and deactivation of the whole receptor complex (R) also depends on the binding of $CheY_p$. The active receptor complex can activate CheA which in turn activates CheY. The phosphorylated CheY binds to the flagellar protein and enhances straight runs of the bacterium. It is also assumed that $CheY_p$ deactivates CheA. $CheA_p$ also activates CheB which adjusts the methylation state of the receptor. A new inhibitor CheV exists in *B. subtilis* network which disrupts the receptor complex when unphosphorylated but can be deactivated by $CheA_p$ phosphorylation.

Through the graph decomposition discussed previously, three main feedbacks are unambuguously detected and displayed in Fig. 5(a) for the *B. Subtilis* chemotaxis network. One is through x_8 ($CheB_p$) which is activated by x_7 ($CheA_p$) and changes the methylation state of the receptor ($x_{1,\cdots,6}$). The feedback through x_{10} ($CheV_p$) is the new controller of the receptor complex while the feedback through x_{12} and x_{13} relays the action of $CheY_p$ (x_9) to regulate the activation and inactivation of the receptor complex. The first six variables $x_{1,\cdots,6}$ representing different methylation states are interconnected, making up a complete subgraph and describing



(a) (b)

Figure 5: (a) The feedback and forward structure through graph decomposition. (b) The MPU of the *B. Subtilis* chemotaxis network.

their interdependence. All of them affect the activation of the whole receptor complex (x_{14}).

We checked the dynamics by using the kinetic equations and parameter values in (*37*). Fig. 6 shows the change of x_9 (CheY$_p$) in the time interval $[0\,,1500]$, starting with $x_9 = 0$, $10\mu M$ attactant added at $t = 500$ and removed at $t = 1000$. When the full network is put to work, the adaption is clearly exhibited in Fig. 6(a) by the elevation of x_9 during $[500\,,1000]$ in the presence of the chemo-attractant. When the feedback back is cut, we get in Fig. 6(b) a stable value of x_9, which does not change with the attractant concentration. So the subgraph shown in Fig. 5(b) is an MPU, which only provides the basic supply of CheY$_p$, devoid of capability of adaptation. Also noticeable is the fast response produced by the MPU and the relatively slow adaptation process controled by the feedbacks.

Comparison of Fig. ?? and 5 reveals the similarity and difference between the chemotaxis pathways of *E. coli* and *B. subtilis*. The forward MPU parts of both pathways are almost identical. Nevertheless, the feedback regulator in *B. subtilis* has extra loops and layers which may provide more subtle control (*37*).

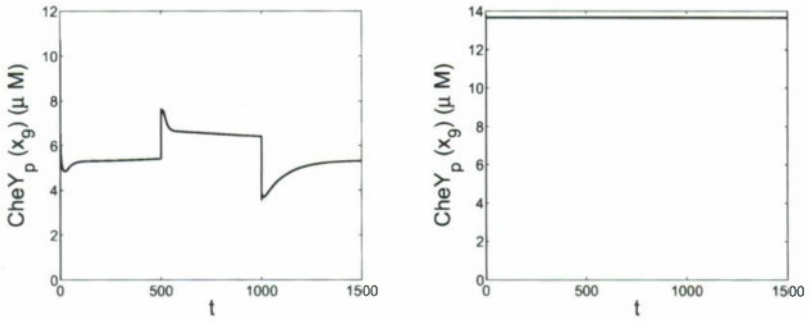


Figure 6: The concentration change of CheY$_p$ (x_9) of the chemotaxis network of *E. coli* (a) with and (b) without feedbacks. External attractant of $10\mu M$ is added at $t = 500$ and removed at $t = 1000$.

C.2 Survival and apoptotic pathways initiated by TNF-α

This model studies the survival and apoptotic pathways initiated by TNF-α and we adopt it from (*38*), which play decisive roles in cell fate decision in response to inflammation and infection. After the external cue TNF-α binds to its receptor TNFR1 (x_2), adaptor proteins TRADD, TRAF2 and RIP-1 are recuited to form an early complex ready for binding and activating other functional proteins. There are two different downstream pathways: the survival pathway mediated by NF-κB and the apoptotic pathway mediated by caspase. NF-κB is usually sequestered by IκB and is released when IκB degrades. IKK binds to the early complex to form a survival complex and is activated with the dissociation of this complex. The activated IKK is able to induce proteolysis of IκB. The released NF-κB translocate to the nucleus, binds to DNA and leads to the transcription of IAP and IκB. c-IAP inhibits apoptosis by binding to caspase-3* and thus preventing DNA fragmentation. The notation is detailed in Table C.2.

Upon application of the graph decomposition routine, we unfold the successfully unfold the underlying modular structure of the TNFα network. The forward production unit is a long cascade involving many different species and reactions. The signal TNFα (x_1) is processed through

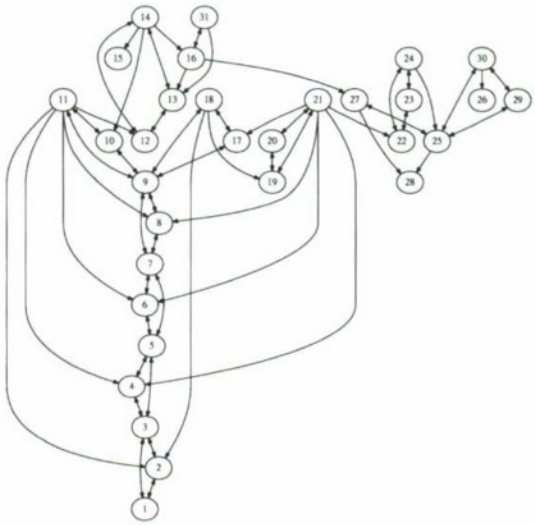


Figure 7: Network representation of the Survival and apoptotic pathways initiated by TNF-α.

the network until DNA fragmentation is induced (x_{26}) as shown Fig. 8(a). The direct HVD identifies one big SCC enclosed in the two boxes in Fig. 8(a). Further analysis distinguishes the forward and backward edges. The whole NFκB pathway is now treated as a feedback back module, which controls the level of the c-IAP (x_{27}) and thus Caspase-3* (x_{25}), and maintain the option for survival. It is intriguing that the NFκB module is produced automatically by our decomposition procedure althougth it has many connections to the rest of the network. The removal of the NFκB module singles out the MPU.

Fig. 9 shows the level of DNA fragment (x_{26}) with or without the presence of the NFκB control module. With the feedback module, the fragmentation of DNA is low (Fig. 9(a)), which may suggest the survival of the cell; without, the DNA cleaverance is high (Fig. 9(b)), which could indicate an apoptotic fate of the cell. So, indeed, here the NFκB modules acts as a controller of the apoptotic pathway. Our decomposition technique accurately captures this information. Again, without the control module, the MPU produces over abundantly the output signal in a relatively fast way. The long feedforward edge from x_{16} to x_{27} may accelerate the

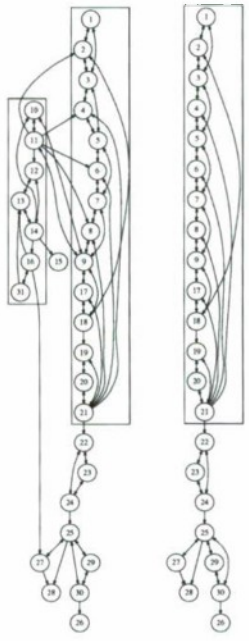
x_1	TNFα	x_{17}	FADD
x_2	TNFR1	x_{18}	$\langle x_7 \rangle$/RIP1/FADD
x_3	TNFα/TNFR1	x_{19}	TRADD/TRAF2/RIP1/FADD
x_4	TRADD	x_{20}	Caspase8
x_5	TNFα/TNFR1/TRADD	x_{21}	TRADD/TRAF2/RIP1/FADD/Caspase8
x_6	TRAF2	x_{22}	Caspase8$*$
x_7	TNFα/TNFR1/TRADD/TRAF2	x_{23}	Caspase3
x_8	RIP1	x_{24}	Caspase8 $*$ /Caspase3
x_9	$\langle x_7 \rangle$/RIP1	x_{25}	Caspase3$*$
x_{10}	IKK	x_{26}	DNA $-$ frag
x_{11}	$\langle x_7 \rangle$/RIP1/IKK	x_{27}	cIAP
x_{12}	IKK$*$	x_{28}	Caspase3 $*$ /cIAP
x_{13}	IκB/NFκB	x_{29}	DNA
x_{14}	IκB/NFκB/IKK$*$	x_{30}	Caspase3 $*$ /DNA
x_{15}	IκBP	x_{31}	IκB
x_{16}	NFκB		

Table 2: TNFα is one tumor necrosis factor which binds to the receptor TNFR1. TRADD, TRAF2 and RIP1 are adaptor proteins which may form complexes with TNFα. IKK, NFκB, IκB belongs to the NFκB module while FADD, caspase8 and caspase3 are on the apoptotic pathway. c-IAP is an inhibitor of apoptosis protein.

control in this case.

C.3 Circadian clock in *Drosophila*

Circadian clock exists in many different organisms ranging from bacteria to human. The regulation pathway adopted from (*39*) and displayed in Fig. 10 models the *Drosophila* circadian clock which mainly contains two interlocked loops. The notations are explained in Table 3. The TIM and PER protein in the first loop may bind to each other in the cytosol or nucleus, but they enter the nucleus separately. They down-regulate their own expression by inhibiting the transcription factor CLK-CYC. The association of TIM and PER in the cytoplasm is mediated by FBM and the dissociation is catalyzed by SM which is generated by the constitutive entering of PER into the nucleus. In the second loop, CLK-CYC activates both VRI and PDP expression.



(a) (b)

Figure 8: (a) Feedback and forward structure through graph decomposition. (b) The minimal production unit of the TNFα network.

VRI represses the expresson of CLK while PDP promotes.

The network graph after the decomposition analysis in Fig. 11(a) clearly shows 5 feedbacks. The one through SM (x_{11}) is the positive feedback that accelerates the dissociation of the PER·TIM complex. The other four are the important regulators of the concentration of PER, TIM, VRI and PDP through DNA expression and protein translation. The feedbacks through x_{12} and x_{13} interact with each other and control the production of CLK (x_{19}). The MPU is very easily obtained by removing the feedback modules and displayed in Fig. 11(b), which indicates how the (sunlight) signal is picked up at x_4, processed via PER·TIM, CLK·CYC interaction and output at x_{21}. With all the feedbacks, the *Drosophila* network is able to generate stable oscil-

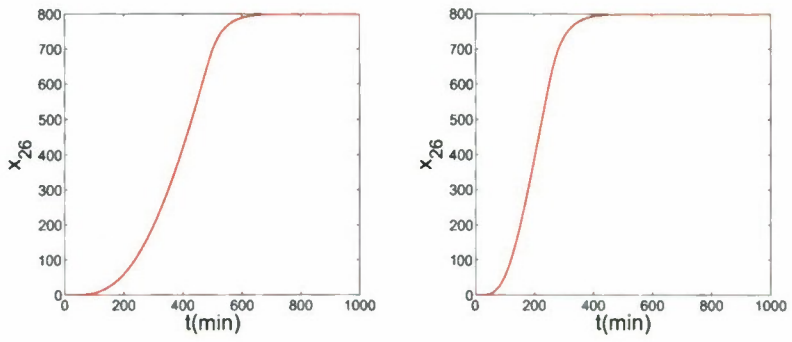


Figure 9: The evolution of the DNA fragment (x_{26}) (a) with and (b) without the NFκB feedback module.

x_1	Per_m	x_7	$\mathrm{PER} \cdot \mathrm{P}_c$	x_{13}	Pdp_m	x_{19}	CLK_c
x_2	Tim_m	x_8	$\mathrm{PER} \cdot \mathrm{P}_n$	x_{14}	Clk_m	x_{20}	$\mathrm{CLK} \cdot \mathrm{CYC}_c$
x_3	PER_c	x_9	TIM_n	x_{15}	VRT_c	x_{21}	$\mathrm{CLK} \cdot \mathrm{CYC} \cdot \mathrm{P}_c$
x_4	TIM_c	x_{10}	$\mathrm{PER} \cdot \mathrm{TIM}_n$	x_{16}	VRI_n	x_{22}	$\mathrm{CLK} \cdot \mathrm{CYC}_n$
x_5	$\mathrm{PER} \cdot \mathrm{TIM}_c$	x_{11}	SM_c	x_{17}	PDP_c	x_{23}	$\mathrm{CLK} \cdot \mathrm{CYC} \cdot \mathrm{P}_n$
x_6	$\mathrm{PER} \cdot \mathrm{TIM}_f$	x_{12}	VRI_m	x_{18}	PDP_n		

Table 3: This *Drosophila* circadian clock model consists of two loops. One contains PER and TIM and the other contains PDP and VRI. They interact through CLK-CYC controlled expression. SM and FBM are two proteins assisting in the first loop.

lations with period being 24 hours. Indeed, employing the kinetic model in (*39*) and starting with a somewhat arbitrary condition, the network soon reached an oscillatory state as shown in Fig. 12,. Without the feedbacks, all the state variables quickly relaxed to a steady state. The network loses its function. So, these feedbacks are essential elements for the generation of the circadian cycles. We note that the drift to stable oscillation in Fig. 12(a) takes much longer than the relaxation in Fig. 12(b), which indicates that the MPU works at a much smaller time scale while the feedback controller is slowly adjusting the motion to the desired one.

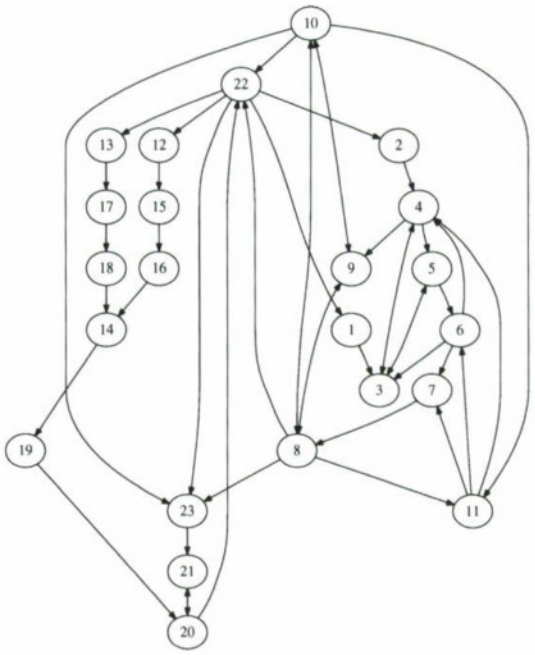


Figure 10: Network representation of the Circadian clock in *Drosophila*.

C.4 The EGF receptor signaling

The epidermal growth factor receptor (EGFR) belongs to the family of protein-tyrosine kinase receptors, which regulate cell growth, survival, proliferation and differentiation (*40, 41*). This is a simplified version of the EGFR signal transduction network adopted from (*42*). The interaction graph is depicted in Fig. 13 and notations are explained in Table C.4. Upon binding of EGF (x_1) to its receptor R (x_2), the occupied receptors EGFR (x_3) will form dimers $EGFR_2$ (x_4) and get phosphorylated $EGFR_{2p}$ (x_5), which triggers the binding of the cytoplasmic proteins, such as Grb2 (x_{20}), Shc(x_{21}) and PLCγ (x_{22}), to the receptor. PLCγ binds to the receptor to form a complex $EGFR_p$-PLC (x_6) and gets phosphorylated $EGFR_p$-PLP (x_7). The activated PLCγ, *i.e.* PLP (x_{18}) then dissociates with the receptor and initiates many subsequent reactions, like binding to cytoskeletal structures PLC-I (x_{19}). The complex $EGFR_p$-G (x_8) of the adaptor pro-

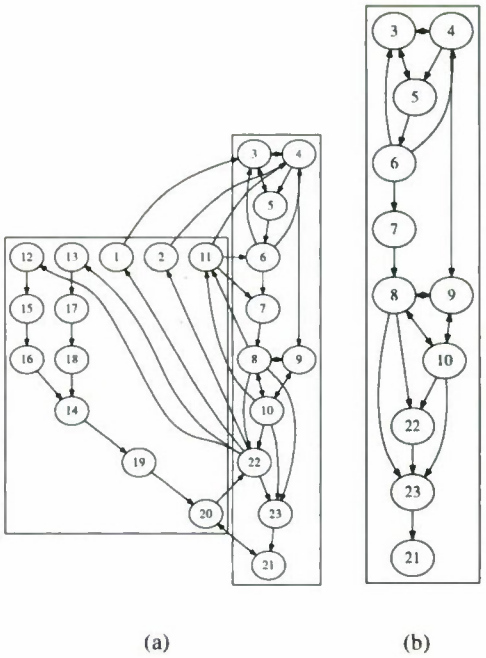


Figure 11: (a) Feedback and forward structure through graph decomposition. (b) The minimal production unit of the *Drosophila* cirdadian network.

tein Grb2 and the receptor may recruit the GDP-GTP exchange factor SOS (x_{22}) and put it in the vicinity of Ras. The binding of Grbs to the receptor could also be mediated by Shc, which makes a complex EGFR$_p$-Shc (x_{10}) first and is consequently phosphorylated into EGFR$_p$-ShP (x_{11}). The binary complex is able to bind Grb2 to form a ternary complex EGFR$_p$-ShP-G (x_{12}) and then binds to SOS to produce a quadruple EGFR$_p$-ShP-G-S (x_{13}), which approaches and affects the Ras activity like as well as in the complex EGFR$_p$-G-S (x_9). All the complexes dissociate at some rates.

In this example, almost all reactions are reversible so the whole network is strongly connected and hence usual simple decompostion schemes are hard to work. Our scheme, however, is still able to generate the biologically meaningful topology if the input and ouput vertices are

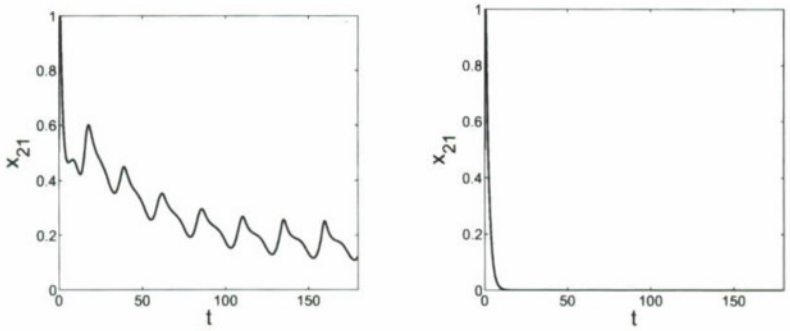


Figure 12: The evolution (a)of the cytoplamic CLK·CYC·P_c (x_{21}) with all the feedbacks present and (b) of $x_{1,\dots,23}$ without the feedback module.

x_1	EGF	x_7	$EGFR_p$−PLP	x_{13}	$EGFR_p$−ShP − G − S	x_{19}	PLC − I
x_2	R	x_8	$EGFR_p$−G	x_{14}	G − S	x_{20}	G
x_3	EGF − R	x_9	$EGFR_p$−G − S	x_{15}	ShP	x_{21}	Shc
x_4	EGF − R_2	x_{10}	$EGFR_p$−Shc	x_{16}	ShP − G	x_{22}	S
x_5	EGF − R_{2p}	x_{11}	$EGFR_p$−ShP	x_{17}	PLC	x_{23}	Shc − G − S
x_6	$EGFR_p$−PLC	x_{12}	$EGFR_p$−ShP − G	x_{18}	PLP		

Table 4: R denotes the EGF binding receptor. PLC and PLP denotes the enzyme PLC_γ and its phosphorylated form, respectively. G and S stand for Grb and SOS. ShP is the phosphorylated form of Shc.

appropriately selected, see the decomposed graph in Fig. 14(a). The binding, dimerization and phosphorylation of the EGFR leads to the production of EGF-R_{2p} (x_5) which becomes the hub of all the signaling processes. The reactions involving PLC (x_{17}) are automatically selected to make one separate module in the small box on the right of Fig. 14(a) since they are not direcly related to the production of $EGFR_{2p}$-G-S (x_9). Displayed in the graph are mainly three routes to the target output x_9. The longest one is through $x_{21} \rightarrow x_{10} \rightarrow x_{11} \rightarrow x_{12} \rightarrow x_{13}$, another is through x_8 and the shortest one is a direct connection from x_5 to x_9. But the shortest one is related to the reverse reaction of the dissociation of the $EGFR_{2p}$-G-S complex and should not

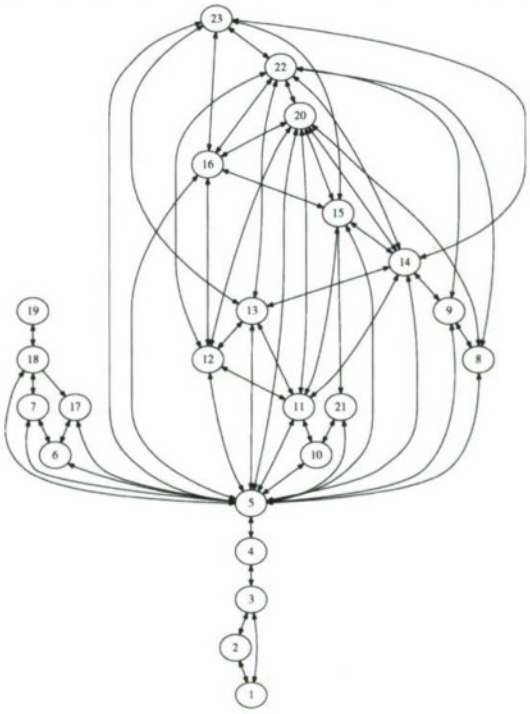
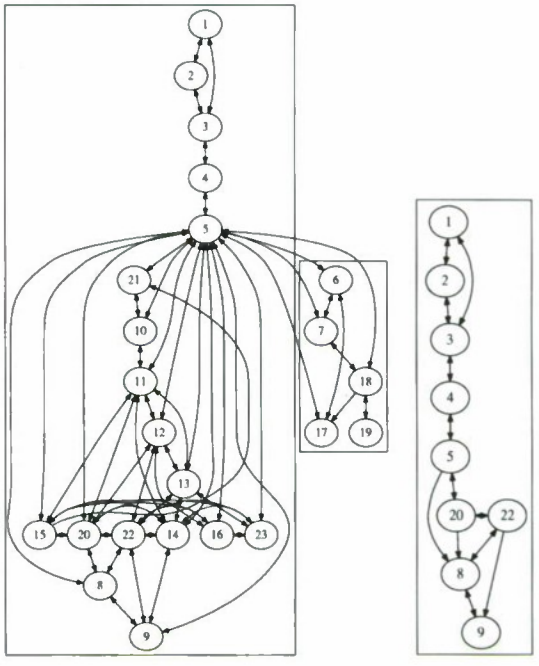


Figure 13: Network representation of the epidermal growth factor receptor model.

be treated as a signaling process. It is removed in the MPU graph shown in Fig. 14(b). So, there are two major pathways as explained before: the short one passing x_8 is through direct binding of Grb and the other long one is mediated by the binding and activation of Shc. The MPU shown only contains the short course.

The time evolution of the output plotted in Fig. 15 shows that with the feedback control (Fig. 15(a)), the signaling arrived at the desired value $x_9 = 0.13$ in a short time and stay there indefinitely long. Working only with MPU (Fig. 15(b)), however, the signal quickly rises to a large steady value $x_9 = 4.5$ and might change the behavior of the downstream reaction as a consequence. As before, the fast production unit responds in a rapid and uncontroled manner while the extra controllers may pin down the output to the desired value at a large time scale.



(a) (b)

Figure 14: (a) Feedback and forward structure through graph decomposition. (b) The minimal production unit of the EGFR network.

C.5 G-protein coupled receptor model

Here, the model is taken from (*43*), for the G protein-coupled receptor (GPCR) signaling modules in macrophage immune cells. The GPCR system responds to multiple external cues, such as: light, hormones, odorants, neurotransmitters, amino acids, which is an important drug target and is one of the most common signaling channels in a cell. Here, a simplified model is used to study the effects of two signaling molecules (C5a and UDP) on the second messenger calcium Ca^{2+}.

The interaction graph of the model network is displayed in Fig. 16 and the notation is ex-

x_1	C5aR	x_{21}	$\mathrm{PLC}\beta 4 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\alpha_q\mathrm{GTP}$	x_{41}	Buf
x_2	C5aC	x_{22}	PIP_2	x_{42}	$\mathrm{Ca}^{2+} \cdot \mathrm{Buf}$
x_3	$\mathrm{GRK}_p \cdot \mathrm{G}\beta\gamma$	x_{23}	$\mathrm{PLC}\beta_4 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\alpha_q\mathrm{GTP} \cdot \mathrm{PIP}_2$	x_{43}	Ca^{2+}_{ER}
x_4	$\mathrm{GRK}_p \cdot \mathrm{G}\beta\gamma \cdot C5aC$	x_{24}	IP3	x_{44}	PKC
x_5	$\mathrm{C5aC}_p$	x_{25}	DAG	x_{45}	$\mathrm{PKC} \cdot \mathrm{DAG}$
x_6	P2YR	x_{26}	$\mathrm{PLC}\beta 3$	x_{46}	$\mathrm{PKC} \cdot \mathrm{Ca}^{2+}$
x_7	UDPC	x_{27}	$\mathrm{PLC}\beta 3 \cdot \mathrm{Ca}^{2+}$	x_{47}	GRK_p
x_8	$\mathrm{G}\beta\gamma \cdot \mathrm{G}\alpha_i\mathrm{GDP}$	x_{28}	$\mathrm{PLC}\beta 3 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\alpha_q\mathrm{GTP}$	x_{48}	GRK
x_9	$\mathrm{G}\beta\gamma$	x_{29}	$\mathrm{PLC}\beta_3 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\alpha_q\mathrm{GTP} \cdot \mathrm{PIP}_2$	x_{49}	$\mathrm{PKC} \cdot \mathrm{DAG} \cdot \mathrm{Ca}^{2+} \cdot \mathrm{GRK}$
x_{10}	$\mathrm{G}\alpha_i\mathrm{GTP}$	x_{30}	$\mathrm{PLC}\beta 3 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\beta\gamma$	x_{50}	DAG_d
x_{11}	$\mathrm{G}\alpha_i\mathrm{GDP}$	x_{31}	$\mathrm{PLC}\beta 3 \cdot \mathrm{Ca}^{2+} \cdot \mathrm{G}\beta\gamma \cdot \mathrm{PIP2}$	x_{51}	$\mathrm{IP3}K_a$
x_{12}	$\mathrm{G}\beta\gamma \cdot \mathrm{G}\alpha_q\mathrm{GDP}$	x_{32}	$\mathrm{PKC} \cdot \mathrm{DAG} \cdot \mathrm{Ca}^{2+}$	x_{52}	IP4
x_{13}	$\mathrm{G}\alpha_q\mathrm{GTP}$	x_{33}	$\mathrm{PKC} \cdot \mathrm{DAG} \cdot \mathrm{Ca}^{2+} \cdot \mathrm{PLC}\beta 4 \cdot \mathrm{Ca}^{2+}$	x_{53}	IP5
x_{14}	$\mathrm{G}\alpha_q\mathrm{GDP}$	x_{34}	$\mathrm{PLC}\beta_4 \cdot \mathrm{Ca}^{2+}_p$		
x_{15}	RGS_a	x_{35}	$\mathrm{PKC} \cdot \mathrm{DAG} \cdot \mathrm{Ca}^{2+} \cdot \mathrm{PLC}\beta 3 \cdot \mathrm{Ca}^{2+}$		
x_{16}	$\mathrm{RGS}_a \cdot \mathrm{G}\alpha_i\mathrm{GTP}$	x_{36}	$\mathrm{PLC}\beta_3 \cdot ca^{2+}_p$		
x_{17}	$\mathrm{RGS}_a \cdot \mathrm{G}\alpha_q\mathrm{GTP}$	x_{37}	$\mathrm{IP3}R$		
x_{18}	$\mathrm{PLC}\beta 4$	x_{38}	$\mathrm{IP3R} \cdot \mathrm{IP3}$		
x_{19}	Ca^{2+}	x_{39}	$\mathrm{IP3R} \cdot \mathrm{IP3} \cdot \mathrm{Ca}^{2+}$		
x_{20}	$\mathrm{PLC}\beta 4 \cdot \mathrm{Ca}^{2+}$	x_{40}	$\mathrm{IP3R} \cdot \mathrm{Ca}^{2+}$		

Table 5: The notations for the G-protein coupled receptor model. Buf represents other calcium buffers in the cell.

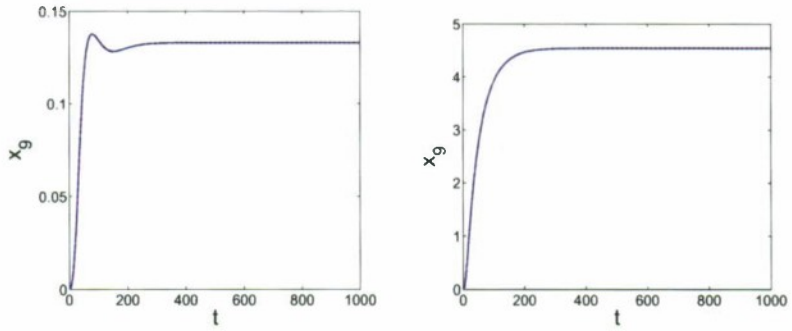


Figure 15: The evolution (a)of the EGFR$_p$-G-S (x_9) with all the feedbacks present and (b) with MPU.

plained in Table 5. The main biochemical process could be explained as following. Upon C5a and UDP binding to specific cell surface receptors, the G-protein heterotrimer dissociates to free Gα-GTP and G$\beta\gamma$, both being able to bind isoforms of PLCβ and catalyze the synthesis of IP3 and DAG from PIP2. IP3 binds to certain ion channels on the membrane of endoplasmic reticulum to induce the release of Ca^{2+} into the cytosol. DAG and Ca^{2+} bind to and activate PKC which then phosphorates and inactivate PLCβ. GRK is localized at the membrane by G$\beta\gamma$ once it is phosphorylated by PKC. GRK phosphorylates and inactivates C5a receptor (C5aR). There are also Ca^{2+} flow from other buffers in the cell or from extracelluar environment.

Upon application of our decomposion technique to the above network, a structured graph is obtained and displayed in Fig. 17. The network consists of four parts:

- The source vertex set $\{x_6, x_7, x_{51}\}$.
- The sink vertex set $\{x_{50}\}$.
- The feedback vertex set $\{x_5, x_{34}, x_{41}, x_{42}, x_{48}, x_{49}, x_{52}, x_{53}\}$.
- The major strongly connected processing unit, the rest of the vertices

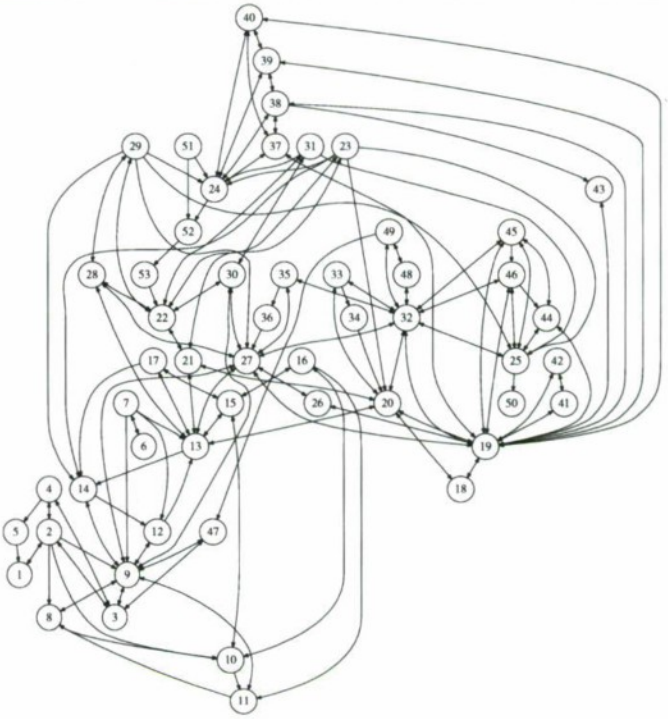


Figure 16: Network representation of the G-protein coupled receptor model.

In the major processing part, the cycle search and selection program decomposed it into horizontal layers, being put in an order which relays the signals input from vertex x_1, the cell surface receptor and terminates at vertex x_{19}, the calcium ion. According to the order given in the figure, we can see that there are many feedbacks. It happens that the feedback vertex set mark the major ones. The feedback through x_{52} , x_{53} marks the feedback of phosphorylation of IP3 back to PIP2; the one through x_{48} , x_{49} marks the feedback of PKC on GRK2; the one through x_{34} marks the action of calcium on isforms of PLCβ; the one through x_{41} , x_{42} marks the exchange of calcium with other calcium buffers. Of course, there are many other important feedbacks. They are visible as feedbacks in the boxed forward processing unit. The MPU is shown in Fig. 17(b) where we have removed the feedback box and some of the feedbacks in the

forward processing unit which are unidirectional. Shown in Fig. 18(a), is the change of Ca^{2+} over time when a concentration $250nM$ of C5a is added at $t = 0$. A fast initial growth of x_{19} and a slow relaxation to a steady value is obserbed. With only the MPU, x_{19} reaches a very high value in a short time and stay there without being brought down to the equilibrium value of the full network. So, those feedbacks act as a controller to set up the output of the system to a controled value. The MPU responds to external signals with fast production as in other examples.

C.6 λ phage decision circuit

When the bacterium *Esherichia Coli* is infected by phage λ, there are two possible subsequent pathways: lysogeny and lysis depending on the concentration of particular proteins. In the lysogenic pathway, the DNA of the λ-phage will be integrated to the host DNA. In the lytic pathway, the host DNA will be excised to provide material for λ-phage duplication. The selection of the pathway is made by the λ lysis-lysogeny decision circuit where the molecular fluctuations play significant roles (*44*).

Here, we present a circuit model due to H. H. McAdams and L. Shapiro (*45*).

Upon infection of *E. coli*, phage λ is faced with selection of two different fates: lysis where more copies of phage λ are assembled and released after the host bacterium is lysed or lysogenesis where the DNA of the phage is integrated into the host's DNA. This fate decision is made by a well-characterized competitive regulatory mechanism of a bistable gene switch. The core of the switch is the P_R and P_{RM} operators which share three operator sites (OR1, OR2, OR3). Translation of P_R transcript produces CI while P_{RM} encodes Cro. The two factors CI and Cro bind to the operators sequentially in the opposite order. When CI binds to OR1, P_R is repressed following which P_{RM} is activated by CI at OR2 and repressed by CI at OR3. P_{RM} is repressed when Cro binds to OR3 and P_R is repressed by Cro at OR2 or OR1. Initially P_{RM} is off. But CII

x_1	RecA	x_{11}	Cro	x_{21}	tL2	x_{31}	Anti − Q
x_2	ci	x_{12}	ant − Cro	x_{22}	tR2	x_{32}	$P_{R'}$
x_3	CI	x_{13}	Ant − Cro	x_{23}	tR2	x_{33}	tR′
x_4	CI*	x_{14}	P_{RM}	x_{24}	o	x_{34}	P_I
x_5	cii	x_{15}	P_R	x_{25}	O	x_{35}	xis
x_6	CII	x_{16}	P_{RE}	x_{26}	p	x_{36}	Xis
x_7	CII*	x_{17}	P_L	x_{27}	P	x_{37}	int
x_8	ciii	x_{18}	n	x_{28}	q	x_{38}	Int
x_9	CIII	x_{19}	N	x_{29}	Q	x_{39}	S
x_{10}	cro	x_{20}	tL1	x_{30}	anti − Q		

Table 6: In the table, P_{RM},P_R,P_{RE},P_L,P_I, P_{AQ},$P_{R'}$ are promoters; tR1, tR2, tL1, tL2, tR' are terminator on the DNA strand; CI, CII, CIII, N, O, P, Q, Cro, Xis, Int, Anti-Q are regulatory proteins; Anti-Q, Ant-Cro are gene transcripts. The corresponding gene is marked by lower case letters.

initiated from P_R and CIII, N initiated from P_L lead to the stimulation of P_{RE}, which induces rapid production of CI and suppresses the production of Cro by the anti-Cro transcript. Both Cro and CI supress P_L. Protein N is able to anti-terminate the terminators tR1, tR2, tL1, tL2, such that CII and CIII could be produced rapidly and the production of proteins Q, Xis, Int is able to start. Regulator Q can anti-terminate tR', which paves the way for the $P_{R'}$ initiated transcription to continue through S and activate the lysis coding genes. However, Q may be repressed by Anti-Q initiated from P_{AQ} which is activated by CII. The concentrations of Xis and Int will determine lysis-oriented excision or lysogenesis-oriented integration, respectively. The CII activated promoter P_I favors the production of Int.

In all, roughly speaking, CI, CII and CIII production leads to lysogenesis while Cro production leads to lysis. Both the bacteria state such as the multiplicity of infection (MOI) and external cues such as ultraviolet light play important roles in the fate decision process. Many components of the network are effective only transiently. Once the decision has been made, they are suppressed. In the model shown in Fig. 19, the operators OR1, OR2, OR3 are not represented explicitly.

This network is quite sparse. x_1 represents the external signal source and sits on the top of the decomposed structure in Fig. 20(a). Five elements $\{x_{25}, x_{27}, x_{36}, x_{38}, x_{39}\}$ stand at the bottom. in which $\{x_{25}, x_{27}\}$ are intermidate products that are not active in the model while the rest are important for the λ lysis-lysogen fate decision. The main processing part is contained in the left long bar where the signal enters through cii (x_5) and finally switches on (off) N (x_{19}) and tR1 (x_{22}). All the subsequent steps depend on these two. The feedback vertices are contained on the small vertical box to the right of the main processing unit. The Cro (x_{11}) autoregulation and the CIII factor (x_9) are treated as feedbacks, which seem plausible. Certainly, different choice of the input or the output vertices may result in different sequence and partitions. Also noticeable is the small loop in the main unit consisting of $\{x_{14}, x_2, x_3, x_4\}$ with one feedback from x_4 to x_{14}. This loop is just the autoregulation module of CI. After removing most feedbacks, the MPU is obtained and displayed in Fig. 20(b) where the signal relaying process is very clearly exhibited. As before, the MPU is very likely losing its biological function without the feedback controllers.

References and Notes

1. B. Alberts, *et al.*, *Molecular Biology of the Cell* (Garland Science, New York, 2002). Fourth Edition.

2. L. You, *Cell Biochem. Biophys.* **40**, 167 (2004).

3. D. Endy, R. Brent, *Nature* **409**, 391 (2001).

4. N. Barkai, S. Leibler, *Nature* **387**, 913 (1997).

5. J. Hasty, J. J. Collins, *Nat. Genet.* **31**, 13 (2002).

6. P. S. Swain, M. B. Elowitz, E. D. Siggia, *Proc. Natl. Acad. Sci.* **99**, 12795 (2002).

7. M. Kærn, T. C. Elston, W. J. Blake, J. J. Collins, *Nat. Rev. Genet.* **6**, 451 (2005).

8. K. Oda, Y. Matsuoka, A. Funahashi, H. Kitano, *Mol. Syst. Biol.* **2005.0010**, 1 (2005).

9. K. Oda, H. Kitano, *Mol. Syst. Biol.* **2006.0015**, 1 (2006).

10. D. J. Watts, S. H. Strogatz, *Nature* **393**, 440 (1998).

11. O. Mason, M. Verwoerd, *IET Syst. Biol.* **1**, 89 (2007).

12. P. D. Kuo, W. Banzhaf, A. Leier, *BioSystems* **85**, 177 (2006).

13. A. M. Walczak, M. Sasai, P. G. Wolynes, *Biophys. J.* **88**, 828 (2005).

14. R. Heinrich, B. G. Neel, T. A. Rapoport, *Molecular Cell* **9**, 957 (2002).

15. M. Chaves, E. D. Sontag, R. J. Dinerstein, *J. Phys. Chem. B* **108**, 15311 (2004).

16. A. L. Barabási, Z. N. Oltvai, *Nature Rev. Gen.* **5**, 101 (2004).

17. S. S. Shen-Orr, R. Milo, S. Mangan, U. Alon, *Nat. Genet.* **31**, 64 (2002).

18. A. Goldbeter, *Proc. Natl. Acad. Sci. USA* **88**, 9107 (1991).

19. M. Kaern, W. J. Blake, J. J. Collins, *Annu. Rev. Biomed. Eng.* **5**, 179 (2003).

20. M. Freeman, J. B. Gurdon, *Annu. Rev. Cell Devel. Biol.* **18**, 515 (2002).

21. M. E. Csete, J. C. Doyle, *Science* **295**, 1664 (2002).

22. E. Ravasz, A.-L. Barabási, *Phys. Rev. E* **67**, 026112 (2003).

23. I. Mezić, *Coupled nonlinear dynamical systems: asymptotic behavior and uncertainty propagation* (2004). 43rd IEEE Conference on Decision and Control.

24. T. Lipniacki, P. Paszek, A. R. Brasier, B. Luxon, M. Kimmel, *Biophys. J.* **228**, 195 (2004).

25. A. Hoffmann, A. Levchenko, M. L. Scott, D. Baltimore, *Science* **298**, 1241 (2002).

26. C. Scheidereit, *Oncogene* **25**, 6685 (2006).

27. K. Fujarewicz, M. Kimmel, A. Swierniak, *Math. Biosci. Engr.* **2**, 527 (2005).

28. G. Craciun, M. Feinberg, *SIAM. J. Appl. Math.* **65**, 1526 (2005).

29. G. Craciun, M. Feinberg, *SIAM. J. Appl. Math.* **66**, 1321 (2006).

30. R. Cheong, *et al.*, *J. Biol. chem.* **281**, 2945 (2006).

31. N. Barkai, S. Leibler, *Nature* **403**, 267 (2000).

32. A. E. C. Ihekwaba, *et al.*, *FEBS J.* **274**, 1678 (2007).

33. A. P. Arkin, *Curr. Opin. Biotech.* **12**, 638 (2001).

34. G. Golub, C. van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, Maryland, 1996).

35. C. M. Bender, S. A. Orszag, *Advanced mathematical methods for scientists and engineers* (McGraw-Hill, Inc., New York, 1978).

36. R. N. Gutenkunst, *et al.*, *PLoS Comput. Biol.* **3**, e189 (2007).

37. C. V. Rao, J. R. Kirby, A. P. Arkin, *PLoS Biol.* **2**, 0239 (2004).

38. P. Rangamani, L. Sirovich, *Biotech. Bioengr.* **97**, 1216 (2006).

39. R. S. Kuczenski, K. C. Hong, J. García-Ojalvo, K. H. Lee, *PLoS Comput. Biol.* **3**, 1468 (2007).

40. J. Schlessinger, *Cell* **103**, 211 (2000).

41. H. Resat, J. A. Ewald, D. A. Dixon, H. S. Wiley, *Biophys. J.* **85**, 730 (2003).

42. B. N. Kholodenko, O. V. Demin, G. Moehren, J. B. Hoek, *J. Biol. Chem.* **274**, 30169 (1999).

43. P. J. Flaherty, A kinetic model for g protein-coupled signal transduction in macrophage cells, Ph.D. thesis, EECS Department, University of California, Berkeley (2007).

44. A. Arkin, J. Ross, H. H. McAdams, *Genetics* **149**, 1633 (1998).

45. H. H. McAdams, L. Shapiro, *Science* **2669**, 650 (1995).

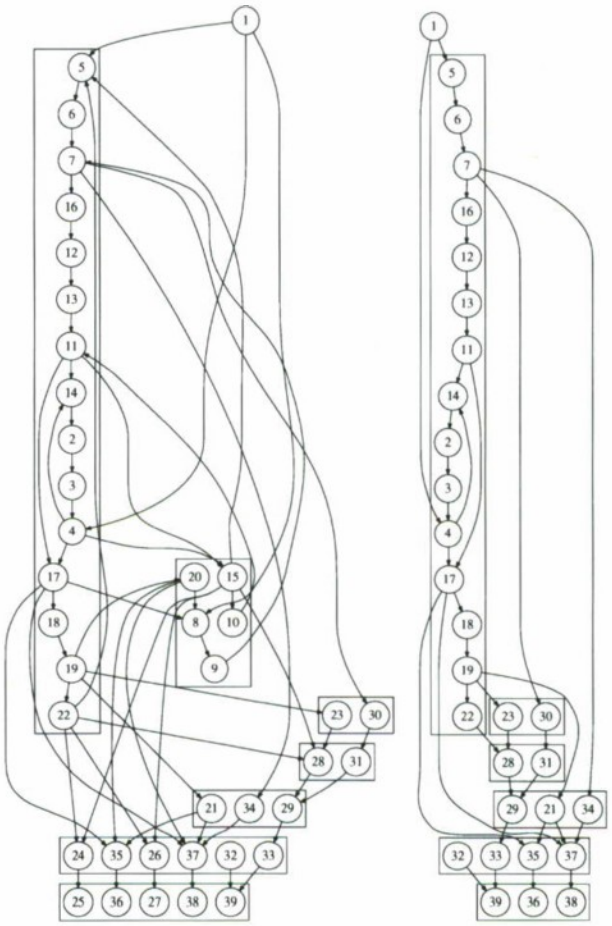


Figure 17: (a) Feedback and forward structure through graph decomposition. (b) The minimal production unit of the G-protein coupled receptor network.

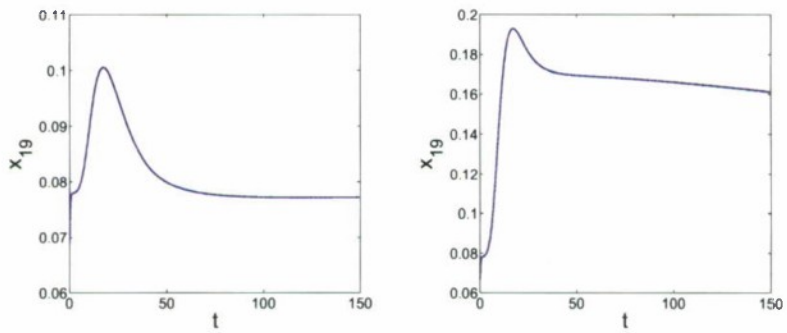


Figure 18: The evolution of the Ca^{2+} (x_{19}) with all the feedbacks present (a)and only with MPU (b).

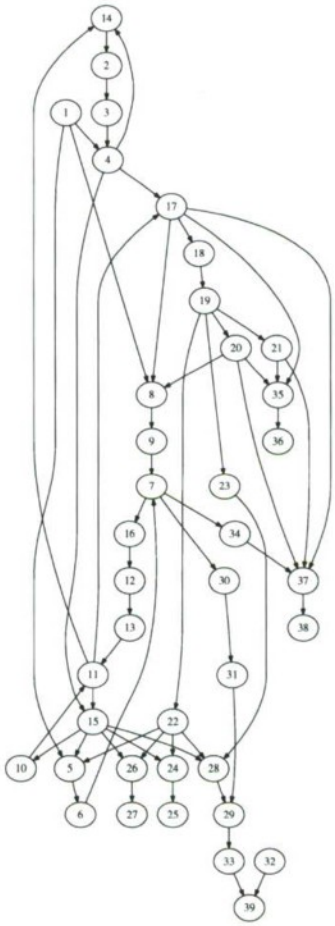


Figure 19: Network representation of the λ phage decision circuit.

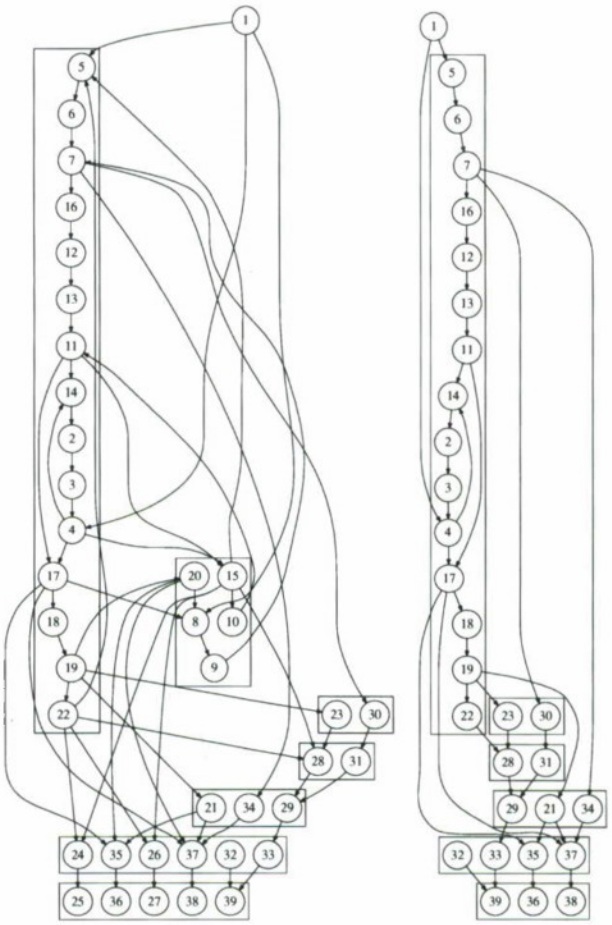


Figure 20: (a) Feedback and forward structure through graph decomposition. (b) The minimal production unit of the λ lysis-lysogen decision network.

B.7 Constrained dynamics lifting

Constrained Dynamics Lifting

Gear, Givon and Kevrekidis

August 3, 2008

Abstract

In this paper we present a novel numerical scheme that aims on integrating a dynamical system which exhibits time scale separation. The novelty of the new algorithm is that it assumes only the separation of scales, without explicitly knowing what variables are the slow ones and what variables are the fast ones.

1 Introduction

Many problems in science can be described by a dynamical system that exhibits time scale separation. The case where the variables can be categorized as slow and fast variables has been successfully treated in the past years mainly due to tools provided by Differential Algebraic Equations and the Averaging Principle. Various kinds of multiscale algorithms have been introduced to effectively integrate systems with scale separation. Following the introduction of a novel auxiliary process by Freidlin and Wentzell [1], different integrators were suggested in [2], [3],[4] which are based on a numerical discretization of the auxiliary process. The use of the averaging principle in this context crucially depends on many layers of a priori knowledge about the evolution equations, for example, the decoupling of the variables into slow variables and fast variables, the full knowledge of the different summands on the right hand side of the evolution equations for the slow variables which in turn implies the a priori knowledge of the nature of the slow variable (deterministic, stochastic,..) and last, the assumption that the observables are the slow and fast variables of the system.

Scientists and engineers are faced with research problems that often have many complex internal feedback processes that defy simple analysis, or that must be studied at scales that are much different than processes occurring in nature. In such, any a priory assumption on the nature of the coarse

description limits the feasibility of the results. These challenges often need massive datasets for simulation, have heterogeneous data sources that must be linked, or generate massive, high-dimensional datasets from experiment or observation, and will soon be beyond today's capabilities.

Computational experimentation allows insight into complex systems by enabling the creation of a virtual description (algorithmic or computational) that can interact with elements from the real world. Simulation and other dynamic modeling techniques allow us to experiment with complex systems in ways that would be unimaginable in the real world, and to constrain our understanding of the system characteristics or underlying physical phenomena. Furthermore, it allows us to guide real world operations and experimentation in cases that have potential for unforeseen or extreme events. Research in this area will provide needed new modeling techniques ranging from mathematical formulations to multi-scale simulation techniques.

A novel approach which is based on computational experimentation is the Projective Integration of Gear and Kevrekidis [5]. In the projective integration method there is no a priori knowledge of the nature of the coarse description. In their coarse projective integration method it is assumed that a parametrization of the slow variable is given and that it can be computed from the observations.

In this paper we consider the problem of integrating systems with scale separation where the separation is not explicitly given in the equations describing the dynamical system and hence no a priori knowledge of the coarse description is at hand. Of course, if additional information about the system is given in advance, one should use it to make the computations simpler and faster.

We start with a simple example of our formulation.

H1. There exists a hidden dynamical system, described by,

$$\begin{aligned} dx &= y\,dt \\ dy &= \frac{1}{\epsilon}(\sin x - y)\,dt + \frac{\sqrt{2}}{\sqrt{\epsilon}}\,dW. \end{aligned} \tag{1.1}$$

where

1. $(x, y) \in \mathcal{R} = \mathbb{R}^2$, $x \in \mathcal{R}_x = \mathbb{R}^1$, $y \in \mathcal{R}_y = \mathbb{R}^1$ and $t \in [0, T]$.
2. $\epsilon \ll 1$.

H2. The observables of the system $(u, v) \in \mathcal{S} = \mathbb{R}^2$ are an unknown, smooth and invertible function of the hidden variables x, y which we set to be

$\mathbb{F} : \mathcal{R} \to \mathcal{S}$, $\mathbb{F}(x, y) = (u, v)$,

$$(u, v) = \mathbb{F}(x, y) = \begin{pmatrix} \cos(\|(x,y)\|) & sin(\|(x,y)\|) \\ -\sin(\|(x,y)\|) & \cos(\|(x,y)\|) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (1.2)$$

An image of the function $\mathbb{F}$ is sketched in Figure 1.

Using results from the averaging principle, we have the following approximation which we now describe. We start by defining the auxiliary process. For every x,

$$dY(t) = \frac{1}{\epsilon}(\sin x - Y(t))\, dt + \frac{\sqrt{2}}{\sqrt{\epsilon}}\, dW(t). \quad (1.3)$$

If we ignore the noise term in (1.1), we have a system of stiff ODEs. The auxiliary process (1.3) corresponds to the fast equation in stiff ODEs when the slow variable is fixed. The next step after fixing the slow variable when solving stiff ODEs is to move ϵ to the left hand side and to set is to zero. This leads to an algebraic equation. The solution of the algebraic equation is the solution at time equals infinity of te fast dynamics when x is kept fixed. Equivalently it is the limiting position of the y variable when evolving by the fast dynamic equation with fixed x. When the auxiliary process is stochastic one follows the same lines. A limiting solution of the auxiliary process is sought, however the stochasticity implies that the limiting solution, when exists, is a distribution which is not limited to a stationary position. Basic knowledge in stochastic processes tells us that in our example the auxiliary process induces a stationary measure $\mathcal{N}(\sin x, 1)$, which is Guassian with $\sin x$ as its mean and one standard deviation.

Recalling again stiff ODEs, the solution of the algebraic equation, y as a function of x, is plugged into the right hand of the equation describing the motion of the slow variable to give in our example $dx \approx \sin x dt$. In the stochastic case we average the right hand side of the slow dynamics over the stationary measure to induce the averaged dynamics in $\mathcal{R}_x = \mathbb{R}^1$,

$$d\bar{x} = dt \int y\, d\left\{\mathcal{N}(\sin \bar{x}, 1)\right\} = dt\, \sin \bar{x}.$$

The averaging principle asserts that for $t \in [0, T]$,

$$(x(t), y(t)) \approx (\bar{x}(t), Y(\infty)) = (\bar{x}(t), \mathcal{N}(\sin \bar{x}(t), 1)).$$

This example is a special demonstration to the case where the knowledge of $Y_x(\infty)$, the invariant measure generated by (1.3), is sufficient to generate $\bar{x}(t)$. This was mainly achieved due to he fact that one knows which variables

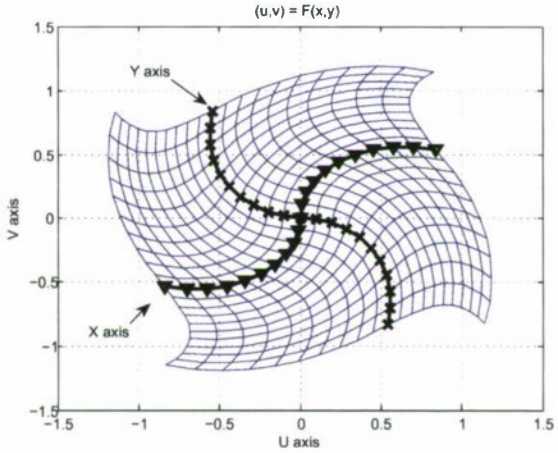


Figure 1: The image of the non linear transformation given in (1.2)

are the fast variables and which are the slow variables. In all other cases which we categorized under Assumption [H2], there is no known scheme that is able to integrate such systems.

This section is organizes as follows. In Subsection 1.1 we describe the (coarse) projective integration method. In Section 1.2 we introduce our algorithm.

1.1 (Coarse) Projective Integration

1.1.1 Start with projective for ODES

Emphasize that the slow manifold exists because of the enslaving. This is a case where the high dimensional evolution reduces to a lower dimensional process. Regardless of the initial position of the particle, it takes the particle very short time (which is usually $\epsilon \log \frac{1}{\epsilon}$) to approach the lower dimensional manifold, and after the time the particle evolves only close to the lower dimensional submanifold. This is variable reduction. For example consider

$$\begin{aligned} dx &= y\,dt \\ dy &= \frac{1}{\epsilon}(\sin x - y)\,dt. \end{aligned} \tag{1.4}$$

If we start at (x_0, y_0), then after short time the particle will be positioned close to $(x_0, \sin x_0)$, and then it will continue evolving along the curve $\{(x, \sin x)\}$ where the evolution is dictated by $dx = \sin x dt$, the speed at which it moves along the curve.

1.1.2 Move to projective for fast stochastic when we know the slow variable

Explain that the motion is in the entire space, the particle is not enslaved to a lower dimensional submanifold. Instead we can find coarse variables that evolve on a lower dimensional submanifold, they are not physical particles.

Consider again (1.1). For every initial value (x_0, y_0) the particle will explore the entire y-space before we can see a change in the x direction. As x evolves the particle will continue to explore the entire y-space that corresponds the the current value of the x component. What the averaging principle gives us is an artificial description, in the sense that the particle itself doesn't follow this description but rather the x-component of the particle follows that reduced evolution. Note that in our examples the coarse description is the same both for the stiff ODEs and for the case where the fast

dynamics is stochastic. However, we emphasize again that in the stochastic case there is no physical particle which evolves along the averaged system.

Conclude with the difficulty with lifting/initilalizing when the coarse variable is not part of the phase space.

1.2 Algorithm

It is often the case for a complex system that one is usually looking for a coarse description, we do the same, however, the algorithm we present is independent of the choice of the coarse variable. A predetermined choice of a coarse variable doesn't change the way the algorithm is implemented.

The averaging principle as given in the xy-space, $\mathcal{R}$, decouples the phase space $\mathcal{R}$ into a product of two spaces $\mathcal{R} = \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, where the first subspace consists of only the subspace where the slow motion occurs and the second one consists of only the subspace where the fast motion occurs. For every $x \in \mathbb{R}^{d_x}$, the fast motion induces a probability distribution $\mathcal{N}(\sin x, 1)$ in $\mathbb{R}^{d_y}$. This is equivalent to saying that in $\mathcal{R}$, the phase space, the probability density is given by,

$$\delta(x)\mathcal{N}(\sin x, 1).$$

We emphasize that although in $\mathcal{R}$ there exists a natural coarse variable and natural coarse space $\mathcal{R}_x$, in $\mathcal{S}$ there are infinitely many "natural" coarse spaces and coarse variables. To see this, recall our system (1.1). The fast dynamics induces a Gaussian measure with $\sin x$ as its mean, and the natural slow variable will evolve in $\mathbb{R}$ like

$$d\bar{x} = \sin \bar{x}\, dt.$$

However, for this dynamical system an important coarse variable might be the evolution of the mean of the distribution which is a vector in $\mathbb{R}^2$ and is given by $(\bar{x}(t), \sin \bar{x}(t))$. Hence in this case the elimination of the y component of a coarse description is quiet brutal.

We now move to introduce the algorithm. We first describe it using hand waving and assuming no predetermined coarse variable is known.

1. Set initial value in the observed world u, v.

2. The current value has an inverse image in the x, y world which we denote by (x, y). Since we have no access to the x, y world, we use this only as a fact.

3. Evolve the dynamical system in the u, v world using microscopic time step to generate a good empirical sample for the image of the distribution in the y space which corresponds to the current x.

4. Calculate the median of the empirical sample. The median will be our coarse variable.

5. Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value.

6. Starting at that value evolve the dynamical system using microscopic time step to generate a second good empirical sample of the stochastic cloud.

7. Calculate the median of the empirical sample. The is the second median point for evaluating the derivative of the median.

8. Make a **Projective step** using the two medians.

9. If $(n < N)$ repeat the procedure, otherwise END.

In order to move to the more formal description of the algorithm, we now define the symbols we use throughout the algorithm:

- Let $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ be the metric in $\mathcal{S}$. This metric is used for determining the medians. This knowledge can be used as variance reduction for the empirical median. However in this paper we do not address this point any further

- Define $P_x : \mathcal{R} \to \mathbb{R}_x$ to be the function ("projection") $P_x(x, y) = x$

- Let $(u_0, v_0) \in \mathcal{S}$ be an initial value

- (U_n, V_n) are the numerical values of the coarse description

- Let (U^m, V^m) be the empirical samples of a cloud

- Recall $\mathbb{F}(x, y) = (u, v)$ to be the unknown function between the two worlds

- Z_n^i for $i = 1$(resp. $i = 2$) is the first (resp. second) median used for the projective step.

Equipped with the definitions we present the algorithm.

Algorithm:

1. Set $n = 0$, $i = 1$ and $(U^0, V^0) = (u_0, v_0)$.
2. Given (U^0, V^0) there exist $(X, Y) = \mathbb{F}^{-1}(U^0, V^0)$.
3. Evolve the dynamical system using microscopic time step δt to generate (U^m, V^m) $m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^M\right)\right) \approx X$
4. Set $Z_n^i = \min_{0 \le l \le M} \sum_{m=0}^M d((U^l, V^l)), (U^m, V^m))$.
5. If $(i = 1)$
   (a) $(U_n, V_n) = Z_n^1$
   (b) Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value as (U^0, V^0).
   (c) Set $i = 2$.
   (d) Go to 2.
6. **Projective step:**
   (a) $(U_{n+1}, V_{n+1}) = (U_n, V_n) + \frac{Z_n^2 - Z_n^1}{\lambda t}\Delta t$.
   (b) $i = 1$.
   (c) If $(n < N)$ go to 2, otherwise END.

The coarse description that we have constructed is given by

$$\mathcal{C} := \{(U_n, V_n)\}_{n=0}^N \subset \mathcal{S}.$$

We have chosen $(U_n, V_n) = Z_n^1$ to be the median [?] point of the experimental data set sampled from the invariant measure $\mathbb{F}(\delta(X)\rho_\infty(y; X))$.

When one is equipped with a predetermined coarse variable that we set to be Q then the following additions need to be made.

Algorithm with predetermined coarse variable:

1. Set $n = 0$, $i = 1$ and $(U^0, V^0) = (u_0, v_0)$.
2. Given (U^0, V^0) there exist $(X, Y) = \mathbb{F}^{-1}(U^0, V^0)$.

3. Evolve the dynamical system using microscopic time step δt to generate (U^m, V^m) $m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^M\right)\right) \approx X$
   (a) Set $Z_n^i = \min_{0 \le l \le M} \sum_{m=0}^M d((U^l, V^l)), (U^m, V^m))$.
   (b) Calculate the current value of the coarse variable and denote it as Q_n^i.
4. If $(i = 1)$
   (a) $(U_n, V_n) = Z_n^1$
   (b) Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value as (U^0, V^0).
   (c) Set $i = 2$.
   (d) Go to 2.
5. **Projective step:**
   (a) i. $(U_{n+1}, V_{n+1}) = (U_n, V_n) + \frac{Z_n^2 - Z_n^1}{\lambda t}\Delta t$.
   ii. $Q_{n+1} = Q_n + \frac{Q_n^2 - Q_n^1}{\lambda t}\Delta t$.
   (b) $i = 1$.
   (c) If $(n < N)$ go to 2, otherwise END.

The coarse description that we have constructed is given by

$$\mathcal{Q} := \{Q_n\}_{n=0}^N \subset \mathcal{S}.$$

The idea in this algorithm is to evolve the predetermined coarse variable using the standard projective schemes. However with that we are still left with the problem of performing the lifting. To overcome the lifting problem we couple the standard projective scheme with the new algorithm that evolves the medians. In such, the lifting is done by using the present value of the median.

2 The same thing in infinitesimal generators language

The relation of $\mathcal{C}$ to the slow/coarse variable x in $\mathcal{R}$ is through the existence of an additional smooth and invertible function $\mathbb{G}$ such that.

$$\mathbb{F}^{-1}(\mathcal{C}) = \mathbb{F}^{-1}\left(\{(U_n, V_n)\}_{n=0}^N\right) \subset \mathbb{G}\left(\mathcal{R}_x\right).$$

We start by formulating the problem.

H1. There exists a hidden dynamical system, described by,

$$\partial_t \rho(x, y, t) = L\rho(x, y, t), \tag{2.1}$$

where

1. $(x, y) \in \mathcal{R} = \mathbb{R}^n$, $x \in \mathcal{R}_x = \mathbb{R}^{d_x}$, $y \in \mathcal{R}_y = \mathbb{R}^{d_y}$ and $t \in [0, T]$.
2. $$L = \frac{1}{\epsilon} L_1 + L_2.$$
3. $\epsilon \ll 1$.
4. the derivatives (or differences when we include jumps) prescribed in L_1 (resp. L_2) are only in the y (resp. x) direction.

H2. The observables of the system $(u, v) \in \mathcal{S} = \mathbb{R}^n$ are an unknown, smooth and invertible function of the hidden variables x, y which we set to be $\mathbb{F} : \mathcal{R} \to \mathcal{S}$, $\mathbb{F}(x, y) = (u, v)$.

Using results from the averaging principle, we have the following approximation,

$$\rho\left((x, y), t\right) \approx \bar{\rho}\left(x, t\right) \rho_\infty(y; x). \tag{2.2}$$

In special cases when the evolution of the slow variable can be modeled by a differential or difference equations which includes ODEs, SDEs and Jump Processes, the knowledge of $\rho_\infty(y; x)$ is sufficient to generate $\bar{\rho}\left(x, t\right)$. In all other cases which we categorized under Assumption [H2], there is no known scheme that is able to integrate such systems.

The paper is organizes as follows. In Section 3 we describe the (coarse) projective integration method. In Section 4 we introduce our algorithm.

3 (Coarse) Projective Integration

3.1 Start with projective for ODES

Emphasize that the slow manifold exists because of the enslaving. This is a case where the high dimensional evolution reduces to a lower dimensional process; the particle evolves only on a lower dimensional submanifold. This is variable reduction.

3.2 Move to projective for fast stochastic when we know the slow variable

Explain that the motion is in the entire space, the particle is not enslaved to a lower dimensional submanifold. Instead we can find coarse variables that evolve on a lower dimensional submanifold, they are not physical particles.

Conclude with the difficulty with lifting/initilalizing when the coarse variable is not part of the phase space.

4 Algorithm

We start our presentation of the algorithm for a simple case where,

$$L_2 = a(x, y) \cdot \nabla_x,$$

i.e., assuming the slow variables x evolve according to an ODE,

$$dx = a(x, y)\, dt,$$

and we later make the necessary generalizations for more complex L_2 generators.

It is often the case for a complex system that one is usually looking for a coarse description, we do the same, however, the algorithm we present is independent of the choice of the coarse variable and in some sense it can be used as a "Coarse Free - Equation Free " approach. We first start with exploring the submanifold in the uv-space, $\mathcal{S}$, which is the image of

$$\mathbb{F}(\mathrm{Supp}(\rho_\infty(y; x))),$$

where Supp is the support of the density in $\mathcal{R}$. The averaging principle as given in the xy-space, $\mathcal{R}$, decouples the phase space $\mathcal{R}$ into a product of two spaces $\mathcal{R} = \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, where the first subspace consists of only the subspace where the slow motion occurs and the second one consists of only the subspace where the fast motion occurs. For every $x \in \mathbb{R}^{d_x}$, the fast motion induces a probability distribution $\rho_\infty(y; x)$ in $\mathbb{R}^{d_y}$. This is equivalent to saying that in $\mathcal{R}$, the phase space, the probability density is given by,

$$\rho_\infty(x, y) := \delta(x)\rho_\infty(y; x).$$

We emphasize that although in $\mathcal{R}$ there exists a natural coarse variable and natural coarse space $= \mathcal{R}_x$, in $\mathcal{S}$ there are infinitely many "natural"

coarse spaces and coarse variables. To see this, consider the fast dynamics to be described by an OU process,

$$dx = y\,dt$$
$$dy = \frac{1}{\epsilon}(\sin x - y)\,dt + \frac{\sqrt{2}}{\sqrt{\epsilon}}\,dB.$$

The fast dynamics induces a Gaussian measure with $\sin x$ as its mean, and the natural slow variable will evolve in $\mathbb{R}$ like

$$dx = \sin x\,dt.$$

However, for this dynamical system an important coarse variable might be the evolution of the mean of the distribution $\rho_\infty(x, y)$ which is in $\mathbb{R}^2$. Hence in this case the elimination of the y component of a coarse description is quiet brutal.

$d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$, to be the metric in $\mathcal{S}$ and $P_x : \mathcal{R} \to \mathbb{R}_x$ to be the function ("projection") $P_x(x, y) = x$. Let $(u_0, v_0) \in \mathcal{S}$ be an initial value.

1. Set $n = 0$, $i = 1$ and $(U^0, V^0) = (u_0, v_0)$.
2. Given (U^0, V^0) there exist $(X, Y) = \mathbb{F}^{-1}(U^0, V^0)$.
3. Evolve the dynamical system using microscopic time step δt to generate (U^m, V^m) $m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^M\right)\right) \approx X$
4. Set $Z_n^i = \min_{0 \le l \le M} \sum_{m=0}^M d((U^l, V^l)), (U^m, V^m))$.
5. If $(i = 1)$
   (a) $(U_n, V_n) = Z_n^1$
   (b) Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value as (U^0, V^0).
   (c) Set $i = 2$.
   (d) Go to 2.
6. **Projective step:**
   (a) $(U_{n+1}, V_{n+1}) = (U_n, V_n) + \frac{Z_n^2 - Z_n^1}{\lambda t}\Delta t$.
   (b) $i = 1$.

(c) If $(n < N)$ go to 2, otherwise END.

The coarse description that we have constructed is given by

$$\mathcal{C} := \{(U_n, V_n)\}_{n=0}^{N} \subset \mathcal{S}.$$

We have chosen $(U_n, V_n) = Z_n^1$ to be the median [?] point of the experimental data set sampled from the invariant measure $\mathbb{F}(\delta(X)\rho_\infty(y; X))$.

The relation of $\mathcal{C}$ to the slow/coarse variable x in $\mathcal{R}$ is through the existenee of an additional smooth and invertible funetion $\mathbb{G}$ sueh that,

$$\mathbb{F}^{-1}(\mathcal{C}) = \mathbb{F}^{-1}\left(\{(U_n, V_n)\}_{n=0}^{N}\right) \subset \mathbb{G}\left(\mathcal{R}_x\right).$$

5 Generalizations

In this section we give different generalization to our novel scheme. Each generalization corresponds to some lack of a priory information about the dynamical system whieh is not known in advanee. The less information we have in advance about the system, the more complexity we need to add to the solution.

In the recent years .... how to efficiently parallel the algorithm

5.1 Effective dimension

We assume here that we do not know in advance the dimension of the slow variables and the dimension of the fast variables. We only show how to find the dimension of the fast variables. The description that we give assumes that ϵ is at the limit 0.

We start by evolving the system according to Step 3 in the algorithm to generate the data set $A = \{(U^m, V^m)\}_{m=1}^{M}$. If A contains no acute triangles then A is a manifold with dimension 1. If A contains acute triangles but no acute three-dimensional polyhedron where the faces are acute triangles, then A is a submanifold of dimension 2. This verification can continue to higher dimension although its complexity becomes exponential in the dimension.

5.2 Hypothesis Validation

In the algorithm we presented only the case where $L_2 = a(x, y) \cdot \nabla_x$, i.e., assuming the slow variables x evolve according to an ODE, $dx = a(x, y)\, dt$, Most of the problems that appear in the literature in the context of scale separation and coarse graining can be categorized into three types:

1. The coarse variable is governed by an ordinary differential equation
2. The coarse variable is governed by an stochastic differential equation that corresponds to a Fokker-Planck equation
3. The coarse variable is governed by a jump process that can be numerically solved using Gillespie's SSA.

Under the assumption that the coarse variable is either one of these types, we now present the generalization of our original algorithm to the two stochastic cases. We emphasize that we do not need to know in advance the nature of the coarse variable and we can verify it from the data. We start by determining to which category the dynamics belong and then we show how to change the algorithm to incorporate this knowledge.

1. $(U^0, V^0) = (u_0, v_0)$.
2. Given (U^0, V^0) there exist $(X, Y) = \mathbb{F}^{-1}(U^0, V^0)$.
3. Evolve the dynamical system using microscopic time step δt to generate $(U^m, V^m) \quad m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^M\right)\right) \approx X$
4. Set $Z^0 = \min_{0 \le l \le M} \sum_{m=0}^M d((U^l, V^l)), (U^m, V^m))$.
5. Repeat for $1 \le k \le K$,
   (a) Set Z^0 to be initial value
   (b) Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value as (U_k^0, V_k^0).
   (c) Given (U_k^0, V_k^0) there exist $(X, Y) = \mathbb{F}^{-1}(U_k^0, V_k^0)$.
   (d) Evolve the dynamical system using microscopic time step δt to generate $(U_k^m, V_k^m) \quad m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^M\right)\right) \approx X$
   (e) Set $Z^k = \min_{0 \le l \le M} \sum_{m=0}^M d((U_k^l, V_k^l)), (U_k^m, V_k^m))$.
6. **Hypothesis test:**
   (a) $\bar{A}_1(Z^0) = \frac{1}{\lambda t K} \sum_{k=1}^K \left(Z^k - Z^0\right)$

(b) $\bar{A}_2(Z^0) = \frac{1}{\lambda t K} \sum_{k=1}^{K} \left(Z^k - Z^0\right)^2$

With that we can test to get the following results:

1. If $\exists k$ such that $Z^k = Z^0$ then the coarse dynamics is governed by a jump process with intensity $\lambda = \frac{\#\{k: Z^k \neq Z^0\}}{K}$. *check whether I can do even the jumps or just approximate them*
2. If $\bar{A}_2(Z^0) = 0$ then the coarse dynamics is deterministic.
3. In all other cases the coarse dynamics is governed by an SDE.

Now that we know the nature of the coarse variable we can change the Projective Step in the algorithm. We present here the algorithm for a coarse variable described by an SDE:

5.2.1 Algorithm for coarse SDE

1. Set $n = 0$, $i = 1$ and $(U^0, V^0) = (u_0, v_0)$.
2. Given (U^0, V^0) there exist $(X, Y) = \mathbb{F}^{-1}(U^0, V^0)$.
3. Evolve the dynamical system using microscopic time step δt to generate $(U^m, V^m) \quad m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^{M}\right)\right) \approx X$
4. Set $Z_n^i = \min_{0 \leq l \leq M} \sum_{m=0}^{M} d((U^l, V^l)), (U^m, V^m))$.
5. $(U_n, V_n) = Z_n^1$
6. Repeat for $1 \leq k \leq K$,
   (a) Set Z_n^1 to be initial value
   (b) Evolve the system for a period of λt where $\delta t \ll \lambda t \ll \Delta t$ and record the final value as (U_k^0, V_k^0).
   (c) Given (U_k^0, V_k^0) there exist $(X, Y) = \mathbb{F}^{-1}(U_k^0, V_k^0)$.
   (d) Evolve the dynamical system using microscopic time step δt to generate $(U_k^m, V_k^m) \quad m = 1, \ldots, M$, where M is large enough to have a good empirical sample of $\mathbb{F}(\delta(X)\rho_\infty(y; X))$, but small enough to insure $P_x\left(\mathbb{F}^{-1}\left(\{U^m, V^m\}_{m=0}^{M}\right)\right) \approx X$

(e) Set $Z^k = \min_{0 \le l \le M} \sum_{m=0}^{M} d((U_k^l, V_k^l)), (U_k^m, V_k^m))$.

7. $\bar{A}_1(U_n, V_n) = \frac{1}{\lambda t K} \sum_{k=1}^{K} \left(Z^k - Z^0\right)$

8. $\bar{A}_2^2(U_n, V_n) = \frac{1}{\lambda t K} \sum_{k=1}^{K} \left(Z^k - Z^0\right)^2$

9. **Projective step:**

   (a) $(U_{n+1}, V_{n+1}) = (U_n, V_n) + \bar{A}_1(U_n, V_n)\Delta t + \bar{A}_2(U_n, V_n)\, dB$.

   (b) If $(n < N)$ go to 2, otherwise END.

5.3 Coarse projective

In this section we discuss the case where we are given a coarse variable D of the system and a function H such that for any value $(u, v) \in \mathcal{S}$ we can calculate $D = H(u, v)$.

The problem

6 Diffusion maps for exploring the coarse sub-manifold

Clustering and low dimensional representation of high dimensional data are important problems in many diverse fields. In recent years various spectral methods [6] to perform these tasks, based on the eigenvectors of adjacency matrices of graphs on the data have been developed. One can think of the diffusion maps as a way to revive a data set by choosing a discrete jump process that can generate the same data set as its stationary distribution and then using characteristics of the jump process to extract information from the data set.

In the context of this paper and under appropriate assumptions one can improve the way the jump process is chosen. If the coarse variable is indeed restricted to $\mathcal{R}_x$, then the following can be done.

1. Discretize $\mathcal{R}_x$ and label the space segments as $\{B_l\}_{l=1}^{L}$. Set a zero matrix $A_{L,L} = 0$.

2. For each step from B_l to B_m set $A(l, m) = A(l, m) + 1$.

3. Normalize the rows to stochastic matrix.

A is now an approximation to the operator of the coarse variable. $A(i, j)$ gives the transition probability between the ith neighborhood and between

the jth neighborhood. Now with this matrix one can continue with the following parts of the diffusion maps methods.

When the above assumptions are not met, for example in the case where the coarse dynamics is not restricted to $\mathcal{R}_x$ rather it is just restricted to a submanifold of $\mathcal{S}$, then the use of the diffusion maps method is very appealing for exploring the coarse submanifold. We demonstrate here a problem in which the coarse variable evolves according to a double well FP equation on an unknown submanifold. Since the dimension of the phase space is much larger compared to the dimension of the submanifold explored by the coarse variable the discretization suggested above becomes inefficient.

7 Acknowledgment

This work was supported in part by DARPA DSO (Dr. Cindy Daniell PM) under AFOSR contract FA9550-07-C-0024 (Dr. Fariba Fahroo PM).

References

[1] M. I. Freidlin and A. D. Wentzell. *Random perturbations of dynamical systems*, volume 260 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 1984. Translated from the Russian by Joseph Szücs.

[2] E. Vanden-Eijnden. Numerical techniques for multi-scale dynamical systems with stochastic effects. *Commun. Math. Sci.*, 1(2):385–391, 2003.

[3] D. Givon, I. G. Kevrekidis, and R. Kupferman. Strong convergence of projective integration schemes for singularly perturbed stochastic differential systems. *Commun. Math. Sci.*, 4(4):707–729, 2006.

[4] D. Givon and I. Kevrekidis. Multiscale integrators for jump-diffusion stochastic differential systems. *SIAM MMS*, 2008.

[5] C. W. Gear and I. G. Kevrekidis. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comput.*, 24(4):1091–1106 (electronic), 2003.

[6] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Difusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.*, 21(1):113–127, 2006.

Appendix C

Design of dynamics for self-assembly

C.1 Fast generation of potentials for self-assembly of particles

PRE/Self-Assembly

Fast Generation of Potentials for Self Assembly of Particles

Philip du Toit, Katalin Grubits*, and Jerrold Marsden

Control and Dynamical Systems, California Institute of Technology, MC 107-81, Pasadena, CA 91125†

Sorin Costiner

United Technologies Research Center, East Hartford, CT 06108

(Dated: April 9, 2009)

Abstract

We address the inverse problem of designing isotropic pairwise particle interaction potentials that lead to the formation of a desired lattice when a system of particles is cooled. The design problem is motivated by the desire to produce materials with pre-specified structure and properties. We present a heuristic computation-free geometric method, as well as a fast and robust trend optimization method that lead to the formation of high quality honeycomb lattices. The trend optimization method is particularly successful since it is well-suited to efficient optimization of the noisy and expensive objective functions encountered in the self-assembly design problem. We also present anisotropic potentials that robustly lead to the formation of the kagome lattice, a lattice that has not previously been obtained with isotropic potentials.

* Current affiliation: Department of Mathematics, Marymount Manhattan College, 221 East 71st St, New York, NY 10021

† Electronic address: pdutoit@cds.caltech.edu; URL: `https://www.cds.caltech.edu/~pdutoit`

I. INTRODUCTION

During the process of self-assembly, randomly distributed components acting under the influence of short-range mutual interactions, arrange themselves into a highly ordered final configuration. An important feature of self-assembly is that the components arrange *themselves* into this more ordered state without the influence of external factors. The ordered arrangement observed at the level of the resulting superstructure is predicated on the design of the individual components and their local interactions. Understanding the self-assembly process and how local properties of the components may be manipulated to influence the resulting global ordering, is an active area of research spanning a broad range of disciplines including materials science, chemical engineering, bioengineering, and nanotechnology.

In a seminal article, Whitesides and Grzybowski [1] provided a broad definition of self-assembly, and addressed promising applications in a wide range of disciplines. They observed that self-assembly of cells to form tissue, organs, and ultimately organisms, is fundamental to life, and motivates a determined study of the self-assembly process. Subsequent studies and applications of self-assembly are as fascinating as they are plentiful. Murr *et al* [2] identified self-assembly as the mechanism by which silicatein monomers combine to form protein fibers in certain marine sponges. Zheng *et al* used shape recognition and selective binding to induce self-assembly and packaging of integrated semiconductor microsystem devices by agitating the components in an aqueous environment [3], and Stauth *et al* [4] have manufactured field-effect transistors via self-assembly of micrometer scale components using similar techniques. In [5], self-assembly is shown to enhance synergistic group transport of autonomous robots. Jakab *et al* recognized that although biological systems are genetically controlled, the formation of biological superstructures is ultimately governed by physical interactions, and demonstrated the formation of prescribed shapes using self-assembling multicellular systems [6].

Typically, studies of self-assembly examine the ordered superstructures that arise from a given fixed interaction potential. For example, Manoharan *et al* used optical and electron microscopy to identify the range of structures produced by self-assembly of colloidal microspheres as fluid is removed from the emulsion droplets containing the spheres [7]. Maksymovych *et al* investigated the formation and reactivity of linear chains of dimethyldisulfide molecules on a gold surface [8]. Engel *et al* ([9], see also [10–13]) have observed the formation of complex crystals and quasicrystals arising from a simple double-well interaction potential.

Interest in the fabrication of nanomaterials and photonic crystals with desired material prop-

erties [14–21] motivates the inverse problem: *design* the shortrange interactions in order to induce self-assembly of the components into a desired lattice structure. Laboratory techniques now available allow for modification and tuning of particle interaction potentials [22], and hence experimentalists are achieving ever-increasing control over the local interactions that influence the global properties of the material formed via self-assembly. These methods typically use colloidal suspensions and optical forcing to alter the chemical environment and screening properties of the solution in which the assembly occurs.

In this paper, we specifically consider the problem of designing short-range pairwise interaction potentials between particles on a planar surface to induce self-assembly of a desired lattice. The main results to be presented are new methods—a heuristic *geometric method* as well as a robust *trend optimization method* for the design of isotropic interaction potentials that lead to high quality honeycomb lattices as the system of particles is cooled. The geometric method is also extended to the case of anisotropic potentials which allows for the formation of more exotic kagome lattices. Another contribution of this work is the development of tools for objective assessment of quality of lattices, that mimic intuitive human perception of lattice quality.

Rechtsman *et al* in [23] have already demonstrated computational methods for finding solutions to the inverse self-assembly problem. They used a simulated annealing optimization procedure to find potentials that lead to the self-assembly of particles into square and honeycomb lattices. To be sure, the intent of [23] was to demonstrate that the inverse problem of potential design for the purpose of inducing the formation of a target lattice can be solved in practice, and to carefully verify that the potentials they proposed do indeed lead to the target lattices through Monte Carlo simulation. Hence, the computational effort required to obtain the potentials was only a marginal consideration in their work. Presently, we consider the straightforward simulated annealing method of [23] as a baseline method with which we may compare the new methods presented here. When compared with the baseline simulated annealing method, the optimization procedure described in this paper leads to a hundredfold speed-up in the generation of potentials, as well as the formation of higher quality lattices. Furthermore, the procedure for finding potentials is more robust, and the resultant potentials form the target lattices more robustly with respect to variations in the initial conditions of the particles. As will be demonstrated, the chief reason for the marked speed-up over the simulated annealing method is the facility of the trend optimization method to optimize objective functions that are both noisy and expensive to evaluate.

The organization of the paper is as follows. In Section II, we precisely define the self-assembly problem, and summarize the method for generating potentials devised previously by [23] that will

serve as a baseline method for purposes of comparison. In Section III, we establish objective metrics for measuring lattice quality so that reasonable comparisons between the methods can be made.

We approach the self-assembly problem by framing it as an optimization problem in which the desired potential optimizes a suitably chosen objective function. This approach requires that we choose both an optimization scheme and an objective function to be optimized. Section IV describes three objective functions that will be used, while Section V describes in detail the trend optimization method. A discussion on the hierarchical nature of the trend method is also provided. We proceed in Section VI to list the five solution methods to be compared and describe their implementation. The final comparison of the methods is presented graphically in the plots of Section VII. In Section VIII, we present an extension of the geometric method to anisotropic potentials that lead to self-assembly of the kagome lattice.

All molecular dynamics simulations performed during the design and testing of the potentials were executed on the CITerra high performance computing cluster housed in the Division of Geophysical and Planetary Sciences at Caltech using the LAMMPS software package [24] from Sandia Laboratories.

Acknowledgments. We would like to thank Andrzej Banaszuk, Ronald Coifman, Yannis Kevrekidis, Alison Marsden, Matthew West, José Miguel Pasini, and Igor Mezić for their interest and helpful comments. This work was supported in part by DARPA DSO under AFOSR contract FA9550-07-C-0024.

II. THE SELF ASSEMBLY PROBLEM

The availability of laboratory methods to tune interaction potentials between components, and hence influence the structure of the resulting self-assembled configurations, motivates the use of self-assembly to produce materials with desired structural properties. The specific self-assembly problem addressed in this paper is defined as follows:

Definition: The Self-Assembly Problem

Design a radially symmetric pairwise interaction potential, $V_{\mathsf{HC}}(r)$, so that when a system of particles interacting with each other in the plane through this potential is cooled, the particles form a honeycomb lattice.

The purpose of the present paper is to compare methods for generating potentials that solve the

self-assembly problem. The methods for generating potentials are compared using three criteria:

C1. The computational effort required by the method to produce the potential;

C2. The quality of the lattices formed by the potential;

C3. The robustness of the quality of the lattices formed to variations in the initial conditions of the particles.

Certainly, for a given method we expect to see trade-offs between these criteria. For example, a faster method may lead to a potential that produces lattices of poorer quality.

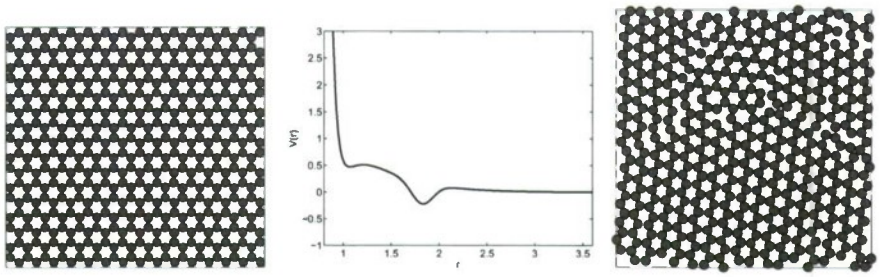
Laboratory techniques for tuning interaction potentials motivated Rechtsman *et al* [23] to consider physically realizable potentials as basis functions for the desired potential, where the basis functions contain parameters that allow for tuning the shape of the total potential obtained from their sum. For the case when the desired final configuration of particles is the honeycomb lattice, [23] proposed an interaction potential consisting of the sum of a Lennard-Jones potential, an exponentially decaying potential, and a Gaussian-shaped potential, parameterized in the following way:

$$V_{\mathsf{HC}}(r; a_0, a_1, a_2, a_3) = \frac{5}{r^{12}} - \frac{a_0}{r^{10}} + a_1\, e^{-a_2 r} - 0.4\, e^{-40(r-a_3)^2}, \tag{1}$$

where a_0, a_1, a_2, and a_3 are four free parameters that can be tuned to adjust the shape of the potential.

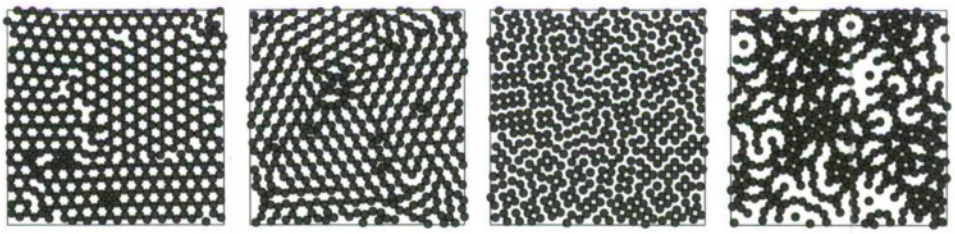
A solution (there are many) to the self-assembly problem has been provided by [23]: namely, choosing $a_0 = 5.89$, $a_1 = 17.9$, $a_2 = 2.49$, and $a_3 = 1.823$ in the expression for V_{HC}. A sample lattice obtained when cooling a system of particles using these parameters for the interaction potential is shown in Figure 1(c). The honeycomb lattice is the dominant structure in the lattice, although there are still visible defects that arise due to the finite duration of the cooling schedule. The cooling simulation used to obtain this lattice, as well as all other simulations referred to in this paper, were performed using periodic boundary conditions.

A reasonable question to ask at this point is: "How difficult is the self-assembly problem?" Experience shows that although easy to state, the self-assembly problem is difficult to solve in that solutions that lead to the honeycomb lattice are difficult to find and possibly very fragile. For instance, adding small perturbations to the parameters in [23] for the honeycomb potential leads to the configuration in Figure 2(b) in which the honeycomb structure is less pronounced. The



(a) A perfect honeycomb lattice. (b) The potential produced by [23]. (c) A sample lattice obtained using the potential provided by [23].

FIG. 1: The self-assembly problem entails finding an interaction potential that leads to the formation of a honeycomb lattice (a). [23] used a simulated annealing optimization procedure to generate a potential (b). A sample lattice formed using this potential is shown in (c) and exhibits defects that result from the finite duration of the cooling simulation.



(a)$a_0 = 5.89$, $a_1 = 17.9$, $a_2 = 2.49$, $a_3 = 1.823$. (b) $a_0 = 5.89$, $a_1 = 17.9$, $a_2 = 2.49$, $a_3 = 1.89$. (c) $a_0 = 5.0$, $a_1 = 17.0$, $a_2 = 2.0$, $a_3 = 1.5$. (d) $a_0 = 6.0$, $a_1 = 18.0$, $a_2 = 3.0$, $a_3 = 2.0$.

FIG. 2: The lattices shown here are obtained by slowly cooling 484 particles using the interaction potential $V_{\mathsf{HC}}(r; a_0, a_1, a_2, a_3)$ provided in Equation 1 with the parameter values indicated. Figure 2(a) uses the parameters proposed by [23], while Figure 2(b) indicates how fragile the honeycomb lattice is with respect to slight changes in the parameters. Figures 2(c) and 2(d) indicate the range of structures obtained for generic values of the parameters.

amorphous configurations shown in Figures 2(c) and 2(d) indicate the range of structures that are possible for generic values of the parameters.

We have identified the honeycomb lattice as the target lattice in the self-assembly problem because it the simplest lattice structure that represents a non-trivial case for self-assembly. The triangular lattice is easily and robustly assembled by a straightforward Lennard-Jones interaction potential. This fact can be readily demonstrated through direct simulation; however, it is interest-

ing that a rigorous mathematical proof that the triangular lattice is the global energy minimizer and ground state of a particle system with a pairwise Lennard-Jones interaction potential was provided only recently by Theil in [25]. Generating a potential that robustly induces the formation of a square lattice in particle simulations is also a simple matter using the geometric method to be presented later. Producing the honeycomb lattice using only a radially symmetric potential is more troublesome chiefly because regions of the domain tend to form the triangular lattice – a lattice that competes strongly with the honeycomb lattice. This is due to the fact that the triangular and honeycomb lattices have identical distances to nearest neighbors, and the triangular lattice is formed by simply adding a particle to the center of each hexagonal cell in the honeycomb lattice.

For concreteness, we constrain the parameter space over which we search for parameter values in $V_{\mathsf{HC}}(r; a_0, a_1, a_2, a_3)$ such that

$$4.0 < a_0 < 8.0, \qquad 0 < a_1 < 30.0, \qquad 0 < a_2 < 3.8, \qquad \text{and} \qquad 1.25 < a_3 < 2.25\,. \tag{2}$$

These ranges yield a large class of function shapes over which the search is performed.

Rechtsman and co-workers developed two computational algorithms to find potentials that lead to the self-assembly of particles into a given target lattice [23, 26].

The first optimization scheme chooses the shape of the potential so that the energy difference between the target lattice configuration and the configuration of the competitor lattices is maximized. The optimization is performed while ensuring mechanical stability by allowing only real phonon frequencies. This method is purely static in that it seeks to ensure that the target lattice configuration is energetically the most preferred final state; the method does not incorporate information about the dynamics of the particles as they tend *towards* this final configuration.

The second optimization scheme considered by Rechtsman *et al* concentrates on choosing potentials so as to maximize the stability of the target lattice near its melting point, while also requiring stability of the the target lattice with respect to changes in density, and mechanical stability by ensuring that the phonon frequencies are real. After placing particles into the target lattice configuration, a short molecular dynamics simulation is performed at a fixed temperature just below the melting temperature of the lattice. The deviation of the the final configuration from the initial target lattice is computed using the Lindemann parameter,

$$\mathsf{LP} := \sqrt{\frac{1}{N}\sum_i \left(\mathbf{r}_i - \mathbf{r}_i^{(0)}\right)^2 - \left(\frac{1}{N}\sum_i \left(\mathbf{r}_i - \mathbf{r}_i^{(0)}\right)\right)^2}\,, \tag{3}$$

where N is the number of particles, and $\mathbf{r}_i^{(0)}$ and $\mathbf{r}_i$ are the initial and final positions of particle i,

respectively. A simulated annealing procedure is used to choose parameters in the expression for $V_{\mathsf{HC}}(r; a_0, a_1, a_2, a_3)$ that minimize the Lindemann parameter.

After generating a potential using these methods, Rechtsman *et al* carefully checked using Monte Carlo simulations that particles starting in a random initial configuration do indeed self-assemble into the target lattice. The determination as to whether or not the target lattice was formed, was made by visual inspection of the final configuration and deciding if the configuration has few enough defects to be considered a lattice, as well as checking the long range order by visually inspecting simulated Bragg diffraction patterns.

III. LATTICE QUALITY METRICS

As described in Section II, one of the criteria used to compare methods for generating interaction potentials is the quality of the resulting lattices. In the work of [23], a simple visual check of the resulting lattice was used to determine if the desired lattice (with a few defects perhaps) was obtained, and since their purpose was to show that the desired lattices *can* be obtained, this visual check was sufficient. The purpose of the present paper is to compare several methods according to the lattice quality criterium, thus we must first introduce objective methods for assessing lattice quality.

A prevalent metric for lattice quality is the *structure factor*, a quantity that assesses long range spectral order in the lattice using diffraction patterns. We have found that this quality metric is inadequate for our purposes, firstly because the structure factor does not provide a scalar value for quality, and secondly because we have observed that long range order is not necessarily strongly correlated with visual perception of lattice quality – a lattice with glaring defects may still exhibit a high degree of long range order, for example, while a lattice that has weak long range order due to a grain boundary may have excellent local lattice structure. Hence, we have developed two lattice quality metrics that mimic nearly as possible the visual assessment of lattices. When the human eye looks at a lattice and makes a judgement with respect to lattice quality, the emphasis is on order within sub-regions of the entire lattice. A lattice that consists of two perfectly formed sub-lattices that have a domain wall where they meet will be judged by the eye to be quite well-formed. Thus, although long range order is important, a determination of local ordering is crucial for assessing lattice quality in a manner similar to the eye. The metrics that we use to quantify lattice quality have this local feature.

The two lattice quality metrics we present here are called the *Template Measure* and the *Defect*

Measure. In both cases, a lower value of the metric corresponds to a higher quality lattice.

A. Template Measure

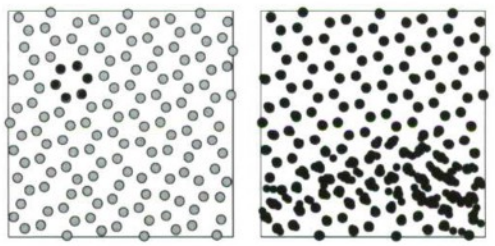
The *Template Measure* uses a small segment of the target lattice as a template with which to locally compare nearby lattice particle positions for each particle in the given lattice. In the honeycomb lattice, a suitable template may be one hexagonal cell composed of six particles, or points. For each particle in a given lattice configuration, the first point in the template is pinned to the particle, and the template is then rotated to find the best fit to other nearby particles. As the template is rotated, each point in the template is paired with the nearest particle in the given lattice. The angle of rotation of the template that produces the least deviation between the template points and lattice particles is considered the best fit. Once this best fit position of the template has been found for each particle in the lattice, the Template Measure (TM) is obtained by summing the deviation in the positions of the template points and lattice particles from the best fit for each particle in the lattice:

$$\mathsf{TM} := \frac{100}{N}\sum_{p=1}^{N}\min_{\theta}\left[\sum_{i=2}^{c}\left(\mathbf{r}_{i,p}^{\theta} - \mathbf{r}_{i,p}^{\theta,\text{template}}\right)^2 + n_{p,\text{extra}}\right], \tag{4}$$

where the index p ranges over all N particles in the given lattice, θ is the angle of rotation of the template, $\mathbf{r}_{i,p}^{\theta,\text{template}}$ is the position of the ith point in the template when the template is attached to particle p and rotated by angle θ, $\mathbf{r}_{i,p}^{\theta}$ is the position of the particle in the given lattice that is closest to $\mathbf{r}_{i,p}^{\theta,\text{template}}$, and c is the number of points in the template (c = 6 for the honeycomb cell template). Notice that since the first point in the template is pinned to the given lattice particle their positions are equal, that is $\mathbf{r}_{1,p}^{\theta} = \mathbf{r}_{1,p}^{\theta,\text{template}}$, and hence the sums over the template points need not consider this first template point. The extra term, $n_{p,\text{extra}}$, is a count of any extra particles in the given lattice that fall inside the hexagonal template, but are not paired with any of the template points. In this way, the best fit of the template seeks not only to match the particle locations, but also the void within the hexagon. This 'opacity' of the template ensures that defects that arise due to the formation of the triangular lattice that has a particle located at the center of the hexagon will be penalized. The prepended scaling factor of $100/N$ is not strictly necessary in the Template Measure, but is included for convenience so that the resulting lattice quality values can be interpreted as a measure of the defectiveness *per* particle, and have a magnitude in a range from zero to roughly 100.

An illustration of how the Template Measure is implemented in practice is provided in Figure 3.

In Figure 3(a), a single hexagonal honeycomb cell template is attached to a particle in a honeycomb lattice and rotated until a best fit with the surrounding particles is achieved. Repeatedly attaching the template to each particle in the lattice and realigning as in Figure 3(b) quickly reveals the locations of the defects.



(a) (b)

FIG. 3: Illustration of the Template Measure. (a) A template consisting of a single hexagonal cell is pinned to a particle in the given lattice, and rotated to the position which minimizes the distance between points in the template and the nearest particles in the lattice. (b) Fitting the template cell to each particle in the lattice and rotating to find the best fit, quickly reveals the locations of defects.

B. Defect Measure

The second lattice quality metric we present is the *Defect Measure*. The Defect Measure provides a weighted count of all the defects in the local neighborhood of each particle in the given lattice. The types of defects considered are shown in Figure 4, and include displaced, missing, and extra particles. Note that all of the possible defects, including global defects, are taken into account by the types of defects shown. For example, extended grain boundaries are taken into account by contributions to the Defect Measure from locally displaced, missing, and extra particles.

Recall that in the perfect honeycomb lattice, the distance to nearest neighbors is unity, while the distance to second nearest neighbors is $\sqrt{3}$. In order to calculate the Defect Measure, we consider only a small circular region around each particle of radius $(1+\sqrt{3})/2 \approx 1.366$ so that only three nearest neighbors, each at unit distance, should be included. Then, using the locations of all particles actually located within this circular region for the given lattice, the contributions from each of the defects is calculated, and then summed with the specified weighting, for each particle in the given lattice to provide a measure of the quality of the lattice as a whole. The weights attributed to each type of defect may be chosen according to the desired properties for the

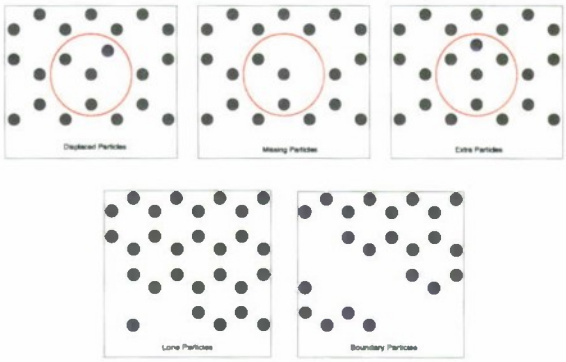


FIG. 4: To compute the Defect Measure, the number of defects in a circular neighborhood of each particle are weighted and summed. The types of defects used in computing the Defect Measure are displaced, missing, and extra particles, as well as lone and boundary particles.

self-assembled lattice. These weights may emphasize correct local densities, correct alignment of particles, correct inter-particle distances, or any other customized weighting.

Accordingly, the Defect Measure (DM) is given by

$$\mathsf{DM} := \frac{100}{N}\sum_{p=1}^{N}\Bigg[\omega_{\text{displaced}}\left(\sum_{j}\chi(d_{pj})\cdot(d_{pj}-1.0)^2\right) + \omega_{\text{missing}}\cdot n_{p,\text{missing}} + \omega_{\text{extra}}\cdot n_{p,\text{extra}} + \omega_{\text{lone}}\cdot \eta_{p,\text{lone}} + \omega_{\text{boundary}}\cdot \eta_{p,\text{boundary}}\Bigg]$$

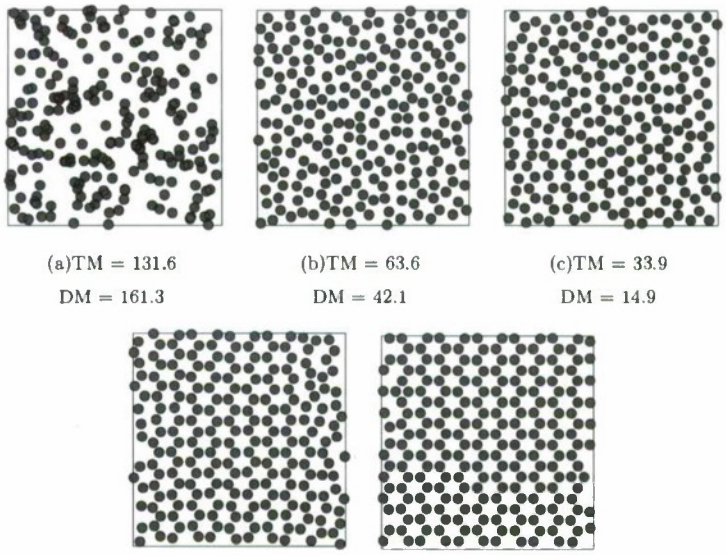
where the ω's are the weights attributed to each type of defect; the n_p's are the integer number of missing and extra particles within the small circular region surrounding particle p; the η_p's are indicator functions that equal unity if particle p is a lone or boundary particle and zero otherwise; the index p ranges over all the N particles in the given lattice; the index j ranges over the particles contained within the small circular region surrounding particle p; and d_{pj} is the positive distance between particle p and particle j. As with the Template Measure, the scaling factor of $100/N$ is included so that the quality values are normalized by the number of particles and fall within a convenient range. In the first term, $\chi(\cdot)$ is a smooth cutoff function that decreases from 1 to 0 at the outer edges of the circular region, or more precisely, as its argument increases from 1.275 to 1.366. This smooth cutoff function ensures that the Defect Measure remains continuous with respect to motion of the particles and as the number of particles entering the circular region of each particle fluctuates.

More explanation of the Defect Measure and its applications can be found in [27], where there is also a discussion of other lattice quality metrics.

For the purposes of this paper, we use the following weight values for the Defect Measure:

$$\omega_{\text{displaced}} = 1.0\,, \quad \omega_{\text{missing}} = 1.5\,, \quad \omega_{\text{extra}} = 0.8\,, \quad \omega_{\text{lone}} = 2.0\,, \quad \omega_{\text{boundary}} = 0.1\,.$$

Sample values of both the Template Measure and the Defect Measure for four lattices of varying quality are shown in Figure 5.



(a)TM = 131.6 DM = 161.3 (b)TM = 63.6 DM = 42.1 (c)TM = 33.9 DM = 14.9

(d)TM = 19.9 DM = 9.6 (e)TM = 10.2 DM = 1.6

FIG. 5: Sample values of the lattice quality metrics are shown for a range of lattices. In each figure, the value of the Template Measure (TM) and the Defect Measure (DM) are provided. Both measures assign a lower value to a lattice of higher quality.

IV. OBJECTIVE FUNCTIONS

A natural way in which to find solutions to the self-assembly problem is to search for a potential that optimizes an appropriate objective function. The objective function must be well-chosen so that 1) optimizing the objective correlates well with the formation of the desired lattice, and 2)

the objective function is not prohibitively expensive to evaluate. In this section, we present the objective functions that will be used in the potential generation methods under comparison.

A. The Lindemann Parameter Objective Function

As noted in Section II, the Lindemann parameter is a measure of how much a lattice configuration has deviated from the perfect lattice configuration after a short simulation. Consequently, the Lindemann parameter cannot be used as a metric for lattice quality *per se* since it measures how far a lattice has deviated when initially placed in the perfect target lattice configuration, whereas a lattice quality metric must assess the quality of a lattice obtained after a slow cooling process from an *arbitrary* initial condition. The Lindemann parameter can nevertheless be used as an objective function that must be minimized in order to find potentials that lead to formation of the target lattice – an approach that was first proposed and developed by [23]. Intuitively, minimizing the Lindemann parameter produces potentials that stabilize the honeycomb lattice to thermal agitation. Used as an objective function, the Lindemann parameter is advantageous since it is much faster to evaluate than other objective functions that require much longer molecular dynamics simulations.

As implemeted in this paper, the Lindemann parameter objective function is computed for a given interaction potential by placing 72 particles in the honeyomb lattice formation, and then performing a brief simulation at a temperature very near the melting temperature of the lattice. The use of 72 particles allows for the construction of a lattice consisting of 30 honeycomb cells that form an infinite honeycomb lattice when the 72-particle configuration is used to tile the plane. The value of the Lindemann parameter is calculated using the initial and final lattice configurations. The wall clock time to compute the Lindemann parameter on a single CPU is approximately 2 seconds.

Optimizing the Lindemann parameter is, however, an *indirect* method in that the quanitity being optimized is not the quantity that will be used to determine the final quality of the potential. A more direct approach is to explicitly optimize the lattice quality metrics. Although the quality metrics require a slow cooling simulation and are consequently more expensive to evaluate, optimization of the quality metrics guarantees optimization of lattice quality. Indeed, an important observation made in this paper is that optimizing the Lindemann parameter is only moderately correlated with lattice quality – potentials can be found that produce a low value of the Lindemann parameter, yet when tested in a slow cooling simulation produce lattices of poor quality.

B. Quality Metric Objective Functions

We also consider direct evaluation of the Template Measure and Defect Measure quality metrics as objective functions. To evaluate these objective functions, we start with a regular Cartesian grid of 64 particles with a spacing that provides a particle density equal to the density of the honeycomb lattice. These particles are initialized with a temperature well above the lattice melting point and then this temperature is slowly reduced until the particles freeze into a lattice configuration. The quality metrics are computed using only the final lattice configuration. Compared with the computation of the Lindemann parameter, these cooling simulations are expensive to carry out; a single call to a quality metric objective function on a single CPU takes approximately 70 seconds.

It should be noted that the computational expense associated with evaluation of the objective functions arises almost entirely from the molecular dynamics simulations. Computation of the actual value of the Lindemann parameter, the Template measure, or the Defect measure after the final configuration has been obtained, requires less than a tenth of a second.

C. Properties of the Objective Functions

The objective functions presented have important features that influence the effectiveness of the optimization schemes employed. One of the chief contributions of this paper is the use of a trend optimization scheme that is better suited to these objective functions over a standard simulated annealing optimization procedure.

The fact that the objective functions require evaluation times on the order of seconds and minutes implies that any optimization method that requires many thousands of evaluations to search the four-dimensional parameter space will necessitate a computation time measured in hours and days. The time taken to run an optimization algorithm will be spent almost entirely evaluating the objective functions, and overhead computation required by the optimization scheme in choosing the next location at which to evaluate the objective, for example, is practically negligible in comparison. The trend optimization method we propose is particularly well-suited for this situation in which the objective functions are expensive to evaluate.

Figure 6 displays repeated evaluations of the Lindemann parameter objective function as each of the parameters in $V_{\mathsf{HC}}(r; a_0, a_1, a_2, a_3)$ is varied in turn, while the remaining parameters are held fixed at the values provided by Rechstman *et al* (namely, a_0=5.89, a_1=17.9, a_2=2.49, and a_3=1.823). Each circle in the plots corresponds to a single evaluation of the the Lindemann

parameter objective function. The red line in each plot represents a trend line that is computed by averaging over 100 samples taken at each of 600 regularly spaced intervals along the axis of the varied parameter.

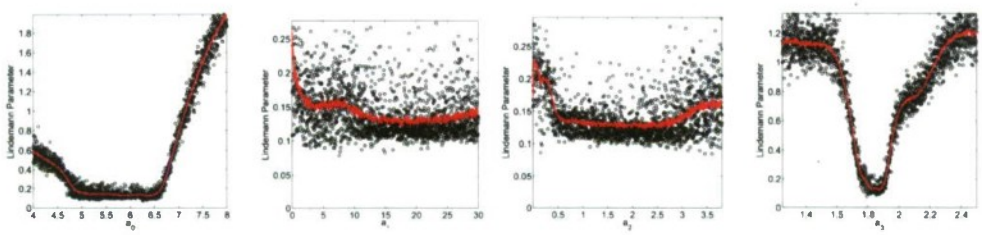


FIG. 6: Each circle in the figures above corresponds to a sample evaluation of the the Lindemann parameter objective function. The parameter shown on the x-axis is varied over the indicated domain, while the remaining parameters are held fixed at the values provided by Rechstman *et al* (a_0=5.89, a_1=17.9, a_2=2.49, and a_3=1.823). The vertical axis in each plot represents the value of the Lindemann parameter. Note that the vertical scale in each case is different. The red line is computed by averaging over 100 trials at each location, and reveals a smooth trend in the data.

Our first observation is that due to randomness induced by the initial conditions and simulation at constant temperature, the objective function does not produce repeatable values for a fixed set of parameter values. Repeated evaluation of the objective functions for the same parameter values leads to a wide range of output values. For this reason, the objective functions are not strictly *functions*, although we continue to use this term. More correctly, due to the randomness introduced by the initial conditions and the simulation with a thermostat, we can think of the objective functions as assigning a one dimensional probability distribution to each set of values that define the potential. Thus, a call to the objective function returns a value drawn from the probability distribution specified by the parameters. We shall see that the trend optimization method exploits the fact that the expectation values of the probability distributions vary smoothly with respect to changes in the parameters of the potential.

Examining the evaluations of the objective functions reveals that they are noisy with respect to variations in the parameters. The functions are not smooth and certainly no derivatives of the objectives are available. Nevertheless, averaging over many evaluations for a fixed set of parameter values reveals that the objective functions do possess a smooth and slowly varying trend. As indicated in Figure 6, the smoothness of the average values (shown in red) indicate that, when averaged, the objective functions do admit a sensible notion of a minimum.

In summary, the salient features of the objective functions in the self-assembly problem are the following:

1. They are expensive to evaluate;
2. They are highly variable—repeated evaluations of the objective functions for the same input potential yields a broad range of values;
3. They are highly non-smooth with respect to changes in the parameters;
4. They have a smooth trend when averaged over many evaluations.

Objectives with these features are often encountered in optimal design problems in which evaluation of the objective function for a specific set of parameter values requires completion of an actual laboratory experiment, or the execution of a computationally expensive simulation using random initial conditions [28, 29]. Optimization of these objective functions is clearly not tractable using standard gradient-based methods. In this paper, we implement an optimization scheme ideally suited for objective functions of this type that has allowed us to construct an efficient procedure for solution of the self-assembly problem, an approach that we call *trend optimization.*

V. OPTIMIZATION METHODS

In the previous section, various objective functions have been introduced that can be used in the search for lattice-forming potentials. In this section, we present two methods for finding optimal values of these objectives. First, we briefly discuss simulated annealing which was used by [23] in the baseline approach. Second, we present in detail the trend optimization approach.

A. Simulated Annealing

Simulated annealing mimics the ability of a thermal process to search a configuration space to find the ground state. We begin the search by evaluating the objective at an initial point, x_0, in the parameter space and denote the value of the objective function at this location by E_0. The search then proceeds by jumping to new locations in a random walk. The probability, p, of making the jump from x_i to a new location x_{i+1} is determined by

$$p = \begin{cases} 1 & \text{if } E_{i+1} \leq E_i\,, \\ e^{-(E_{i+1}-E_i)/(T)} & \text{if } E_{i+1} > E_i\,. \end{cases} \tag{5}$$

The small probability of moving to a location where the objective is in fact higher than the current value, allows for the search to escape from a local minimum. In analogy with true simulated annealing in metals, the temperature, T, is slowly reduced so that at first the search eagerly traverses the search space by easily escaping from local minima, and then settles to a specific local minimum as the temperature decreases. This simulated anncaling optimization scheme was implemented using the GNU Scientific Library `GSL_SIMAN_SOLVE()` routine [30].

When applied to the self-assembly problem, the simple simulated annealing method described here fails to converge to a sensible minimum because of noise in the objective functions. Understandably, the method gets stuck in ephemeral local minima that appear and disappear due to noise in the objectives. Convergence can be obtained if the objectives are sufficiently smoothed. Smoothing necessitates averaging over at least 20 independent simulations and thus requires added computational expense, and even then a large proportion of optimizations fail to converge within a reasonable number of objective evaluations.

Admittedly, the simulated annealing method described here represents a very simple approach, and more complex methods involving adaptive search, for example, could be employed. The straightforward simulated annealing approach serves, therefore, as a modest but easily understood baseline method. We have investigated the simulated annealing method using many different cooling regimens including adaptive methods, and have found little improvement in each case.

B. Trend Optimization

Trend optimization is well-suited for problems in which the objective functions are expensive to evaluate, derivatives of the objectives are not available, and the objectives are noisy yet exhibit a simple underlying trend when averaged over many evaluations.

In the trend approach, we use a few well-distributed evaluations of the objective function to generate a smooth global approximation to the averaged objective function. This smooth approximating surface, referred to as the *surrogate*, attempts to capture the underlying smooth trend in the noisy objective evaluations. The optimization then proceeds by finding the optimal values of this surrogate function that is computationally cheap to evaluate. Trends in the surrogate quickly reveal regions of the parameter space in which the optimal parameters are most likely to be found and hence the search is greatly accelerated. The insight provided by the smooth surrogate function then informs the choice of parameters at which subsequent evaluations of the objective should be made.

Booker *et al* [31] combined the speed and facility with which the trend approach zooms in on regions of optimal parameter values with the rigorous convergence guarantees of patterned search methods developed previously by Torczon [32], to produce what is known in the literature as the *Surrogate Management Framework*. In this two-pronged approach, the trend method is used in the *global search* step of the patterned search method to accelerate the search, while the use of a *polling* step on a patterned conceptual grid provides the guarantees of convergence. In this seminal paper, Booker *et al* also applied the Surrogate Management Framework approach to the optimal design of a helicopter rotor blade with thirty-one design variables. More recently, Audet *et al* [33, 34] have provided a generalization of Torczon's pattern search method, which they refer to as Mesh Adaptive Direct Search, and have applied it to optimization of the chemical treatment of discarded potliners to minimize release of toxic waste in the production of aluminum [35]. Mesh Adaptive Direct Search has since been incorporated into the Surrogate Management Framework by Marsden *et al* in [36].

The range of design problems to which the trend optimization approach has been applied is starting to grow. Marsden *et al* have used the Surrogate Management Framework in the optimal design of airfoils to reduce noise generated in the trailing turbulent flow [37], and a computational framework has been provided in [36] for optimizing design of surgeries for improved blood flow and cardiovascular geometry. Siah *et al*, have used Kriging surrogate models to design optimal configurations and shapes of automobile antennae to minimize electromagnetic coupling [38], and Raza *et al* have compared methods for generating surrogate functions in a design problem seeking the optimal arrangement of fuel rods in a liquid metal reactor [39].

A unifying theme in all these applications is the parsimonious way in which trend optimization is able to optimize expensive, noisy objective functions. Moreover, trend optimization is robust to noise in that the trends approximate the general shape of the objective function with smooth surfaces that quieten the noise and anomalous evaluations of the objective function. Hence, trend-based approaches survey the parameter landscape for large depressions, and are not distracted by superficial deep spikes that may arise due to noise. Because the surrogate prioritizes regions that consistently perform well, rather than a single "flash-in-the-pan" evaluation, the parameter values returned by the trend are robust to uncertainties and are more likely to reliably reproduce near-optimal values of the objective upon repeated evaluation.

Trend optimization can be performed in a coarse-to-fine hierarchical manner by recursively building a hierarchy of trend-fitting surfaces. Each successive iteration of the procedure yields a new trend that focuses on the most optimal region of the search space. Successive trend surfaces

utilize all objective function evaluations obtained in previous iterations to more accurately model the objective function. After an initial global trend is developed, the search area is refined to the area surrounding the global minimum of the surrogate – recall that since the surrogate is smooth and cheap to evaluate, the global minimum of the surrogate can easily be found. Refinement of the search area helps to ensure that subsequent function evaluations are chosen in locations that are most relevant and promising. As the search area becomes more refined, successive iterations may use a larger basis of fitting functions, or use more sophisticated trend construction methods to more accurately pinpoint the location of the minimum. These features of iterative trend optimization yield a hierarchy of coarse to fine trends that enable the method to initially make large strides toward the optimal value, and then to focus ever more tightly on the exact location of the optimal value.

In our discussion thus far, it remains to be described how the surrogate functions are generated from a small number of function evaluations. This topic lies within the province of *data approximation* and fills a large body of literature. Needless to say, there are a great number of interpolation and fitting approaches available. Popular methods include polynomial interpolation, splines, Kriging, distance-based interpolation, linear and nonlinear regressions, radial basis functions, neural networks, and kernel-based approaches [40–42]. For a thorough survey, please see the monograph by Hastie [43].

Reviewing the Lindemann parameter objective evaluations depicted in Figure 6, we plainly see that an interpolating scheme is not appropriate for the noisy objective functions of the self-assembly problem. For this reason, we have chosen the *ridge regression* method for generating surrogate functions that is particularly well-suited for noisy data in high dimensions. Ridge regression was originally developed by Tikhonov (hence the method is sometimes referred to as Tikhonov regularization) for ill-conditioned linear regression problems [44]. His approach was to introduce diagonal stabilization to regularize the linear interpolation system. The added regularization improves the conditioning, and introduces smoothing.

In the current context, we are provided with a vector of M noisy measurements, $\mathbf{y} = [y_1, \cdots, y_M]$, of the objective function at the vector of locations $\mathbf{x} = [x_1, \cdots, x_M]$. We want to find a smooth function that best represents the smooth trend in this data. This trend, denoted $T(x)$, is constructed as the weighted sum of basis functions:

$$T(x) = \sum_{k=1}^{M} c_k \Phi(x, x_k),$$

where we must now solve for the vector of coefficients $\mathbf{c} = [c_1, \cdots, c_M]$. Gaussian radial basis

functions are chosen for the basis since they are well-suited for regression problems in high dimensions [45–47]. To be specific, we choose radial basis functions of the form

$$\Phi(x, x_j) = \phi(\|x - x_j\|_2)$$

where

$$\phi(r) = e^{-(\epsilon r)^2}, \quad r \in \mathbb{R}.$$

Proceeding in the standard manner for linear regression, the vector of coefficients, $\mathbf{c}$, in the expression for the trend are obtained by solving the linear system

$$\mathbf{A}\mathbf{c} = \mathbf{y} \tag{6}$$

where the elements of the square symmetric matrix $\mathbf{A}$ are given by

$$\mathbf{A}_{ij} := \Phi(x_i, x_j).$$

This procedure assumes that $\mathbf{A}$ is full rank. In ridge regression, the conditioning of $\mathbf{A}$ is improved by adding diagonal regularization. The linear system in (6) above is replaced by

$$\left(\mathbf{A} + \frac{1}{2\omega}\mathbf{I}\right)\mathbf{c} = \mathbf{y} \tag{7}$$

in which $\omega \in \mathbb{R}$, and $\mathbf{I}$ is the identity matrix. Regularization is obtained at the expense of introducing the new free parameter ω. In the traditional usage of ridge regression, the analyst must choose an optimal value of ω that balances the need for improved conditioning with the desire to keep the departure from the least-squares solution small. In our context, since we are not immediately concerned with conditioning, we use ω to control the amount of smoothing introduced. Smaller values of ω yield larger smoothing, while larger values of ω ensure increased pointwise accuracy to the noisy data. In practice, this approach is straightforward and robustly produces smooth trends to noisy objective functions. In Figure 7, an illustration of smooth trends generated using ridge regression are given for noisy evaluations of the Lindemann parameter. It must be remembered though that these surrogates are constructed as one-dimensional curves for illustration, whereas in the full self-assembly problem the surrogate functions are smooth four-dimensional hypersurfaces.

The ridge regression method can be extended to include adaptive control of the amount of smoothness introduced in response to local conditions. This *local ridge regression* approach is implemented by replacing Equation (7) with

$$\left(\mathbf{A} + \mathbf{diag}\left[\frac{1}{2\omega_1}, \cdots, \frac{1}{2\omega_M}\right]\right)\mathbf{c} = \mathbf{y}$$

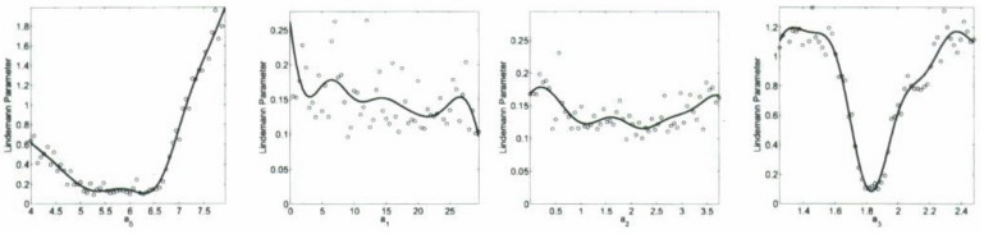


FIG. 7: Generating surrogate functions for noisy data. Each circle represents an evaluation of the Lindemann parameter as in Figure 6. The solid trend line is generated using Gaussian radial basis functions and ridge regression.

where the vector of M free parameters $[\omega_1, \cdots, \omega_M]$ can be chosen independently to adjust the amount of smoothing local to each measurement. In this way, a low value of ω can be chosen to locally increase the smoothing in a region in which the noise in the data is high, and similarly, a large value of ω can be chosen for measurements in regions in which there is high confidence in the data and noise is low. This facility is not currently implemented in our trend optimization algorithm, but is mentioned here to indicate the flexibility that the ridge regression method affords.

As previously mentioned, there are many candidate approaches for trend fitting besides ridge regression that could be uses in the self-assembly problem. For instance, simple quadratic fitting is very easily implemented and guarantees convexity of the surrogate. In practice, this method also works remarkably well for the objectives of the self-assembly problem.

Trend Optimization Algorithm used in the Self-Assembly Problem

In using the trend optimization approach for the design of the potentials in the self-assembly problem, we have implemented it without coupling to a Direct Search method. Consequently, we lose rigorous guarantees of convergence, but in practice trend optimization alone performs remarkably well and reliably converges to an optimal value, as will be demonstrated over repeated trials.

Throughout the computations, we scale the four-dimensional parameter search space to form a unit hypercube. We use the hierarchical approach described above with three levels of recursion. Hence, during each optimization three trend surfaces will be constructed at finer and finer resolution. The basic steps of the procedure are as follows.

Let U_1 denote the four-dimensional unit hypercube of the parameter space to be searched, and let H be the number of levels in the trend hierarchy (we use $H = 3$).

FOR each iteration in the hierarchical trend optimization method, indexed by $k = 1, \cdots, H$:

Step 1. **Generate Sample Locations:**

Select M points, $[x_1^k, \cdots, x_M^k]$, from U_k using Latin Hypercube Sampling [48].

Step 2. **Evaluate the Objective:**

Evaluate the objective at each of the M locations x_i^k, and store each corresponding result in y_i^k.

Step 3. **Build the Trend:**

Use all data $\left\{(x_i^j, y_i^j) : j = 1, \cdots, k\,;\, i = 1, \cdots, M\right\}$ obtained during the optimization so far to construct the trend surface T_k via ridge regression.

Step 4. **Optimize the Trend:**

Quickly find x_*^k, the location in the parameter space that globally minimizes the trend surface T_k.

Step 5. **Refine the Search Domain:**

Generate a new search domain, U_{k+1}, by reducing the size of the current search domain, U_k, by a factor of 2 along each dimension centered about the point x_*^k.

END

After H iterations, declare $x_* := x_*^H$ as the parameter location that minimizes the objective function. If desired, the objective function can be evaluated repeatedly at x_*, and then averaged, to generate y_*, the expected value of the objective at x_*.

The most computationally expensive step is Step 2, the evaluation of the objective function. In comparison, optimization of the surrogate performed in Step 4 is extremely fast. To effect Step 4, we simply evaluated the surrogate at 5000 points randomly distributed throughout the search space and selected the point that produces the lowest value of the surrogate. The surrogate provides repeatable values so there is no need to average over many evaluations of the surrogate.

At the completion of the algorithm, the total number of required objective function evaluations is $M \cdot H$, that is, M objective evaluations during each of the H trend iterations. An important point is that, at each iteration, the M objective evaluations required by Step 2 are *independent*, meaning, therefore, that *these M evaluations can be computed in parallel.* In our optimization source code, we have parallelized Step 2 so that the total wall clock time required for the entire optimization algorithm to complete is $H \cdot T_{\text{obj}}$, where T_{obj} is the CPU time required to run the

optimization with a single objective function evaluation. An important point, therefore, is that if the number of available computing processors is greater than M (and we consider any overhead communication costs between the M processors as negligible), then the execution time of the algorithm is independent of M. Thus, trend optimization provides an effective method to harness parallel computation resources for fast global optimization.

Since in our specific implementation of the trend algorithm we recursively generate three surrogate functions, and since the evaluation of even the most expensive objective function is approximately a minute; the trend algorithm terminates after just three minutes of computation on 40 processors having used information gathered from 120 objective evaluations. The CITerra cluster at Caltech has 4096 processes that could conceivably be used to find an optimal solution; however, we have observed that trend optimization provides reliable convergence when M (the number of objective evaluations made at each step) is ~ 40 or above, so that a far more modest number of processors is actually required. In summary, the trend optimization scheme as we have applied it to the self-assembly problem is able to provide parameters that optimize the most expensive objective functions in just over three minutes when run on a cluster of 40 parallelized processors.

It should be noted that the hundredfold speed-up obtained by trend optimization over simulated annealing in the time taken to generate potentials (as stated in the introduction), is assessed using the *total* CPU time summed over all processors, and not the wall clock time. The speed-up is attributed to the robust manner in which trend optimization accelerates search of a noisy objective with a smooth trend. Hence, the speed-up attributed to the facility with which the trend optimization admits parallelization of the computation represents an additional time savings over and above the hundredfold speed-up.

VI. METHODS FOR GENERATING POTENTIALS

The five methods for generating potentials that we compare in this paper are described below. The first is a heuristic geometric method that requires no computation. The other four methods utilize an optimization procedure.

A. Geometric Method

The geometric method (abbreviated as GM) that we present here is an optimization-free procedure that exploits differences between the geometry of the desired target lattice and competitor

lattices that we hope to discourage. The design of the potential is based on four main principles:

GM1. The potential must have local minima located at each of the radial distances to nearest neighbors in the desired lattice and nowhere else;

GM2. The potential may include local maxima at radial distances at which competitor lattices have nearest neighbors but the target lattice does not;

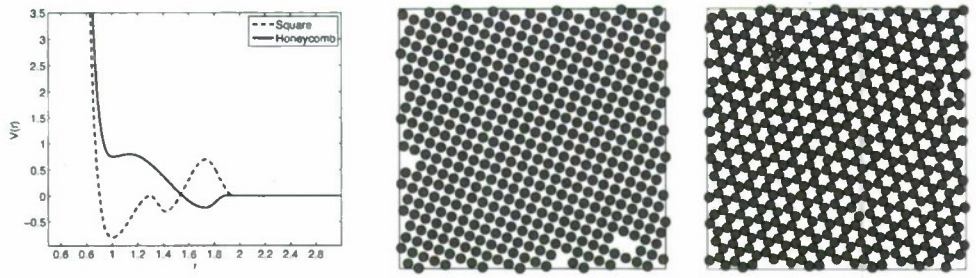
GM3. When the target and competitor lattices have nearest neighbors at the same radial distances, the energy levels of the local minima identified in **GM1** are chosen to energetically prefer the target lattice;

GM4. A Lennard-Jones potential is spliced into the potential at the origin to provide a hard core potential.

GM1 and **GM2** use information about the geometric structure of the target and competitor lattices to stabilize the target lattice, and to discourage a competitor lattice that has different distances to nearest neighbors. **GM3** uses energetics to discriminate between lattices that have identical distances to nearest neighbors and differ only in the numbers of particles at those distances.

When the target lattice is the square lattice, and the identified competitor is a triangular lattice, the fact that these lattices have different distances to nearest neighbors makes this method of potential generation a very robust approach. By explicit construction, the locations of the local minima and maxima discourage the triangular lattice and stabilize the square lattice. The shape of the potential generated for this case is shown in Figure 8(a). Notice in particular that the potential has a very simple form with local minima at distances of 1.0 and $\sqrt{2}$ to encourage the square lattice, and a local maximum located at $\sqrt{3}$ to discourage the triangular lattice. In simulation this potential performs remarkably well. The potential robustly forms the square lattice, and defects (except for voids) are seldom observed. A sample square lattice obtained with this potential is shown in Figure 8(b).

The honeycomb lattice represents a more difficult self-assembly problem since both this target lattice and the competitor triangular lattice have identical distances to nearest neighbors. The only difference between these lattices that must be exploited is the *different number of neighbors at each distance*. Hence, in **GM3**, the relative heights of the local minima are chosen to ensure that the target lattice represents a lower energy state than the competitor lattice and is thus more likely to form upon cooling.



(a) (b) (c)

FIG. 8: Potentials developed using the heuristic geometric method for both the square and honeycomb lattices are shown in Figure (a). Figures (b) and (c) show typical lattices obtained using these potentials.

The application of the GM method for the specific case of constructing a potential that favors the honeycomb lattice and discourages the triangular lattice proceeds as follows:

1. Recognize that both the triangular and honeycomb lattices have nearest neighbors at a distance of 1.0, and second nearest neighbors at a distance of $\sqrt{3}$. The number of neighbors at those distances (the lattice coordination numbers) for the honeycomb lattice are 3 and 6, while the triangular lattice has 6 and 6.

2. Assign local minima at distances of 1.0 and $\sqrt{3}$.

3. Since the triangular lattice has more particles at distance 1.0 than the target lattice, raise the first minimum at 1.0 with respect to the height of the minimum at $\sqrt{3}$.

4. Splice in a 6–12 Lennard-Jones type repulsive potential at the origin to simulate a rigid core.

5. Use cubic splines to piece together the repulsive core at the origin and the local minima at their various locations and heights.

One other consideration that we include when designing this potential is that we are careful not to make the local maximum between the first and second minima too high. This ensures that particles that are stuck in the triangular formation have increased likelihood of escaping to the lower potential well of the honeycomb lattice.

The cubic splines are constructed so as to have zero derivative at the local minima, thus ensuring that the potential is continuously differentiable on the whole positive real line. A sample potential constructed using this method for the generation of honeycomb lattice potentials is shown in

Figure 8(a). Notice that the potential has local minima located at distances of 1.0 and $\sqrt{3}$, with the first minimum located above the second minimum. Since the triangular lattice has more particles located at distance 1.0 than the honeycomb lattice, this ensures that the triangular lattice is a higher energy state than the honeycomb lattice. A sample lattice produced using this potential is shown in Figure 8(c). The fact that this heuristic geometric method is able to produce honeycomb lattices of this quality without having to use any computation and optimization is quite remarkable and until now has been overlooked in the self-assembly literature.

We refer to the geometric method as a heuristic method since we do not provide formulas or algorithms for choosing the relative heights of the local minima and maxima. The only prescribed constraints are the locations of the local minima that, by construction, ensure stability of the target lattice. In practice, we find that any sensible choice of the heights that avoids sharp gradients in the potential works reasonably well. In the implementation of our computer code, the user need only input the location and heights of the desired minima and maxima, and the spline fitting and splicing of the Lennard-Jones potential are computed automatically, which makes the approach very simple to execute. Given the simplicity of the geometric method, the ease with which it is implemented, and the relatively high quality of the resulting lattices, it represents a "rough-and-ready" approach that anyone needing to design a potential would do well to consider before pursuing more computationally expensive schemes.

Whereas the geometric method utilizes only the geometries of the static target lattice and competitor lattice, the optimization methods discussed next incorporate information about particle dynamics by optimizing parameters in the potential with respect to dynamic particle simulations.

B. Baseline Method

The baseline simulated annealing method of [23] uses the simulated annealing optimization procedure with the Lindemann parameter as the objective function. As mentioned previously, the simulated annealing procedure fails to converge to a meaningful minimum unless the objective function is averaged over many trials. We refer to this method with the abbreviation SA-LP20 to indicate that the simulated annealing approach is applied to the Lindemann parameter averaged over 20 independent evaluations. The large number of samples required to sufficiently reduce the noise in the objective function, means that it is impractical to apply the simulated annealing optimization procedure to the more expensive quality metric objective functions. The simulated annealing procedure is initiated with initial guess solutions chosen randomly and uniformly from

Summary of Potential Generation Methods			
Method	**Optimization**	**Objective**	**Cost** (s)
GM	Geometric method		0
SA-LP20	Sim. anneal.	Lindemann (20 sample avg.)	40
T-LP	Trend	Lindemann	2
T-TM	Trend	Template Measure	70
T-DM	Trend	Defect Measure	70

TABLE I: Each of the five methods for generating potentials is listed here with the associated computational cost of evaluating the objective function. Notice that the Geometric Method is not an optimization-based method and incurs no computational cost.

the parameter space described by the inequalities in line (2) of Section II.

C. Trend Optimization Methods

The remaining three methods that we consider utilize the trend optimization method applied to the Lindemann parameter, the Template Measure, and the Defect Measure as objective functions. These methods are abbreviated T-LP, T-TM, and T-DM respectively. Since the trend optimization method is well-suited to noisy objective functions, the objectives do not need to be averaged over many runs when evaluated.

A summary of all the potential generation methods under study is provided in Table I.

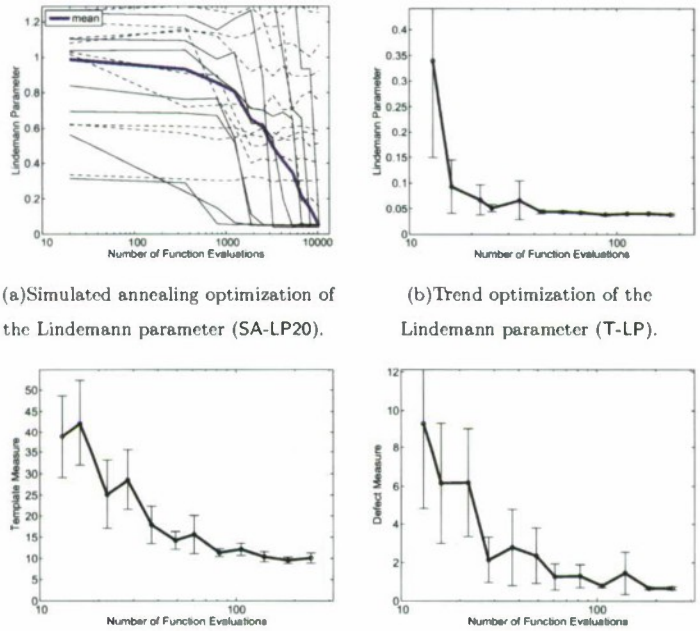
VII. RESULTS

In this section, we compare the effectiveness of the proposed methods for generating interaction potentials that lead to the self-assembly of the honeycomb lattice. In particular, in accordance with the comparison criteria enumerated in Section II, we seek answers to the following questions:

1. How much computational effort is required to generate the potentials?
2. What is the quality of the lattices generated by the potentials?
3. How reliably do the potentials form high quality lattices given uncertainty in the initial conditions of the particles?

Answers to these questions, as well as a comparison of the five potential generation methods discussed in Section VI, are summarized in the suite of plots shown in Figures 9 and 10. In

brief. Figure 9 shows the superior effectiveness of trend optimization over simulated annealing in minimizing the respective objectives as a function of the number of objective evaluations required, while Figure 10 shows the quality of the lattices obtained by the potentials that were generated by the optimizations.



(a)Simulated annealing optimization of the Lindemann parameter (SA-LP20).

(b)Trend optimization of the Lindemann parameter (T-LP).

(c)Trend optimization of the Template Measure (T-TM).

(d)Trend optimization of the Defect Measure (T-DM).

FIG. 9: Averaged objective function values versus number of objective function evaluations for four optimization methods are shown here. In the results for simulated annealing shown in (a), dashed lines indicate optimizations that failed to find an optimal value. The single bold line represents the mean of the optimizations that did converge to an optimal value (indicated by solid lines). Most notably, trend optimization reliably converges to a minimum value of the Lindemann parameter with a one-hundredfold reduction in the number of required objective function evaluations when compared with the simulated annealing approach.

The geometric method (GM) is the simplest method to implement. It does not require any objective optimization to search for parameters and consequently does not incur any computational cost. In order to evaluate the quality of the lattices produced by the geometric method, one-hundred cooling simulations were performed using the honeycomb potential shown in Figure 8(a). After each simulation, the Template Measure and the Defect Measure were computed. Averaging over

all these runs yielded the following scores:

1. Template Measure = 17.9 with standard deviation 3.4,

2. Defect measure = 11.8 with standard deviation 4.5.

Comparison with the lattice qualities obtained using the optimization-based methods as shown in Figure 10 indicates that these quality scores are exceptionally good, and that assiduous computational optimization is required to produce potentials that produce lattices of higher quality.

The four optimization-based methods (SA-LP20, T-LP, T-TM, and T-DM) were each run for an increasing number of allowed objective function evaluations. For each number of objective function evaluations, each trend optimization method was executed in twenty independent trials, with each trial generating parameters for the interaction potential that seek to minimize the respective objective function. In Figure 9, the average of the values of the objective function obtained in the twenty trials is plotted for each number of function evaluations (the error bars indicate the standard deviation over the twenty trials for each number of function evaluations).

After all the optimizations are completed and the potentials have been generated, the quality of each potential must be tested. For each number of function evaluations, and for each of the twenty independent trials, the potential produced by a method was quality tested by running twenty cooling simulations on a system of 225 particles, and then measuring the quality of the final lattices using both the Template Measure and the Defect Measure. For a single trend optimization method, this requires

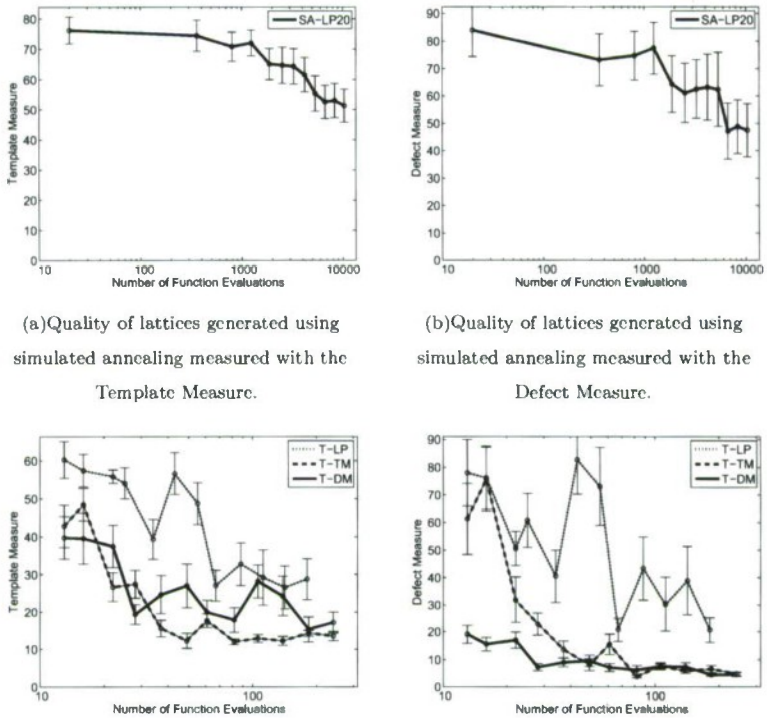
$$\left(\frac{\text{12 different numbers of function evaluations}}{\text{each trend method}}\right) \times \left(\frac{\text{20 independent trials}}{\text{each number of function evaluations}}\right) \times \left(\frac{\text{20 cooling simulations}}{\text{each independent trial}}\right) = \left(\frac{\text{4800 cooling simulations}}{\text{each trend method}}\right).$$

In other words, 240 independent optimization trials are required to produce a single curve in Figure 9, and 4800 cooling simulations are required to produce a single curve in Figure 10. The cooling simulations were initialized with a temperature approximately 1.5 times the melting temperature of the lattice and then slowly cooled using a Nosé-Hoover thermostat to less than ten percent of the melting temperature. At the completion of each cooling simulation, both the Template Measure and the Defect Measure were computed to measure the quality of the resulting lattice. Averaging over the 20 cooling simulations yields two quality scores for each potential – one for each quality metric. Results for the Template Measure are shown in Figure 10(c) while results for the Defect Measure are shown in Figure 10(d).

After reviewing the results in Figure 9, we see that the trend approach clearly provides a faster and more robust method for optimizing the objective functions. Figure 9(a) shows the results from twenty simulated annealing optimizations for different randomly chosen starting points in the search domain, and indicates that the success of the method is highly variable. Simulated annealing often fails to find optimal parameter values and remains stuck in local minima even after 10,000 evaluations of the objective. Of the twenty simulated annealing optimization trials performed, only nine were able to find an optimal value of the Lindemann parameter less than 0.15. The bold line in Figure 9(a) represents the mean over only these nine best-performing trials. Moreover, many of these simulated annealing optimizations that do converge require more than 6,000 objective evaluations. In contrast, we see from Figure 9(b) that trend optimization reliably finds optimal values after sixty evaluations of the Lindemann parameter. We conclude that when trend optimization is used to optimize the Lindemann parameter, we obtain a one-hundred-fold reduction in computation time over the simulated annealing method of [23], and that the optimization is more robust. The speed-up can be attributed to the fact that the objective function is noisy yet has a simple trend – two properties for which trend optimization is ideally suited.

Furthermore, the accelerated search of the trend method makes it possible to use trend optimization with the more expensive quality metrics as objective functions. Doing so provides the extra guarantee that the generated potentials produce high-quality lattices. As indicated in Figures 9(c) and 9(d), trend optimization reliably finds optimal values of the lattice quality objectives in less than 120 function evaluations, although it must be remembered that these objectives are approximately 35 times more expensive to evaluate than the Lindemann parameter. Nevertheless, the total CPU time required to perform these optimizations is still less than the time taken by simulated annealing to optimize the Lindemann parameter.

Figure 10 shows the quality of the lattices generated by the potentials produced by the various methods, as measured using both quality metrics. Several important observations are to be made after reviewing these plots. First, optimization of the Lindemann parameter leads to lattices of modest and unreliable quality, indicating that the Lindemann parameter and lattice quality are only moderately correlated. Second, even when simulated annealing is successful in optimizing the Lindemann parameter, the corresponding potential may yield lattices of poor quality. This occurs because simulated annealing may find minima corresponding to very narrow wells that do not provide robustness against uncertainty in initial conditions. In contrast, since trend optimization seeks out the general trend over the entire search space, this method finds minima that more robustly



(a)Quality of lattices generated using simulated annealing measured with the Template Measure.

(b)Quality of lattices generated using simulated annealing measured with the Defect Measure.

(c)Comparison of lattice quality using the Template Measure.

(d)Comparison of lattice quality using the Defect Measure.

FIG. 10: Here we compare the quality of the lattices produced using the four optimization methods. In (a) and (b), the quality of the lattices produced using the simulated optimization method is shown. It should be noted that only the nine potentials generated by the convergent simulated annealing trials in Figure 9(a) were used to produce these plots. Figures (c) and (d) show the quality of lattices produced using the three trend optimization methods. Using trend optimization directly on the quality metrics reliably produces high quality lattices. Optimization of the Lindemann parameter is only moderately correlated with improved lattice quality. In these Figures, recall that values of the Template Measure and the Defect Measure for purely random configurations are 132 and 161, respectively, and that the computation-free Geometric Method produces lattice quality scores of 18 and 12, respectively.

lead to high quality lattices. Finally, the two lattice quality metrics are reasonably consistent in that a potential generated by optimizing one of the quality measures is also considered high-quality in the other measure.

Figure 11 shows the range of potentials generated by the trend optimization methods. The potential provided by [23] is also shown for reference. Most striking is that trend optimization un-

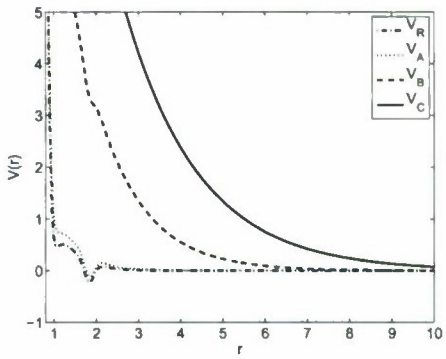
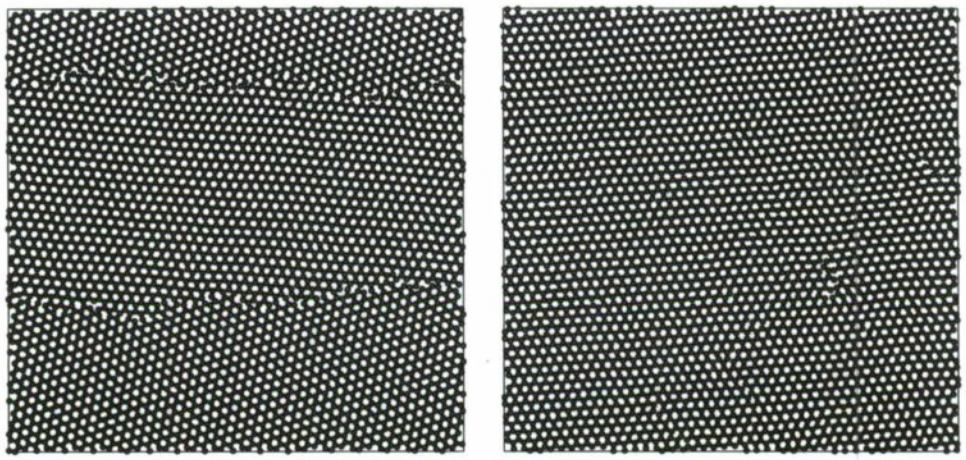


FIG. 11: Potentials for the generation of honeycomb lattices. V_R is the potential previously provided by [23]. Trend optimization generates potentials similar to [23] (labelled V_A), but also uncovers a new family of solutions that have a more repulsive shape and no local minima (sample potentials in this family are labelled V_B and V_C). Remarkably, these repulsive potentials robustly form large regions of honeycomb lattice without defects.

covers an entirely new family of potentials not previously considered. In this family of potentials, the exponential term is dominant and consequently, their shape does not admit a local minimum. The repulsive shape of these potentials leads to higher quality lattices since particles do not get stuck in local minima associated with local potential wells. In simulation, we observe that the defects continue to move through the configuration until they leave the domain entirely, or annihilate one another through collisions.

Figure 12 shows two sample lattice configurations obtained for 4128 particles simulated using the potential labelled V_C in Figure 11 that was generated with trend optimization on the Template Measure. The parameter values for this potential are a_0=5.771, a_1=23.594, a_2=0.574, and a_3=1.816. The final lattices exhibit large regions of extremely well-formed honeycomb lattice that we have not previously observed using potentials that contain local minima. In Figure 12(a), a prominent grain boundary lies between two large areas of well-formed honeycomb lattice. As in Nature, this grain boundary forms when the cooling is not sufficiently slow. In Figure 12(b), isolated defects are visible in the lattice; however, it should be noted that these defects are not yet "frozen" into the lattice, and given enough time will eventually leave the domain or disappear through mutual collision.

The repulsive potentials found using trend optimization are remarkably effective at producing large regions of almost defect free honeycomb lattice despite their relatively simple shape. The effectiveness of these potentials suggests that a far simpler basis of potential functions can be



(a) (b)

FIG. 12: Honeycomb lattices formed using the repulsive family of potentials discovered using trend optimization. (a) Two large regions of extremely well-formed honeycomb lattice meet to form a grain boundary. (b) Large regions of the honeycomb lattice are formed with a few isolated defects. These defects continue to move, even after the lattice is frozen, until they exit the boundary or are annihilated by collisions with other defects.

used in the parameterization of $V_{\mathsf{HC}}(r)$. In these potentials it is the exponential decay term that dominates over the Gaussian term. The numerical simulations seem to suggest, and it would be interesting to pursue rigorously, that for a fixed density, the honeycomb lattice is a global minimizer (or ground state) of the exponential decay pairwise interaction potential. This path is consistent with the numerical findings of Jagla *et al* in [50] for a ramp type potential.

VIII. ANISOTROPIC POTENTIALS

The methods presented thus far can be used to quickly generate isotropic potentials that produce high quality lattices. However, the potentials for self-assembling honeycomb lattices are not robust to variations in density – the lattices only form if the initial density of the particles is very near the ideal density of the target lattice. If the initial density is much less than that of the target lattice, then a triangular lattice will be favored. [23] addressed this issue by searching for optimal potentials over a range of densities. However, the tolerance for variation in density is still very narrow and may indeed be a fundamental limitation of isotropic potentials. Allowing the potentials to have angular dependence leads to the robust formation of high-quality lattices from initial conditions

with large variations in density. Moreover, admitting potentials with angular dependence allows for the construction of potentials that form more exotic lattices, such as the kagome lattice, which has not yet been accomplished with purely isotropic potentials [23]. Certainly, the use of anisotropic potentials is well-motivated by abundant natural examples of anisotropic interaction potentials in Nature – the water molecule serving as an ubiquitous prototype.

In order to introduce potentials with angular dependence, we must first extend the configuration space of the particle system to include an angular coordinate. We no longer consider the particles as point particles, but rather as two-dimensional sliding disks, each with radius R and uniform mass density. The configuration manifold of each particle is now $\mathbb{R}^2 \times S^1$. The configuration of the ith particle is described by the coordinate chart (x_i, y_i, θ_i) as indicated in Figure 13, and we write the vector of coordinates describing the configuration of all N particles as $\mathbf{x} = \left[(x_1, y_1, \theta_1), \cdots, (x_N, y_N, \theta_N)\right]$. The total interaction potential over all particles now has the

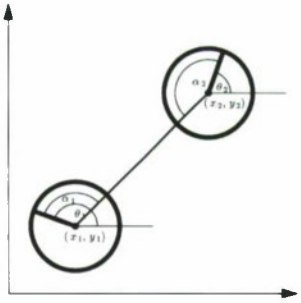


FIG. 13: To implement an anisotropic potential, particles are now modeled as two-dimensional disks whose configuration is described by a location for the center of the disk, (x, y), as well as a heading angle θ. The interaction potential between two particles is a function of the distance between the particles, as well as the angle, α, that lies between the heading angle and the bearing toward the second particle.

form

$$V(\mathbf{x}) = \sum_{i<j}^{N} V_{\text{pair}}(x_i - x_j, y_i - y_j, \theta_i, \theta_j) \tag{8}$$

where $V_{\text{pair}} : \mathbb{R} \times \mathbb{R} \times S^1 \times S^1 \to \mathbb{R}$ is the pairwise interaction potential between particles that depends not only on the relative displacement between particles, but also the angular displacement of each particle relative to the angular bearing of the other particle. By construction, we make V_{pair} symmetric with respect to particle interchange. Specifically,

$$V_{\text{pair}}(x_i - x_j, y_i - y_j, \theta_i, \theta_j) := \psi(x_i - x_j, y_i - y_j, \theta_i) + \psi(x_j - x_i, y_j - y_i, \theta_j)\,. \tag{9}$$

It remains to choose the functional dependence of $\psi(\cdot,\cdot,\cdot)$ to reflect the desired symmetry in the target lattice. Consider the interaction potential

$$\psi(\Delta x, \Delta y, \theta_k) := \frac{1}{2r^{12}} - \frac{1}{r^6}\left[1 - \nu \sin^2\left(\frac{n\alpha_k}{2}\right)\right] \tag{10}$$

where

$$r := \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{11}$$

is the radial distance between the particles, and

$$\alpha_k := \theta_k - \arctan(\Delta y, \Delta x) \tag{12}$$

is the angle between the heading of particle k and the bearing toward the second particle (see Figure 13.) This potential is simply a Lennard-Jones potential with added amplitude modulation in the azimuthal direction on the attractive term. The periodicity of the trigonometric functions ensures that the potential has n-fold radial symmetry, and that each particle has preferred directions along which it feels the attractive pull of neighboring particles. The free parameter ν is chosen to adjust the shape of the potential. When ν is zero, the potential collapses to the isotropic Lennard-Jones potential. For values of ν between zero and unity, the potential has n potential wells symmetrically distributed in the azimuthal direction. Taking values of ν greater than unity raises the repulsive regions between the potential wells. Hence, ν is a parameter that determines how strongly the anisotropic potential prefers the binding site directions. In simulations, we have observed that taking larger values of ν produces lattices with less defects since particles that do not align with the preferred binding directions fall into these more repulsive regions between the wells and create a configuration with much higher energy. These defects are quickly removed by vibrations in the lattice. In practice, a compromise must be met since the larger regions of repulsion created by higher values of ν increase the time taken for self-assembly to occur – particles only bind with one another if they approach each other along an ever more narrower binding direction. In the simulations that follow, we have used a value of $\nu = 1.5$.

By changing the integer value of n, we can induce the formation of lattices with desired n-fold symmetry. Surface plots of potentials with 3-fold and 4-fold symmetry (n=3 and n=4) as a function of the radial coordinate r and the azimuthal angle α, are provided in Figures 14(a) and 14(b). A honeycomb lattice and a square lattice produced with these potentials using particles initialized at low density are shown in Figures 14(c) and 14(d).

For the creation of more exotic lattices, we alter the azimuthal modulation of the potential even further. Each particle in the kagome lattice, for example, has binding sites at angles of 0°,

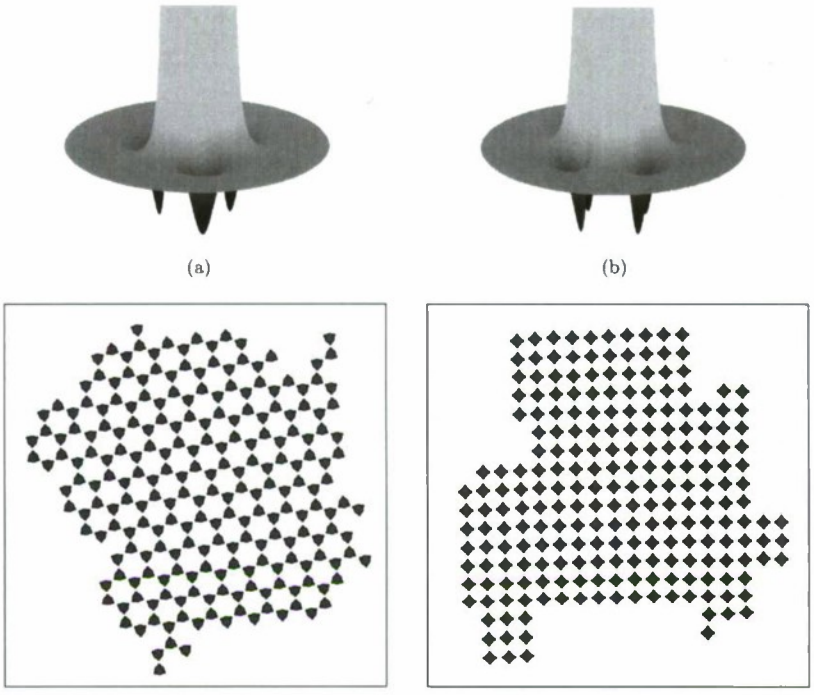


FIG. 14: The potential with three-fold symmetry shown in (a) yields the honeycomb lattice shown in (c). Similarly, the potential with four-fold symmetry shown in (b) yields the square lattice shown in (d).

60°, 180°, and 240° [49]. To ensure preferred binding along these directions, we must modify the azimuthal dependence of the Lennard-Jones potential accordingly. Before doing so, we first express the potential function ψ of line (10) in the following equivalent way:

$$\psi(\Delta x, \Delta y, \theta_k) = \frac{1}{2r^{12}} - \frac{1}{r^6}\left[1 - \nu\, S(\alpha_k)\right] \tag{13}$$

where

$$S(\alpha_k) := \sin^2\left(\frac{n\alpha_k}{2}\right). \tag{14}$$

As written here, ψ produces a potential with n-fold symmetry. In order to produce a potential that favors the kagome lattice, we must simply introduce a new definition for the functional form of $S(\cdot)$.

Let $B := [b_1, \cdots, b_n]$ denote the ordered list of n desired binding directions measured in radians satisfying

$$0 = b_1 < \cdots < b_n < 2\pi\,. \tag{15}$$

Note that without loss of generality, we may prescribe that the first binding direction lies along the ray corresponding to zero radians. For the honeycomb lattice, $B_{\mathsf{HC}} := \left[0, \frac{2\pi}{3}, \frac{4\pi}{3}\right]$, while for the kagome lattice we have $B_{\mathsf{kagome}} := \left[0, \frac{\pi}{3}, \pi, \frac{4\pi}{3}\right]$. Then, for $\alpha \in [0, 2\pi)$, we use the elements in the list B to define $S(\alpha)$ piecewise as follows:

$$S(\alpha) := \begin{cases} \sin^2\left(\frac{\alpha - b_i}{b_{i+1} - b_i}\pi\right) & \text{if } b_i \leq \alpha < b_{i+1}\,, \quad \text{for } i = 1, \cdots, n-1\,, \\ \sin^2\left(\frac{\alpha - b_n}{2\pi - b_n}\pi\right) & \text{if } b_n \leq \alpha < 2\pi\,. \end{cases} \tag{16}$$

When $S(\alpha)$ defined in this way is substituted into the expression for the anisotropic potential function ψ in line (13), it provides the necessary azimuthal modulation of the Lennard-Jones potential to produce potential wells along the binding directions specified in the list S. A plot of $S(\alpha)$ using B_{kagome} is provided in Figure 15(a). Notice that $S(\alpha)$ has local minima precisely at the binding site angles of 0°, 60°, 180°, and 240° consistent with the kagome lattice (recall from line (13) that minima in $S(\alpha)$ lead to minima in the interaction potential). Consequently, the angular dependence in the resulting interaction potential favors the bond structure peculiar to the kagome lattice. The potential and a lattice resulting from this potential are shown in Figures 15(b) and 15(c) respectively.

IX. CONCLUSIONS AND FUTURE WORK

We have presented and compared methods for generating potentials that lead to the self-assembly of specified target lattices. In particular, we have addressed the problem of designing pairwise interaction potentials that induce the formation of a honeycomb lattice when a planar system of particles is cooled.

We have demonstrated that reasonably high quality lattices can be produced using a heuristic computation-free geometric method. The geometric method provides principles that utilize purely geometric information to design potentials that by construction favor and stabilize the target lattice.

A trend optimization algorithm has been introduced that quickly and robustly finds optimal shapes of the interaction potential that lead to the self-assembly of lattices of high quality. The

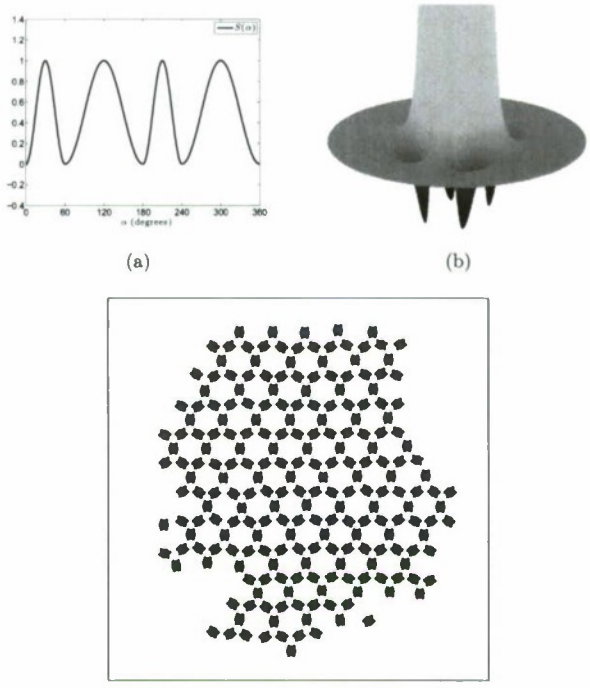


FIG. 15: The function in (a) is used to azimuthally modulate the amplitude of a Lennard-Jones potential to encourage binding directions that favor the kagome lattice. The resulting potential surface is shown in (b), while a kagome lattice formed with this potential is shown in (c).

success of the trend method lies in its ability to quickly locate minima in a noisy and expensive objective function. Moreover, the potentials discovered using the trend optimization procedure robustly form high quality lattices with respect to variations in the initial conditions of the particles. We have seen that the trend optimization method has discovered a family of potentials characterized by very simple exponential decay profiles that routinely lead to the formation of the honeycomb lattice.

Our trend optimization algorithm robustly and routinely finds optimal values of the objective, although it must be noted that as currently implemented, the algorithm does not provide rigorous guarantees of convergence. Convergence can be guaranteed, however, by incorporating a *polling* step as required in the Surrogate Management Framework. It would be interesting to investigate the effects of polling on the optimal results and the efficiency of the method. One expects that the

addition of a polling step to guarantee convergence will represent only a marginal increase in the overall computational cost of the algorithm.

The geometric method has also been extended to the design of anisotropic potentials. Azimuthal dependence of the interaction potential allows for the formation of the kagome lattice which has not previously been performed using isotropic potentials. Incorporating anisotropy into the potentials allows for the formation of lattices over a wide range of particle densities.

An auxiliary contribution of this paper is the development of two metrics for objective analysis of lattice quality. The development of these metrics was necessary for comparison of the lattices produced by the proposed potentials.

We anticipate that the methods presented here will naturally extend to three dimensions without impediment. Rechtsman *et al* have recently investigated the design of potentials for self-assembly of three-dimensional lattice structures [51]. Most notably, they have demonstrated the formation of the diamond and wurtzite lattices. As a matter of course, we intend to apply the methods presented here to the design of potentials in the three-dimensional self-assembly problem and expect the trend optimization method to provide a significant computational savings in the design of potentials. Of particular interest is to determine if trend optimization recovers the three-dimensional result of [51], or if a new, simpler, and perhaps more robust family of solutions is discovered.

A natural extension of the methods presented is to investigate the use of multi-specie and multi-body potentials for the self-assembly of quasicrystals and Penrose tilings. Generating lattices with prescribed geometry and structure is highly motivated by the desire to produce photonic crystals with specified optical properties. Also, we intend to explore the use of the trend optimization method for designing potentials that lead to the the formation of hexahedral meshes over complicated three-dimensional volumes. Producing hexahedral meshes is a notoriously difficult problem that is currently a major stumbling block for continuum mechanics computations that use a finite element method.

More broadly, we expect to use trend optimization to understand more deeply the fundamental limitations and extraordinary possibilities of self-assembly. It will be interesting to investigate, for example, the role of optimal potential design in natural systems. In this regard, a study of self-assembly is a study of life itself. Are there mechanisms at the level of local interactions that allow for the differentiation of cells that self-assemble to form bone, for instance, from those that form brain tissue? How much control authority over the superstructure formed via self-assembly is provided by control over the local interactions? Are there natural systems in which a small amount of flexibility in the properties of the local interactions allows for large and beneficial changes in

structure at the macroscopic level? We believe that insights to these questions may be afforded by finding optimal potentials through the method of trend optimization.

---

[1] G. M. Whitesides and B. Grzybowski, Self-Assembly at All Scales, Science, 295(5564), 2418–2421 (2002).

[2] M. M. Murr and D. E. Morse, Fractal intermediates in the self-assembly of silicatein filaments, Proceedings of the National Academy of Sciences of the United States of America, 102(33), 11657-11662 (2005).

[3] W. Zheng, P. Buhlmann, and H. Jacobs, Sequential shape-and-solder-directed self-assembly of functional microsystems, Proceedings of the National Academy of Sciences of the United States of America, 101(35), 12814-12817 (2004).

[4] S. A. Stauth and B. A. Parviz, Self-assembled single-crystal silicon circuits on plastic, Proceedings of the National Academy of Sciences of the United States of America, 103(38), 13922-13927 (2006).

[5] R. Gross and M. Dorigo, Evolution of Solitary and Group Transport Behaviors for Autonomous Robots Capable of Self-Assembling, Adaptive Behavior, 16(5), 285-305 (2008).

[6] K. Jakab, A. Neagu, V. Mironov, R. R. Markwald, and G. Forgacs, Engineering biological structures of prescribed shape using self-assembling multicellular systems, Proceedings of the National Academy of Sciences of the United States of America, 101(9), 2864-2869 (2004).

[7] V. Manoharan, M. Elsesser, and D. Pine, Dense packing and symmetry in small clusters of microspheres, Science, 301(5632), 483-487 (2003).

[8] P. Maksymovych, D. C. Sorescu, K. D. Jordan, and J. Yates, John T., Collective Reactivity of Molecular Chains Self-Assembled on a Surface, Science, 322(5908), 1664-1667 (2008).

[9] M. Engel and H. Trebin, Self-Assembly of Monatomic Complex Crystals and Quasicrystals with a Double-Well Interaction Potential, Physical Review Letters, 98(22), 225505 (2007).

[10] M. A. Glaser, G. M. Grason, R. D. Kamien, A. Kosmrlj, C. D. Santangelo, and P. Ziherl, Soft spheres make more mesophases, EPL (Europhysics Letters), 78(4), 46004 (5pp) (2007).

[11] V. V. Hoang and T. Odagaki, Molecular dynamics simulations of simple monatomic amorphous nanoparticles, Physical Review B (Condensed Matter and Materials Physics), 77(12), 125434 (2008).

[12] Y. H. Lin, L. Y. Chew, and M. Y. Yu, Self-assembly of complex structures in a two-dimensional system with competing interaction forces, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), 78(6), 066405 (2008).

[13] A. Quandt and M. P. Teter, Formation of quasiperiodic patterns within a simple two-dimensional model system, Phys. Rev. B, 59(13), 8586–8592 (1999).

[14] O. Sigmund and S. Torquato, Composites with extremal thermal expansion coefficients, Applied Physics Letters, 69(21), 3203-3205 (1996).

[15] K. M. Ho, C. T. Chan, and C. M. Soukoulis, Existence of a photonic gap in periodic dielectric structures, Phys. Rev. Lett., 65(25), 3152–3155 (1990).

[16] T. A. Mary, J. S. O. Evans, T. Vogt, and A. W. Sleight, Negative Thermal Expansion from 0.3 to 1050 Kelvin in ZrW_2O_8, Science, 272(5258), 90–92 (1996).

[17] B. Xu, F. Arias, S. Brittain, X. Zhao, B. Grzybowski, S. Torquato, and G. Whitesides, Making negative Poisson's ratio microstructures by soft lithography, Advanced Materials, 11(14), 1186-1189 (1999).

[18] S. Hyun and S. Torquato, Designing composite microstructures with targeted properties, Journal of Materials Research, 16(1), 280-285 (2001).

[19] G. Ferey and A. Cheetham, Porous materials - Prospects for giant pores, Science, 283(5405), 1125-1126 (1999).

[20] B. Chen, M. Eddaoudi, S. Hyde, M. O'Keeffe, and O. Yaghi, Interwoven metal-organic framework on a periodic minimal surface with extra-large pores, Science, 291(5506), 1021-1023 (2001).

[21] A. Greer, Condensed matter - Too hot to melt, Nature, 404(6774), 134-135 (2000).

[22] C. A. Murray and D. G. Grier, Video Microscopy of Monodisperse Colloidal Systems, Annual Review of Physical Chemistry, 47(1), 421-462 (1996).

[23] Rechtsman, M., F. Stillinger, and S. Torquato [2006], Designed interaction potentials via inverse methods for self-assembly. *Phys. Rev. E* **73**, 011406.

[24] S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, Journal of Computational Physics, 117(1), 1 - 19 (1995). See also the website at http://lammps.sandia.gov.

[25] Theil, F. [2006], A proof of crystallization in two dimensions. *Comm. Math. Phys* **262**, 209.

[26] Rechtsman, M., F. Stillinger, and S. Torquato [2005], Optimized interactions for targeted self-assembly: Application to a honeycomb lattice. *Phys. Rev. Lett.* **95**, 228301.

[27] Grubits, K. A., and J. E. Marsden (2008), Lattice Quality Assessment Tools and their Applications. *in preparation.*

[28] Booker, A. J. (1996), Case studies in design and analysis of computer experiments. *Proceedings of the Section on Physical and Engineering Sciences*, American Statistical Association.

[29] Myers, R., and D. C. Montgomery (1995), Response Surface Methodology: Process and product optimization using designed experiments. John Wiley & Sons, New York.

[30] Galassi, Davies, T. G. J. A. B. R. (2009). GNU Scientific Library Reference Manual - Third Edition (Third). Network theory Ltd..

[31] Booker, A. J., J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torzon, and M. W. Trosset (1999), A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* **17**, 1.

[32] Torczon, V. [1997], On the convergence of pattern search algorithms. *SIAM J. Optimization* **7**, (1), 1.

[33] Audet, C., and J. E. Dennis Jr. (2003), Analysis of generalized pattern searches. *SIAM Journal on Optimization* **13**, 889.

[34] C. Audet and J. J. E. Dennis, Mesh Adaptive Direct Search Algorithms for Constrained Optimization,

SIAM Journal on Optimization, 17(1), 188-217 (2006).

[35] C. Audet, V. Bchard, and J. Chaouki, Spent potliner treatment process optimization using a MADS algorithm, Optimization and Engineering, 9(2), 143–160 (2008).

[36] A. L. Marsden, J. A. Feinstein, and C. A. Taylor, A computational framework for derivative-free optimization of cardiovascular geometries, Computer Methods in Applied Mechanics and Engineering, 197(21-24), 1890 - 1905 (2008).

[37] A. L. Marsden, M. Wang, J. E. Dennis, and P. Moin, Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation, Journal of Fluid Mechanics, 572(-1), 13-36 (2007).

[38] E. S. Siah, M. Sasena, J. L. Volakis, P. Y. Papalambros, and R. W. Wiese, Fast parameter optimization of large-scale electromagnetic objects using DIRECT with Kriging metamodeling, Microwave Theory and Techniques, IEEE Transactions on, 52(1), 276-285 (2004).

[39] W. Raza and K. Kim, Evaluation of Surrogate Models in Optimization of Wire-Wrapped Fuel Assembly, Journal of Nuclear Science and Technology, 44(6), 819-822 (2007).

[40] Cressie, N. (1990), The origins of kriging. *Mathematical Geology* **22**, 239.

[41] Simpson, T. W., J. J. Korte, T. M. Mauery, and F. Mistree [1998], Comparison of response surface and kriging models for multidisciplinary design optimization. AIAA Paper, 98-4755.

[42] Giunta, A. A., and L. T. Watson (1998), A comparison of approximation modeling techniques: polynomial versus interpolating models. AIAA Paper, 98-4758.

[43] Hastie, T., R. Tibshirani, and J. H. Friedman (2001), *The Elements of Statistical Learning*, Springer.

[44] Tikhonov, A. N. and Arsenin, V. Y. (1977). Solutions of Ill-posed Problems. New York: Halsted Press.

[45] Dyn, N., D. Levin, and S. Rippa (1986), Numerical procedures for surface fitting of scattered data by radial basis functions. *SIAM J. Sci. Statist. Comp.* **7**, 639.

[46] Fasshauer, G. E. (2007). Meshfree Approximation Methods with MATLAB. Singapore: World Scientific Publishing .

[47] Gutmann, H.-M. (2001), A radial basis function method for global optimization. *Journal of Global Optimization* **19**(3), 201.

[48] M. D. McKay, R. J. Beckman, and W. J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics, 21(2), 239–245 (1979).

[49] M. Mekata, Kagome: The Story of the Basketweave Lattice, Physics Today, 56(2), 12-13 (2003).

[50] Jagla, E. A. (1999), Minimum energy configurations of repelling particles in two dimensions. *J. Chem. Phys.* **110**, 451.

[51] M. C. Rechtsman, F. H. Stillinger, and S. Torquato, Synthetic diamond and wurtzite structures self-assemble with isotropic pair interactions, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), 75(3), 031403 (2007).

C.2 Tools for design of potentials for particle self-assembly

Tools for design of potentials for particle self-assembly

Symeon Grivopoulos George Mathew Gunjan Thakur
Marko Budisić Igor Mezić

October 24, 2008

Department of Mechanical and Environmental Engineering
University of California, Santa Barbara, CA 93106-5070

Abstract

Motivated by the work in [1] on design of potentials for spontaneous self-assembly of particle systems into 2D target structures, we propose three tools to apply to this problem. First, we introduce a new pseudo-distance between planar point sets, that can be used to compare particle configurations with the target structure. In conjunction with Molecular Dynamics simulations, it provides an objective function to be graphed or optimized over the space of parameters of the potential, and thus, it provides a tool that quantifies the effectiveness of a potential to assemble a target configuration. The second tool, is the use of infinitesimal mechanical stability of the target structure with a given inter-particle potential as a necessary condition for self-assembly. The third one, characterizes the target structure (with given inter-particle potential and Rayleigh dissipation forces) as an asymptotic fixed point of the system dynamics or not, by examining the behavior of initial conditions close to the structure in backward time. The first tool provides a sufficient condition to identify assembling potentials, but it can be computationally expensive, while the other tools are computationally fast and provide necessary conditions that can be used to exclude a priori "bad" regions of the parameter space.

1 Introduction

In a recent paper [1], Rechtsman et al discuss "inverse methods" for design of potentials for self-assembly. The implied "direct" problem is the determination of the minimum energy configuration of a system of identical, classical interacting particles (at a fixed density), given their interaction. The "inverse" problem is to design the interactions between the particles, so that their minimum energy configuration (again, at fixed density) is a desired one. In particular, these authors consider isotropic two-body potentials with fixed functional form, but tunable parameters, and seek to determine optimal parameter sets so that a

particle system with these interactions (in 2 dimensions) would "self-assemble", as closely as possible, into a (piece of a) square lattice or a honeycomb structure of given density, as the temperature decreases to $T = 0$. The inspiration for their work comes from a host of examples in Biology, Chemistry and Materials Science where smaller "units" arrange themselves to form larger structures through their interactions. If one can manipulate these interactions to some extent, one may be able to form structures with desired properties and functions, see [1] and references therein.

The general methodology of [1] is as follows: Starting with a two-body isotropic potential of given functional form, but with tunable parameters a_i, $V(r, \{a_i\})$ (r is the inter-particle distance), they propose *objective functions* of the a_i whose minimization implies that the particle system has the desired structure (square lattice or honeycomb of a given density) as its ground state, and thus, it will equilibrate to it as its temperature is brought down to 0. They propose two such functions, one based on energy considerations and one based on dynamical ones ("zero-temperature" and "near-melting" schemes). After each of the optimization schemes is terminated, the obtained potential is tested for effectiveness: An annealed Monte Carlo (MC) simulation is performed in an NVT ensemble starting from a random configuration, and the resulting minimum energy configuration is compared to the desired structure.

In this work, we propose some tools to attack the problem of self-assembly. Before we introduce these tools, we want to comment on the approach of [1] and justify the need for our methods. As pointed out by the authors, the energy-based zero-temperature scheme ignores all other competing structures other than the ones it explicitly considers. Though physically intuitive, it does not provide a sufficient or necessary condition for self-assembly. The near-melting scheme offers only a necessary condition: To quote, there is an "inherent bias in this scheme towards the target lattice", meaning that the scheme guarantees non-local stability of the structure but not self-assembly from any (or most) initial conditions. Also, the metric used by the authors (and referred to as the "Lindemann parameter") to compare the particle system configuration near its melting point with the target structure, is not entirely satisfactory. If the particle system has rotated as a whole, or two particles have exchanged positions, the distance of the particle configuration from the target will be increased. Similarly, the structure assembled by the final MC simulation (to check the outcome of the optimization) is *qualitatively* compared to the target, either directly, or by the plot of the structure function $S(\mathbf{k})$, see section 2 and [1].

In our approach, we seek to provide clear-cut necessary and sufficient conditions on whether a set of parameters renders a potential capable to assemble a structure. The first tool we introduce is a Fourier space-based distance for point sets in $\mathbb{R}^2$. This new distance function allows us to compare two planar point sets modulo translations and rotations, i.e. a point set and a translated and rotated copy of itself are considered the same. It allows us to *quantitatively* compare a particle configuration with a target structure, rather than relying on qualitative estimations. After running an MD simulation of the particle system with a given potential, while slowly lowering the temperature to $T = 0$ (the

freezing has to be slow enough, so that the system is not trapped in a metastable state, i.e. a local minimum of the interaction energy), we calculate the distance of the equilibrium configuration from the target structure. Averaged over all initial conditions, this provides an objective function, whose minimum determines the best set of parameters (of the potential) for self-assembly from most initial conditions. We will refer to this function as the *Defect Metric*. In practice, see section 2, it suffices to consider just a few initial conditions per function evaluation (the most we used was 5). Note that the Defect Metric being equal to zero (or, at least, very small) for a set of parameter values provides a *sufficient* condition for self-assembly of the target structure. We consider this a major advantage of this method. Also, the method is not in any way biased towards the target structure. We will not attempt any optimization in the examples considered here, but rather plot the Defect Metric in parameter neighborhoods around the solutions of [1] in order to illustrate the effectiveness of our approach.

The second tool we introduce, is the use of the (infinitesimal) mechanical stability of a structure as a necessary condition for its self-assembly from a given potential. In [1], the authors emphasize the necessity for mechanical stability of a target structure if it is to be considered as a ground state for the particle system with a given potential interaction. They also check that the potentials furnished by their optimization schemes guarantee mechanical stability in each case. However, this idea can be exploited further to delineate portions of the parameter space where the resulting potential does not guarantee mechanical stability of the target structure. The relevant calculations are very fast and require no optimization. This provides a simple, fast test that can drastically reduce the parameter space one has to search over for a solution. Note that a potential that guarantees mechanical stability of the target structure does not necessarily assemble it, hence this condition is only necessary.

The third tool, is the use of *backward (in time) integration* to identify an asymptotic stable fixed point of a dynamical system. It is based on the property of an asymptotic stable fixed point that, trajectories slow down more and more the closer they get to it. Contrast this with the behavior of trajectories around a regular point of phase space, where they don't, at least in some directions. The idea is to consider initial conditions around a presumed fixed point and study their behavior in reverse time. If the presumed fixed point is indeed one, the initial conditions will move away from it slowly. If it isn't, at least some of them will move away from it fast enough. The way we implement this idea in the self-assembly problem, is the following: For a given target structure, we create an ensemble of particle configurations uniformly distributed according to their distance from it (with the Defect Metric taking values between 0 and a small upper bound). We propagate this ensemble backwards in time for a short time and calculate the average of the Defect Metric. A large value of this average Defect Metric means that the target structure is not an asymptotic stable fixed point for the potential. As with mechanical stability, this test provides a necessary condition only, however, it is fast and simple to implement, and requires no optimization. Again, the goal is to exclude portions of the parameter space and thus speed up the search for good parameter sets.

We will use three examples to demonstrate our methods: Self-assembly of a square structure using two candidate potentials with parameters, V_{squ1} and V_{squ2},

$$V_{\text{squ1}}(r; a_0, a_1, a_2) = \frac{1}{r^{12}} - \frac{2}{r^6} + a_0 \exp[-a_1 (r - a_2)^2], \quad (1)$$

$$V_{\text{squ2}}(r; a_1, a_2) = \frac{1}{r^{12}} + \tanh(a_1 r - a_2) - 1, \quad (2)$$

and self-assembly of a honeycomb structure using V_{hon},

$$V_{\text{hon}}(r; a_0, a_1, a_2, a_3) = \frac{5}{r^{12}} - \frac{a_0}{r^{10}} + a_1 \exp[-a_2 r] - .4 \exp[-40(r - a_3)^2]. \quad (3)$$

V_{squ1} and V_{hon} are taken from [1], while V_{squ2} is a generalization of a potential found in [2]. These three potentials are plotted in figure 1 for parameter values that make them assembling. The physical motivation for these choices of functional forms is the following: V_{squ1} and V_{squ2} energetically favor neighbors at distances 1 and $\sqrt{2}$ (first and second neighbors in the case of a square lattice of lattice constant 1) while they strongly disfavor neighbors at distance $\sqrt{3}$ (second neighbors in the case of a triangular lattice of constant 1, the triangular lattice being the main competitor of any 2D structure assembled by an isotropic potential). V_{hon} energetically favors neighbors around $r = \sqrt{3}$. It disfavors neighbors around $r = 1$, but makes them locally stable (see small dip in V_{hon} close to $r = 1$; the actual lattice constant of the honeycomb structure for this example is 1.0565). At the right density, this competition between energy and mechanical stability favors the honeycomb structure which has only 3 first neighbors at distance 1.0565 versus the triangular lattice that has 6 first neighbors at this distance (they both have 6 second neighbors at distance $\sqrt{3} \times 1.0565$).

The organization of the rest of the paper is as follows: In section 2, we introduce our Fourier space-based distance of point sets in $\mathbb{R}^2$ and use it to compute the Defect Metric as a function of the potential parameters in each example. In sections 3 and 4, we present our mechanical stability and backward integration calculations, respectively. Section 5 concludes. Some properties of Fourier space-based distance functions of point sets in $\mathbb{R}^2$ are discussed in an Appendix.

2 Defect Metric

In this section we introduce a new metric that compares point configurations (sets) in $\mathbb{R}^2$ modulo translations and rotations. It allows us to compare the "shapes" of two point sets as structures, without the need to use an a priori correspondence of points of the two sets. The use of this metric, combined with Molecular Dynamics simulations will provide a sufficient test for the assembling capability of potentials. Note that the constructions in this section (as well as the derivations in the Appendix) generalize immediately to $\mathbb{R}^m$.

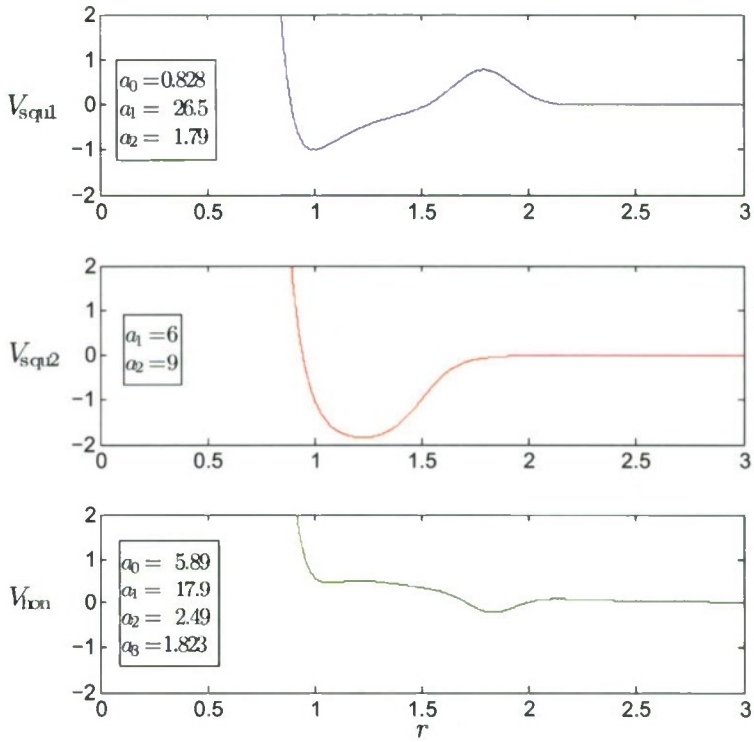


Figure 1: Potentials V_{squ1}, V_{squ2} and V_{hon} for parameter values that make them assembling.

Consider a configuration of points $\{\mathbf{r}_i\}$, $i = 1, \ldots, N$, in $\mathbb{R}^2$. Assigning a mass of 1 to each point, we may define the (singular) mass distribution of this point set to be

$$\rho(\mathbf{r}) \doteq \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i).$$

The Fourier transform of ρ is given by

$$c(\mathbf{k}) = \sum_{i=1}^{N} e^{-i\mathbf{k}\cdot\mathbf{r}_i}, \ \mathbf{k} \in \mathbb{R}^2. \tag{4}$$

The $c(\mathbf{k})$'s were introduced in [3] under the name *collective coordinates*, with the purpose of capturing collective aspects of the dynamical behavior of a system of identical particles . For some of their uses, see [4, 5, 6] and references therein.

The quantity $|c(\mathbf{k})|^2$ is the Fourier transform of the pair correlation function of the mass distribution of this point set, i.e.

$$\int \frac{d^2\mathbf{k}}{(2\pi)^2} e^{i\mathbf{k}\cdot\mathbf{r}} |c(\mathbf{k})|^2 = \sum_{i,j} \delta(\mathbf{r} - (\mathbf{r}_i - \mathbf{r}_j)).$$

Notice that it is invariant to translations of the point set, $\mathbf{r}_i \to \mathbf{r}_i + \mathbf{c}$. Since the pair correlation function of a finite point set (i.e. the knowledge of the set of relative positions $\{\mathbf{r}_i - \mathbf{r}_j\}$) determines the point set modulo translations, so does the *structure function* $S(\mathbf{k}) \doteq |c(\mathbf{k})|^2/N$. Note that this fact does not resolve the Crystallographic Phase Problem in general, i.e. a distribution cannot be reproduced by knowledge of the moduli of its Fourier coefficients only, but for the restricted class of distributions we are considering, namely finite superpositions of delta measures, knowledge of $|c(\mathbf{k})|^2$ is enough to determine the point set modulo translations.

Given two point sets, $\{\mathbf{r}_i^{(1)}\}$ and $\{\mathbf{r}_i^{(2)}\}$, $i = 1, \ldots, N$, we define a distance $d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\})$ between them (modulo translations) by

$$d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) \doteq \frac{1}{N} \int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) \left| |c^{(1)}(\mathbf{k})|^2 - |c^{(2)}(\mathbf{k})|^2 \right|, \tag{5}$$

where w is a positive, continuous and integrable function that weighs the relative importance of Fourier modes. In the Appendix, it is shown that d is well defined, and satisfies the properties of a distance function. Different choices of w, emphasize different spatial characteristics of the point set. Consider, for example, the point sets a and b of figure 2: a is a piece of square lattice of constant 1 made up of 10×10 sites and b is made by shifting the left half of a (10×5 sites) by 1. Structures a and b appear much the same in their bulk (i.e. away from their boundaries), but globally the sets are distinct. We can capture the similarities and the differences of these two sets at different length scales by using different weight functions. For example,

$$w_1(\mathbf{k}) = \begin{cases} 1, & k_x \in [-10\pi, 10\pi],\ k_y \in [-10\pi, 10\pi], \\ 0, & \text{otherwise}, \end{cases}$$

penalizes differences in the structures in length scales from .2 and up, and gives a value of $d = 17$ for the distance of a and b.

$$w_2(\mathbf{k}) = \begin{cases} 1, & k_x \in [-10\pi, -\pi] \cup [\pi, 10\pi],\ k_y \in [-10\pi, -\pi] \cup [\pi, 10\pi], \\ 0, & \text{otherwise}, \end{cases}$$

penalizes differences in length scales from .2 up to 2 only (i.e. inside a 5×5 box centered at every lattice site) and thus gives a much smaller value of $d = 1.8$ (the calculations are done using a discrete version of (5) with a and b inside a 20×20 box). So, w provides a flexibility in the comparison of characteristics of two point sets at different scales.

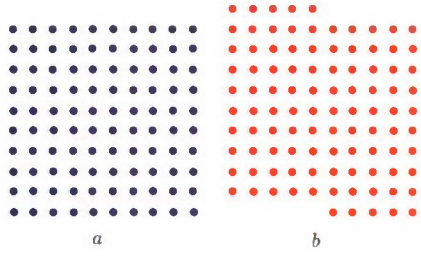


Figure 2: Left: A piece of square lattice. Right: The left half of a is shifted up by one lattice constant.

To compare the "shapes" of two point sets, one needs to mod out not only translations, but also rotations. To this end, we define the quantity

$$I(s) \doteq \frac{1}{N\pi s^2} \int_{\|\mathbf{k}\| \leq s} \frac{d^2\mathbf{k}}{(2\pi)^2} |c(\mathbf{k})|^2. \tag{6}$$

The crucial step here is the angular $\mathbf{k}$-integration that eliminates all information in the structure function except for the pairwise distances in the point set. In the Appendix, it is shown that

$$I(s) = \frac{1}{2N\pi^2} \sum_{i \neq j} \frac{J_1(r_{ij}s)}{r_{ij}s},$$

where $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$, and hence I(s) only retains information about the pairwise distances of points. It is shown in [7, Theorem 2.6] that the distribution of pairwise distances of points in a point configuration uniquely determines it modulo translations and rotations, except for a set of configurations of measure zero in the space of all possible point configurations.

We define now the following *pseudo-distance* function between two two point sets, $\{\mathbf{r}_i^{(1)}\}$ and $\{\mathbf{r}_i^{(2)}\}$, $i = 1, \ldots, N$:

$$u(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) \doteq \int_0^\infty w(s)\, |I_1(s) - I_2(s)| ds. \tag{7}$$

w must be a positive, continuous and integrable function in $[0, \infty)$. In the Appendix, we show that u is well-defined, and it is a pseudo-distance. As in (5), w weighs various length scales differently. Consider, for example, the point sets a and c in figure 3 (c is made from a by slightly shifting and rotating its upper right quarter). For $w_1(s) = 1$, $s \in [0, 10\pi]$, their distance is $u = 0.0038$. Using

$$w_2(s) = \begin{cases} 0, & s < \pi, \\ 1, & \pi \leq s \leq 10\pi, \end{cases}$$

reduces the distance to $u = 0.0011$. As expected, when the large scale features are ignored, the difference between a and c is much less pronounced.

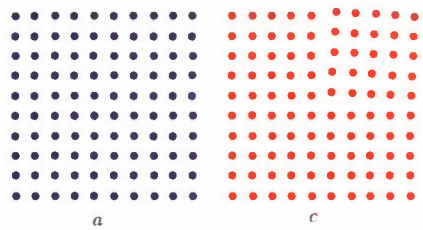


Figure 3: Left: A piece of square lattice. Right: The upper right quarter of a is slightly shifted and rotated.

In each of the three examples mentioned in the introduction (two candidate potentials for the square lattice and one for the honeycomb structure), we performed MD simulations for particles in a box with periodic boundary conditions, with density fixed to the appropriate value for each target structure. The simulations were performed using the MD software package LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [8]. The initial positions of the particles formed a square lattice and the initial velocities were randomly generated. The system temperature was slowly decreased close to $T = 0$ using a Nose-Hoover thermostat [9, 10, 11]. The Defect Metric (distance u of the target structure from the final configuration of the MD) was computed for a grid of parameters in each case, using a small number of runs per evaluation (average over runs). The weight function

$$w(s) = \begin{cases} 0, & s < \pi/l, \\ 1, & s \geq \pi/l, \end{cases}$$

(l being the lattice constant in each case) was used in its computation, so that large-scale differences between the achieved structures and the target were not penalized. The integral (6) was replaced with the sum over Fourier modes appropriate to the container box of each simulation.

The results of the simulations are plotted in figure 4. The top plot represents the Defect Metric as a function of parameters a_1 and a_2 of V_{squ1}, with a_0 set equal to 0.828. It is immediately seen that the parameter values $(a_1, a_2) = (26.5, 1.79)$ of [1] for V_{squ1} fall exactly in the parameter region where the Defect Metric is close to zero. The middle plot contains the Defect Metric for V_{squ2}. As a side remark, compare the two wedge-like domains of the parameter space where the Defect Metric is (approximately) zero for V_{squ2} with the corresponding narrow strip of parameter values for V_{squ1}. It is obvious that V_{squ2} is more robust to changes in its parameters than V_{squ1}. Finally, for V_{hon}, the parameters $(a_0, a_2) = (5.89, 2.49)$ of [1] fall into a shallow and narrow minimum of the Defect Metric, see bottom plot. As stated in the Introduction, we

did not attempt any optimization here, but rather, we wanted to demonstrate the effectiveness of our approach. If desired, optimization over the parameters a_i can be done using simulated annealing as in [1], or other methods, such as trend optimization [12, 13] that are very effective when one is optimizing over functions that are expensive to evaluate and noisy (as is the case here, because the objective function is evaluated using a few MD runs with different initial conditions). We plot the same results in the top plots of figures 7-9, along with the results of sections 3 and 4 for easy comparison.

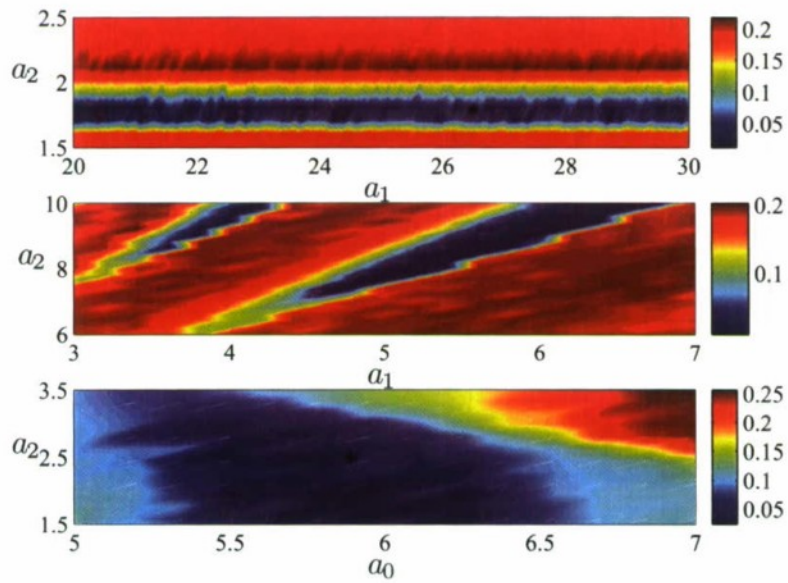


Figure 4: Top: Defect Metric as a function of parameters a_1 and a_2 of V_{squ1}, with $a_0 = 0.828$. Middle: Defect Metric as a function of parameters a_1 anda_2 of V_{squ2}. Bottom: Defect Metric as a function of parameters a_0 and a_2 of V_{hon}, with $a_1 = 17.9$ and $a_3 = 1.823$. The stars correspond to the solutions of [1]

3 Mechanical stability

Infinitesimal mechanical stability of an equilibrium configuration of a particle system is a test that the said configuration is a (local, in general) minimum of the potential interaction energy. Hence, it is a natural condition to check in the search for the global energy minimum. Any potential (and hence, set of parameters) that renders the desired structure infinitesimally mechanically unstable, cannot have that structure as a ground state. This provides a simple and fast

test that excludes portions of the parameter space before any search or optimization. The quantity to be calculated, is the Hessian of the potential energy of the system, evaluated at the equilibrium configuration. Semi-definiteness of the Hessian is equivalent to mechanical stability. For infinite periodic configurations of identical particles, the eigenvalues of the Hessian are the squared frequencies of the phonon excitations (modes) of the structure.

We plot the minimum squared frequency ω^2 of the phonon modes of the target structure, as a function of the parameters in the potential in each case, in figure 5. When the minimum ω^2 is negative, the potential renders the target structure infinitesimally mechanically unstable. For the case of the square lattice with V_{squ1} and V_{squ2}, the top and middle plots of figure 5 delineate the "a priori good" portions of the parameter space with with very good accuracy. In the case of the honeycomb structure with V_{hon}, the bottom plot of figure 5 shows that about 30% of the parameter space depicted can be excluded beforehand. The same results are plotted in the middle color plots of figures 7-9 for comparison with the results of sections 2 and 4.

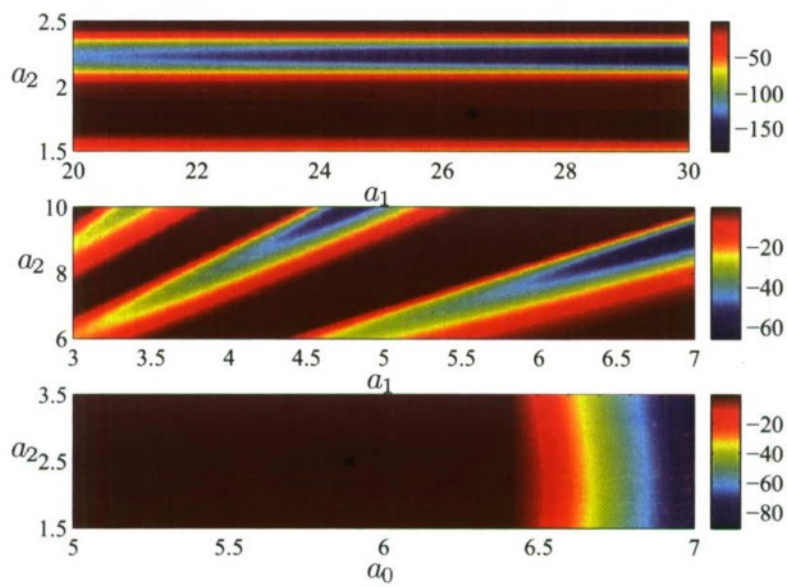


Figure 5: Top: Minimum squared phonon frequency ω^2 in a square lattice of constant 1 as a function of parameters a_1 and a_2 of V_{squ1}, with $a_0 = 0.828$. Middle: Minimum squared phonon frequency ω^2 in a square lattice of constant 1 as a function of parameters a_1 and a_2 of V_{squ2}. Bottom: Minimum squared phonon frequency ω^2 in a honeycomb structure of constant 1.0565 as a function of parameters a_0 and a_2 of V_{hon}, with $a_1 = 17.9$ and $a_3 = 1.823$. The stars correspond to the solutions of [1]

4 Backward integration

Consider a dynamical system with a globally attractive fixed point and a small neighborhood around that point. If we reverse the arrow of time, this small neighborhood will expand, and in the limit $t \to -\infty$, it will encompass the whole space. This is an equivalent way to state that the fixed point is globally attractive. Notice, that this does not happen for regular points in the phase space: A neighborhood of a regular point will be mapped only to a subset of the phase space as $t \to -\infty$. This idea could be used, in principle, in the self-assembly problem in the following way: If a target structure is to be the ground state of a particle system for a given potential, then a a neighborhood of initial conditions (particle positions/velocities) around the target structure at equilibrium would expand to the full phase space of the particle system in the limit $t \to -\infty$. However, for an N particle system in 2D, this would require propagating a $4N$-dimensional probability distribution function, a daunting task even for small N. Instead, we use a different manifestation of the asymptotic stability of a fixed point in backward evolution. Consider an asymptotic stable fixed point (not necessarily globally attractive). Initial conditions in a small neighborhood of it, will move away from it when we reverse time evolution, but very slowly. This is equivalent to the asymptotic approach of trajectories to the fixed point in forward evolution. This "slowing down" of trajectories, however, does not happen for regular points in phase space, no matter how close we get to that point. Hence, in reverse time, initial conditions around a regular point, tend to move away from it faster (at least some of them). We use this fact to construct a simple test for detecting non-assembling potentials: For a given target structure, we create an ensemble of 50 particle configurations uniformly distributed according to their distance from it. We propagate this ensemble backwards in time for a short time and calculate the average of the Defect Metric. Large magnitudes of this average, signify non-assembling potentials. As with mechanical stability, this test provides a necessary condition only, however, it is fast and simple to implement, and requires no optimization. The goal, again, is to exclude portions of the parameter space and thus speed up the search for good parameter sets.

Figure 6 contains the plots of the average Defect Metric for the particle ensembles that were integrated backwards for a short time, for each example, as functions of the potential parameters. The same results are plotted in the bottom color plots of figures 7-9 for comparison with the results of sections 2 and 3. Notice the very good agreement with the mechanical stability plots.

5 Conclusion

Motivated by the recent work [1], we presented three new tools for optimizing potentials for self-assembly. The first tool, is a new (pseudo)metric for comparison of particle configurations. In conjunction with MD simulations, it provides an automated sufficient criterion for detection of good parameter values of the

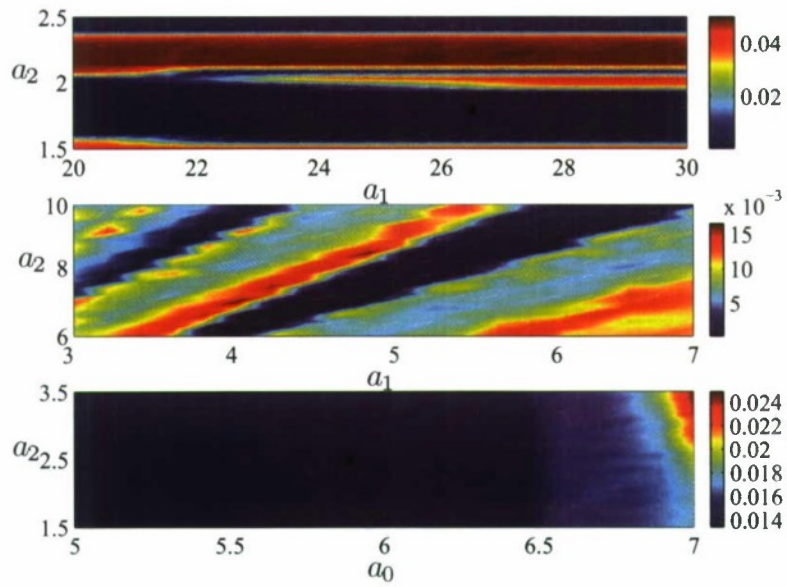


Figure 6: Top: Average Defect Metric as a function of parameters a_1 and a_2 of V_{squ1}, with $a_0 = 0.828$. Middle: Average Defect Metric as a function of parameters a_1 and a_2 of V_{squ2}. Bottom: Average Defect Metric as a function of parameters a_0 and a_2 of V_{hon}, with $a_1 = 17.9$ and $a_3 = 1.823$. The stars correspond to the solutions of [1]

potentials. Since it is based on Fourier space considerations, it is ideal for target structures with simple spectra, like crystals and quasi-crystals (see [14] for a recent work on assembly of crystals and quasi-crystals with a simple double-well potential). The other two tools, infinitesimal mechanical stability and backward integration, provide simple (no optimization required) and fast (no MD's or very short run MD's needed) criteria that can a priori exclude "bad" portions of the parameter space to be searched. This makes them invaluable when the potential depends on 3 or more parameters.

References

[1] M. Rechtsman, F. Stillinger, and S. Torquato, "Designed interaction potentials via inverse methods for self-assembly," *Physical Review E*, vol. 73, p. 011406, January 2006.

[2] F. Theil, "A proof of Crystallization in Two Dimensions," *Communications in Mathematical Physics*, vol. 262, pp. 209–236, 2006.

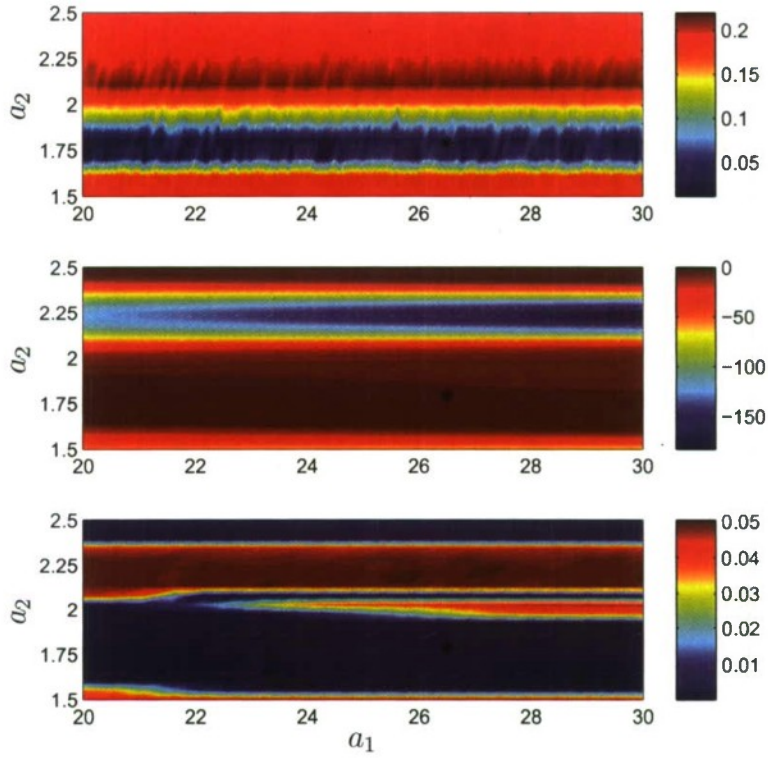


Figure 7: Top: Defect Metric as a function of parameters a_1 and a_2 of V_{squ1}, with $a_0 = 0.828$. Middle: Minimum frequency squared ω^2 from mechanical stability analysis. Bottom: Average Defect Metric from backward integration. The star corresponds to the solution of [1].

[3] D. Pines and D. Bohm, "A collective description of electron interactions: II. collective *vs* individual particle aspects of the interactions," *Physical Review*, vol. 85, pp. 338–353, January 1952.

[4] J. Percus and G. Yevick, "Analysis of classical statistical mechanics by means of collective coordinates," *Physical Review*, vol. 110, pp. 1–13, April 1958.

[5] S. Edwards and M. Schwartz, "Statistical mechanics in collective coordinates," *Journal of Statistical Physics*, vol. 110, pp. 497–502, 2003.

[6] O. Uche, S. Torquato, and F. Stillinger, "Collective coordinate control of density distributions," *Physical Review E*, vol. 74, p. 031104, September 2006.

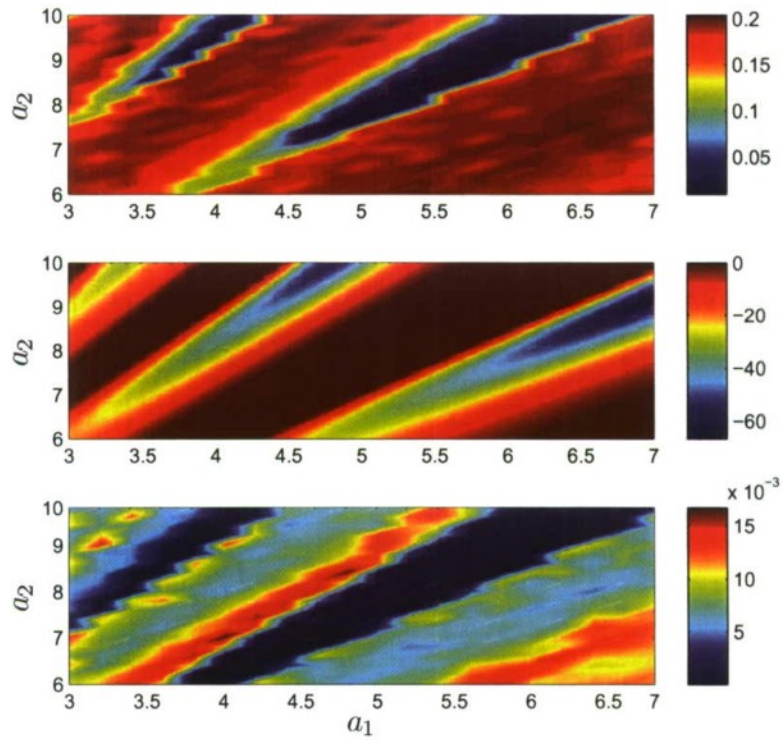


Figure 8: Top: Defect Metric as a function of parameters a_1 and a_2 of V_{squ2}. Middle: Minimum frequency squared ω^2 from mechanical stability analysis. Bottom: Average Defect Metric from backward integration.

[7] M. Boutin and G. Kemper, "On reconstructing n-point configurations from the distributions of distances or areas," *Advances in Applied Mathematics*, vol. 32, pp. 709–735, 2004.

[8] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics*, no. 117, 1995. LAMMPS is freely distributed as an open source code at http://lammps.sandia.gov.

[9] S. Nose, "A molecular dynamics method for simulations in the canonical ensemble," *Molecular Physics*, vol. 52, no. 2, pp. 255–268, 1984.

[10] S. Nose, "A unified formulation of the constant temperature molecular dynamical methods," *Journal of Chemical Physics*, vol. 81, pp. 511–519, July 1984.

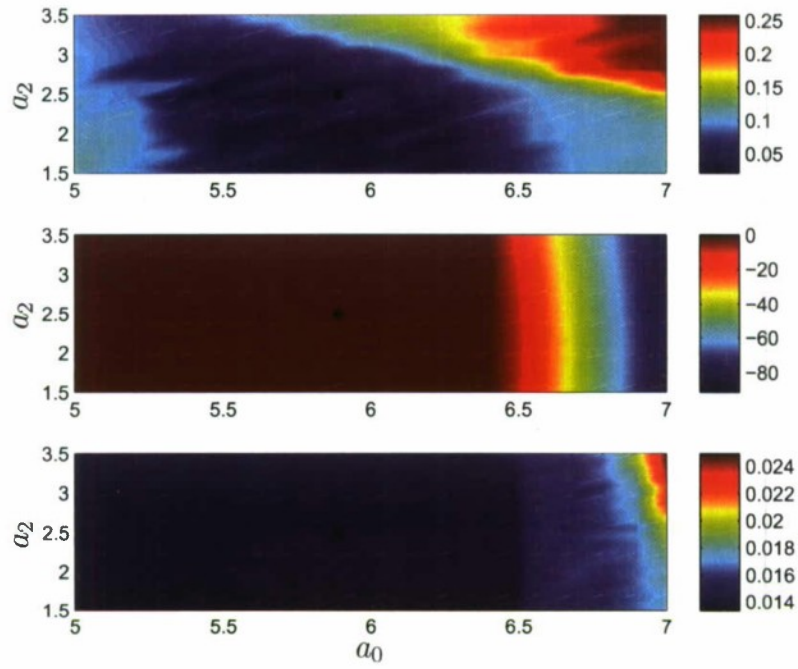


Figure 9: Top: Defect Metric as a function of parameters a_0 and a_2 of V_{hon}, with $a_1 = 17.9$ and $a_3 = 1.823$. Middle: Minimum frequency squared ω^2 from mechanical stability analysis. Bottom: Average Defect Metric from backward integration. The star corresponds to the solution of [1].

[11] W. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Physical Review A*, vol. 31, pp. 1695–7, March 1985.

[12] d M. Schonlau and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[13] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization*, vol. 17, pp. 1–13, 1999.

[14] M. Engel and H.-R. Trebin, "Self-assembly of monatomic complex crystals and quasicrystals with a double-well interaction potential," *Physical Review Letters*, vol. 98, p. 225505, June 2007.

Appendix

In this appendix, we present proofs that the functions d and u defined, respectively, in (5) and (7), are metrics (actually, u is a pseudometric). From the definition (4), it is immediately seen that $|c(\mathbf{k})| \leq N$. We have, for d:

1. $d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\})$ is well-defined:

$$\begin{aligned} d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) &= \frac{1}{N}\int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) \left| |c^{(1)}(\mathbf{k})|^2 - |c^{(2)}(\mathbf{k})|^2 \right| \\ &\leq \frac{1}{N}\int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) (|c^{(1)}(\mathbf{k})|^2 + |c^{(2)}(\mathbf{k})|^2) \\ &\leq 2N \int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) < \infty. \end{aligned}$$

2. d is symmetric in its arguments (obvious)

3. d satisfies the triangle inequality:

$$\begin{aligned} & d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(3)}\}) \\ = & \frac{1}{N}\int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) \left| |c^{(1)}(\mathbf{k})|^2 - |c^{(3)}(\mathbf{k})|^2 \right| \\ \leq & \frac{1}{N}\int \frac{d^2\mathbf{k}}{(2\pi)^2} w(\mathbf{k}) (\left| |c^{(1)}(\mathbf{k})|^2 - |c^{(2)}(\mathbf{k})|^2 \right| + \left| |c^{(2)}(\mathbf{k})|^2 - |c^{(3)}(\mathbf{k})|^2 \right|) \\ = & d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) + d(\{\mathbf{r}^{(2)}\}, \{\mathbf{r}^{(3)}\}). \end{aligned}$$

4. If $d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) = 0$, the two point sets are identical up to a global translation.

$$\begin{aligned} & d(\{\mathbf{r}^{(1)}\}, \{\mathbf{r}^{(2)}\}) = 0 \\ \Rightarrow\ & |c^{(1)}(\mathbf{k})|^2 - |c^{(2)}(\mathbf{k})|^2 = 0,\ \forall \mathbf{k} \in \mathbb{R}^2 \\ \Rightarrow\ & \int \frac{d^2\mathbf{k}}{(2\pi)^2} e^{i\mathbf{k}\cdot\mathbf{r}} (|c^{(1)}(\mathbf{k})|^2 - |c^{(2)}(\mathbf{k})|^2) = 0 \\ \Rightarrow\ & \sum_{i\neq j}^{N} \delta(\mathbf{r} - (\mathbf{r_l} - \mathbf{r_j})) = \sum_{m\neq n}^{N} \delta(\mathbf{r} - (\mathbf{r_m} - \mathbf{r_n})), \end{aligned}$$

   which means that the relative positions in the first point set are the same as in the second point set, including multiplicities. This implies the statement, i.e. the second point set is just a translation of the first.

Now, for u:

1. $u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(2)}\})$ is well-defined:
   From the definition of $I(s)$, we have that $I(s) \leq N$. Then,

$$\begin{aligned} u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(2)}\}) &= \int_0^\infty w(s)\,|I_1(s) - I_2(s)|ds \\ &\leq \int_0^\infty w(s)\,(I_1(s) + I_2(s))ds \\ &\leq 2N \int_0^\infty w(s)ds < \infty. \end{aligned}$$

2. u is symmetric in its arguments (obvious)

3. u satisfies the triangle inequality:

$$\begin{aligned} & u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(3)}\}) \\ = & \int_0^\infty w(s)|I_1(s) - I_3(s)|ds \\ \leq & \int_0^\infty w(s)(|I_1(s) - I_2(s)| + |I_2(s) - I_3(s)|)ds \\ = & u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(2)}\}) + u(\{\mathbf{r}^{(2)}\},\{\mathbf{r}^{(3)}\}). \end{aligned}$$

4. If $u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(2)}\}) = 0$, the pairwise distances in the first point set are the same as in the second point set, including multiplicities. This implies [7] that the point configurations are identical modulo translations and rotations, except for a set of configurations of measure zero in the space of all possible point configurations. Hence u is only a pseudometric.

   First, we compute $I(s)$ analytically. We have,

$$\begin{aligned} I(s) &= \frac{1}{N\pi s^2}\int_{\|\mathbf{k}\|\leq s} \frac{d^2\mathbf{k}}{(2\pi)^2}|c(\mathbf{k})|^2 \\ &= \frac{1}{N\pi s^2}\int_0^s \frac{kdk}{2\pi}\frac{1}{2\pi}\int_0^{2\pi} d\phi|c(k\cos\phi\,\hat{\mathbf{x}} + k\sin\phi\,\hat{\mathbf{y}})|^2 \\ &= \frac{1}{N\pi s^2}\int_0^s \frac{kdk}{2\pi}\frac{1}{2\pi}\int_0^{2\pi} d\phi\sum_{i,j} e^{-ikr_{ij}\cos(\phi-\phi_{ij})} \\ & \quad [\text{Note}: \mathbf{r}_i - \mathbf{r}_j = r_{ij}\cos(\phi_{ij})\hat{\mathbf{x}} + r_{ij}\sin(\phi_{ij})\hat{\mathbf{y}}] \\ &= \frac{1}{N\pi s^2}\int_0^s \frac{kdk}{2\pi}\sum_{i\neq j} J_0(r_{ij}k) \\ &= \frac{1}{2N\pi^2 s^2}\sum_{i\neq j}\frac{1}{r_{ij}}\int_0^s dk(kJ_1(r_{ij}k))' \\ &= \frac{1}{2N\pi^2 s^2}\sum_{i\neq j}\frac{1}{r_{ij}}sJ_1(r_{ij}s) = \frac{1}{2N\pi^2}\sum_{i\neq j}\frac{J_1(r_{ij}s)}{r_{ij}s}. \end{aligned}$$

Then,

$$
\begin{aligned}
& u(\{\mathbf{r}^{(1)}\},\{\mathbf{r}^{(2)}\}) = 0 \\
\Rightarrow\ & I_1(s) = I_2(s)\ \forall s \in [0,\infty) \\
\Rightarrow\ & \sum_{i\neq j} \frac{1}{r_{ij}^{(1)}} J_1(r_{ij}^{(1)} s) = \sum_{n\neq m} \frac{1}{r_{nm}^{(2)}} J_1(r_{nm}^{(2)} s)\ \forall s \in [0,\infty) \\
\Rightarrow\ & \sum_{i\neq j} \frac{1}{r_{ij}^{(1)}} \int_0^\infty dss J_1(rs) J_1(r_{ij}^{(1)} s) \\
=\ & \sum_{n\neq m} \frac{1}{r_{nm}^{(2)}} \int_0^\infty dss J_1(rs) J_1(r_{nm}^{(2)} s) \\
\Rightarrow\ & \sum_{i\neq j} \frac{1}{r_{ij}^{(1)}} \delta(r - r_{ij}^{(1)}) = \sum_{n\neq m} \frac{1}{r_{nm}^{(2)}} \delta(r - r_{ij}^{(2)}),
\end{aligned}
$$

which implies that the pairwise distances in the first point set are the same as in the second point set, including multiplicities.

Appendix D

Correct and fast computation of phase diagrams in the presence of uncertainty

D.1 Constructing the phase diagram for krypton on graphite by detecting pattern boundaries

Constructing the Phase Diagram for Krypton on Graphite by Detecting Pattern Boundaries

Katalin Grubits

27 October 2007

Contents

1 Introduction

The phase diagram for krypton atoms on a graphite substrate can be constructed by running molecular dynamics simulations at a variety of temperature and density conditions, and observing some measure of the order of the system. Rather than using such an order parameter for the system as a whole, the approach taken here is to focus on local properties. In particular, the density of the different phases that are in co-existence in our system.

Using snapshots from molecular dynamics simulations, the fluid and solid regions are identified based on the local density of particles. The overall density in these regions is then calculated and this yields points on the phase co-existence curve at the temperature of the snapshot. The phase diagram can be constructed by analysing snapshots from different final temperatures in this way.

In order to calculate the extent of the fluid region, the boundary between the fluid and solid phases needs to be accurately identified. This is achieved by looking not only at the local density of particles but also at the local density around nearest neighbour particles.

The motivation for this approach is Gelb and Müller's temperature quench method [1] described in the following section.

The results of the pattern boundary detection method will be compared with the phase diagram for krypton on graphite that was found by Vladimir Fonoberov using molecular dynamics simulations. They will also be compared against experimental results due to Butler [Butler] and Larher [Larher], as well as 2D calculations by Sander [Sander].

2 Gelb and Müller's Method

Gelb and Müller's method consists of two parts: the molecular dynamics simulation using the temperature quench method and the analysis of the simulation results.

2.1 Temperature Quench Simulations

Figure 2.1 illustrates the temperature quench method for the molecular dynamics simulations. The system starts in a single-phase equilibrated state, shown as point A. The temperature is then suddenly dropped to a point B at a lower temperature. This places the system in an unstable state and causes a spontaneous separation into two co-existing phases. At this point, it is not necessary for the simulation to continue until the system is once again in equilibrium. Only local equilibrium is required.

2.2 Analysis of simulation results

Analysis of a snapshot of the system after the above temperature quench simulation gives two points on the phase co-existence curve at the final temperature. One will be for the lower density phase (vapour in the example) and one for the higher

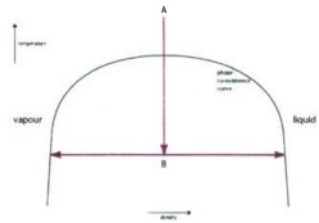


Figure 2.1: Example phase co-existence curve with temperature in the vertical direction and density in the horizontal direction. The temperature is dropped from an equilibrated initial condition at A to a lower temperature at B, causing a separation of phases. Figure based on [1] Fig. 1.

density phase (liquid in the example). The lower portion of the phase diagram can be constructed in this way, however the method is not successful in the broad, flat region of the phase diagram. In this region there is a range of densities for one temperature at which the phases co-exist and thus it is difficult to identify the phases.

Gelb and Müller analysed a snapshot from the simulation by first dividing the simulation box into sub-cells. The density of each of these sub-cells was then calculated, as well as the local co-ordination number of each particle. Boundary particles were identified as those particles that had a local co-ordination number greater than the low-density phase and less than the high-density phase. Sub-cells that consisted of more than 30% boundary particles were excluded from the analysis.

A histogram of the remaining cell densities was constructed. Examples of such histograms for our krypton on graphite system are shown in Figure 2.2. The histogram of densities separates into a bi-modal distribution, reflecting the densities of the phases that co-exist in the system. The density of each of these phases is identified as the most common density within the range of densities for that phase.

As can be seen from Figure 2.2(b), the histogram can have more than one prominent peak in the range of densities that correspond to one of the co-existing phases. It is the highest peak which is taken to be the relevant density. Further, for high (or low) total densities, there may not be enough sub-cells with densities in the range of densities of the other phase for a confident assessment of the peak in the histogram. This is illustrated in Figure 2.2(c).

The results obtained by Gelb and Müller's method of analysis are shown in Figure 2.3. For comparison, results of molecular dynamic simulations with 10 000 atoms (instead of the 2496 particles used here) are indicated, as well as experimental results and numerical 2D results. The method was repeated for total densities of 0.467 and 0.612.

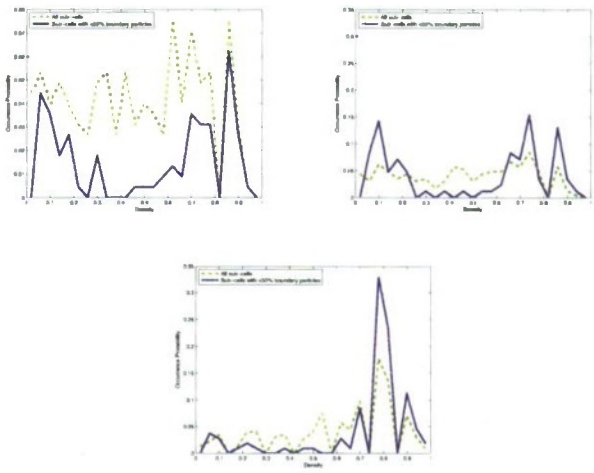


Figure 2.2: (a) Histogram of densities including all sub-cells (dashed green line) and including only sub-cells with less than 30% of particles being boundary particles (solid blue line). Temperature=75K, total density=0.467. (b) As for (a) but area under solid blue line normalised to 1. Temperature=80K, total density=0.467. (c) As for (b). Temperature=80K, total density=0.612.

3 Pattern Boundary Detection Method

Instead of following Gelb and Müller's method of excluding sub-cells with too many boundary particles and constructing histograms of densities, in the pattern boundary detection method we use the boundary particles to define the fluid regions. The areas of such fluid regions are then calculated by summing the area of the Voronoi cells of the constituent particles. This yields a density for the fluid region that does not exclude any particles in its calculation. It also allows us to get over- and under-estimates of the fluid density by accounting for the boundary particles in different ways.

The major differences from Gelb and Müller's method are thus:

1. Find boundary particles by counting nearest neighbours and nearest neighbours of nearest neighbour particles.

2. Calculate area of each phase by finding the Voronoi cell of each particle and not by using areas of sub-cells.

3. Include all particles and regions and do not arbitrarily eliminate regions.

4. Find over and under estimates for points on phase co-existence curve by attributing different areas to fluid regions.

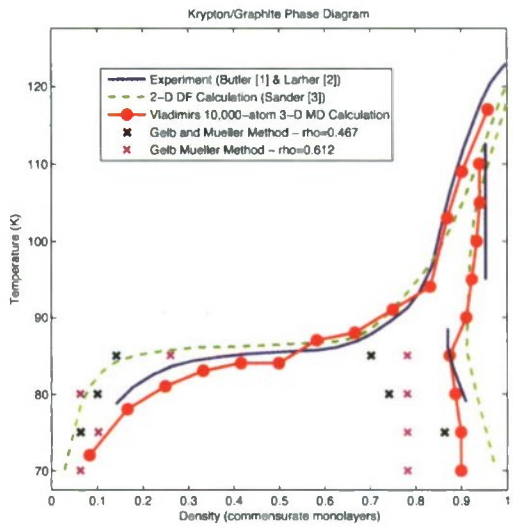


Figure 2.3: Comparison of phase co-existence points found by Gelb and Müller's method of analysis with molecular dynamics, experimental and 2D calculation results.

Figure 3.1 shows a typical snapshot from a simulation of 2496 particles at a final temperature of 70K with a total density of 0.467. The highlighted particles are within the simulation box. This box has periodic boundary conditions in both directions. In order to see the neighbouring particles and calculate the Voronoi cells on the edges, the particles have been repeated outside the simulation box. Fluid particles are coloured blue, boundary particles are coloured green, and solid particles are coloured red.

In the solid phase, particles are in a triangular lattice configuration with 6 nearest neighbours. Fluid particles have less than 6 nearest neighbours. Boundary particles have less than 6 nearest neighbours and have at least one nearest neighbour that has 6 nearest neighbours. Thus, by counting the number of nearest neighbours of each particle and keeping track of which neighbours have fewer than 6 nearest neighbours, each particle can be identified as either a fluid particle, a solid particle, or a boundary particle.

3.1 Calculating Densities

Having identified the type of each particle, the density of the fluid regions can be calculated in the following way. Let n_x denote the number of particles of type x and let A_x denote the sum of the area of all Voronoi cells corresponding to particles of

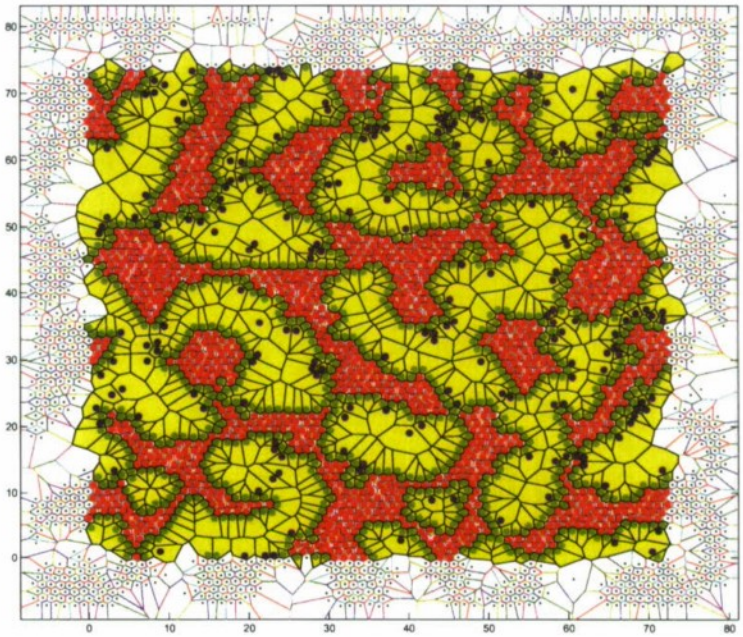


Figure 3.1: Typical snapshot from simulations, showing Voronoi cells of particles in doubly periodic box. Particles within the simulation box are highlighted: blue = fluid particle, green = boundary particle, red = solid particle.

type x. The fluid density is then given by

$$\text{Fluid Density} = \frac{n_{fluid}}{(A_{fluid} + A_{bdry}) - \left(n_{bdry} \times \frac{A_{solid}}{n_{solid}}\right)}. \tag{3.1}$$

The area in the denominator is the area attributed to the fluid particles. The first term corresponds to the area of the Voronoi cells coloured yellow in Figure 3.1. The second term is an estimate for the area occupied by the boundary particles. The Voronoi cells of the boundary particles can not be used to calculate the area occupied because the boundary particle is to be treated as the edge of the solid phase. The boundary particles are each attributed an area equal to the average area occupied by a particle in the solid phase.

An under-estimate for the fluid density does not take into account the area occupied by the boundary particles and thus over-estimates the area occupied by

the fluid region:

$$\text{Fluid Density Under} - \text{Estimate} = \frac{n_{fluid}}{A_{fluid} + A_{bdry}}. \tag{3.2}$$

An over-estimate for the fluid density can be obtained by using a different definition of fluid particles. Designate a particle as a *large cell* particle if its Voronoi cell has an area greater than 1.5 times the average area of the solid particle cells. Then

$$\text{Fluid Density Over} - \text{Estimate} = \frac{n_{large\ cells}}{A_{large\ cells}}. \tag{3.3}$$

Figure 3.2 shows the particles that are designated as *large cell* particles in black, with their Voronoi cells coloured yellow.

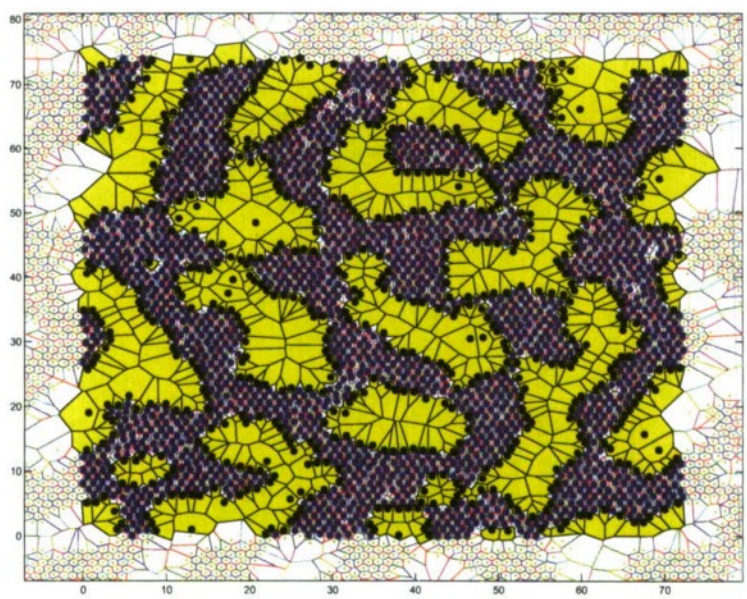


Figure 3.2: Particles defined to be *large cell* particles are coloured black for the same simulation as shown in Figure 3.1. The Voronoi cell of such particles are coloured yellow.

To calculate the density of the solid phase, only particles in the inner core of the solid regions were used. Such inner core particles have the correct number of nearest neighbours and each of their nearest neighbours also has the correct number of nearest neighbours. The inner core of the solid region is less likely to have defects

than locations close to the edge of the solid region. The density is given by:

$$\text{Solid Density} = \frac{n_{solid\ inner\ cluster}}{A_{solid\ inner\ cluster}}. \tag{3.4}$$

The inner core regions can be seen in Figure 3.3 for the same simulation as shown in Figure 3.1.

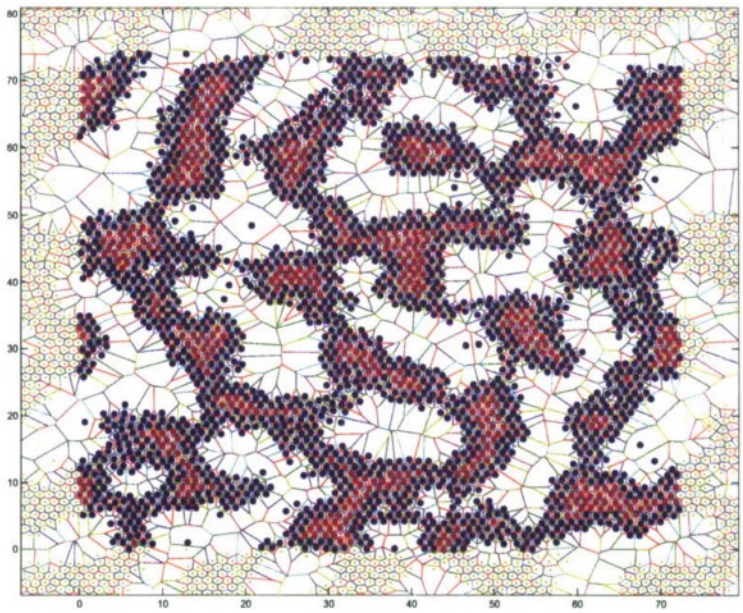


Figure 3.3: Particles in the inner core of the solid regions are coloured magenta, for the same simulation as shown in Figure 3.1.

4 Results

The points obtained on the phase co-existence curve for krypton on graphite using the Pattern Boundary Detection method are shown in Figure 4.1. The molecular dynamics simulation, experimental and 2D calculation results are shown for comparison. The different colours (yellow, magenta and cyan) refer to different total densities that were used for the simulations. The small coloured dots on the left side of the phase diagram correspond to the under- and over-estimates of the fluid density.

It can be seen that the points on the left of the phase co-existence curve agree well with the molecular dynamics simulation results at low temperatures. As the temperature increases and the phase co-existence curve becomes flatter, the method performs worse, as expected. The results for the right side of the phase diagram (density of the solid phase) have an unexplained constant offset from the molecular dynamics simulation results. Note that simulations were carried out at each temperature for each of the total densities – coloured points that can not be seen in the figure are obscured by other points, indicating a good agreement between results from different total densities.

4.1 Speed-up

The main advantage of the Pattern Boundary Detection Method is that it is faster than a long molecular dynamics simulation that requires the system to be in equilibrium. The temperature quench method needs only one equilibrated single-phase initial condition. This can then be used to obtain results at a variety of lower temperatures. After the temperature is dropped suddenly, only local equilibrium is necessary and so there is no need to wait for the whole system to equilibrate.

The Pattern Boundary Detection method yields two points on the phase diagram from one final temperature, thus halving the number of simulations that need to be carried out. Furthermore, identifying the phases requires only one snapshot from the simulation. There is no global quantity, such as an order parameter, that needs to be tracked over successive snapshots (as for example in methods that observe the heat capacity).

A further advantage of the Pattern Boundary Detection Method is that it provides under- and over-estimates of the fluid density. Up until the broad, flat region of the phase diagram (where the method is not expected to work anyway), these estimates bounded the results that were obtained by using any of the total densities considered.

4.2 Extension to the neck region of the phase diagram

The method, as described above, identifies the fluid and solid phases by counting nearest neighbours. This is effectively a measure of the local density. At higher total densities, the difference in density between the fluid and solid phases is no longer large enough for this approach to be successful. These high total density simulations must be considered so as to extend the Pattern Boundary Detection method to the neck region of the phase diagram (in the upper right of Figure 4.1).

The distinguishing feature of the solid phase in these situations is the order of the particles. They form a triangular lattice by adsorption of atoms onto the graphite substrate. Thus a measure of the order, or geometry, of the particles must be used, such as the Defect Measure. This is ongoing work.

A further complication in the neck region of the phase diagram is that the temperature is high enough such that particles that have been adsorbed onto the substrate move significantly, while remaining adsorbed. There is a spiral motion

about the adsorption site. To identify the solid particles, this motion about the adsorption site must be averaged out by looking at a number of snapshots.

5 Conclusions

The Pattern Boundary Detection method provides a way of constructing the lower portion of the phase diagram of krypton on graphite. Speed-up over equilibrated molecular dynamics simulations comes from using the temperature quench method in simulations, from finding two points on the phase co-existence curve for every final temperature, and from needing only one snapshot per final temperature.

References

[1] Gelb, Lev D., and Erich A. Müller [2002], Location of phase equilibria by temperature-quench molecular dynamics simulations. *Fluid Phase Equilibria* **203**, 1-14.

[Butler]

[Larher]

[Sander]

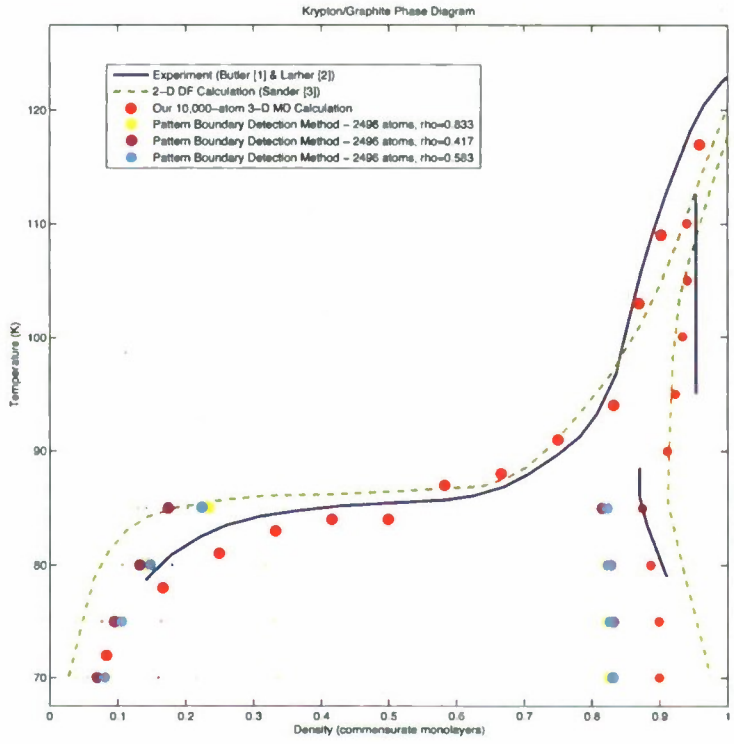


Figure 4.1: Phase diagram for krypton on graphite showing the results of the Pattern Boundary Detection method (in yellow, magenta and cyan), as well as molecular dynamics, experimental and 2D calculation results. Small coloured dots indicate under- and over-estimates of the fluid density.

D.2 Quality assessment tools for lattices

Quality Assessment Tools for Lattices

Katalin Grubits

31 July 2007

Contents

1 Introduction

The structure of crystalline solids has been an interest of the condensed matter and materials science fields for a long time. The defects present in these solids determine many of the physical properties of the material. Investigations have been both theoretical and experimental, concentrating on how different types of defects, and the number of them, affect mechanical, electrical and optical properties. A large number of experiments have been performed that attempt to make materials that are free of defects, or examine the formation energies and movement of defects. The number of defects and their type is commonly deduced from bulk properties of the material, such as diffraction patterns or shear stress. The focus is on the material as a whole rather than on the environment of each constituent particle.

This report is concerned with a quantitative assessment of the quality of two-dimensional lattices. We are interested not in the way that a lattice responds to some external stimulus that would measure a bulk property of the lattice as a whole, but rather assessing the positions of the constituent particles in a manner similar to the human eye. The aim is to quantify what the eye sees when comparing two lattices and deciding that one is better than the other. The notion of "better" may depend on which property of the lattice is more important to the assessor, or the goal of assessing the lattice. The measures of the defects of a lattice that we have developed is thus concerned with the local neighbourhood of each particle, reflecting the eye's propensity to judge sub-regions of the lattice and how these regions combine.

Four defect metrics that have this local nature are described and compared. The most versatile of these, the Defect Measure, is used as a tool in applications that arise from the challenge of designing an isotropic potential that leads to the self-assembly of particles into a lattice. All particles are identical and move in a finite two dimensional area.

The self-assembly of particles is of importance in the diverse fields of understanding how biological or chemical components form a coherent whole and multi-vehicular surveillance. Large numbers of small vehicles moving in a lattice formation has been proposed as an efficient way of surveying the landscape. (Ref?) The local deviations from a perfect lattice formation must be understood in this context. Vehicles that communicate only with their neighbouring vehicles should be able to assemble into and maintain a lattice formation, eliminating the need for each vehicle's specific trajectory to be programmed. The design of the ideal isotropic potential for doing so requires an effective measure of the defects in the lattice, as does the evaluation of the robustness of that potential.

2 Types of Lattice Defects

There are a number of different types of defects that can occur in a monatomic two-dimensional lattice. The defects most commonly referred to are listed below.

- *Vacancy*: a lattice site that should have a particle is unoccupied.

- *Interstitial*: a particle occupies a lattice site that should not have a particle occupying it.
- *Frenkel pair*: a vacancy and an interstitial are nearby. A particle is at a lattice site which should be unoccupied, leaving a nearby lattice site that should be occupied empty.
- *Topological defect*: a region in a lattice where the ordered structure is different to the rest of the lattice. For example, in a honeycomb lattice a region that has five particles in a ring rather than the required six.
- *Split interstitial*: two particles share a lattice site, typically by having their centre of mass at the lattice site where there should be one particle.
- *Edge dislocation*: an extra line of particles inserted part of the way into the lattice. The adjacent lines of correctly ordered particles bend around the line that terminates. Dislocations are breaks in the translational symmetry of the lattice.
- *Disclinations*: a line defect that results in a rotation if the orientation of the lattice around the defect is tracked.
- *Grain boundaries*: regions, typically lines, where the orientation of the lattice changes abruptly. Frequently caused by two lattices growing separately and then meeting.

3 Metrics for Assessing the Quality of Lattices

We describe four defect metrics that focus on the local configuration of particles. These metrics are compared in the following section.

3.1 Defect Measure

The Defect Measure is a tool that was developed in order to compare the quality of lattices. The human eye is frequently a good judge of the quality of a lattice, however, a more qualitative assessment was sought in order to efficiently assess lattices that are formed during the optimization procedure used for finding a potential that leads to the self-assembly of particles in a plane.

Given particle positions in a plane, the quality of the lattice that is formed is determined by the desired or target lattice. If the target lattice is known then calculating the Defect Measure requires only Step 3 in the procedure described below.

If the target lattice is not known, or if it is necessary to find the type of lattice that the system of particles is forming, then Steps 1 and 2 can be applied, to determine what the target lattice is. Identification of a target lattice means identifying both the type of lattice (honeycomb, triangular, etc.) and the minimum lattice

spacing, which we shall call the *lattice constant*, a. That is, identifying the target lattice involves identifying the shape and the scaling.

The need to identify the target lattice arises when optimizing the potential for the self-assembly of a honeycomb lattice. The competing lattice is a triangular lattice that may have one of two different lattice constants, depending on the density of particles. When searching over parameters for the honeycomb potential, if it can be seen that the lattice that is forming for a set of parameters is heading towards a triangular lattice rather than a honeycomb lattice, the simulation can be stopped and another set of parameters may be tried. In this way, the optimization procedure can be sped up.

Identifying the lattice constant is important not only because it contributes to the identification of the target lattice but also because for some potentials, the lattice that is formed has a different lattice constant than that suggested by the isotropic inter-particle potential that led to the formation of the lattice. For example, a potential designed to form a honeycomb lattice with a lattice constant that is equal to the distance to the first minimum of the potential, may form a lattice with a different lattice constant due to the sensitivity to the density of particles. In assessing the quality of the resulting lattice, if only the shape of the lattice is important, then the actual lattice constant must be found.

The algorithm for identifying the target lattice and computing the Defect Measure does not need the positions of all particles in the plane. It only requires a list of distances to the nearest 20 neighbours of each particle. To distinguish between the four types of lattices considered – triangular, honeycomb, square and Kagomé, it is only necessary to consider the distances to the nearest 15 neighbours of each particle, however results are more reliable (identification of target lattice is improved) if more nearest neighbours are considered.

This is an important feature for our surveillance example. Vehicles would only need to detect other vehicles that are in a certain range that covers an area in which there would be approximately 20 other vehicles. The direction of each detected vehicle is unimportant, only the distance to that vehicle. In this way, the position of each vehicle does not need to be tracked.

In the identification steps (Steps 1 and 2), it is assumed that the lattice is reasonably well formed. This assumption effectively means that the human eye would be able to distinguish the type of target lattice.

Any number of particles may be in the lattice being assessed, if periodic boundary conditions are imposed. If this is not the case, then the number of particles must be large enough such that the number of inner particles of the lattice out-number the number of boundary particles. The larger the number of inner particles with respect to boundary particles, the better the algorithm is able to identify the target lattice.

3.1.1 Procedure for identifying the target lattice and computing the Defect Measure

Step 1: *Identify lattice constant, a*

1. List inter-particle distances d_{pj} for particle p in ascending order.
2. Find clusters of values for each particle.
3. Average over values in clusters of lowest values to find a.

Step 2: *Identify type of lattice*

1. Count how many values in first few clusters identified in Step 1.
2. Find mode of number of particles in clusters of lowest values, then clusters of next lowest values, etc (for a majority of particles, the number of values in each cluster should be the same).
3. Compare number of particles at each distance (modes of clusters) with known values for possible types of lattices.

   Number of particles in perfect lattice:

Mode of:	cluster 1	cluster 2	cluster 3
Triangular	6	6	6
Square	4	4	4
Honeycomb	3	6	3
Kagomé	4	4	6

Step 3: *Compute the Defect Measure*

Compare the given lattice to a perfect lattice of the same type, with the same lattice constant, to find a measure of the defects, i.e. the quality of the lattice.

For a particle p in the target lattice, find the distance r that is halfway between the distance to the closest neighbours and the next closest neighbours. This distance is shown as the red circle in Figure 3.1. For a triangular lattice, there are 6 closest neighbours at a distance of a and 6 next closest neighbours at a distance of $\sqrt{(3)}a$. The red circle has a radius of $r = (1 + \sqrt{(3)})a/2$.

Define the *nearest neighbours* of a particle p to be those particles that are within a radius r of particle p.

1. Choose weights, $\omega_{defect\ type}$ for each type of defect (see discussion below).
2. For each particle, p, construct the nearest neighbours circle and compute the Defect Measure of that particle according to which of the following types of defects apply (shown in Figure 3.2):

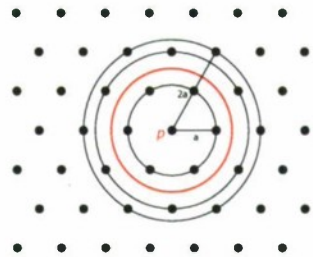


Figure 3.1: The nearest neighbours circle (red circle) of particle p is halfway between the closest particles and the next closest particles.

- *Displaced particles*

$$(\text{Defect Measure})_p = \omega_{displaced} \times \left(\sum_{j \in \text{nearest neighbours}} ff_{jp} \times (d_{pj} - a)^2 \right)$$

- *Missing particles*

$$(\text{Defect Measure})_p = \omega_{missing} \times n_{missing} \times a^2$$

- *Extra particles*

$$(\text{Defect Measure})_p = \omega_{extra} \times n_{extra} \times a^2$$

- *Lone particles*

$$(\text{Defect Measure})_p = \omega_{missing} \times 6 \times a^2 + \omega_{lone} \times a^2$$

- *Boundary particles*

$$(\text{Defect Measure})_p = \omega_{boundary} \times a^2$$

3. The Defect Measure for the lattice is given by summing over all particles p:

$$(\text{Defect Measure for Lattice}) = \sum_p (\text{Defect Measure})_p$$

ff_{jp} is the *fade factor* for particle j with respect to particle p. The fade factor allows particles to fade out of view of particle p rather than disappear as they cross the nearest neighbours circle. The fade factor is equal to 1 for $d_{pj} < (5 + 3\sqrt{3})a/8$ and is equal to 0 for $d_{pj} > (1 + \sqrt{3})a/2$ (the nearest neighbours circle). The fade factor decreases from 1 to 0 over a distance that is equal to a quarter of the distance from a to the nearest neighbours circle. Within this region, the fade factor decreases in a cubic polynomial fashion with horizontal tangent at the end points of the region.

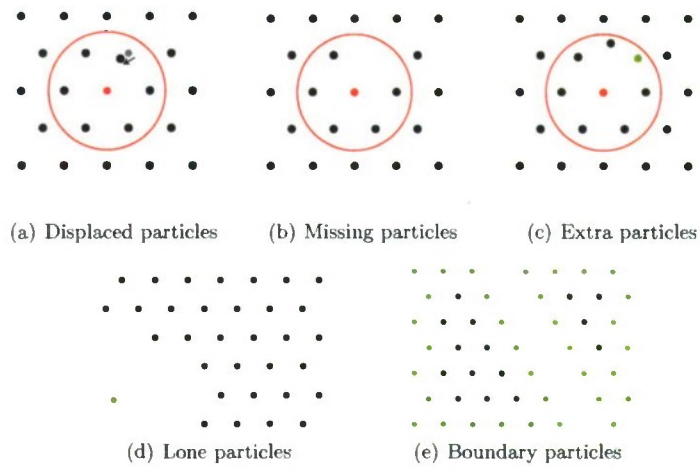


Figure 3.2: Type of defects used in computing the Defect Measure

The weights, $\omega_{defect\ type}$, for each type of defect are chosen according to the severity of the defect. This depends on the goal. For example, in the surveillance situation, if the collision avoidance of the vehicles is an issue, then a larger weight would be given to the weight for extra particles, ω_{extra}, in order to deter more strongly lattices with extra particles. When optimizing the potential for the self-assembly problem, a larger weight for boundary particles, $\omega_{boundary}$, may be necessary to penalize the formation of distinct sub-lattices. Note that boundary particles are not penalized for missing particles in their nearest neighbours circle. The effect of changing the weights for the various types of defects will be discussed further in Section 4.

One of the advantages of the Defect Measure is that it allows this flexibility to penalize different types of defects more heavily. In this way, it is a tool that can be shaped for the specific task at hand.

The Defect Measure is lower for higher quality lattices. A perfect lattice will only have a Defect Measure equal to zero if the weight for boundary particles is set to zero.

Note that all of the varieties of defects discussed in Section 2 are taken into account by the types of defects listed in Step 3.2. For example, grain boundaries are taken into account by contributions to the Defect Measure from displaced, missing and extra particles.

The Defect Measure essentially compares the local density around each particle to that of the target lattice, with a correction for the distance between particles, added on.

For this reason, only the inter-particle distances are required. The geometry of the particles is not considered. This is sufficient because, for any potential that could

lead to self-assembly, if the correct number of particles are put into a region the size of the nearest neighbours circle, they will arrange themselves into approximately the correct configuration due to the potential between them.

3.1.2 Advantages of the Defect Measure

The Defect Measure has a number of advantages over the methods for quantifying the quality of lattices surveyed at the beginning of Section 3.

- The Defect Measure gives a *local* assessment of the quality of a lattice. Apart from the versatility of the Defect Measure in applications that this leads to, a local assessment of a lattice is closer to the qualitative assessment that a human eye would make of a lattice.
- Each particle's contribution to the quality of the lattice can be quantified. In this way, regions of the lattice that are not well formed can be identified. This is useful in applications such as the one discussed in Section 5.4.
- The primary types of defects, and the number of such defects, that occur in a lattice can be easily identified.
- The flexibility of the Defect Measure due to the assignment of weights to defects, leads to a versatility that may be exploited in applications.
- The target lattice does not need to be specified.
- The Defect Measure is invariant under rotations, reflections and translations of the lattice.

3.1.3 Identifying the boundary of a lattice

To implement the procedure for calculating the Defect Measure (with non-zero weights for the boundary particles), the particles that form the boundary of the lattice must be identified, using only the distances to the nearest 20 neighbours.

The boundary particles of a perfect lattice can be identified by counting the number of particles at a distance of a, and the number of particles within a distance of $2a$ from each particle. For a triangular lattice, an inner particle has 6 neighbours at a distance of a, 6 neighbours at a distance of $\sqrt{3}a$, and 6 neighbours at a distance of $2a$. Boundary particles can have a maximum of 5 particles at a distance of a and 15 particles within a distance of $2a$. This is shown in Figure 3.3. Particles that satisfy these conditions are designated as boundary particles. Note that these requirements specify the maximum concavity of the boundary that can be detected. A similar construction applies to different types of lattices.

For an imperfect triangular lattice, the condition of having a maximum of 5 particles at a distance of a is loosened to having a maximum of 5 particles within the nearest neighbours circle. The second condition is relaxed to having a maximum of 15 particles within a distance $(1 + \sqrt{7}/2)a$ of the candidate boundary particle.

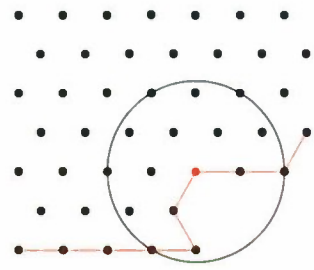


Figure 3.3: A boundary particle in a triangular lattice can have a maximum of 5 particles at a distance of a and 15 particles within a distance of $2a$. The red particle shows such a particle. The red line indicates the boundary of the lattice. The circle encloses particles that are within a distance $2a$ of the red particle.

This distance is halfway between $2a$ and the distance to the next nearest neighbours (at $\sqrt{7}a$). Similarly relaxed conditions apply to other types of lattices.

Identifying the boundary of a lattice given the positions of particles

If a system of particles is not in a reasonably well-formed lattice, yet the goal is to compute the Defect Measure of the particles with respect to some specified target lattice, then it will be necessary to use a different method to identify the boundary particles (for non-periodic boundary conditions). In this situation, the position of each particle is required, as the number of particles in the circles described above may not be at all comparable to that of the target lattice.

The convex hull of a set of points in a plane is the minimal convex set containing all the points. It may be visualised as the shape of an elastic band that has been stretched to encompass all the points and then allowed to collapse around them. The convex hull for a set of points and the points identified as boundary points by this convex hull are shown in Figure 3.4.

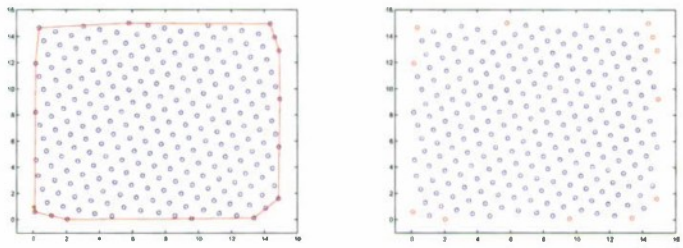
Clearly, this is not what the human eye identifies as the boundary. It is the non-convex hull that correctly identifies the boundary particles. First, a minimum concave curvature, ρ, must be chosen. The non-convex vertices of the boundary are then those particles that are touched by a disk of radius ρ as it is rolled around the outside of the set of points. The following algorithm for finding the non-convex hull is due to E. Boje [Boje [2000]].

Algorithm for finding the non-convex hull (Boje):

1. Find the Delaunay triangulation[1] of the set of points.

2. Find the outside triangles, i.e. those triangles with an edge that does not touch another triangle's edge. Such edges together form the convex hull.

[1]The Delaunay triangulation of a set of points is a triangulation such that no point is inside the circumcircle of any triangle in the triangulation. It is the dual graph of the Voronoi tessellation of the points.



(a) Convex hull (b) Boundary points

Figure 3.4: (a) The convex hull of a set of points in the plane. (b) The points identified by the convex hull as boundary points (shown in red).

3. Recursively delete any outside triangle that has an outside edge longer than 2ρ.
4. Recursively delete outside triangles whose outside edge is the longest and whose circumscribing circle[2] has a radius greater than ρ.
5. Iterate until all triangles pass steps 3 and 4.

The points identified by this algorithm as the particles on the non-convex hull of the set of points in Figure 3.4 is shown in Figure 3.5 in red. These are the boundary particles. Note that the boundary points identified by the convex hull corresponds to the points found by rolling a disk of infinite radius around the set of points.

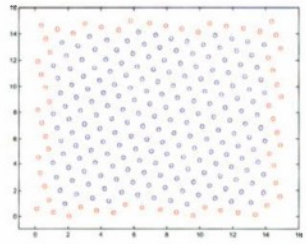


Figure 3.5: The points identified by the non-convex hull as boundary points (shown in red).

[2]The circumcircle of a polygon is a circle that passes through all of the vertices of the polygon.

3.2 Geometric Defect Measure

The Defect Measure takes into account the local density of particles and the distance between particles. It does not consider the local geometry of particles. Despite this, as discussed above, the Defect Measure provides a good assessment of the quality of a lattice. The Geometric Defect Measure was developed as an alternative quality assessment tool that could be compared to the Defect Measure to check that looking at local densities (with a correction for displacements) does indeed lead to a lattice with the correct geometry. It can also be combined with the Defect Measure as a correction, to yield a quality assessment tool that considers local densities, distances between particles and local geometry.

Computation of the Geometric Defect Measure requires the position of each particle in the system as well as the type of target lattice. It focuses on the shape of the lattice and not the scaling. Thus two lattices that differ only by a scaling of the lattice constant will have identical values for the Geometric Defect Measure. It is computationally more expensive than the calculation of the Defect Measure.

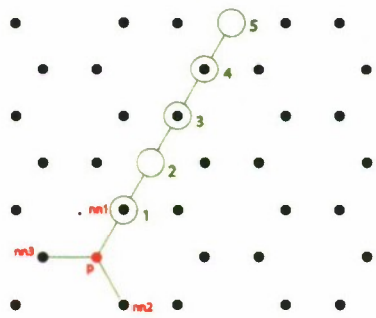


Figure 3.6: Construction used in computing the Geometric Defect Measure for particle p.

The algorithm for computing the Geometric Defect Measure of a honeycomb lattice is outlined below and illustrated in Figure 3.6. The procedure is similar for other types of lattices.

Algorithm for computing the Geometric Defect Measure of a honeycomb lattice:

1. Find the nearest neighbour of particle p. Label it $nn1$. Let the distance between particles p and $nn1$ be d_{nn1}.

2. Extend the line from p to $nn1$ a distance of d_{nn1}. Determine whether there is a particle within a distance $d_{nn1}/8$ of this point, i.e. whether there is a particle in region 2 in Figure 3.6.

3. Continue extending the line from p to $nn1$ in units of d_{nn1} and determining whether a particle is within a region around the end points. Do this for a total of 5 regions or steps from particle p.

4. Compare whether or not a particle is found in each region to that expected from a perfect lattice. For the honeycomb lattice, there should be particles in regions 3 and 4, but not in regions 2 and 5. (Region 1 will have a particle by construction.)

5. If a region j does not have the correct number of particles then $n_{p1j} = 1$, else $n_{p1j} = 0$.

6. Repeat for the second and third nearest neighbours, $nn2$ and $nn3$.

7. Calculate the angle between pairs of nearest neighbours of p: θ_{p12}, θ_{p13}, and θ_{p23}.

8. Sum over all particles p in the lattice to obtain the Geometric Defect Measure of the lattice.

$$\text{Geometric Defect Measure} = \sum_{p} \sum_{i=1}^{3} \left(\sum_{j=1}^{5} n_{pij} + \sum_{k=2,k>i}^{3} |\cos 120^\circ - \cos \theta_{pik}| \right) \tag{3.1}$$

If the system of particles does not have periodic boundary conditions then the extension of the lines from particles to their nearest neighbours should be cut off when a boundary is reached.

A number of modifications to the algorithm outlined above are appropriate for most lattices. Firstly, the number of steps that the lines are extended may be increased or decreased, depending on the type of lattice. For the triangular lattice, looking at only 3 regions is sufficient to give a quality assessment that is comparable to what the human eye would judge. However, 5 regions is more appropriate for a honeycomb lattice. Changing the number of steps taken alters how local the quality assessment is. It is the local nature of the Geometric Defect Measure that makes it useful for detecting regions with many defects.

Secondly, the size of the regions used to determine whether a particle is in the correct position relative to the base particle p may be adjusted. It is appropriate to increase the size of the detection region the further the region is from the base particle. Doing so is compatible with judgements made by the human eye. It is also affected by the importance of having a correctly aligned lattice rather than a skewed lattice. For the honeycomb lattice, a good choice is allowing the radius of the detection region for region s to be $(s-1) \times d_{nn1}/8$ for $s > 1$.

Lastly, as regions further away from the base particle are less important in quantifying the local geometry, the contribution of the more distant regions to the Geometric Defect Measure can be reduced. The first term in parentheses in Equation 3.1 then becomes $\sum_{j=1}^{5} n_{pij} \times (6-s)/4$, for $s > 1$ where s denotes the number of the detection region.

3.3 Voronoi Metric

The Voronoi Metric finds the Voronoi tessellation of the particles in the lattice and compares the area of each Voronoi cell to the area of a Voronoi cell of the target

lattice. The Voronoi tessellation of a set of particles in the plane is the partition of the plane into regions such that any point of the plane in the region corresponding to particle p is closer to p than to any other particle. The Voronoi tessellation of an imperfect lattice can be seen in Figure 3.7(a).

The Voronoi Metric is straightforward to apply to lattices with periodic boundary conditions. The positions of the particles near the bounding box are mapped outside the bounding box in an appropriate way that is consistent with the periodic boundary conditions. This is illustrated in Figure 3.7(b).

The particles of the lattice are shown in blue; the lattice has a bounding box specified by $0 < x < 1$ and $0 < y < 1$. The particles that have been mapped outside the bounding box are coloured cyan. The particle on the left that is coloured magenta lies inside the bounding box. It is mapped to the particle position on the right that is coloured red. Such a construction allows the Voronoi cells of the particles close to the boundary to be calculated without edge effects, for periodic boundary conditions.

Algorithm for computing the Voronoi Metric:

1. Map particles that are close to the bounding box outside the bounding box, respecting the periodic boundary conditions.

2. Find a Voronoi tessellation of the particles.

3. Compute the area of each Voronoi cell that contains a particle in the original lattice.

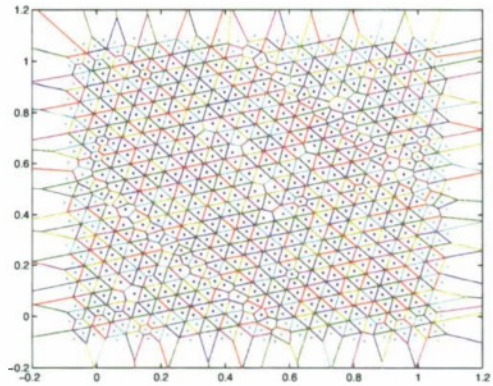
4. The Voronoi metric is given by

$$\text{Voronoi Metric} = \sum_{p}^{N} \left| \text{area}(V(p)) - \frac{\text{area of bounding box}}{N} \right| \tag{3.2}$$

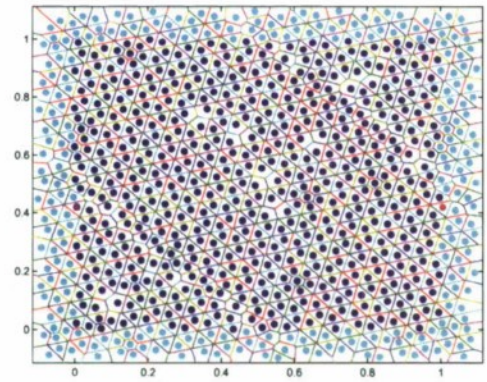
   where N is the number of particles in the original lattice and $V(p)$ is the Voronoi cell containing particle p.

An alternative expression for the area of a Voronoi cell in the perfect lattice may be obtained from the geometry of the target lattice and the best estimate for the lattice constant. The lattice constant may be estimated by Step 1 of the procedure for computing the Defect Measure described in Section 3.1.1. This is particularly important for the honeycomb lattice, for which the lattice constant depends not only on the first minima of the potential but also the density of particles. For a perfect honeycomb lattice (with periodic boundary conditions), the area of each Voronoi cell would be $3\sqrt{3}\,a^2/4$. The second term in Equation 3.2 may be replaced by this expression.

For lattices formed in a bounding box that does not have periodic boundary conditions, an assessment of the quality of the lattice based on the Voronoi Metric can be made by ignoring the contribution from boundary particles. The boundary particles can be identified by using Boje's algorithm for finding the non-convex hull



(a)



(b)

Figure 3.7: (a) Voronoi tessellation of a set of points in the plane. (b) The same Voronoi tessellation as in (a) with the particles of the original lattice coloured blue and the particles that are mapped outside the periodic boundaries coloured cyan. The particle on the left coloured magenta is mapped outside the bounding box to the particle position coloured red on the right.

of a set of points. In this case, the alternative expression for the area of a Voronoi cell in the target lattice, explained in the preceding paragraph, should be used.

To include a contribution from the boundary particles of the lattice, there are a number of options. Modified Voronoi cells for the boundary particles can be formed by taking the area enclosed by the lines of the Voronoi tessellation and the bounding

box. These areas for the boundary particles can then be compared to the areas of the similarly modified Voronoi cells that would be formed for the target lattice. Since the modified Voronoi cells of the boundary particles in the target lattice depend a great deal on the construction of the target lattice, it is best to sum the areas of the modified Voronoi boundary cells of the lattice and compare this to the sum of the areas of the modified boundary cells of the target lattice, rather than compare the areas of individual cells. However, this is not an effective way to assess the quality of a lattice because of the many possible ways of constructing the target lattice.

Another way to include the contribution of the boundary particles to a Voronoi-like metric is to define the Voronoi cells of the boundary particles to be the area enclosed by the lines of the Voronoi tessellation and the non-convex hull of the particles. This method suffers from the same drawback as the previous one, but has the advantage that for a lattice that is not aligned with the edges of the bounding box, the contribution of the boundary particles to the value of the metric will be smaller. This effect is important for lattices that do not require the pressure from the domain walls to form. For example, this method is the appropriate way to include the boundary particles in the Voronoi metric of a triangular lattice that is formed in a domain that is larger than the area occupied by the lattice.

The boundary particles can also be dealt with in the following manner. Assign to each boundary particle of the lattice the area of an inner Voronoi cell in the target lattice. For example, each boundary particle of a honeycomb lattice formed in a domain without periodic boundary conditions, would be assigned an area of $3\sqrt{3}\,a^2/4$. The area occupied by the cells of the boundary particles is then $A_{bdry} = n_{bdry}3\sqrt{3}\,a^2/4$, where n_{bdry} is the number of boundary particles. Compute the area, A_{inner}, of the domain that is occupied by the inner particles of the lattice. The contribution of the boundary particles to the Voronoi-like metric is then $|A_{bdry} - A_{inner}|$.

None of these options for including the contribution of the boundary particles of the lattice to a Voronoi-like metric are satisfactory, for the reasons mentioned above as well as the variability at the edges of computed Voronoi tessellations. The Voronoi Metric is thus of limited use for lattices formed in domains without periodic boundary conditions.

3.4 Cumulative Distribution Function Metric

The cumulative distribution function of the inter-particle distances of a lattice can be used to assess the quality of the lattice. This metric was proposed by Mezić and Runolfsson in a different setting [Mezic and Runolfsson [2004]]. The cumulative distribution function (CDF) metric is most effective when only inter-particle distances up to a distance of slightly above $2a$ are considered. In a perfect lattice, this includes the nearest neighbours, the next nearest neighbours and the third nearest neighbours of each particle.

Algorithm for computing the Cumulative Distribution Function Metric:

1. Let the distance to a point half way between the third nearest neighbours

circle and fourth nearest neighbours circle of particles in the target lattice be d_{max}.

2. For each distance d from $d = 0$ to $d = d_{max}$, find all inter-particle distances of the lattice being assessed that are less than or equal to d.

3. Then
$$CDF_{lat}(d) = \sum_{d_{ij} \leq d,\ i>j} d_{ij} \tag{3.3}$$
for $0 < d < d_{max}$, where the d_{ij} are the inter-particle distances of the lattice.

4. CDF_{target} is defined in a similar way using the inter-particle distances of the target lattice.

5. The Cumulative Distribution Function Metric is given by
$$\int_{l=0}^{l=d_{max}} |CDF_{lat}(l) - CDF_{target}(l)|\, dl \tag{3.4}$$

The CDF of a lattice and its target lattice is shown in Figure 3.8. CDF_{lat} is in blue and CDF_{target} is in green. The value of the Cumulative Distribution Function Metric is the area between the curves.

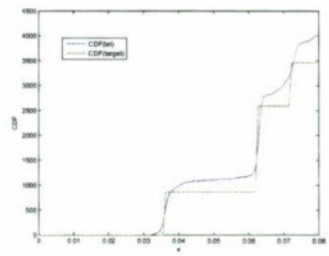


Figure 3.8: The Cumulative Distribution Function Metric computes the area between CDF_{lat} shown in blue and CDF_{target} shown in green.

The question of how to treat the boundary particles of a lattice is also an issue for this metric, especially for domains without periodic boundary conditions. For lattices formed in domains with periodic boundary conditions, the only concern is whether the domain can indeed be filled with a perfect lattice with the specified number of particles. Not all domains have a perfect lattice that completely fills the domain for an arbitrary number of particles. However, this introduces only a very minor error into the value of the CDF metric.

For domains without periodic boundary conditions, a target lattice must be constructed that can be used to find CDF_{target}. The inter-particle distances of the target lattice depend on where the boundary particles are placed, i.e. the shape of the boundary.

A more serious problem with the CDF metric is that there is a cancellation between two different types of defects: missing particles and extra particles. If a particle is missing a neighbouring particle at a distance d' from it and another particle has an extra particle at a distance d' from it, then there will be some cancellation between these two defects, and the value of the CDF metric at d' will be lower than it should be. However, these two defects also affect the surrounding particles and this will add to the value of the CDF metric. How much is added depends on the arrangement of particles around the defect and not the defects themselves.

The CDF metric was designed to only look at inter-particle distances less than d_{max} partially for this reason. There is less opportunity for such cancellation of defects to occur. Another reason for limiting the CDF metric's horizon to d_{max} is that considering all inter-particle distances would put too much emphasis on the long range order of the lattice. When assessing the quality of a lattice, the human eye tends to focus more on the order within regions that have a radius of a few lattice constants, rather than the long range order of the lattice as a whole.

A feature of the CDF metric is that it tends to judge lattices with grain boundaries relatively harshly. This may or may not be a concern depending on the goal and how severe such a defect is considered to be.

4 Comparison of Quality Metrics

A natural question to ask is: Which of the metrics for assessing the quality of a lattice discussed in Section 3 is the best?

Evidently, this depends on which properties of a lattice are more important. Different metrics focus on different aspects, such as having the correct number of particles in approximately the right positions, or having the correct alignment of particles. This will be discussed further below.

There are two straightforward ways to compare metrics that assess the quality of a lattice: whether it can identify the best lattice from a set of lattices, and the computational time taken to compute the value of the metric for a lattice.

In order to compare the metrics from Section 3, a set of 20 lattices were generated, with 576 particles in a domain with periodic boundary conditions. The target lattice was the honeycomb lattice. Each metric was used to assess the lattices and rank them from best to worst. The time taken to calculate the value of the metric for each lattice was averaged over the 20 lattices. The results are shown in Figure 4.1 and Table 4.

Figure 4.1 has the metrics discussed in Section 3 along the vertical axis and the number of each of the 20 lattices used for the comparison along the horizontal axis. Defect Measure 1 and Defect Measure 2 differ only in the weights assigned to the different types of defects. [3] The colours in the figure represent the ranking of the 20 lattices, with a rank of 1 in red being the best lattice and a rank of 20 in dark blue

[3] Weights for Defect Measure 1: $\omega_{displaced} = 1.0, \omega_{missing} = 1.0, \omega_{extra} = 0.8$.
Weights for Defect Measure 2: $\omega_{displaced} = 1.0, \omega_{missing} = 0.02, \omega_{extra} = 0.015$.
There were no boundary or lone particles in these lattices.

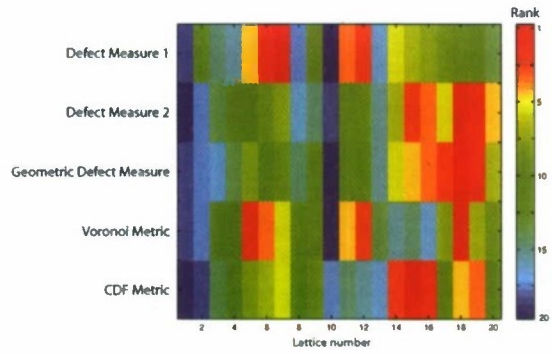


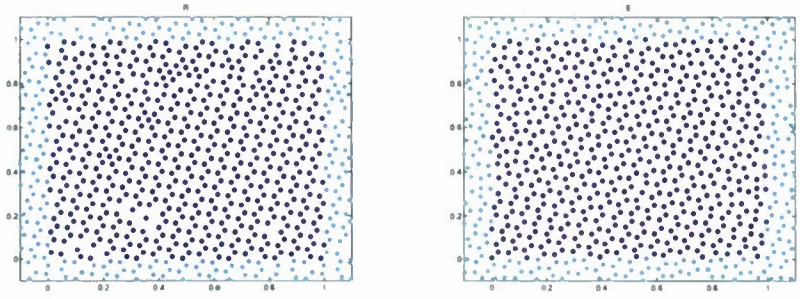
Figure 4.1: Comparion of quality assessment metrics. Each metric along the vertical axis ranked the 20 test lattices (along horizontal axis). The colourbar indicates which colour corresponds to which ranking. Red signifies the best lattice (rank 1) and dark blue the worst lattice (rank 20).

being the worst lattice. For example, along the top row it can be seen that Defect Measure 1 ranked lattice number 7 as the best lattice and lattice number 10 as the worst lattice.

	Time to compute (s)
Defect Measure	0.117
Geometric Defect Measure	6.69
Voronoi Metric	0.338
CDF Metric	0.163

Figure 4.1 shows that Defect Measure 1 ranks the test lattices in an order that is similar to the ranking of the Voronoi Metric (with the notable exception of the identification of the best lattice). Rows 2 and 3 of Figure 4.1 show that Defect Measure 2 assigns similar rankings to the test lattices as the Geometric Defect Measure. This highlights the flexibility of the Defect Measure which results from the freedom to choose the weights for the defects. It will be shown below that the Geometric Defect Measure and the Voronoi Metric consider different aspects of lattices to be important and thus apply to different situations. The CDF Metric identifies the same few lattices as being the worst lattices that the other metrics identify. However, the lattices that are judged to be the best lattices by the CDF Metric are not judged to be that way by the other metrics. The Geometric Defect Measure ranks the CDF Metric's best lattices as being only moderately good.

Lattices number 18 and 5 are shown in Figure 4.2. It can be seen that in lattice 18, although there are defects like missing and extra particles, the particles tend to be aligned with each other. For a majority of particles, the angles between nearest neighbour particles are close to that of a perfect honeycomb lattice. There also seems to be more medium range structure than in lattice 5. This is precisely what



(a) Lattice 18 (b) Lattice 5

Figure 4.2: Test honeycomb lattices with 576 particles and periodic boundary conditions. The dark blue particles are inside the bounding box. The cyan particle positions show the structure at the edge of the bounding box.

the Geometric Defect Measure focuses on. However, in lattice 18, there are obvious defects. In lattice 5 on the other hand, the density of particles is much more uniform across the domain.The particles are not aligned well into a honeycomb lattice but a majority of particles have the correct number of nearest neighbours and form rough rings of 6 particles. It is this local area of each particle aspect of lattices that is the focus of the Voronoi Metric. The shape of the Voronoi cells is not considered, only their area compared to a Voronoi cell of a perfect lattice.

Which of lattices 18 and 5 is judged to be the better lattice depends on the goal. If the purpose of having a honeycomb lattice is to cover an area evenly, with each particle having 3 nearest neighbours and forming rings of 6 particles, then the Voronoi Metric is the one to use. If the goal is to form as much of a close-to-perfect honeycomb lattice as possible then the Geometric Defect Measure should be used. [4] It is interesting to note that the Defect Measure can be used to achieve both of these goals simply by adjusting the weights for the different types of defects.

Both the Voronoi Metric and the Geometric Defect Measure ranked lattice 18 as the best lattice. The Geometric Defect Measure did so because of the regular alignment of the particles. The Voronoi Metric ranked it so highly because of the peculiar coincidence of the error in the area of the Voronoi cells around the gross defects that can be seen in lattice 18, summing to a similar error in area that is spread out across all Voronoi cells. This can be seen by comparing the lattices in Figure 4.2; lattice 18 was ranked as the best and lattice 5 was ranked as the second best. It is because of a few large errors in area summing to a similar total as the sum of many small errors in area, and not because the Voronoi Metric particularly looks at geometry, that lattice 18 has the best ranking. It can be seen from Figure 4.1 that Defect Measure 1 avoids this anomaly.

[4]Lattice 18 was generated using a polynomial potential while lattice 5 was generated using a Rechtsman style potential Rechtsman, Stillinger, and Torquato [2006]

The Voronoi Metric whose results are shown in Figure 4.1 used the alternative expression for the area of a Voronoi cell in the perfect lattice that is explained in Section 3.3. The alternative expression computes the average latttice constant and then sets the perfect Voronoi cell area to be $3\sqrt{3}\,a^2/4$, based on geometry. The assessment of this quality metric thus depends greatly on how accurate the estimate of the lattice constant is. The average lattice constant is calculated using Step 1 of the algorithm for computing the Defect Measure.

The Voronoi Metric can also be implemented as in the algorithm in Section 3.3, with the area of a perfect Voronoi cell being given by (area of bounding box)/(number of particles). The results for this metric are shown in Figure 4.3 under the label Voronoi Metric 2. Lattice 18 is no longer ranked highly, while most of the other rankings remain the same. Using this version of the metric concentrates more on how much of the available space a particle's Voronoi cell covers rather than comparing it to the space a particle in a perfect lattice would cover.

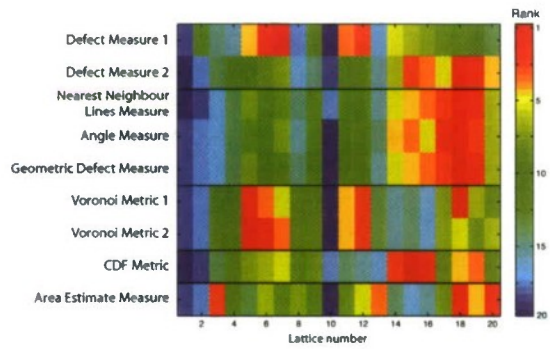


Figure 4.3: Comparison of quality assessment metrics. Each metric along the vertical axis ranked the 20 test lattices (along horizontal axis). The colourbar indicates which colour corresponds to which ranking. Red signifies the best lattice (rank 1) and dark blue the worst lattice (rank 20).

Figure 4.3 also shows the two components of the Geometric Defect Measure: one focusing on whether particles are where they should be along the nearest neighbour lines and one focusing on how close the angles between nearest neighbours are to what they should be in a perfect lattice. The rankings for these two components are very similar.

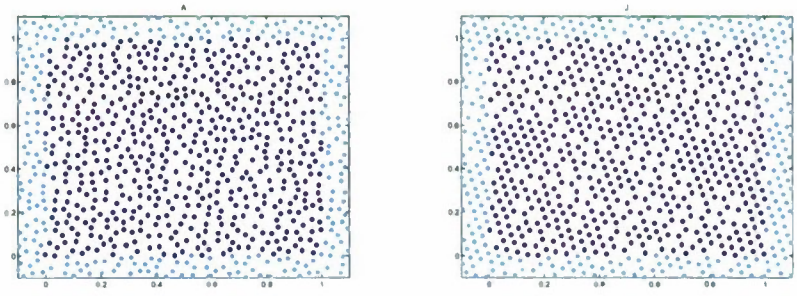
The last row of Figure 4.3, labeled Area Estimate Measure, is a very rough quality assessment metric. It compares the area a perfect lattice would cover if it had the average lattice constant, to the area of the bounding box. That is,

$$\text{Area Estimate Measure} = \left|\text{Area of bounding box} - (3\sqrt{3}\,a^2/4) \times (\text{number of particles})\right|.$$

This quick calculation can identify the best lattice and the worst lattice but does not perform well in between. It's only advantage is that it requires nothing more

than the calculation of the average lattice constant. All of the other metrics, require this computation and then other computations.

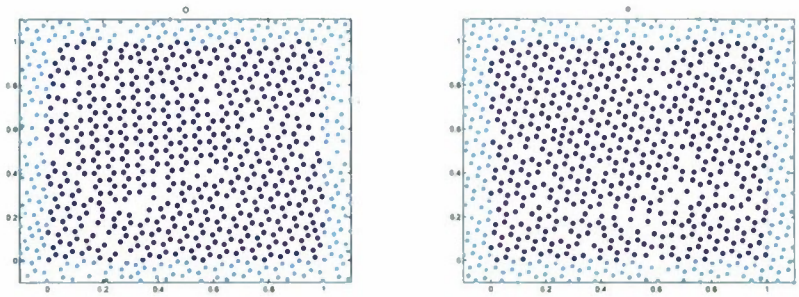
Table 4 shows the time taken to compute the value of each metric. These times were obtained by averaging the time taken to compute the metric for each of the 20 test lattices. Each metric was given the minimum information it needed in order to compute the value of the metric. The Defect Measure calculations were given the distances to the nearest twenty neighbours of each particle. The CDF Metric was given a list of all nearest neighbour distances up to a distance of $2.2a$. The Geometric Defect Measure and the Voronoi Metric were given the particle positions. The rationale for giving each quality metric only the minimum information that it needs stems from the applications of the metrics. If the Defect Measure is used to assess the lattice formed by vehicles flying in formation then these vehicles need only detect their nearest twenty neighbours – this limits the range necessary for their relative distance sensors and the amount of information that must be transmitted. When using the Defect Measure to assess the quality of lattices formed in a LAMMPS simulation, the distance to the nearest twenty neighbours is easily accessible due to the structure of the LAMMPS simulation code. This code calculates particle positions in parallel by dividing up the domain into smaller regions, thus keeping more detailed information about a particle's nearest neighbours. Similar reasoning holds for the CDF Metric.



(a) Lattice 1 (b) Lattice 10

Figure 4.4: The two worst test honeycomb lattices with 576 particles and periodic boundary conditions. The dark blue particles are inside the bounding box. The cyan particle positions show the structure at the edge of the bounding box.

Figure 4.1 shows that all of the metrics, apart from the CDF Metric, found lattices 1 and 10 to be the two worst lattices. These lattices are shown in Figure 4.4. It can be seen that these two worst lattices also exhibit the two different types of lattice that the best lattices in Figure 4.2 did. Namely, one has particles that are well aligned (albeit in the wrong locations) and the other has a more uniform number of particles per area (though not aligned into the honeycomb pattern at all).



(a) Lattice 15 (b) Lattice 16

Figure 4.5: The CDF Metric's two best test honeycomb lattices with 576 particles and periodic boundary conditions. The dark blue particles are inside the bounding box. The cyan particle positions show the structure at the edge of the bounding box.

The CDF Metric assigns rankings that are quite different from all of the other metrics. It judges the worst lattices, similarly to the other metrics but chooses different lattices as the best lattices. These best lattices, shown in Figure 4.5, received a moderately good ranking from the Geometric Defect Measure but were rated as quite bad lattices by the Voronoi Metric. One feature that they exhibit is having regions that are well-formed that are separated from other well-formed regions by regions with many defects. Lattices 15 and 16 have large regions with missing particles and also curves with a small distance between the particles (more evident in Lattice 16). This combination leads to some cancellation, causing the lattice to be ranked highly, as discussed in Section 3.4.

Thus, it seems that the CDF Metric does not appear to assign rankings that are similar to what a human observer would assign, whether particle alignment or density is the focus. The CDF Metric is an indicator of how many inter-particle distances (within a limited range set by d_{max}) are correct. Since the distances are the focus rather than the local density or alignment (properties that are important for forming a lattice structure), the structure is less important with this metric. This renders the CDF Metric less useful as a quality metric for the particular problem of the self-assembly of particles into a target lattice.

5 Applications of the Defect Measure

The Defect Measure is a tool that can be used to quantitatively assess the quality of lattices. It is useful in a variety of situations, such as finding potentials that lead to the self-assembly of particles, assessing the robustness of such potentials, and helping to speed up simulations of lattice formation.

5.1 Design of Potential for Self-Assembly

The Defect Measure can be used as a tool not only to assess the quality of lattices that result from simulations with a particular isotropic potential between particles, but also to refine such potentials to achieve the best potential. For example, the isotropic inter-particle potential necessary for the self-assembly of a honeycomb lattice will have a strongly repulsive component at short inter-particle distances, a local minimum at the lattice constant of the target lattice, a global minimum at $\sqrt{3}$ times the lattice constant, and a tail that goes to zero at long inter-particle distances. A functional form for such a potential can be designed with a number of parameters that may be varied to achieve the potential that yields the best lattices, as measured by the Defect Measure.

Rechtsman and co-workers [Rechtsman, Stillinger, and Torquato [2006]] found the following expression for the self-assembly of particles into a honeycomb lattice:

$$V_{HC} = \frac{5}{r^{12}} - \frac{a_0}{r^{10}} + a_1 \exp[-a_2 r] - 0.4 \exp[-40(r - a_3)^2] \tag{5.1}$$

with $a_0 = 5.89$, $a_1 = 17.9$, $a_2 = 2.49$, and $a_3 = 1.823$ as the best parameter values for good lattices. The form of Equation 5.1 can be used to refine the potential and find the best value of parameter a_2 for self-assembly. The other parameter values and conditions (such as the density and cooling schedule) were kept constant in the 1000 simulations that were run with 65 particles in a domain without periodic boundary conditions. The parameter a_2 values used for the 1000 simulations, formed a Gaussian distribution with mean $a_2 = 1.49$ and standard deviation of 0.6. Figure 5.1 shows the Defect Measure of the final lattice of each simulation versus the value of the a_2 parameter used to generate the lattice. The values of the weights for the Defect Measure are: $\omega_{displaced} = 1.0$, $\omega_{missing} = 1.0$, $\omega_{extra} = 0.8$, $\omega_{boundary} = 0.2$, and $\omega_{lone} = 2.0$. These weights are the same weights as for Defect Measure 1 in Section 4. The lattices corresponding to the red and green points are shown in Figure 5.2. The green point has the lowest Defect Measure of all the lattices. Note that each point in Figure 5.1 is the result of one simulation with random initial velocities for the particles. Ideally, for each value of a_2 a number of simulations would be run and the average Defect Measure of the final lattices used.

Figure 5.1 indicates that for the conditions under which the simulations were run, the best choice for parameter a_2 is a value of 2.6. Note that the honeycomb lattice is quite fragile and particularly sensitive to the density of particles and the boundary conditions. Thus for different densities or boundary conditions, other values of a_2 may be more appropriate. However, the method for designing an isotropic potential has been illustrated by the above example.

As mentioned in Section 4, the weights of the Defect Measure should be chosen according to the properties of the lattice that are most important.

The Geometric Defect Measure can also be used to design the potential for the formation of lattices with the correct alignment of particles. Figure 5.3(a) shows the Geometric Defect Measure versus the parameter a_2 for the same simulations as above. The best lattice, shown in Figure 5.3(b), was formed with $a_2 = 2.23$. Thus which value of a_2 is chosen depends on whether the focus is the correct alignment or

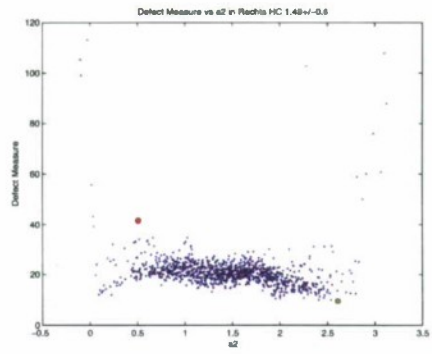
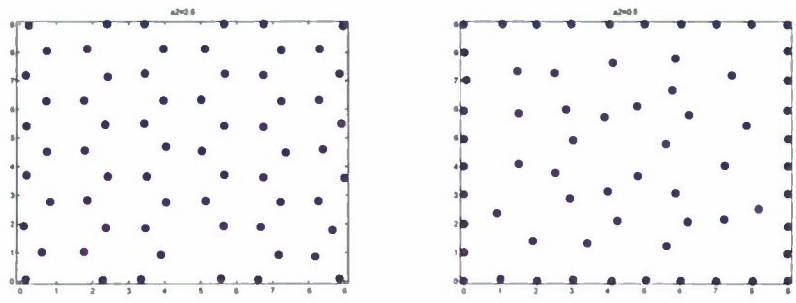


Figure 5.1: Defect Measure versus parameter a_2 in Equation 5.1. The 1000 simulations shown here used 65 particles in a domain without periodic boundary conditions. The a_2 values have a mean of 1.49 and a standard deviation of 0.6. The lattices corresponding to the red and green points are shown in Figure 5.2.



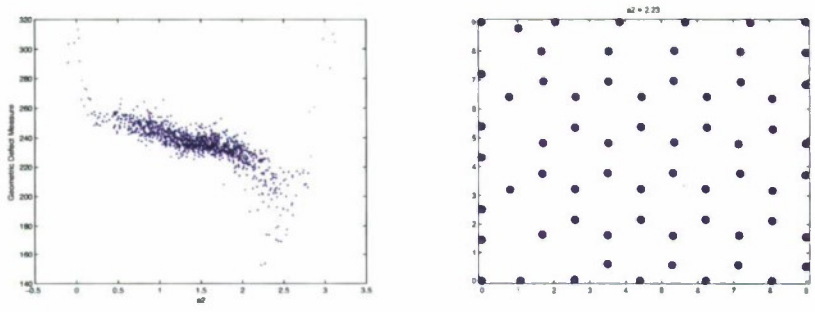
(a) (b)

Figure 5.2: (a) Lattice corresponding to the green point in Figure 5.1, with $a_2 = 2.6$ and Defect Measure= 9.6. (b) Lattice corresponding to the red point in Figure 5.1, with $a_2 = 0.5$ and Defect Measure= 41.6.

local density of particles. Figure 5.4 depicts the two components of the Geometric Defect Measure: the Nearest Neighbour Lines Measure and the Angle Measure.

5.2 Optimization of Potential for Self-Assembly

When employing an optimization procedure to search over parameter space in order to find the best potential for the self-assembly of particles, the Defect Measure can be a useful tool in a number of ways. Firstly, it quantitatively measures the quality of each lattice, thus removing the need for the user to visually assess each



(a) (b)

Figure 5.3: (a) Geometric Defect Measure versus parameter a_2. (b) Lattice corresponding to the lowest Geometric Defect Measure in (a), with $a_2 = 2.23$.

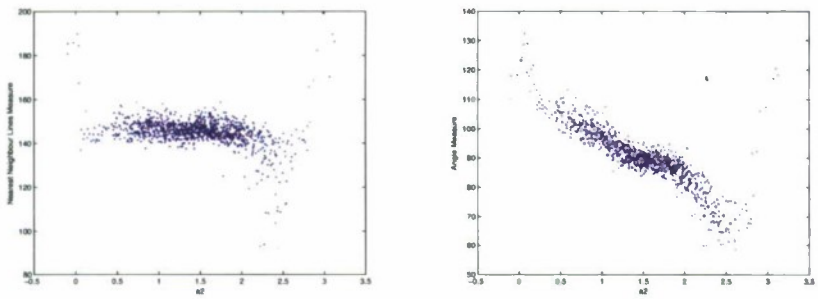
lattice. Secondly, the procedure for computing the Defect Measure, explained in Section 3.1, can identify the type of lattice (if it is reasonably well-formed). This is particularly useful when the target lattice is the honeycomb lattice because the competitor lattice is the very stable triangular lattice. In the parameter space of the honeycomb self-assembly potential, the Defect Measure may have local minima corresponding to triangular lattices. Step 2 of the Defect Measure algorithm can identify such local minima and the optimization procedure can be constructed so as to avoid parameters leading to triangular lattices.

The third use of the Defect Measure in an optimization scheme is indicating when the simulations can be ended. For each set of parameters, a simulation will have to be run to determine whether those parameters lead to a good potential for self-assembly. Rather than having to run each large simulation for a long time, the Defect Measure of the particles can be computed during the simulation and when the Defect Measure levels off, the simulation can be cut short. Figure 5.5 shows the Defect Measure plotted against the time step for a single simulation. From time step 110 until the end of the simulation at time step 150, the Defect Measure stays relatively constant, indicating that the simulation could have been cut off at time step 110. Naturally, the Defect Measure would not need to be computed at every time step, and not at the beginning of the simulation. Shortening the simulation time in this way in an optimization scheme would speed up the procedure.

5.3 Quantifying Robustness of Potentials

A good potential for the self-assembly of particles will be robust to uncertainty in the parameters of the potential, the density of particles and the cooling schedule. The larger the range of values over which the final lattice formed is acceptably good, the more robust the potential is.

There are two ways to define what an acceptably good lattice is. Lattices that



(a) (b)

Figure 5.4: The two components of the Geometric Defect Measure. (a) Nearest Neighbour Lines Measure versus parameter a_2. (b) Angle Measure versus parameter a_2.

have a Defect Measure below a certain value can be accepted as good lattices. The actual value of the Defect Measure chosen for this purpose will depend on the weights assigned to the different types of defects. This is because the value of the Defect Measure for a lattice only has meaning with respect to another value determined with the same weights.

The second way to define the acceptably good lattices is by using Step 2 of the procedure to compute the Defect Measure. If the lattice can be identified by the algorithm to be of the same type of lattice as the target lattice, then it is an acceptably good lattice. Such lattices will have a majority of particles that have the correct number of nearest neighbours, next nearest neighbours and third nearest neighbours. This definition of a good lattice works best for lattices with periodic boundary conditions or lattices with many more inner particles than boundary particles.

Figure 5.6 plots the Defect Measure versus density of particles for two honeycomb potentials: Rechtsman's potential and a piecewise polynomial potential. The weights of Defect Measure 1 were used in the computation. Note that 15 points were omitted from Figure 5.6(a). These points all had a Defect Measure greater than 495 and a density greater than 1. Each point in the plot corresponds to a simulation of 65 particles in a domain without periodic boundary conditions. There were 1000 simulations with densities given by a Gaussian distribution with mean of 0.8 and standard deviation of 0.1.

Defining acceptably good lattices to be those having a Defect Measure less than 30, Rechtsman's potential yields good lattices from densities of 0.6334 to 0.9578. The polynomial potential yields good lattices over a range of densities from 0.6884 to 1.0690. This corresponds to a spread of 0.3244 for Rechtsman's potential and a spread of 0.3806 for the polynomial potential. Thus the polynomial potential can be said to be more robust to uncertainty in the density of particles. It forms good

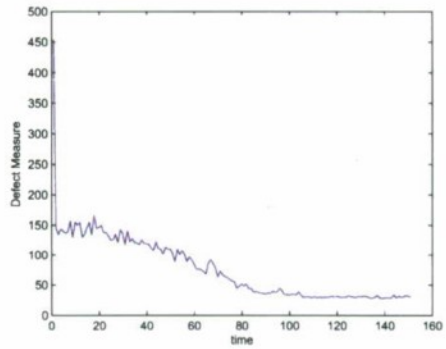


Figure 5.5: Defect Measure versus time for a single simulation.

lattices over a wider range of densities.

The sensitivity of potentials to the parameters in the potential and the cooling schedule can be examined by a similar procedure.

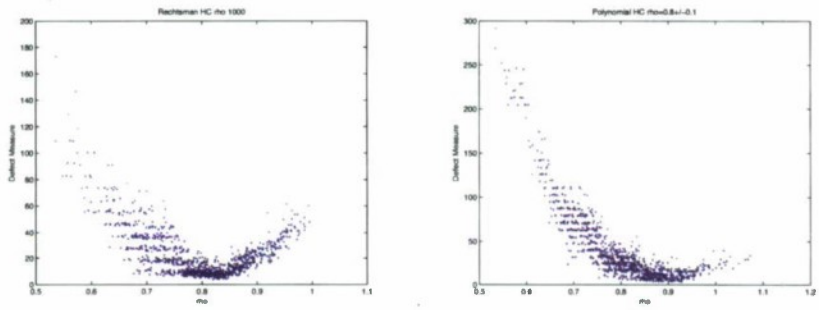
5.4 Identification of Clusters to be treated as Rigid Bodies

The self-assembly of particles with pairwise isotropic inter-particle potentials into a lattice requires long and expensive simulations. The temperature of the particles must be decreased slowly to allow a lattice to form. One way to speed up such simulations is to identify particles that have already formed a lattice structure and treat these particles as a rigid body, thus decreasing the number of evaluations of the potential function. This procedure works best for lattices that form nuclei that grow and join up to form a lattice. The triangular lattice is ideal because of its stability once formed. The polynomial potential discussed above for the honeycomb potential also forms in this way.

A cluster of particles that may be treated as a rigid body can be identified by observing how much the inter-particle distances change over time. If these distances have not changed significantly for an appropriate length of time then the particles are most likely in the lattice configuration and may be treated as a rigid body.

However, there may be some defects in this cluster of particles. A lattice with a defect is typically less stable than a lattice free of defects. Over time, the particles will try to eliminate the defect or move it towards the edge of the otherwise well-formed cluster. This takes more time than the free movement of particles that are not a part of a cluster. So a cluster that has a defect may be identified as a cluster to treat as a rigid body because the inter-particle distances have not changed much.

To allow defects within clusters of particles to be eliminated, the Defect Measure can be calculated for each candidate rigid body particle. Those particles with a high Defect Measure (those close to the defect) would then not be included in the cluster of particles to be treated as a rigid body.



(a) Rechtsman's potential (b) Polynomial potential

Figure 5.6: Defect Measure versus density of particles for (a) Rechtsman's potential and (b) a piecewise polynomial potential. Each point corresponds to one simulation of 65 particles in a domain without periodic boundary conditions. The weights corresponding to Defect Measure 1 were used.

This application of the Defect Measure has not yet been implemented. See Sun Hwan Lee's work.

References

Boje, E. [2000], Finding nonconvex hulls of QFT templates. *J. Dynamic Systems, Measurement and Control* **122**,1, 230.

Mezic, I. and T. Runolfsson [2004], Uncertainty analysis of complex dynamical systems. *Proc. Amer. Control Conference 2004* **3**, 2659.

Rechtsman, M., F. Stillinger, and S. Torquato [2006], Designed interaction potentials via inverse methods for self-assembly. *Phys. Rev. E* **73**, 011406.

D.3 Uncertainty as stabilizer of the head-tail ordered phase in carbon monoxide monolayers on graphite

Uncertainty as stabilizer of the head-tail ordered phase in carbon monoxide monolayers on graphite

Tuhin Sahai
United Technologies Research Center, 411 Silver Lane, MS 129-85, East Hartford, CT 06108

Vladimir A. Fonobcrov and Sophie Loire
Aimdyn, Inc., 1919 State Street, Suite 207, Santa Barbara, CA 93101
(Dated: June 2, 2009)

$(CO)_{1-x}(Ar)_x$ mixtures physisorbed on graphite experimentally display a novel phenomenon of increasing phase transition temperature (stabilizing the system) with increasing Ar impurity concentration or uncertainty [H. Wiechert and K.-D. Kortmann, Surf. Sci. **441** (1999)]. We develop a 2D Ising model that accurately captures the phase transition and its temperature dependence. The anomaly in transition temperature is due to formation of pinwheel regions of CO around Ar atoms. The dilemma of whether the ground state is head-to-head or head-to-tail ordered is reconciled in favor of the latter. The phase transition curve in the presence of uncertainty in Ar impurity is computed using Monte Carlo (MC) and Probabilistic Collocation Method (PCM). PCM computes the first two moments ≈ 2000 times faster than MC.

PACS numbers: 68.35.Rh, 68.55.Ln, 05.50.+q, 02.70.Jn

Uncertainty is an important factor in the design of physical models. Usually significant effort is needed to minimize uncertainty in the output of a model subject to input uncertainty. Phase transitions in statistical thermodynamics of condensed matter are some of the most vivid manifestations of the effect of uncertainty on the state of a system.[1,2] In particular, defects in the form of vacancies, interstitial and quenched impurities give rise to random fields which are known to catalyze phase transitions.[3,4] Disorder and nonequilibrium effects are known to modify structural phase transitions in pure periodic systems.[5] Even small amounts of impurities in monolayers of adsorbed gases can induce significant changes in the phase behavior and phase transitions. Impurities tend to distort the sublattices of the adsorbed phase, and hence lead to a phase transition into an "intermediate" phase significantly prior to that observed for a pure monolayer.[6] Low temperature phase transitions in some systems can be regarded as realizations of two-dimensional (2D) Ising systems.[7]

Recently, Carbon Monoxide (CO) monolayers with Argon (Ar) impurities physisorbed on graphite have been studied experimentally and found to exhibit unique physical properties.[4,8] When adding Ar impurities to head-tail ordered CO monolayers, the order in the system is slowly destroyed and the phase transition is found to be completely suppressed when the impurity concentration reaches $\approx 7\%$.[4] Unlike any other known physical system, the disorder induced in a CO monolayer by Ar impurities results in a higher phase transition temperature, thereby stabilizing the head-tail ordered phase.[4] The phase transition of interest (called head-tail ordering transition) occurs at ≈ 5.18K.[9,10] In this Rapid Communication we develop a model of the CO-Ar system and explain the origin of the observed phenomena.

CO-Ar mixtures physisorbed on graphite can be considered as experimental realizations of 2D Ising models. In the following we design an Ising model that captures the head-tail ordering transition along with the anomalous shift in the transition temperature with increasing Ar concentration. The stabilization of the phase with uncertainty is successfully captured by the Ising model when the experimentally observed pinwheel structure[4] of CO molecules around the Ar sites is correctly modeled (see Fig. 1). We study the phase transition curve as a function of Ar concentration in the presence of uncertainty. The unpredictability in the exact concentration of Ar atoms is another source of uncertainty. The latter uncertainty transforms the phase transition curve into a phase transition region, which can be captured by computing the moments of phase transition region at every nominal concentration of Ar. The moments are obtained using Monte Carlo (MC) and Polynomial Chaos (PCH) techniques. The two methods are compared, and it is found that PCH captures the moments of the uncertain phase transition curve ≈ 2000 times faster than standard MC.

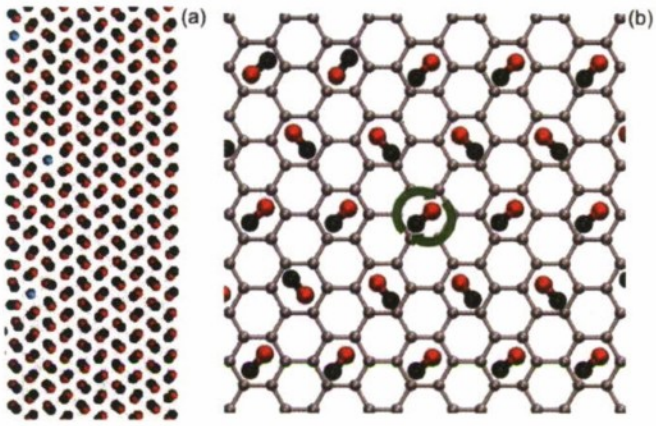


FIG. 1: CO-Ar on graphite. (a) Pinwheel structure formed around the Ar impurities. (b) Flip of CO molecule in Metropolis-Hastings.

We model the potential energy of the system as a sum of pairwise interactions between CO molecules and their six nearest neighbors: $U = \sum_{i=1}^{N} \sum_{j=1}^{6} U_{ij}$, where N is the total number of CO molecules. Pairwise interactions U_{ij} take into account electrostatic interactions of point charges on both molecules, Van der Waals interactions between the two molecules and interactions with the graphite substrate. The center of mass of each molecule is shifted by a distance d_c from the center of the hexagonal lattice cell in which it is located (see Fig. 1).

To simulate the system at different temperatures we employed the Metropolis-Hastings algorithm. A CO site is picked at random and its spin is flipped (i.e. C and O are swapped, see Fig. 1). If the total energy of the modified system decreases, the spin flip is accepted, otherwise the flip is accepted with probability associated with the Boltzmann distribution.[11] For all simulations reported in this Rapid Communication, 2×10^4 flips are performed per CO site. At each temperature, the fluctuation in energy U gives us the heat capacity (C_v).[11] The center of mass offset d_c is a very important parameter of the model. In the absence of impurities, we observe head-to-tail ordering for $d_c = 0.202$, while $d_c = 0.205$ results in head-to-head ordering. Both values of d_c correspond to the experimentally observed position of the heat capacity peak.

All results are obtained for a 10^4-site Ising system. The impurities are chosen randomly on the two-dimensional lattice with the restriction that no CO molecules can have more than one impurity in their six nearest neighbors. The locations of impurities on the lattice are found to make no difference for the transition temperatures. When the center of mass offset d_c lies outside of the interval $[0.195; 0.212]$, the system exhibits "classical" behavior with high phase transition temperatures which decrease with increased impurity concentration as the system becomes less stable. For the head-to-head ordered monolayer with $d_c = 0.205$, the phase transition temperature does not depend on the impurity concentration. Finally, for the head-to-tail ordered monolayer with $d_c = 0.202$, the phase transition temperature shifts to higher temperatures with the inclusion of impurities. In addition to the fact that our model is able to explain the anomalous increase in transition temperature with impurity concentration, it also reconciles the dilemma of whether the ground state is head-to-head or head-to-tail ordered in favor of the latter.[4]

For each concentration of Ar, a heat capacity curve is computed with $d_c = 0.202$ (see Fig. 2). It can be seen in Fig. 2 that the effect of Ar impurities on the C_v curves match experimental observations.[4] Both the suppression of the heat capacity curve along with its anomalous shift to higher transition temperatures[4] are captured. To the best of our knowledge, the Ising model implemented in this paper is the first model that captures the anomalous shift of the phase transition temperature caused by formation of pinwheel structures around Ar impurity sites.

From Fig. 2 the transition temperatures can be plotted against the concentration of Ar (we pick

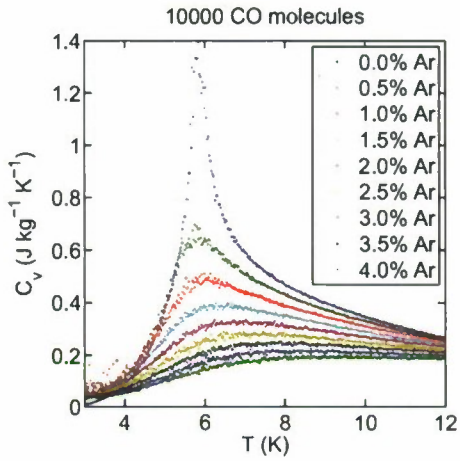


FIG. 2: Heat capacity C_v for different concentrations of Ar impurities.

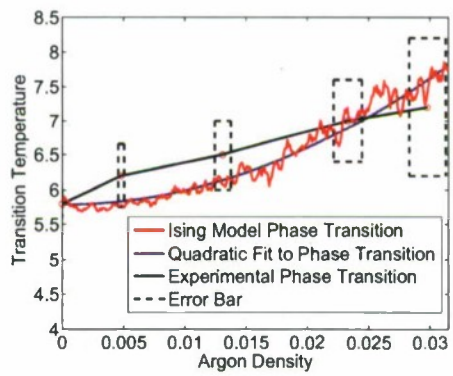


FIG. 3: Phase transition temperature as a function of Ar impurity density extracted from the Ising model (red curves) and its quadratic fit (blue curve). Experimental data (black curve) and error bars are taken from Ref. 4.

the transition temperature at the peak for every curve in Fig. 2). The transition temperatures with increasing Ar density are plotted in Fig. 3, demonstrating the anomalous shift found experimentally.[4] The noisy curve obtained from the Ising model (see Fig. 3) is approximated by a quadratic fit to ease the computation of the phase transition in the presence of uncertainty. Moreover, our theoretical predictions are found to lie within experimental error of Ref. 4 as shown in Fig. 3.

In the experiments, there is always uncertainty in the concentration of impurities. This, in turn, makes the phase transition curve uncertain. Hence, different experimental realizations of CO-Ar mixtures on graphite will give rise to slightly different phase transition curves. In the following we quantify the variability in the phase transition curve in the presence of uncertainty in the concentration of Ar impurities. To compute the variability we use two different approaches. The first approach, the MC technique, is a fairly standard statistical analysis method. This method is, however, very inefficient. A significant speedup in the computation of the uncertainty in the phase transition curves can be obtained by employing PCH techniques.[12,13]

For each value K_0 of Ar concentration in Fig. 3 we assume that the concentration of Ar is

distributed binomially[14] with mean at K_0 and standard deviation of 3% ($\epsilon = 0.03$). The parameters of the binomial distribution N and p can be found from the following system:

$$\begin{aligned} Np &= K_0 \\ Np(1-p) &= (\epsilon K_0)^2. \end{aligned} \tag{1}$$

For each value of mean Ar concentration, we sample the corresponding binomial distribution in Ar concentration using MC techniques and compute the mean and standard deviation of the transition temperature. The results of the MC procedure are shown in Fig. 4 for 10^4 samples for each binomial distribution. As expected, the mean transition curve is the same as the quadratic fit in Fig. 3. The region between the outer curves in Fig. 4 is one standard deviation around the mean transition temperature. As the concentration of Ar atoms increases, the uncertainty in the phase transition temperature also increases. Although MC methods succeed in computing the phase transition curve in the presence of uncertainty, a very large number of samples (10^4) are necessary for a reliable estimate.

PCH techniques are used to quantify output uncertainty by expanding the output random variable of interest in an optimally-chosen orthogonal basis.[13,15] Let us consider the following system:

$$\dot{x} = f(x, \lambda), \tag{2}$$

where x is the system output and λ is a vector of uncertain system parameters with associated probability distribution $w(\lambda)$. In PCH the output random variable is expressed as

$$x(t;\lambda) = a_0(t)\psi_0(\lambda) + a_1(t)\psi_1(\lambda) + \dots \tag{3}$$

Here $\{\psi_i(\lambda) : i \in 1, \dots, \infty\}$ forms an orthogonal basis with respect to $w(\lambda)$.

The coefficients $a_i(t)$ can be determined using Galerkin projections[13] when the equations of the system are explicitly known. In the case of the Ising model, the phase transition temperature is a random variable and f is the Metropolis-Hastings code. Since f is not known explicitly, one can apply Probabilistic Collocation Methods (PCM),[16] where the output random variable is expanded using Eq. 3. However, the system parameters are sampled using zeros of a polynomial orthogonal to the basis used in expansion Eq. 3 (typically if the order of expansion is n, ψ_{n+1} is chosen). A Lagrange interpolating polynomial is passed through the output and the resulting moments of the distribution are computed. The orthogonal polynomials corresponding to the binomial distribution are the Krawtchouk polynomials K_n.[17] The zeros of the polynomials K_n take non-integer values. Since the number of Ar atoms has to be discrete, we use the quadratic fit in Fig. 3 to compute the phase transition temperature at a fractional number of Ar atoms.

The results of using PCM can be seen in Fig. 4. PCM captures the mean transition temperature curve along with the one standard deviation curves exceedingly well with just 4 input samples. The error in the mean and variance at the Ar concentration of 3% can be seen in Figs. 5(a) and 5(b). The results from PCM are obtained with the same magnitude of error as MC 2000 times faster.

In this Rapid Communication we studied the effect of argon impurities on the head-tail ordering phase transition in CO monolayers physisorbed on graphite. We developed an Ising model that captures the head-tail ordering transition in CO-Ar mixtures in agreement with experimental data. The unique physical properties of the CO-Ar system have been explained and attributed to the formation of pinwheel regions of CO around the Ar impurities. To quantify the uncertainty in the number of Ar atoms, we have applied PCM and found it to be far superior to MC in this problem. This approach can be used to quickly bound the variation in phase transition curves when the impurity concentration is not known accurately.

Acknowledgments

The authors thank Dr. H. Wiechert for useful suggestions and discussions. This work was supported in part by DARPA DSO under AFOSR contract FA9550-07-C-0024. This document was

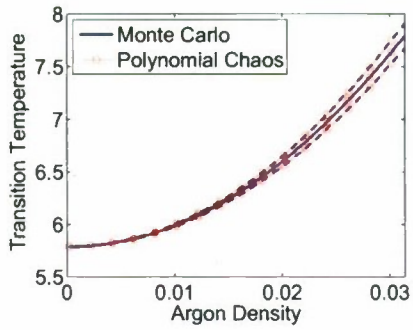


FIG. 4: Comparison of the mean and one standard deviation phase transition curves computed using Monte Carlo and Polynomial Chaos.

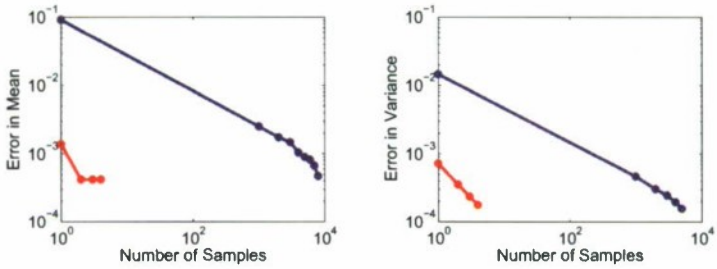


FIG. 5: Comparison of the number of samples needed by Monte Carlo (blue dots) and Polynomial Chaos (red dots) to obtain mean (left) and variance (right) of the phase transition temperature in the CO-Ar system with 3% of Ar impurities.

cleared by DARPA on 4/28/2009 (approved for public release, distribution unlimited).

---

[1] Y. Imry, J. Stat. Phys. **34**, 849 (1984).
[2] R. B. Stinchcombe, *Phase Transitions and Critical Phenomena*, vol. 7 (London: Academic, 1983).
[3] V. Pereyra, P. Nielaba, and K. Binder, J. Phys.-Condens. Mat. **441**, 65 (1999).
[4] H. Wiechert and K.-D. Kortmann, Surf. Sci. **441**, 65 (1999).
[5] A. Sengupta, S. Sengupta, and G. I. Menon, Phys. Rev. B **75**, 180201(R) (2007).
[6] C. Pint and M. Roth, Phys. Rev. B **73**, 115404 (2006).
[7] H. Wiechert, K.-D. Kortmann, and N. Stüßer, Phys. Rev. B **70**, 125410 (2004).
[8] H. You and S. C. F. Jr., J. Stat. Phys. **34**, 849 (1984).
[9] A. Inaba, N. Sakisato, and T. Matsuo, Chem. Phys. Lett. **340**, 400 (2001).
[10] H. Wiechert and S. A. Arlt, Phys. Rev. Lett. **71**, 2090 (1993).
[11] J. M. Yeomans, *Statistical Mechanics of Phase Transitions* (Oxford Science Publications, 1992), 1st ed.
[12] N. Wiener, Am. J. Math. **60**, 897 (1938).
[13] D. Xiu and G. E. Karniadakis, SIAM J. Sci. Comput. **24**, 619 (2002).
[14] W. H. Papoulis, *Probability, Random Variables and Stochastic Processes* (McGraw-Hill, 1984), 2nd ed.
[15] D. Xiu, Commun. Comput. Phys. **5**, 242 (2009).
[16] J. Foo, X. Wan, and G. E. Karniadakis, J. Comp. Phys. **227**, 9572 (2008).
[17] A. F. Nikiforov, V. B. Uvarov, and S. S. Suslov, *Classical Orthogonal Polynomials of a Discrete Variable* (Spring-Verlag, 1992).

Appendix E

Learning algorithms

E.1 Learning macroscopic dynamics for optimal prediction

Learning macroscopic dynamics for optimal prediction*

Sean Meyn† George Mathew‡

June 4, 2009

Abstract

The goal of this paper is to develop modeling techniques for complex systems for the purposes of control, estimation, and inference:

(i) A new class of Hidden Markov Models is introduced, called the finite-rank optimal-prediction (FRO) model. It is similar to the Gaussian mixture model in which the *actual marginal distribution* is used in place of a Gaussian distribution. This structure leads to simple learning algorithms to find an optimal model.

(ii) The FRO model provides a unification of other modeling approaches including the projective methods of Shannon, Mori and Zwanzig, and Chorin, as well as a version of the binning technique for Markov model reduction.

(iii) Several general applications are surveyed, including optimal control, and the dynamical analysis of complex systems via Markov spectral theory. Computation of the spectrum, or solutions to dynamic programming equations are possible through a finite dimensional matrix calculation without knowledge of the underlying marginal distribution on which the model is based.

(iv) A detailed application to molecular dynamics is presented: Spectral theory for the low dimensional Markov model is applied to predict phase transitions for helium adsorbed on a graphite substrate.

Keywords: Markov process, Computational methods in Markov processes, Mori-Zwanzig theory, Spectral theory

AMS subject classifications: Primary: 37A30 60J20, 70K70
Secondary: 37A60, 82B31, 60J22

*This paper is based upon work supported by DARPA DSO (Carey Schwartz PM) through AFOSR contract FA9550-07-C0024 (Fariba Fahroo, PM), *Robust Uncertainty Management*. S.M. is supported in part by the National Science Foundation ECS-0523620.

†Department of Electrical and Computer Engineering and the Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A. (meyn@uiuc.edu)

‡Systems Department, Dynamics and Optimization Group, United Technologies Research Center, 411 Silver Lane, East Hartford, CT 06108, U.S.A.(mathewga@utrc.utc.com)

Contents

1 Introduction

Complex systems can be found throughout engineering, and the social, physical or life sciences, and every academic community that must confront complexity has developed specialized tools for modeling complex systems. In this paper we survey some of these methods, and provide a unifying framework for the purposes of estimation and control.

Markov models are adopted as a basic model since they can capture a range of *non-stationary* behavior. They are a valuable modeling technique even for models that are *not Markovian.* This observation is initially due to Claude Shannon.

Shannon introduced the idea of low dimensional Markov models to replicate features of English language. This appears as the motivation for the notion of entropy in his famous 1948 paper *A mathematical theory of communication*, which is regarded as the birth of modern information theory [26]. These early ideas led to modern techniques in source coding (e.g., the Lempel-Ziv algorithm used in every computer for compression [18]).

The main idea is simply described as follows. Let $\boldsymbol{Z}$ denote a discrete-time stochastic process for which we seek a Markovian description, evolving on a subset Z of Euclidean space. For the purposes of modeling it is assumed that the process is stationary. Shannon's (first-order) Markov model is obtained using Baye's rule to construct a one-step transition kernel,

$$T(z, A) = \frac{\mathsf{P}\{Z(t) \in dz,\ Z(t+1) \in A\}}{\mathsf{P}\{Z(t) \in dz\}}, z \in \mathsf{Z},\ A \subset \mathcal{B}(\mathsf{Z}). \tag{1}$$

where the ratio denotes the Radon-Nikodym derivative. That is, T describes the average dynamics of $\boldsymbol{Z}$ in steady-state. The Markov model with transition kernel T also captures a large part of the steady-state behavior: It follows from the definitions that the marginal distribution of $\boldsymbol{Z}$ is invariant for T: Letting μ denote the marginal, we have, for any set $A \subset \mathcal{B}(\mathsf{Z})$,

$$\mu(A) = \int \mu(dz) T(z, A) \tag{2}$$

Or, in operator-theoretic notation, $\mu T = \mu$.

Shannon's model was developed independently for continuous-time processes in the physics research community: It is a component of the famous Mori-Zwanzig projection method [22, 27]. The model (1) also coincides with the Markov model developed by Chorin (see e.g. [8, 7]). In this work the Markov model is said to exhibit *optimal prediction* due to the solidarity of the steady-state marginals captured in the invariance equation (2).

While providing powerful and general approaches to model reduction, Shannon's approach can still result in a highly complex model. In particular, if Z is not finite, then the Markov model with transition kernel T can be regarded as an infinite-dimensional nonlinear dynamical system. To avoid this complexity we seek a finite dimensional setting. The approach taken in this paper is to enforce the optimal prediction property on a finite-dimensional space of functions. There are many ways to enforce this constraint — Motivation for the class of models adopted in this paper comes from recent spectral theory of Markov processes.

Various theories from Markov processes suggest a model whose transition kernel has *finite rank*: For a collection of functions $\{h_i\}$ and measures $\{\mu_i\}$ the kernel can be expressed as the finite sum,

$$T = \sum_{i=1}^{m} h_i \otimes \mu_i \tag{3}$$

so that $T(z, A) = \sum h_i(z)\mu_i(A)$ for any z and A. For example, if $\boldsymbol{Z}$ is a finite state space Markov model whose transition matrix has distinct eigenvalues, then (3) holds with $\{h_i\}$ and $\{\mu_i\}$ right and left eigenvectors. For general, infinite dimensional models it is still possible to obtain a reduced order model based on the eigenvectors corresponding to eigenvalues near unity. Based on the sign structure of the eigenvectors, it is possible to approximate a complex Markov model by a much simpler hidden Markov model (HMM) [14]. This construction is rooted in the asymptotic theory of Freidlin and Wentzell [12, 6], the theory of quasi-stationarity for Markov chains [11], and the approximate modeling approaches based on spectral theory developed in [13, 9] — see also recent approaches in [15, 19].

Finite rank Markov models with transition kernel of the form (3) are used to approximate complex Markov processes in [3, 17] for the purposes of verifying the existence of a spectral gap, and for establishing limit theory such as large deviations.

Other common approaches to model reduction also lead to a finite rank Markov model of this form. If $\boldsymbol{Y}$ is the output process of an HMM with finite state process $\boldsymbol{I}$, then the joint

process $(\boldsymbol{I}, \boldsymbol{Y})$ is Markov with a finite-rank transition kernel, even if the observations evolve in a complex space.

The use of binning to create a finite-dimensional Markov model, as in [13, 9], results in a finite state-space model, or a simple refinement (given in (8)) leads to a finite-rank model of the form (3).

The approach advocated in this paper combines all of the points of view surveyed above, based on the *finite-rank optimal-prediction* (FRO) model, introduced here for the first time. Let $\{r_i, s_i : 1 \le i \le m\}$ denote measurable functions on Z, each in $L_2(\mu)$, and for a collection of constants $\{\Theta_{ij}\}$ denote,

$$s_\Theta(z_0, z_1) = \sum_{i,j=1}^{m} \Theta_{ij} s_i(z_0) m_j(z_1), \qquad z_0, z_1 \in \mathsf{Z}. \tag{4}$$

The FRO model is Markovian, whose transition kernel is finite rank with density s_Θ,

$$T_\Theta(z_0, A) := \int_{z_1 \in A} s_\Theta(z_0, z_1) \mu(dz_1). \tag{5}$$

The subscript in the density is used to stress that this is to be learned. *The task of learning is greatly simplified by choosing to avoid estimating the entire dynamics.*

The FRO model has unique features, not found in generic HMM models, that are developed in this paper. To summarize,

(i) The structure (3.2) in which μ is the (unknown) marginal distribution of the vector $Z(t)$ leads to a simple characterization of the best s_Θ through L_2 or information-theoretic methods. Complexity of computation of the best parameter is greatly reduced when compared to, say, the EM method [18].

(ii) In particular, this class of models can be chosen to capture the *optimal prediction* property: For a given collection of functions $\{\phi_i\}$ in $L_2(\mu)$ we can choose $\Theta = \Theta^*$ to guarantee,

$$\mathsf{E}_{\Theta^*}[\phi_i(\widehat{Z}(t))\phi_j(\widehat{Z}(t+1))] = \mathsf{E}[\phi_i(Z(t))\phi_j(Z(t+1))], \qquad i, j = 1, \ldots, m, \tag{6}$$

where the left hand side denotes the expectation with respect to the model obtained using Θ^*, and the right hand side is the actual expectation in steady state. This is a generalization of the optimal prediction property described by Chorin for the Markov models of Mori and Zwanzig [8, 7, 22, 27]. In fact, these models were introduced by Shannon for the first time in [26] — Some history is contained in Section 2.

(iii) The FRO model can be adapted to obtain a model based on binning, as in [13, 9], while again preserving the optimal prediction property (6).

(iv) Knowledge of the full marginal μ is unnecessary in all applications of interest in this paper. In particular,

(a) Only finite-dimensional statistics (means and covariances) are required to obtain the optimal parameter in (3.2). These statistics are easily estimated through Monte-Carlo techniques.

(b) It is shown in Proposition 3.3 that computation of the spectra of the kernel T is also possible through knowledge of finite-dimensional statistics. An application of this result is used in Section 5, which contains a detailed treatment of the prediction of phase transitions in molecular models.

(c) Similarly, the solution to dynamic programming equations that arise in optimization as well as simulation can be computed based on easily estimated finite-dimensional statistics.

We believe that the FRO model is ideal for applications to optimization, distributed control, simulation variance reduction, as well as prediction of phase transitions.

The remainder of the paper is organized as follows. Contained in the next section is a review of Shannon's approach to Markov modeling, and results demonstrating how more recent methods fall into Shannon's framework. Section 3 contains a development of the finite-rank Markov model. Several different techniques are introduced to obtain a model exhibiting the optimal prediction property on a finite-dimensional function class. Section 4 contains aa Monte-Carlo methods to compute the parameters in a finite-rank Markov model and provide a statistical analysis of the estimates.

Section 5 contains an in-depth application of the spectral-theoretic techniques to the phase-transition prediction problem of molecular dynamics.

Conclusions and extensions are summarized in Section 7.

Notational conventions An upper case symbol such as X denotes a random variable, and lower case x deterministic. Bold italic $\boldsymbol{X}$ indicates a stochastic process.

To indicate relative dimension we let # denote a very large integer. Hence, if we say that "$\boldsymbol{x}$ evolves on $\mathbb{R}^{\#}$" this signifies that $\boldsymbol{x}$ is a deterministic process evolving on a state space so large that simulation is extremely difficult on an ordinary computer.

We let $\boldsymbol{z}$ or $\boldsymbol{Z}$ denote a process on a simpler state space. A Markov model is constructed to approximate the behavior of $\boldsymbol{Z}$. Its state process is denoted $\widehat{\boldsymbol{Z}}$, each evolving on the state space Z.

2 Shannon's Markovian projection

In the treatment of deterministic dynamical systems, model reduction is typically addressed through singular-value decomposition to eliminate 'fast variables'. Hence the choice of coarse variables in a reduced-complexity description is specified by relative dynamics. In the probabilistic framework considered here we have much greater freedom: *Regardless of what variables are chosen to build a Markov model, the resulting nonlinear system is stable* (see Proposition 2.1 that follows). Moreover, the model replicates exactly whatever statistics are input as constraints in the construction of the transition kernel.

The following result is based on Shannon's construction (1). It is a component of the model reduction techniques pioneered by Mori and Zwanzig in the area of statistical mechanics [22, 27]. The conclusion of the proposition is described by Chorin as *optimal prediction* since the model T captures exact marginal statistics of $\boldsymbol{Z}$.

Proposition 2.1. *Suppose that the Radon-Nikodym derivative* (1) *exists for each z and A to define a transition kernel on $\mathsf{Z} \times \mathcal{B}(\mathsf{Z})$. That is, $T(\,\cdot\,, A)$ is a measurable function on $\mathcal{B}(\mathsf{Z})$ for*

each $A \in \mathcal{B}(\mathsf{Z})$, and $T(z, \cdot)$ is a probability measure on $\mathcal{B}(\mathsf{Z})$ for each $z \in \mathsf{Z}$. Then the Markov chain with this transition kernel describes these aspects of the stationary process $\boldsymbol{Z}$:

(i) One-step dynamics: $T(z, A) = \mathsf{P}\{Z(t+1) \in A \mid Z(t) = z\}$, $z \in \mathsf{Z}$, $A \in \mathcal{B}(\mathsf{Z})$.

(ii) Steady-state: *The probability μ is invariant for T,*

$$\mu(A) = \int_{z \in \mathsf{Z}} \mu(z) T(z, A), \qquad A \in \mathcal{B}(\mathsf{Z}).$$

□

This result is a trivial consequence of the definitions, yet its implications are surprisingly rich. A roadblock to its application is that the transition kernel T is not known. Moreover, in general it remains an infinite dimensional object, in which case learning the entire transition kernel is not feasible. In Section 3 we turn to kernels of finite rank to approximate the optimal prediction model. First we consider some special cases in which the form of the approximate model is relatively transparent.

2.1 Model construction using binning

Let $\{\mathsf{X}_1, \dots, \mathsf{X}_n\}$ denote a partition of the state space for the complex, stationary process $\boldsymbol{X}$: These sets are assumed disjoint, with $\bigcup \mathsf{X}_i = \mathbb{R}^{\#}$. This partition is used to define a coarse variable that indicates the particular partition that $X(t)$ occupies at each time t.

Let $\boldsymbol{Z}$ denote the integer-valued process taking values in the set $\{1, 2, ..., n\}$ defined by $\{\boldsymbol{Z}(t) = i \text{ iff } X(t) \in \mathsf{X}_i, 1 \le i \le m\}$. Shannon's construction has a simple interpretation in this case because $\boldsymbol{Z}$ takes on only m possible values.

Proposition 2.2. *With $\boldsymbol{Z}$ defined using a state space partition, the transition matrix (1) can be expressed for $z_0 = i$ and $z_1 = j$ by the ratio of the probabilities,*

$$T(z_0, z_1) = \frac{\mathsf{P}\{X(0) \in \mathsf{X}_i,\ X(1) \in \mathsf{X}_j\}}{\mathsf{P}\{X(0) \in \mathsf{X}_i\}}. \tag{7}$$

That is, $T(z_0, z_1) = \big(\pi\{\mathsf{X}_i\}\big)^{-1} \int_{x \in \mathsf{X}_i} \pi(dx) P(x, \mathsf{X}_j)$, where π is the marginal distribution for $\boldsymbol{X}$. □

If desired, a finite-rank Markov model on the original state space can be obtained using the transition kernel,

$$T'(x_0, dx_1) := T(z_0, z_1) \frac{\pi(dx_1)}{\pi\{\mathsf{X}_j\}}, \qquad x_0 \in \mathsf{X}_i,\ x_1 \in \mathsf{X}_j. \tag{8}$$

The definition (7) is used in [13, 9] to construct a finite state space Markov chain. In these papers the distribution of $X(0)$ is arbitrary, so that the invariant measure for T will not be consistent with $\boldsymbol{X}$.

2.2 Markov spectral theory

The binning approach can be justified based on the results from spectral theory surveyed above: If the sets $\{\mathsf{X}_i\}$ are selected so that the transition times are approximately geometrically distributed, then an accurate Markov chain approximation can be obtained. Results in [14, 24] show that the geometric distribution approximation holds for a sampled diffusion, provided the sets $\{\mathsf{X}_i\}$ are sublevel sets of the associated eigenfunctions.

We next consider a more general setting: Suppose that $\boldsymbol{Z}$ is an m-dimensional vector-valued function of a Markov chain $\boldsymbol{X}$ evolving on $\mathbb{R}^{\#}$. There is a function $\psi\colon \mathbb{R}^{\#} \to \mathbb{R}^m$ such that $Z(t) = \psi(X(t))$ for each t. If the spectra of $\boldsymbol{X}$ and $\boldsymbol{Z}$ coincide, then a decomposition of the state space $\mathsf{Z} = \mathbb{R}^m$ appears justifiable based on the spectrum of T. Under special conditions Shannon's model does capture a part of the overall spectrum.

Proposition 2.3. *Suppose that h is an eigenfunction for $\boldsymbol{X}$ with real or complex eigenvalue λ, and suppose moreover that it can be expressed as a function of ψ: For some function $g\colon \mathbb{R}^m \to \mathbb{C}$,*

$$h(x) = g(\psi(x)), \qquad x \in \mathbb{R}^{\#}.$$

Then g is also an eigenfunction for T with eigenvalue λ.

Proof. For any function $f\colon \mathbb{R}^m \to \mathbb{C}$ we have from the definitions and the Markov property,

$$\mathsf{E}\big[f(Z(t))g(Z(t+1))\big] = \mathsf{E}\big[f(Z(t))h(X(t+1))\big] = \mathsf{E}\big[f(Z(t))Ph\,(X(t))\big].$$

Based on the eigenvector equation we obtain, $\mathsf{E}\big[f(Z(t))g(Z(t+1))\big] = \mathsf{E}\big[f(Z(t))\big(\lambda g(Z(t))\big)\big]$. The optimal prediction property implies that the same identity holds with $\boldsymbol{Z}$ replaced by $\widehat{\boldsymbol{Z}}$,

$$\mathsf{E}\big[f(\widehat{Z}(t))g(\widehat{Z}(t+1))\big] = \mathsf{E}\big[f(\widehat{Z}(t))\big(\lambda g(\widehat{Z}(t))\big)\big].$$

From the definition of conditional expectation we conclude that $\mathsf{E}[g(\widehat{Z}(t+1)) \mid \widehat{Z}(t)] = \lambda g(\widehat{Z}(t))$, or $Tg = \lambda g$. □

3 Finite-rank optimal-prediction models

In this section we introduce several approaches to model construction for the FRO model with transition kernel (3.2). Stationarity of $\boldsymbol{Z}$ is assumed throughout, even though the ultimate goal is to construct a Markovian approximate model whose realizations are not necessarily stationary.

Some general properties of finite rank models are summarized in Section 3.1.

The main result of Section 3.2 shows how to construct $\Theta^\star$ satisfying the optimal prediction property (6). The remainder of this section develops optimization techniques for the construction of a model. In Section 3.3 a parameter is found that minimizes an L_2 error criterion. It is shown in Proposition 3.4 that an optimizer satisfies the optional prediction property (6) with $\{\phi_i\} \equiv \{r_i\}$, provided the functions $\{r_i\}$ and $\{m_j\}$ coincide. An alternative maximum likelihood criterion is described in Section 3.4, and similar conclusions are obtained.

The following conventions are adopted throughout this section. For any $\mathbb{R}^m$-valued functions f, g whose components lie in $L_2(\mu)$ we denote the auto-correlation functions,

$$R_{ij}^{f,g}(k) = \mathsf{E}[f_i(Z(0))g_j(Z(k))], \qquad i,j = 1,\dots,m,\ k \in \mathbb{Z} \tag{9}$$

or in matrix form, $R^{f,g}(k) = \mathsf{E}[f(Z(0))g^{\mathsf{T}}(Z(k))]$. When $f = g$ we write $R^f(k)$ instead of $R^{f,f}(k)$.

Recall that μ denotes the marginal distribution of $\boldsymbol{Z}$. We let μ^2 denote the stationary bivariate distribution describing the joint statistics of $(Z(t), Z(t+1))$, μ_Θ the stationary distribution for T_Θ, and μ^2_Θ the stationary bivariate distribution under the transition law T_Θ. That is,

$$\mu^2_\Theta(dz_0, dz_1) := \mu_\Theta(dz_0)T_\Theta(z_0, dz_1) \tag{10}$$

The bivariate distribution possesses a density with respect to the product distribution $\mu \times \mu$, denoted p^2_Θ, which can be expressed as a finite sum of the form,

$$p^2_\Theta(z_0, z_1) = \sum_{i,j=1}^{m} \Theta_{ij} r_i(z_0) m_j(z_1). \tag{11}$$

To obtain a tractable framework for approximation, *we assume throughout the remainder of the paper that the functions* $\{r_i, m_j\}$ *are given.* The functions $\{s_i\}$ in (4) are obtained from the functions $\{r_i, m_i\}$ via,

$$s_i = \Big(\sum_{i',j'=1}^{m} \Theta_{i'j'} r_{i'} \mu(m_{j'})\Big)^{-1} r_i. \tag{12}$$

We begin with some comments on the general finite rank model.

3.1 Properties of finite-rank Markov models

Consider a general transition kernel of the form,

$$T(z, A) = \sum_{i=1}^{m} s_i(z)\mu_i(A), \qquad z \in \mathsf{Z},\ A \in \mathcal{B}(\mathsf{Z}), \tag{13}$$

where $\{s_i\}$ are non-negative valued functions, and $\{\mu_i\}$ are probability measures. Let $\widehat{\boldsymbol{Z}}$ denote the Markov chain with this transition kernel,

$$\mathsf{P}\{\widehat{Z}(t+1) \in A \mid \widehat{Z}(t) = z\} = T(z, A).$$

Some properties of T and $\widehat{\boldsymbol{Z}}$ are summarized in the following:

Proposition 3.1. *The Markov chain* $\widehat{\boldsymbol{Z}}$ *with finite-rank transition kernel* (13) *has the following properties:*

(i) $\widehat{\boldsymbol{Z}}$ *is, of course, a Markov chain on the state space* Z.

(ii) *The* m*-dimensional stochastic process* $W(t) = (s_1(\widehat{Z}(t)), \ldots, s_m(\widehat{Z}(t)))^{\mathsf{T}}$, $t \geq 0$, *is a Markov chain on* $\mathbb{R}^m$.

(iii) $\widehat{\boldsymbol{Z}}$ *is also a Hidden Markov Model: There is a finite state space Markov chain* $\boldsymbol{I}$ *on the finite set* $\{1, \ldots, m\}$, *an i.i.d. process* $\boldsymbol{N}$ *on* $\mathbb{R}$, *and a function* $\varphi\colon \{1, \ldots, m\} \times \mathbb{R} \to \{1, \ldots, m\}$ *such that,*

$$\widehat{Z}(t+1) = \varphi(I(t), N(t+1)), \qquad t \geq 0.$$

What are the eigenvalues of T? If (h, λ) solve the eigenfunction equation $Th = \lambda h$ with λ a possibly complex scalar, and $h\colon \mathsf{Z} \to \mathbb{C}$, it follows that h can be written as a linear combination of the functions $\{s_i\}$: For some complex scalars $\{\varrho_i\}$ we have,

$$h(z) = \sum \varrho_i s_i(z), \qquad z \in \mathsf{Z}. \tag{14}$$

The form of T given in (13) and the eigenfunction equation then give,

$$\lambda \sum \varrho_i s_i(z) = \sum_{j,k=1}^{m} s_j(z) \mu_j(s_k) \varrho_k$$

Linear independence of the functions $\{s_i\}$ implies that the coefficients coincide. Consequently, the coefficients $\{\varrho_k\}$ and the eigenvalue λ are obtained as the solution to the finite matrix eigenvalue problem,

$$M\varrho = \lambda\varrho,$$

with $M_{jk} := \mu_j(s_k)$.

The constraints on $\{\varrho_k\}$ and λ in the FRO model are expressed,

$$\lambda \varrho_i = \sum_{j,k} \Theta^*_{ij} M_{jk} \varrho_k, \qquad 1 \le i \le m,$$

where $M_{jk} = \mu(r_j s_k) = R^{rs}_{jk}(0)$. In conclusion, we obtain

Proposition 3.2. *The eigenvalues of T_{Θ^*} in the FRO model correspond with those of the $m \times m$ matrix $\Theta^* R^{rs}(0)$. If $\lambda \in \mathbb{C}$ is an eigenvalue, and ϱ an eigenvector,*

$$\Theta^* R^{rs} \varrho = \lambda \varrho$$

then the function (14) is an eigenfunction for T. □

We next consider a general approach to optimal prediction.

3.2 Optimal prediction on a subspace

Suppose that $\{\phi_i\}$ are a collection of functions in $L_2(\mu)$,

$$\mu(\phi_i^2) := \mathsf{E}[\phi_i^2(Z(t))] < \infty, \qquad 1 \le i \le m.$$

Our goal is to choose $\Theta = \Theta^*$ to guarantee the optimal prediction property (6). We first express the left hand side as follows: For each $i, j = 1, \dots, m$,

$$\begin{aligned} \mathsf{E}_{\Theta^*}[\phi_i(\widehat{Z}(t)\phi_j(\widehat{Z}(t+1)] &= \int \phi_i(z_0)\phi_j(z_1) p_\Theta(z_0, z_1)\mu(dz_0)\mu(dz_1) \\ &= \sum_{k,\ell=1}^{m} \Theta_{k\ell} \mu(\phi_i r_k)\mu(\phi_j m_\ell) \end{aligned} \tag{15}$$

Based on the notation (9), it follows that the equation (6) has the equivalent matrix formulation,

$$R^{\phi,r}(0)\Theta^* R^{m,\phi}(0) = R^{\phi}(1). \tag{16}$$

We thereby arrive at a formula for the optimal parameter.

Proposition 3.3. *Suppose that the covariance matrices $R^{\phi,r}(0)$ and $R^{m,\phi}(0)$ are invertible. Then the unique value of Θ^* satisfying (6) is defined by the matrix product,*

$$\Theta^* = [R^{\phi,r}(0)]^{-1} R^{\phi}(1)[R^{m,\phi}(0)]^{-1}. \tag{17}$$

□

3.3 Optimal prediction and L_2 projection

We now show that the optimal prediction property holds for an optimal model obtained under a natural optimization criterion.

Recall that μ^2_Θ possesses a density p^2_Θ with respect to the product distribution $\mu \times \mu$. Assume that μ^2 also possesses a density, denoted p^2. The L_2 mismatched criterion considered here is defined for any Θ by,

$$\mathcal{E}(\Theta) = \tfrac{1}{2} \int (p_\Theta(z_0, z_1) - p(z_0, z_1))^2 \mu(dz_0)\mu(dz_1) \tag{18}$$

Proposition 3.4 asserts that the L_2 optimal model exhibits optimal prediction on the finite-dimensional subspace of functions spanned by the basis. If $\{r_i\} = \{m_j\}$ are indicator functions of a partition of Z, then (6) coincides with the constraints defining the transition matrix described in Proposition 2.2.

Proposition 3.4. *Suppose that $r_i = m_i$ for each i, that $r_1 = m_1 \equiv 1$, and that these m functions are linearly independent in $L_2(\mu)$. That is, the covariance matrix $R^r(0)$ is full rank. Then, the vector Θ^* minimizes $\mathcal{E}$ if and only if the optimal-prediction constraints (6) hold with $\{\phi_i\} = \{r_i\}$. The unique solution is expressed uniquely,*

$$\Theta^* = \{\Theta^*_{ij}\} = [R^r(0)]^{-1} R^r(1)[R^r(0)]^{-1} \tag{19}$$

Proof. First consider the general setting in which the $\{r_i\}$ and $\{m_i\}$ may differ. On setting the derivative of $\mathcal{E}$ with respect to Θ_{i_0,j_0} equal to zero for each i_0 and j_0 we find that an optimal parameter is characterized by the optimal-prediction constraints,

$$\begin{aligned} \mathsf{E}[r_i(Z(t)m_j(Z(t+1)] &= \int r_i(z_0)m_j(z_1)p(z_0,z_1)\mu(dz_0)\mu(dz_1) \\ &= \int r_i(z_0)m_j(z_1)p_{\Theta^*}(z_0,z_1)\mu(dz_0)\mu(dz_1) \qquad i,j = 1,\dots,m \end{aligned} \tag{20}$$

Under the assumption that r_1 and m_1 are identically equal to unity, the constraints (20) imply that,

$$1 = \mathsf{E}[r_1(Z(t)m_1(Z(t+1)] = \int p_{\Theta^*}(z_0,z_1)\mu(dz_0)\mu(dz_1)$$

so that $\mu^2_{\Theta^*}$ has total mass one.

To complete the proof it is necessary to demonstrate that the bivariate distribution $\mu^2_{\Theta^*}$ has equal marginals $\mu_{\Theta^*1} = \mu_{\Theta^*2}$, with

$$\mu_{\Theta^*1}(\,\cdot\,) = \mu^2_{\Theta^*}(\,\cdot\,,\mathsf{Z}), \qquad \mu_{\Theta^*2}(\,\cdot\,) = \mu^2_{\Theta^*}(\mathsf{Z},\,\cdot\,)$$