



AFRL-RI-RS-TR-2010-128

**SIMULATION CONCEPT – HOW TO EXPLOIT TOOLS FOR COMPUTING
HYBRIDS**

June 2010

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE**

■ AIR FORCE MATERIEL COMMAND

■ UNITED STATES AIR FORCE

■ ROME, NY 13441

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2010-128 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/
MICHAEL J. HAYDUK, Chief
Emerging Computing Technology Branch

/s/
EDWARD J. JONES, Deputy Chief
Advanced Computing Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) JUNE 2010		2. REPORT TYPE Final		3. DATES COVERED (From - To) January 2007 – December 2009	
4. TITLE AND SUBTITLE SIMULATION CONCEPT – HOW TO EXPLOIT TOOLS FOR COMPUTING HYBRIDS			5a. CONTRACT NUMBER In House		
			5b. GRANT NUMBER N/A		
			5c. PROGRAM ELEMENT NUMBER 62702F		
6. AUTHOR(S) Clare D. Thiem, Morgan A. Bishop, Joseph M. Hertline, Andrew Gorczyca, Andrew Flack, Sean A. Cain, and Maximillion T. McMahon			5d. PROJECT NUMBER 459T		
			5e. TASK NUMBER SC		
			5f. WORK UNIT NUMBER HE		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFRL/RITC 525 Brooks Road Rome, NY 13441-4505				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RITC 525 Brooks Road Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) N/A	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2010-128	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release; Distribution Unlimited. PA# 88ABW-2010-3401 Date Cleared: 21-June-2010					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The Simulation Concept – How to Exploit Tools for Computing Hybrids (SCHETCH) project explored the design modeling and simulation (M&S) process for developing advanced computing technology for future information systems. The desire is to integrate new alternative computing concepts with existing silicon-based computing technology in hybrid computing architectures. This final report provides an overview of what was accomplished over duration of the project. During this effort several alternative computing concepts were examined. A modeling and simulation process for the design and development of hardware implementation of alternative computing concepts was defined. Technology gaps were also identified. Commercially available software was found to be applicable in the examination of nanotechnology and components of quantum computing concepts. This success is being transitioned to support other in-house efforts pursuing quantum computing. Initial concepts for computer based design tools to exploit structural DNA nanotechnology for future systems were also developed.					
15. SUBJECT TERMS Design, Modeling, Simulation, Computing, Finite Element Analysis, Parametric Modeling, Microelectromechanical Systems, MEMS, Nanoelectromechanical Systems, NEMS, Quantum Computing, Nanotechnology					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 120	19a. NAME OF RESPONSIBLE PERSON Clare D. Thiem
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) N/A

TABLE OF CONTENTS

1.0 INTRODUCTION	1
2.0 BACKGROUND	2
3.0 NANOTECHNOLOGY	6
3.1 General Thoughts	6
3.2 Carbon Nanotubes	7
3.2.1 SWCNT Background	8
3.2.2 Modeling Assumptions	9
3.2.3 SWCNT Modeling in SolidWorks	11
3.2.4 SWCNT Modeling in COMSOL Multiphysics 3.3	12
3.2.5 SWCNT Concluding Remarks	18
4.0 BIOTECHNOLOGY	19
4.1 Membrane Computing	20
4.2 DNA Self-assembly	20
4.3 Structural DNA Nanotechnology	21
4.3.1 Structural DNA Nanotechnology Background	21
4.3.2 Optimizing DNA Selection	22
4.3.3 Stochastic Simulation of DNA Hairpin Kinetics	41
4.3.4 Modeling Strand Hybridization	67
4.3.5 Structural DNA Nanotechnology Concluding Remarks	75
4.4 Biotechnology Concluding Remarks	77
5.0 QUANTUM SCIENCES	78
5.1 Nanomechanical Resonators	78
5.1.1 Initial Assumptions and Constraints	78
5.1.2 Mechanical Model	79
5.1.3 CAD Drawing and Analysis	82
5.1.4 Finite Element Modeling and Analysis	83
5.1.5 Nanomechanical Resonator Concluding Remarks	85
5.2 Ion Trap	87
5.2.1 Ion Trap Model Generation	87
5.2.2 Ion Trap Analysis	88
5.2.3 Scripting and Customization	94
5.2.4 Circuit Models	97
5.2.5 Ion Trap Concluding Remarks	97
6.0 CLOSING REMARKS	99
7.0 REFERENCES	100
LIST OF ACRONYMS	106
APPENDIX	108

LIST OF FIGURES

Figure 1: Graphical representation of basic question behind this project.....	3
Figure 2: MEMS M&S design process.....	4
Figure 3: M&S process for hybrid computing architectures	5
Figure 4: Continuum cylinder.....	10
Figure 5: Initial SWCNT heat transfer model.....	11
Figure 6: SolidWorks model of a stack.....	12
Figure 7: Cross sectional view of SWCNT model.....	14
Figure 8: Initial SWCNT model fully meshed.....	14
Figure 9: Thermal contours for initial SWCNT model.....	15
Figure 10: Illustration of the mesh variation for the different subdomains	16
Figure 11: Thermal contours for stacked model	17
Figure 12: Close up of interconnect interface with larger layers.....	17
Figure 13: Calculation of free energy for WC complementary sequences.....	24
Figure 14: Folding of single stranded sequences (a) into hairpin secondary structure (b).....	25
Figure 15: Folding of single stranded sequences (a) into double hairpin secondary structure (b).....	26
Figure 16: HCR sequence species.....	27
Figure 17: Linear growth HCR process	28
Figure 18: Exponential growth HCR process	29
Figure 19: Free energy distributions for various DNA sequence lengths.....	31
Figure 20: Internal cross-hybridization matrix	32
Figure 21: Unique barcode test matrix.....	33
Figure 22: HCR Probe Finder Stage 1 output.....	34
Figure 23: Elimination vector matrix for a 6 probe set.....	35
Figure 24: Tree structure for maximizing target identification	36
Figure 25: Algorithm test cases	37
Figure 26: Algorithm effectiveness for finding maximum possible subset.....	39
Figure 27: Stage 2 double hairpin output.....	40
Figure 28: Binding site thermo lookup table output	40
Figure 29: Consecutive biomolecular reactions.....	42
Figure 30: Overview of MATLAB Implementation.....	48
Figure 31: Assumed binding site configurations	49
Figure 32: Combinatorial enthalpy and entropy array	50
Figure 33: Distribution of free energies and probabilities for hairpin:hairpin combinations	51
Figure 34: Length of time step vs. simulation time	54
Figure 35: Two example trees formed.....	55
Figure 36: Free binding sites as a function of reaction time.....	57
Figure 37: Semi-log plot	58
Figure 38: log-log plot	58
Figure 39: Tree size vs. target concentration	60
Figure 40: Average tree size vs. time.....	60

Figure 41: Testing by Dirks and Pierce [19], 2004 Copyright by The National Academy of Sciences of the USA.....	61
Figure 42: Baseline results.....	62
Figure 43: Delta hairpin with large number of target strands.....	63
Figure 44: Delta hairpin with small number of target strands.....	63
Figure 45: No delay in addition of target DNA strands to the system.....	64
Figure 46: Delayed addition of target DNA to the system.....	65
Figure 47: Experimental hairpin set designed to work poorly.....	66
Figure 48: Close-up showing detail of depleting hairpins.....	67
Figure 49: A progressive two-dimensional manual build of dendritic tree formed from the hairpins depicted in the upper left hand corner of the figure.....	68
Figure 50: Adenine graphed using MATLAB (left) and OpenGL (right).....	70
Figure 51: An overhead view of a thymine and adenine base-pair in OpenGL.....	71
Figure 52: Illustration of the Sugar-Phosphate backbone in OpenGL.....	73
Figure 53: Hairpin Loop.....	74
Figure 54: Illustration of basic beam.....	80
Figure 55: Coordinate reference system for beam.....	80
Figure 56: Initial SolidWorks model of nanomechanical beam.....	82
Figure 57: First mode response for beam.....	84
Figure 58: Snapshot of subdomain material definition for beam geometry.....	85
Figure 59: Third mode response for beam.....	86
Figure 60: CAD model of individual linear ion trap.....	88
Figure 61: COMSOL resonance frequency of electrode.....	89
Figure 62: Mesh case study results.....	90
Figure 63: Mesh structure for resonance frequency.....	91
Figure 64: Solution for resonance frequency.....	91
Figure 65: Simplification of ion trap geometry for solution.....	92
Figure 66: View of potential field in YZ plane.....	93
Figure 67: View of solution in XZ plane.....	93
Figure 68: Response frequency solution from MATLAB.....	95
Figure 69: Dependence of solution on mesh refinement with MATLAB.....	96
Figure 70: Model comparison of solution to ideal case for hyperbolic potential.....	97

LIST OF TABLES

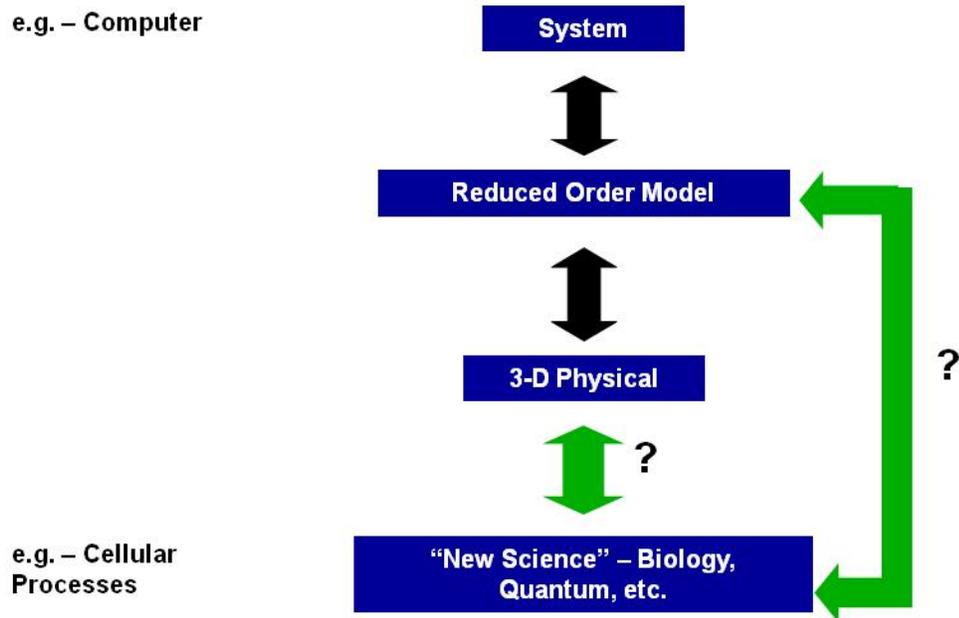
Table 1: SWCNTs compared to copper vias.....	8
Table 2: Thermodynamic properties of WC NN base pair sequences [27].....	24
Table 3: Bond lengths found during research.....	72

1.0 INTRODUCTION

The Simulation Concept – How to Exploit Tools for Computing Hybrids (SCHETCH) project is exploring the design modeling and simulation (M&S) process for developing advanced computing technology for future intelligent systems. It represents a necessary step in the broader Air Force Research Laboratory's (AFRL's) research and development (R&D) activity which seeks to provide new warfighting capabilities to the Air Force. A main premise behind this project is that, for an alternative computing concept to move from laboratory novelty to a technology ready for the field, the proper M&S process must be in place. This requires some understanding as to whether or not existing M&S tools and techniques can be applied to the new technology or if new M&S paradigms need to be developed. Adaptation and integration of commercially-available software provides an opportunity to take advantage of existing functionality without investing time into developing new tools for new concepts. Thus, if the tools and techniques are well defined for modeling a technology then the technology's maturity level is at a point where the broader community can begin to explore ways to exploit it. The number of alternative computing concepts, however, is too large for this project to address all of them. Therefore, it was decided to focus on "hardware" concepts and not software implementations. A decision was also made to initially look at only three concepts: nanomechanical quantum computing, membrane computing, and deoxyribonucleic acid (DNA) computing. This report provides an overview of progress that has been made since the project started. The design M&S process was explored utilizing a combination of literature reviews, discussions, and modeling and simulation trade studies leveraging past experience whenever possible. During this examination some effort was made to identify limitations of existing M&S technology. Naturally, as the project progressed, and the directorate's and division organizational focus evolved, adjustments were made. These adjustments will be explained at the appropriate places throughout the report. This report starts by providing additional background information for the project then moves on to discuss M&S issues related to nanotechnology, biotechnology and approaches to quantum computing. Along the way technology gaps will be identified before wrapping up with some closing remarks.

2.0 BACKGROUND

One research thrust of the Advanced Computing Architectures Core Technical Competency (CTC) Area when this effort began was to examine novel information processing paradigms. [1] Over the past several years members of the Advanced Computing Division have supported several Defense Advanced Research Projects Agency (DARPA) Programs that fall under the broad category of biotechnology and quantum sciences. These programs were designed to demonstrate the integration of bio-fluidics with electronics (Bio-Fluidic Chips - BioFlips), expand the capability of multiphysics design tools (Simulation of Biological Systems - SIMBIOSYS), provide an open source environment for biological simulation tools (Bio-Computation - BioCOMP), and examine quantum computing architectures and algorithms (Quantum Information Science and Technology - QuIST). The results of these programs will play a role in demonstrating how biotechnology and quantum sciences can provide new capabilities for information dominance. Before the capability of a biologically-based or quantum-based information system concept can be demonstrated, however, more research is required to develop the technology to the level of maturity commensurate with practical application. The previous project, Establishing Tools for Computing Hybrids (ETCH), examined roles for which biotechnology could be leveraged for information dominance. ETCH determined that computational biology tools do have a role in maturing the technology, and showed that tools in the Biological Simulation Program for Intra-Cellular Evaluation (Bio-SPICE) software environment could be used in the development of bio-molecular computing concepts. [2] ETCH did not examine the quantum aspects of hybrid architectures. The basic question, as illustrated in Figure 1, is how do you bring new science into existing M&S process? Will existing M&S paradigms work, or new ones required? The baseline for multi-physics, multi-scale M&S is the process that was developed under the DARPA Design for Mixed Technology Integration Program, shown in Figure 2, which was used by the AFRL in-house team during its baseline effort.



Modeling and Simulation Levels for Architecture Development.

Figure 1: Graphical representation of basic question behind this project

Figure 3 illustrates the process that will be pursued for this project utilizing specific commercially available tools. The geometry is introduced in a computer-aided design (CAD) environment. CAD software for three-dimensional model generation allows for the establishment of geometry and physical characteristics of the desired structure. SolidWorks is a parametric CAD package, meaning it allows for manipulation of features such as dimensions or constraints to change the relationships in a single part, or in an assembly of components. This is useful in the design process where ideas are often refined and physical characteristics and geometry relationships change. A model can easily be updated by changing constraints and dimensions without starting from the very beginning of the model generation process. Similarly, visualization in this step of the design process also allows for clear communication of ideas to the user and their audience. SolidWorks allows for material assignment to components in assemblies. This is often useful in electronics because of differing materials such as copper and silicon. Finally, SolidWorks offers a variety of file formats for saving a model that is created, allowing for versatility and compatibility with a number of different programs for viewing, importing, manipulation and analysis of the model generated. For this particular effort three-dimensional physical model geometry from SolidWorks can be imported into COMSOL Multiphysics for analysis.

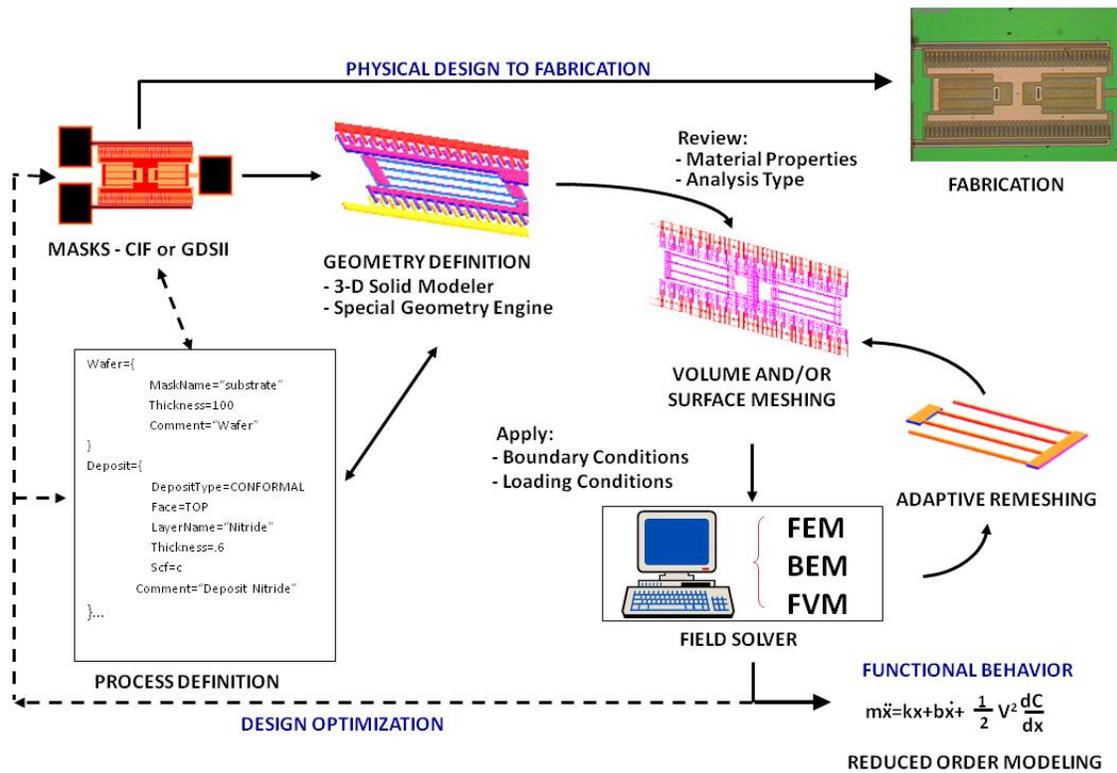


Figure 2: MEMS M&S design process

COMSOL can be applied to a variety of problems such as heat transfer, stress/strain analysis and modal analysis that can be solved using partial differential equation-based methods like Finite Element Analysis (FEA). FEA is a numerical technique used to find an approximate solution to a given partial differential equation (PDE) with pertinent boundary conditions and parameters. Geometry is reduced into a discrete mesh of similar “elements”, and properties and conditions are assigned likewise. Elements may be two-dimensional or three-dimensional. The process is computationally intensive as an increase in the number of elements for a given geometry increases the amount of calculations that are necessary to perform. Therefore, it is critical that care is to be taken to define the proper element mesh density to reduce computational resource consumption and solution time while obtaining reasonable results. Trade studies using FEA allow for the examination of output characteristics and modification of boundary conditions to meet desired performance goals. Solver adaptability is important in this step, as new technologies often require new governing equations and essentially new science. Some finite element solvers offer a fixed set of modes for analysis, i.e. structural, heat, etc., and lack the ability to adapt these solvers for new technologies. COMSOL software was created to address multiphysics problems so it is more adaptable to solving a variety of problems than some of its competitors. COMSOL’s interface with MATLAB allows for customization of governing equations and model refinement with minimal manual manipulation.

Systems integration allows for the implementation of the physical device into an overall architecture for operation. Definition of the control circuits relevant to the individual device can be performed in the proposed process. Simulation Program with Integrated Circuit Emphasis (SPICE) software allows for circuit element definition and is compatible with select FEA programs. Iteration of analysis in both environments allows for refinement to achieve desired circuit characteristics and device performance. COMSOL has an integrated SPICE solver and provides the unique capability to integrate a finite element model with a SPICE model. Systems-modeling is shown as the final step in the M&S process shown in Figure 3 with the initial desire to utilize Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) software if possible. The ability to use VHDL would facilitate the integration of new alternative computing concepts with existing silicon-based computer technologies to create hybrid computing architectures. At this point very little effort has been spent on examining the ability to model alternative computing concepts in VHDL.

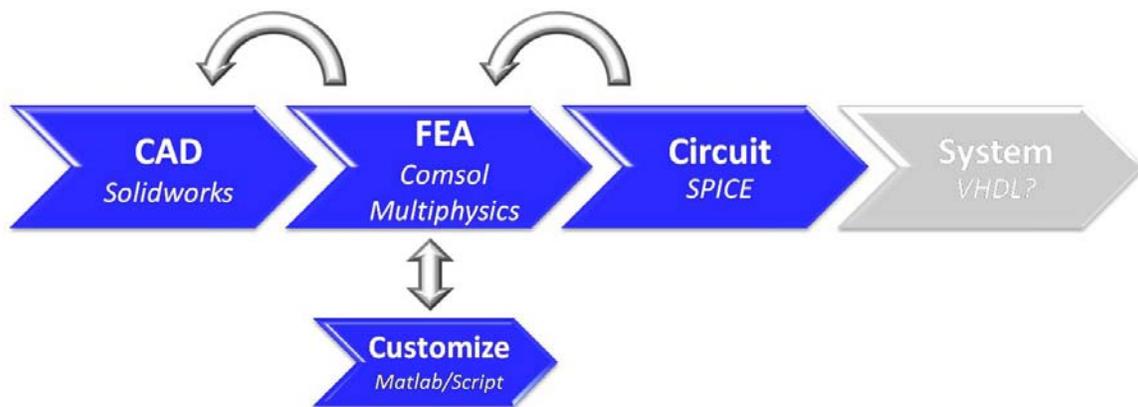


Figure 3: M&S process for hybrid computing architectures

3.0 NANOTECHNOLOGY

Numerous discussions and debates have occurred regarding the role of nanotechnology and how to define nanocomputing. Part of the problem is the hype associated with nanotechnology and the pursuit of new funding. People like to be part of the latest “hot” technology interest. This raises the question of whether or not the promise of the capability of nanotechnology is being oversold in order to get additional funds for existing R&D activity? Technology evolution is taking Complementary Metal Oxide on Silicon (CMOS)-based computing technology into the nanoscales. The fact that computer technology is being fabricated on a 45 nm line does not necessarily make it nanocomputing. If there is no new computing principle being exploited, then it is just an issue of the size scale for classical computing technology getting smaller. Similarly, while the much smaller size may subject devices to quantum effects this also does not mean that one is doing quantum computing. All of this is based on the perspective that we are looking at alternative computing concepts for novel information processing paradigms that will provide revolutionary changes not evolutionary progression. By their very nature when biotechnology such as DNA computing and quantum information science such as quantum computing are finally realized they will most likely be implemented with nanotechnology, and thus could be considered examples of nanocomputing. Thus, for this project there was no major effort made to carve out a nanotechnology focus separate from biotechnology or quantum information sciences except for a quick look at carbon nanotubes.

3.1 General Thoughts

At the beginning of this effort some thought was given to as to whether or not existing M&S concepts are applicable to nanotechnology. There is a history of success at the Rome Research Site using a variety of M&S software to examine microelectronics and microelectromechanical systems (MEMS). Can this experience be brought to bear on nanotechnologies and other alternative computing concepts? The short answer is yes, but only after some effort has been made to understand the proper application of this experience.

In past work with finite element analysis, for instance, it was found for the particular software that was being used at the time, that if one kept the aspect ratio of the longest side to the shortest side of an element equal to or less than 5 to 1 for a static analysis one could get very good results. In the case of conduction heat transfer it was found that one could use an aspect ratio of 10 to 1 with well-formed elements. This understanding of the capabilities of this particular finite element software came as a result of years of experience and validation by several engineers. The pursuit of new technology to exploit for hybrid information systems requires the researchers to address multiphysics problems for which the rules of thumb for modeling and simulation are not broadly known. Different commercial vendors implement finite element analysis with their own biases

and preferences necessitating the need to conduct trade studies to explore peculiarities of the software and understand how to utilize it properly. Some of these products may only address one type of problem well, but not others. Thus, one part of this project's effort was spent to look at how well commercially available software tools could be applied to the development of the technology for alternative computing concepts. For this project, SolidWorks for three-dimensional model construction and COMSOL Multiphysics for FEA were used.

During this quest for knowledge and understanding it was discovered that there are some "new" books available to help one begin to understand the issues related to the multiphysics modeling and simulation of MEMS and nanoelectromechanical systems (NEMS). [3, 4] It was also found that guidelines for the proper application of the Finite Element Method (FEM) for thermal and structural analysis are also available to the broader community. [5] This knowledge was previously limited to practitioners with years of experience behind them.

3.2 Carbon Nanotubes

Developers of advanced computing architectures constantly seek performance improvements in current devices while exploring the capabilities of novel concepts and the potential of new materials to provide superior properties, such as thermal transport, over existing material choices. As the sizes of devices continue to shrink, the density of these devices on a chip increases and in some case so does the heat generated. The correct thermal management scheme must then be employed to maintain the desired performance and reliability of systems. One of the benefits of having the proper M&S software tools is the ability to examine the potential of emerging materials and technologies, such as the Single Walled Carbon Nanotube (SWCNT), prior to any major investment in them. SWCNTs were examined to explore their potential for thermal management in future systems through the use of three-dimensional physical modeling.

3.2.1 SWCNT Background

SWCNTs are a focal point for researchers because of their high strength and exceptional transport properties. The SWCNT was “recently” discovered as a single sheet of graphite rolled into a tube. [6] The diameter of this tube is on the order of one nanometer (1 nm), while the aspect ratio of length to diameter can be greater than one thousand (1000x). [7] The orientation, also known as the chirality, of hexagonal units of the graphene determines whether the SWCNT is a metal or a semiconductor. [7,8] A Young’s Modulus (E) on the order of 1 TPa has been experimentally demonstrated [7,9], supporting claims for its extreme stiffness and strength. As for the energy transport properties, a maximum current density of $\sim 10^{10}$ A/cm² at a temperature of 250 °C [8] and a thermal conductivity demonstrated as high as 6600 W/m K [10] show promise for electronic applications. Table 1 below shows that SWCNTs appear to have superior thermal properties for heat transfer when compared to copper vias. [8] SWCNT supporters are quick to point out that, as copper interconnect technology shrinks down with the evolution of Very Large Scale Integration (VLSI) technology, issues with electromigration and excessive delay associated will be a concern. The resistivity of the copper interconnects increases as device size approaches the nanometer scale. Higher current densities could also lead to reliability issues when the capabilities of copper are exceeded. SWCNTs may provide an alternative to current technology to address the increase in heat generation and current density, eliminating the existence of electromigration and excessive delay.

Table 1: SWCNTs compared to copper vias

	SWCNT	Copper Via at the 22 nm node (expected year 2012)
Thermal conductivity (W/m K)	6600	400
Max Current Density (A/cm²)	$\sim 1 \times 10^{10}$	$\sim 1 \times 10^7$

There are a few issues with carbon nanotubes that must be resolved in order to take advantage of their potential benefits. Initial studies have shown that it is difficult to establish a solid contact to a single SWCNT, and therefore, the contact resistance is of significant concern. A proposed solution for this is to produce a bundled array of SWCNT as an interconnect. [8] This reduces contact resistance with a larger contact area while still exhibiting the material properties of interest. Another issue is that it is difficult to control the chirality of the SWCNTs in the bundle. [8] This means that there is uncertainty in the amount of conducting and semi-conducting SWCNTs which corresponds to unknown contributions to electrical properties. Other obstacles to carbon

nanotubes come to light when the technology is looked at from the modeling and simulation perspective. Material properties, specifically the thermal conductivity [10], lack a concrete definition with experimental values from different research groups ranging from several hundred to 6600 W/m K. Though most work has shown that it clearly is an excellent conductor of heat, uncertainty in the magnitude of the thermal conductivity yields difficulties in accurately quantifying this value when comparing theoretical models to experimental data. Finally, it is not yet clear if the SWCNTs may be modeled simply as a scaled continuum device, or if molecular mechanics need to be taken into account. [11] The Boltzmann Transport Equation has been used to model some devices on the nanoscale [11], however, it takes considerable knowledge to apply this concept. For this effort's brief look at SWCNTs, a simplified model will be used to obtain a general understanding of the characteristic response of SWCNTs with approximations and adjustments to classical theory.

3.2.2 Modeling Assumptions

To understand the application of carbon nanotubes for thermal management, a heat transfer model can be used. In general, there are three basic types of heat transfer: conduction, convection and radiation. Equations 1, 2, 3, and 4 below outline the three heat transfer modes and the corresponding contribution to overall heat transfer.

$$q_{cond} = -k \frac{dT}{dx} \quad k = \text{thermal conductivity} \quad (1)$$

$$q_{conv} = h(T_{surf} - T_{\infty}) \quad h = \text{convection heat transfer coefficient} \quad (2)$$

$$q_{rad} = \varepsilon \sigma (T_{surf}^4 - T_{surround}^4) \quad \varepsilon = \text{Emissivity} \quad (3)$$

$\sigma = \text{Stefan- Boltzmann Constant}$
($5.67 \times 10^{-8} \text{ W/m}^2 \text{ K}^4$)

$$q_{total} = q_{cond} + q_{conv} + q_{rad} \quad (4)$$

In order to simplify the modeling of the heat transfer potential of SWCNTs, assumptions were made. First, one potential application of SWCNTs for heat transfer would be for electronics operating in a vacuum so convection was neglected. The contribution of radiation to the overall heat transfer was also considered very small and therefore could be ignored, leaving a heat transfer problem involving conduction. Another approximation for the modeling was to represent the carbon nanotube as a continuum cylinder with wall thickness of 0.34 nm. [12] Figure 4 below shows a diagram of the continuum cylinder with the attached reference frame coordinate axis.

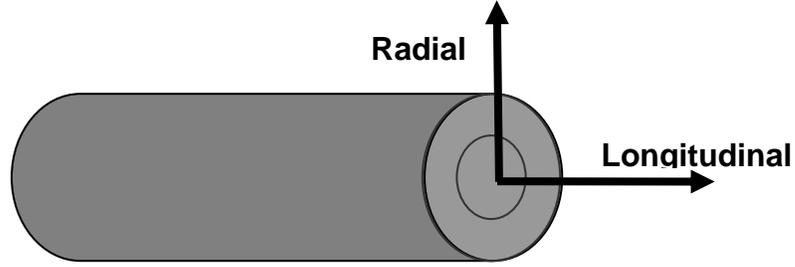


Figure 4: Continuum cylinder

Considering the significant aspect ratio of SWCNTs, the thermal conductivity has been shown to be anisotropic, meaning that the thermal conductivity in the longitudinal direction is considerably larger than the thermal conductivity in the radial direction. [12] An approximation within 5% for the anisotropic thermal conductivities of nanoscale objects has been documented in a fundamental heat transfer book. [13] This can be applied to carbon nanotubes, especially since the diameter is much smaller than the length. The approximation is based on the mean free path, or average distance traveled by an electron before it collides with either an imperfection in the material or with a phonon. This data is available from experiments, and all that is required for calculation is the bulk thermal conductivity for the material, or the material property of a macro-sized sample. Equations 5 and 6 below show the thermal conductivity approximations for the longitudinal and radial heat flow, respectively. All material properties were assumed to be taken at 300 K due to the lack of availability of experimental data outside of the room temperature range. Early results have also demonstrated a temperature-dependence [10] for the thermal conductivity of carbon nanotubes as well as a length-dependence, and with simplicity in mind, this can be ignored by keeping a constant cylinder length at a temperature of 300 K.

$$k_{longitudinal} = k_{bulk} \left[1 - \frac{\lambda}{3L} \right] \quad (5)$$

λ =electron/phonon mean free path

$$k_{radial} = k_{bulk} \left[1 - \frac{2\lambda}{3\pi L} \right] \quad (6)$$

The proposed heat transfer model adjusts the thermal conductivity to an anisotropic property in heat conduction for a SWCNT cylinder approximation. The following equation, Equation 7, highlights the model broken up into the radial and longitudinal components.

$$q_{Total} = - \left[k_{radial} \frac{dT}{drad} + k_{longitudinal} \frac{dT}{dlong} \right] \quad (7)$$

3.2.3 SWCNT Modeling in SolidWorks

There were two heat transfer modeling simulations used for SWCNT study. The first deals with a SWCNT connecting a silicon heat source and heat sink. This model was intended to demonstrate the difference in heat transfer capabilities for silicon and SWCNTs while highlighting the potential of a SWCNT to transfer heat efficiently from a source to sink. A 20 nm-long SWCNT with a diameter of 1 nm is connected to square blocks of silicon of dimensions 3 nm x 3 nm x 3 nm. The geometry for this model, created in SolidWorks, can be found in Figure 5 below. One issue of significance to modeling nanodevices, with the particular version of SolidWorks installed at the time, was encountered. SolidWorks had a seemingly arbitrary setting that would not allow any dimension to be entered that was less than 100 nm. This made it difficult to incorporate nanodevices with characteristic dimensions on the order of 1 nm. It was discovered, however, that a scaling procedure used in COMSOL during the importation of the model could compensate for this limitation and will be discussed later. The broader concern about this limitation is that other analytical software programs will not be able to compensate for this anomaly.

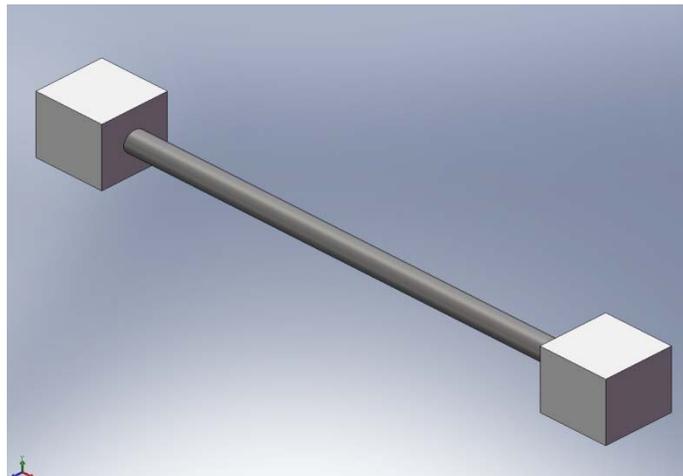


Figure 5: Initial SWCNT heat transfer model

The next model was meant to represent the three-dimensional stacking of chips in an VLSI electronics packaging concept in order to investigate the ability of SWCNT bundled interconnects to dissipate heat between layers. The model includes four 100 μm x 100 μm x 1 μm silicon layers connected by bundles of SWCNTs with a plate of copper 50 μm x 50 μm x 1 μm on the top layer as a “source”. The bundle “interconnects” are approximated as rectangular with a cross sectional area equal to that of a cylinder with a diameter of 1 μm and have a length of 5 μm . This approximation was done to make the importation of the SolidWorks model into COMSOL easier since the interconnects were considerably smaller than the other components in the assembly. This will be discussed in more detail in the COMSOL modeling portion to follow. Figure 6 below shows the three-dimensional stacking model as created in SolidWorks.

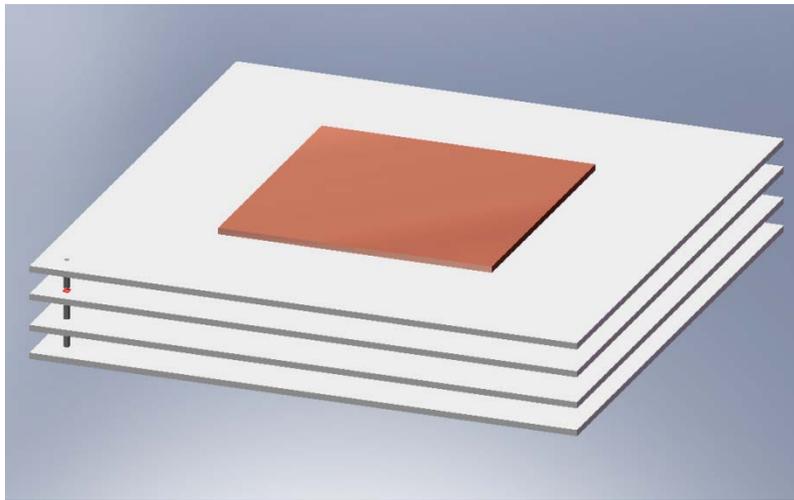


Figure 6: SolidWorks model of a stack

3.2.4 SWCNT Modeling in COMSOL Multiphysics 3.3

Two particular models are analyzed, as mentioned in the previous section. To import these models into COMSOL, special considerations were made due to issues with an arbitrary boundary of 100 nm in SolidWorks. The assemblies were created in SolidWorks at a scale of 1 nm = 1 m for the first model and 1 μm = 1 m for the second model, and this allows for a simple scaling of 10^{-9} and 10^{-6} , respectively, in COMSOL by use of the “Scaling” feature in the drawing application. This scaling was done in all three directions and this brings the component down to the characteristic size of interest. As stated previously, other analytical programs may not have this capability and therefore this feature must be taken into consideration during the geometry development in SolidWorks.

The first model uses the material properties from the bulk thermal conductivity of both silicon and SWCNTs. A thermal conductivity of 1.38 W/m K [13] and mean free path of 300 nm [8] are used for silicon. A thermal conductivity of 2000 W/m K [10] is used to illustrate an average of experimental and theoretical values in the range found by current researchers, while a mean free path of 1000 nm [8] is also used. These material properties are applied to the assembly at the sub domain level, which represents the entire volume of each component in the assembly. The anisotropic conditions as represented in Equations 5 and 6 are incorporated into the material properties by use of the “constants” function and according to the directional frame of reference. This allows for the adaptation of the modeling tool to solve problems on the nanoscale. The conduction problem is incorporated into the “General Heat Transfer” module in COMSOL by neglecting the convection and radiation components of the heat equation. The anisotropic conditions also modify the heat equation per Equation 7 and the temperature distribution is solved for once the appropriate mesh was generated. For boundary conditions, an arbitrary heat flux of 1.0×10^7 W/m² is applied to illustrate a realistic temperature distribution, with the magnitude per unit area being large due to the small cross sectional area at which it is applied. This value can easily be manipulated in an actual design model. Other boundary conditions include insulation where there is no heat transfer and an initial temperature of 300 K set at the trailing edge of the heat sink silicon block. This initial temperature provides a base point that can be analyzed after the heat flux is added for the solution.

A mesh was fitted to the geometry assembly sub domains to solve the heat transfer problem. One advantage found in the meshing application was the ability to mesh each sub domain separately. This allows for different mesh types to be applied to specific geometries. For this case, a triangular mesh is applied to the silicon block sub domains while a rectangular mesh was applied to the SWCNT with a triangular cross section. Some difficulty, however, can be experienced in this procedure. In Figure 7 below, a cross sectional view of the SWCNT cylinder is shown fitted with a triangular mesh. The issue with this is that a very fine resolution must be used in order to get the representative curvature of the cylinder. This becomes a computational problem since this creates a considerable amount of elements. COMSOL also does not allow for much variation from default mesh types for element size, shape, and resolution. The mesh can only be applied uniformly to a component without control over concentration of elements in a single area. There is a necessity for a high resolution at critical points, especially at boundaries; however, the remainder of a large entity does not require as many elements. Figure 8 shows how the fine resolution is distributed in excess over the entire assembly. A user-defined mesh feature would be helpful in future versions of COMSOL, especially when models with a significant characteristic feature size difference are needed.

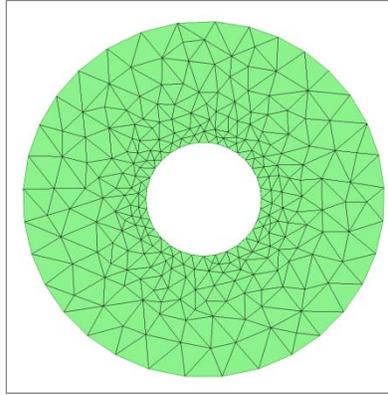


Figure 7: Cross sectional view of SWCNT model

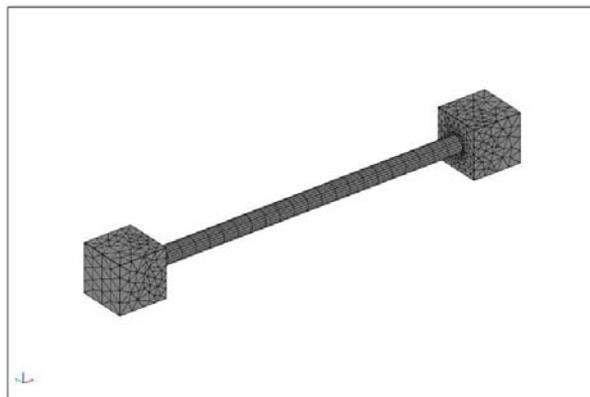


Figure 8: Initial SWCNT model fully meshed

After fitting the mesh, it is possible to solve the heat transfer problem in COMSOL for the geometry. Figure 9 below shows the resulting temperature distribution for the first model. It is evident in Figure 9 that the temperature distribution in the SWCNT is relatively constant. This can be viewed as illustrating the superior thermal conducting capabilities as compared to silicon since the heat is distributed more evenly throughout the SWCNT and there are no visible hot spots. Also, the anisotropic properties are apparent in the model since the temperature distribution is clearly only in the longitudinal direction and there is little variance in the radial direction for both the SWCNT and silicon.

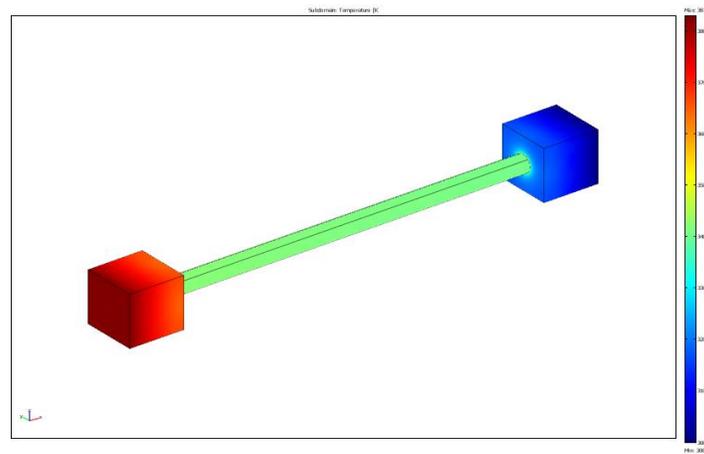


Figure 9: Thermal contours for initial SWCNT model

The second heat transfer model involves a more complicated assembly structure, and therefore, simplifications were made to analyze the model in COMSOL. A square array of SWCNTs was used to represent the interconnect as opposed to a cylinder to simplify the geometry. This simplification still provides the same effective area and theoretically, the same amount of heat flux for the problem while making it easier to generate the model and mesh. Properties used for the previous model are preserved for SWCNT interconnects and silicon in conjunction with anisotropic conditions for the components of small length scales. A bulk thermal conductivity of 400 W/m K [10] and mean free path of 40 nm were used for copper [8]. The scaling procedure was again performed to link the SolidWorks geometry with COMSOL. The arbitrary heat load of 1.0×10^7 W/m K was applied at the copper plate. The appropriate boundary conditions in areas of no heat flow were applied with a temperature of 300 K initialized at the bottom of the lowest silicon layer.

A mesh was applied to each subdomain to allow for some control of the element type. Problems associated with the previous model in meshing exist in this model. COMSOL, however, makes it possible to fit a dense mesh on the interconnects while fitting a coarse mesh to the other components. This is possible because the components are located in different subdomains. The smaller interconnects, which are the areas of interest, can have a finer concentration of elements, than the silicon and copper layers as shown in Figure 10 below.

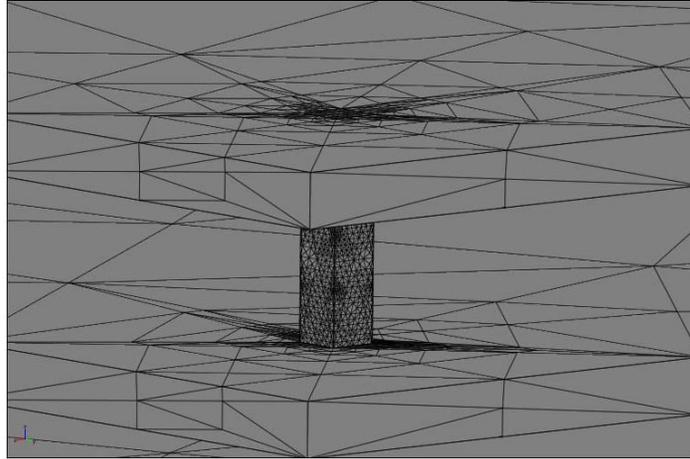


Figure 10: Illustration of the mesh variation for the different subdomains

After the mesh generation was completed, and the appropriate load and boundary conditions applied, COMSOL could solve the heat transfer problem. Figure 11 below shows the resulting temperature distribution for the stacked model. The results illustrate the largest temperature difference between the layers, with the lowest layer remaining at the coldest temperature. Figure 12 is a close-up view of a SWCNT bundle interconnect in the three dimensional stacked model. The results are consistent with the previous model in that the temperature distribution is relatively constant through the bundles.

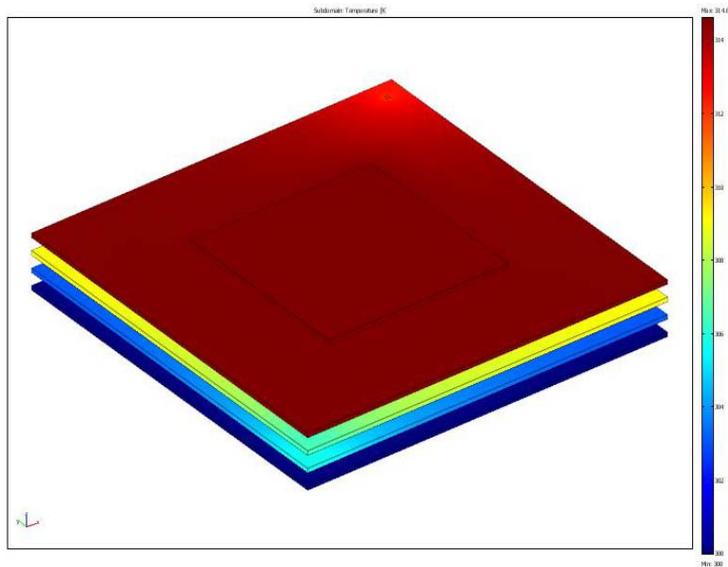


Figure 11: Thermal contours for stacked model

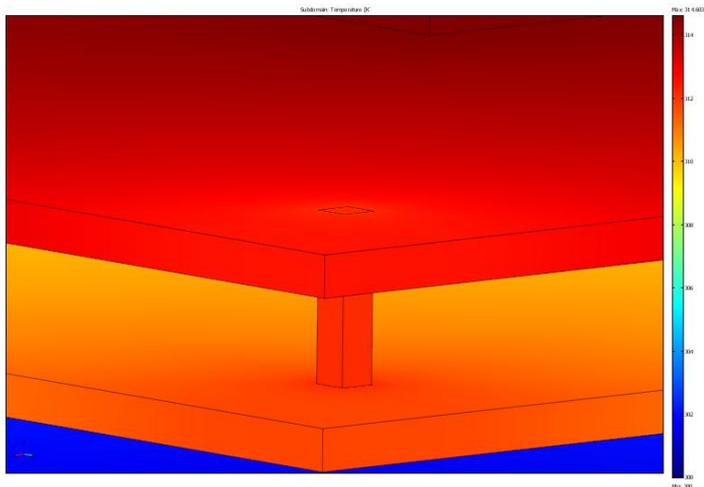


Figure 12: Close up of interconnect interface with larger layers

3.2.5 SWCNT Concluding Remarks

In the examination of SWCNTs assumptions were made to facilitate heat transfer modeling trade studies of this technology. These simplifications when done correctly during model creation permit the use of existing commercial software packages in the exploration of the applicability of new technology. The results from this portion of the effort suggest that SWCNTs might be beneficial for future computing technology. Three gaps in technology stood out in this portion of the project. First, a solid understanding of the characteristic material properties of carbon nanotubes appears to be missing. Second, additional work is needed to understand the role of molecular effects in heat transfer modeling of nanodevices. Third, users of commercial software packages need to be aware of potential limitations in a particular package's ability to model the feature sizes of the nanodevices of interest.

4.0 BIOTECHNOLOGY

Biotechnology opens the door to several different approaches to not only alternative computing concepts, but also alternative means of fabrication. There are many ways to view biotechnology and to help categorize the various research areas the following definitions, defined in a previous project [2], are used:

- a. Biomolecular Computing is computing with biological material.
- b. Bio-inspired computing is the mimicking of biological processes on silicon.
- c. Bioinformatics is, “an interdisciplinary field bringing together biology, computer science, mathematics, statistics, and information theory to analyze biological data for interpretation and prediction.”[14] Basically, this involves the application of computer science to process data for medical and biological research.
- d. Biocomputing is the broad term that covers the pursuit of computing technology which encompasses biomolecular computing, bio-inspired computing, and bioinformatics.

The original intent of looking at this area of alternative computing concepts was to further explore how one would conduct the proper modeling and simulation process for designing and building a biomolecular computer. This portion of the project, however, changed directions a few times as a result of becoming wiser about the limitation of some concepts along with an organizational change resulting in a change in technical direction for the CTC and support for this area of research. M&S work in the area of Biomolecular Computing, for instance, never materialized as envisioned. This section of the report will provide a quick snapshot of what was examined, an area of interest to watch, and an overview of work conducted in-house.

4.1 Membrane Computing

Membrane Computing, also known as P-Systems, is a relatively new concept, coming into existence in 1998 when Gheorghe Paun published his first paper on the subject. “Membrane Computing is a bio-inspired branch of natural computing, abstracting computing models from the structure and functioning of living cells and from the organization of cells in tissues or other high order structures.” [15] This concept was initially looked at for two reasons. First, could computational biology tools such as those found in the Bio-SPICE environment be used to model a membrane computing function? It was quickly determined that the available systems biology tools were not appropriate for modeling P-Systems. Second, there was some research that indicated that there was a little interest in a hardware implementation of P-Systems [16], at the time this area was examined for this project. Thus, it seems like the role for P-Systems is more along the nature of software or algorithm development. Since this effort is focused on examining how alternative computing concepts could be implemented in hardware no further research in this area was pursued. If the reader is interested in learning more about P-Systems, a good starting place for further reading is at: <http://ppage.psyste.ms.eu/>.

4.2 DNA Self-assembly

Part of the problem with the biomolecular computing area was that research and development activity in other organizations seems to get diverted from pure information technology related applications into medical applications. This diversion is a barrier to the sound development of a clear path forward for the broad community for biomolecular computing. It is hard to maintain local support for a technology area when there is not a larger community asking for it. One area in biotechnology that has some potential and will need to be watched is the area of DNA self-assembly. Simply stated, this is the use of synthetic DNA to fabricate nanoelectronics. Interest in this area comes from the debate of Moore’s Law, and whether or not existing lithographic processes, currently used in electronics fabrication lines, can be refined to be applicable for the small size scales of future electronic technologies. Researchers feel that the use of DNA self-assembly may provide an alternative approach for nanoelectronic fabrication. There are numerous sources on the internet for readers interested in learning more about this area. The book, “Introduction to DNA Self-Assembled Computer Design,” by Professors Christopher Dwyer and Alvin Lebeck at Duke University is an example of a resource from which one could learn more about this area and how it could apply to computer design. [17]

4.3 Structural DNA Nanotechnology

This aspect of the project sought to establish the foundation for utilizing structural DNA nanotechnology in future Air Force systems. DNA's natural ability to encode, store, and process information allows DNA to be used as taggant material. Thus, the focus was on enhanced situational awareness through advanced taggant design. The desire was to leverage past work with computational design tools and DNA to capture the fundamentals for the design of optimized, complex, covert molecular taggants utilizing structural DNA nanotechnology.

4.3.1 Structural DNA Nanotechnology Background

Reliable recognition and reporting of extremely small numbers of target DNA strands plays an important role in diverse areas from medical diagnosis to molecular taggants for controlled materials. One problem is that molecular recognition events provide low reporting signal per event, requiring significant instrumentation for detection of small amounts of target material. DNA recognition amplification techniques such as the Polymerase Chain Reaction (PCR) significantly increase the available report signal by increasing the availability of recognition strands and their associated reporting mechanism. [18] Amplification occurs as long as there is more target material for the increased recognition strands to react with. When the target material is exhausted, amplification ends for recognition amplification. A relatively new technique, Hybridization Chain Reaction (HCR), exists that can provide an autocatalytic amplification of the reporting material without requiring an amplification of the recognition material or large amounts of target material. The initial HCR technique involved the use of two complimentary catalyst hairpin strands, one of which was tailored to recognize an initiator or trigger strand. [19] The complimentary hairpin strands then open each other in turn by a chain reaction that is fueled by the energy released as the hairpins open. [20] A large concatenated duplex chain's results can then be isolated by gel electrophoresis. Amplified reporting signal is achieved in the growing chain by attaching an emitter such as a fluorophore to each catalyst strand. Attaching a quenched fluorophore allows the use of real-time detection as the signal strand grows. [21]

The chain-based HCR amplification rate is limited to a simple linear progression depending on the number of trigger molecules and eventually, bio-processing errors limit the size of the concatenated strand. A way around this is to design the catalyst strands to form dendrimer branches. As each catalyst strand opens, more than one new initiation site can be formed to react with a complimentary strand. The resulting structure grows at an exponential rate. [22, 23]

There were three design improvements to improve HCR yield in dendritic self-assembly systems addressed in this project. First, the false positive rate is reduced by creating a specific external toehold for the initiator hairpin. Thus, the toehold length may be optimized to maximize signal-to-noise ratio without negatively affecting the HCR amplification phases. In addition, the concentration of this hairpin will establish a tree-to-

branching ratio which is related to the expected number of growing trees within a reaction versus the expected amount of branching within a tree. The second modification facilitates exponential amplification beginning at the first stage of the reaction and creates more equally stable hairpin structures before a HCR begins. More equally stable hairpins, or those with similar melting temperatures, will help ensure that the different types of hairpins will have similar reaction rates regardless of environmental conditions. Finally, the third modification alleviates an identified potential structural issue while minimizing the required number of extra distinct hairpins. If this structure being prevented was to arise merely once in a single dendritic tree it would cut any future potential signal amplification from that branch site in half. Additional details can be found in the poster paper in the Appendix that was presented at the 2009 Foundations of Nanoscience Conference. [24]

Selection of the appropriate sequence, up to this point, was a manually intensive process requiring considerable individual expertise to implement correctly and lacked a means of verification prior to experimentation. The process also did not consider how the whole structure would recursively unfold and grow. Interaction between the Air Force and Duke research teams [25] clarified the complexity of defining the proper dendritic structure for HCR experiments. The next step was to bring together the software tools to capture the fundamentals for sequence design in order to establish a robust approach for generating libraries of optimized hairpins.

4.3.2 Optimizing DNA Selection

This portion of the project explored a method for optimizing DNA selection in terms of maximizing structural integrity, signal-to-noise ratio and information content by combining computer science techniques and mathematical methods routed in hybridization efficiency. This work presents significant improvement over state-of-the-art since, at the time of this project, there was no known target identification schema using these new dendritic HCR trees. It is a key step in the optimization of taggants to encode and process information reliably. Foundational information for this part of the project will first be presented before presenting the computational tool.

4.3.2.1 Basis for the Optimization Approach

The basis for the optimization approach pursued under this task involves understanding DNA thermodynamics, the impact of secondary structures, and HCR. Key published works of other researchers were leveraged for this project, and are referenced at the appropriate places, so that the focus could be on the development of the computational tool.

It should be noted at this point that, without going into great detail at this point, strands of DNA are commonly made up of a sequence of some combination of four bases adenine (A), guanine (G), thymine (T), and cytosine (C). When hybridization occurs A only pairs with T and G only pairs with C.

DNA Thermodynamics

The ability to predict DNA hybridization depends on having an accurate model of DNA thermodynamics. *The Thermodynamics of DNA Structural Motifs* [27] provides one with an accurate model upon which to base one's calculations of the enthalpy and entropy of nearest neighbor (NN) base pair sequences. It also provides us with a framework to calculate the thermodynamics of almost any DNA structures by providing us with the thermodynamic penalties associated with anomalies such as mismatched base pairs or various hairpin, bulge and internal loop structures. [27] This model is the product of decades of research on polynucleotide structure and thermodynamics [28], and has been extensively verified. [29]

Table 2 provides an example of the nearest neighbor model for Watson Crick (WC) base pair sequences. Values for the enthalpy (ΔH°) and entropy (ΔS°) of a select set of sequences are given from which the Gibbs free energy (thermodynamic strength, ΔG_T°) for WC base pair hybridizations at given temperature can be calculated. Calculations were made using Equation 8, resulting in the right-most column on Table 2.

$$\Delta G_T^\circ = \Delta H^\circ - T\Delta S^\circ \quad (8)$$

Figure 13 illustrates the application of the NN Gibbs free energy values for DNA sequence hybridization. In this case, the sequence is a six base pair hybridizing with its own WC reverse complement. To start this calculation a 1.96 initiation penalty is assumed. Next, the respective free energy from Table 2 for each NN base pair is inserted. Finally, a terminal penalty is applied based upon how the sequence terminates on either end. In this case it is 0.05. All these values are then summed up for a total of $-5.35 \text{ kcal mol}^{-1}$. This represents the amount of potential energy lost when these two sequences hybridized, and inversely, the amount of energy needed to break them apart. The implication of this is that the lower the free energy the stronger the sequence hybridization, or the higher the energy the weaker the hybridization.

It's important to note that Table 2 and Figure 13 only apply to strict WC hybridizations. For non-WC hybridizations, called cross-hybridizations (CHs), the calculation becomes much more involved. Single base pair mismatches involve looking up thermodynamic penalties on a separate table. Multiple mismatches can further complicate things by forming complex structures such as internal or bulge loops. In these cases, the backbone of the DNA structure must also be taken into consideration. Without going into all the details, what is most important to note about cross-hybridizations is that they form much weaker hybridizations, and are more likely to fall back apart. Using this knowledge one can leverage the difference in thermodynamic strengths of various sequence hybridizations to predict and build complex structures for target identification. Maximizing the differences in free energies between wanted hybridizations and unwanted cross-hybridizations, are key in successful target identification using DNA.

Table 2: Thermodynamic properties of WC NN base pair sequences [27]

Sequence	ΔH° (kcal mol ⁻¹)	ΔS° (e.u.)	ΔG_{37C}° (kcal mol ⁻¹)
AA/TT	-7.6	-21.3	-1.00
AT/TA	-7.2	-20.4	-0.88
TA/AT	-7.2	-21.3	-0.58
CA/GT	-8.5	-22.7	-1.45
GT/CA	-8.4	-22.4	-1.44
CT/GA	-7.8	-21.0	-1.28
GA/CT	-8.2	-22.2	-1.30
CG/GC	-10.6	-27.2	-2.17
GC/CG	-9.8	-24.4	-2.24
GG/CC	-8.0	-19.9	-1.84

$$\begin{aligned}
 &5\text{'-CGTTGA-3'} \\
 &3\text{'-GCAACT-5'} \\
 \Delta G_{37C}^\circ &= \text{initiation} + \text{CG} + \text{GT} + \text{TT} + \text{TG} + \text{TA} + \text{terminal} \\
 \Delta G_{37C}^\circ &= 1.96 - 2.17 - 1.44 - 1.00 - 1.45 - 1.30 + .05 \\
 \Delta G_{37C}^\circ &= -5.35 \text{kcal mol}^{-1}
 \end{aligned}$$

Figure 13: Calculation of free energy for WC complementary sequences

Secondary Structures

Secondary structures are the name for structures formed when a sequence of DNA folds back upon itself. The idea is, that a denatured (single stranded) sequence of DNA in an attempt to reduce its “free” potential energy, will fold upon itself into a thermodynamically favorable hybridization. This folded structure can easily be predicted, allowing specially designed DNA sequences to fold into specific secondary structures. An example of this can be seen in Figure 14.



(a) Single Stranded Structure



(b) Folded Secondary Structure

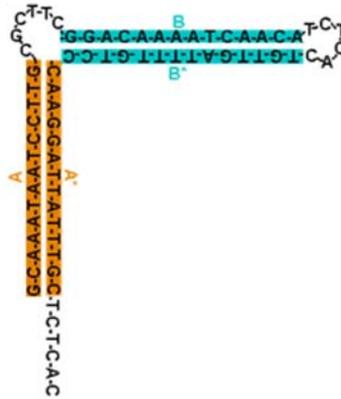
Figure 14: Folding of single stranded sequences (a) into hairpin secondary structure (b)

Figure 14 shows a long sequence of adenines and thymines separated by a few random bases of adenine, thymine, guanine, and cytosine. In an attempt to reduce its free energy, the thymines will fold back upon the adenines. The most probable manner, in which the sequence will fold, a four base pair hairpin loop, is illustrated in the bottom of Figure 14. Other structures such as a five or six base pair hairpin loop could also form, but would be less likely due to the larger thermodynamic penalties associated with them.

Figure 15 shows a slightly longer and more complicated sequence than Figure 14. To deal with these longer sequences, letters are assigned to the subsequences at the macro level to avoid dealing with all the intricacies that lie at the base pair level. For example, Figure 14 contains subsequences A and B. Subsequences A* and B* (sometimes A' B') are denoted such that they are the reverse WC complements of A and B. For example, if a B* folded back upon a B, a perfect WC hybridization occurs and would be very thermodynamically favorable, while an A sequence folding upon a B sequence should be a weak cross-hybridization and not so thermodynamically favorable. In the case of Figure 15, the secondary structure is relatively obvious. B will fold back on B* and A will fold back on A*. The structure that forms is what will be referred to as a double hairpin structure. The double hairpin structure is what is used in the dendritic hybridization chain reaction which will be covered next.

G-C-A-A-T-A-A-T-C-C-T-T-G C-G-C-T-T-C G-G-A-C-A-A-A-T-C-A-A-C-A-T-C-T-C-A-C T-G-T-T-G-A-T-T-T-T-G-T-C-C-A-A-G-G-A-T-T-A-T-T-T-G-C-T-C-T-C-A-C

(a) Single Stranded Structure



(b) Folded Secondary Structure

Figure 15: Folding of single stranded sequences (a) into double hairpin secondary structure (b)

Hybridization Chain Reaction (HCR)

The premise of the HCR is that potential energy is stored in closed and unbound hairpin loops. [20] Normally inaccessible, this energy cannot be activated until the hairpin is opened and the loop sequence is fully exposed. This new exposed energy can act as a catalyst to open another hairpin yet exposing more free energy to open more hairpins. What results, is a structure whose overall potential energy is at a far lower and more stable state than within individual hairpins.

The simplest of the linear growth HCR consists of three different DNA sequence species. The first two are thermodynamically stable hairpin structures that can coexist with each other within a solution with minimal reactions between species. These hairpins are non-reactive due to the long bound neck of the hairpin that is difficult to break apart. The third structure is an initiator strand which ideally contains no large self complementary subsequences so it will not react with itself and form any sort of stable secondary structure. These three structures and their macro level sequences are outlined in Figure 16. [19]

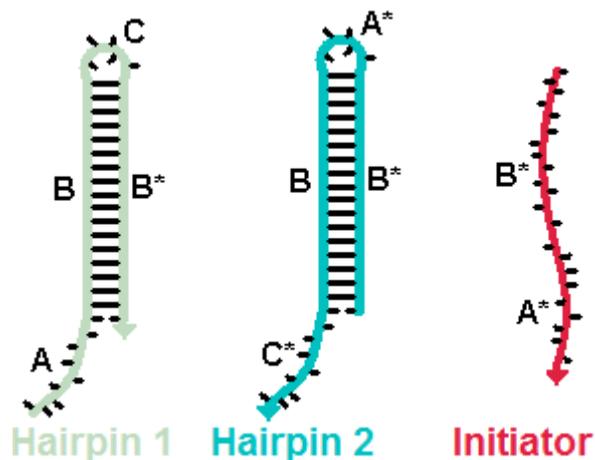


Figure 16: HCR sequence species

Figure 17 shows the actual HCR process. Imagine a “well stirred” solution of Hairpin 1’s and 2’s free flowing and bumping into each other in a solution. The hairpins are designed to have stable secondary structure and be relatively un-reactive which each other, so on the rare chance cross-hybridization does occur between hairpins, they should be weak – and break apart easily. Essentially these hairpins are waiting in the solution until they come in contact with the initiator strand.

The process begins with the application of the initiator strands, which is analogous to the presence of a target. The newly applied initiator strands will float around until they come in contact with the A subsequence of Hairpin 1. Subsequently, the A* subsequence of the initiator will hybridize with the A subsequence of Hairpin 1 (Figure 17, Step 1). Now that the initiator has a “toehold” on Hairpin 1, the subsequence B* of the initiator will displace, and “un-zipper” Hairpin 1’s own B* subsequence. This is because with the toehold, the complete initiator hybridizing with the complete Hairpin 1, is more thermodynamically favorable than the folded hairpin structure (Figure 17, Step 2). Now that the Hairpin 1 is unraveled, a CB* subsequence is exposed which will act as an initiator on Hairpin 2 (Figure 17, Step 3). When Hairpin 2 opens up an A*B* subsequence is exposed which is the same as the initiator sequence, and this opens up another Hairpin 1 (Figure 17, Step 4). The HCR is now in place.

Another approach to exploiting HCR is through exponential growth HCR which is being pursued under this phase of the project. Unlike the linear growth HCR, the exponential HCR initiator strand is made up of three parts instead of two. These initiator strand subsequences are named B*, A*, and T*. Another difference is that exponential growth HCR requires the use of a nine double hairpin set (Figure 15) instead of the pair of hairpins set used in linear HCR. It should also be noted that when these double hairpins unfold they expose two binding sites (initiators) instead of the usual one. An example of this exponential hybridization reaction is seen in Figure 18. [24]

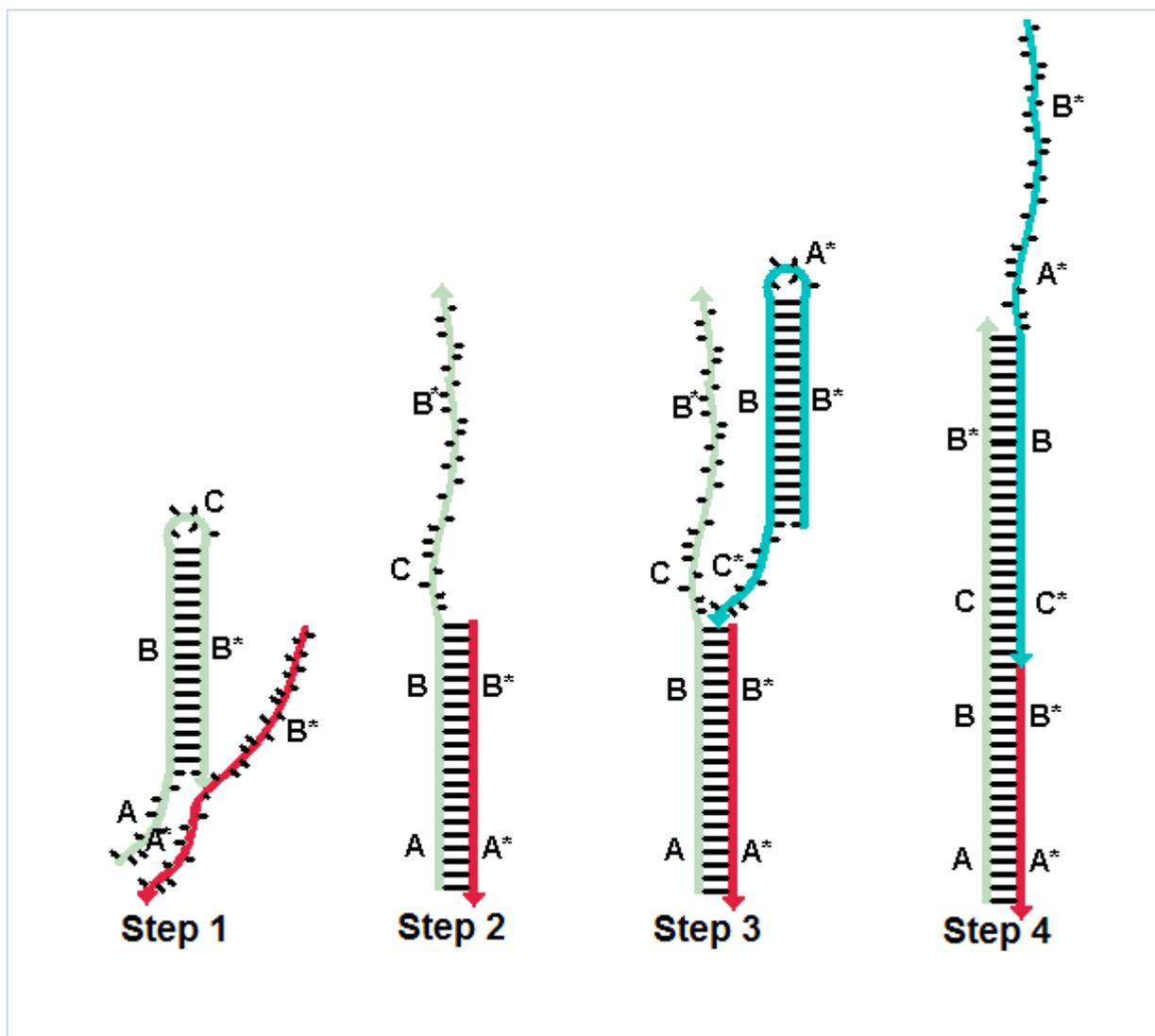


Figure 17: Linear growth HCR process

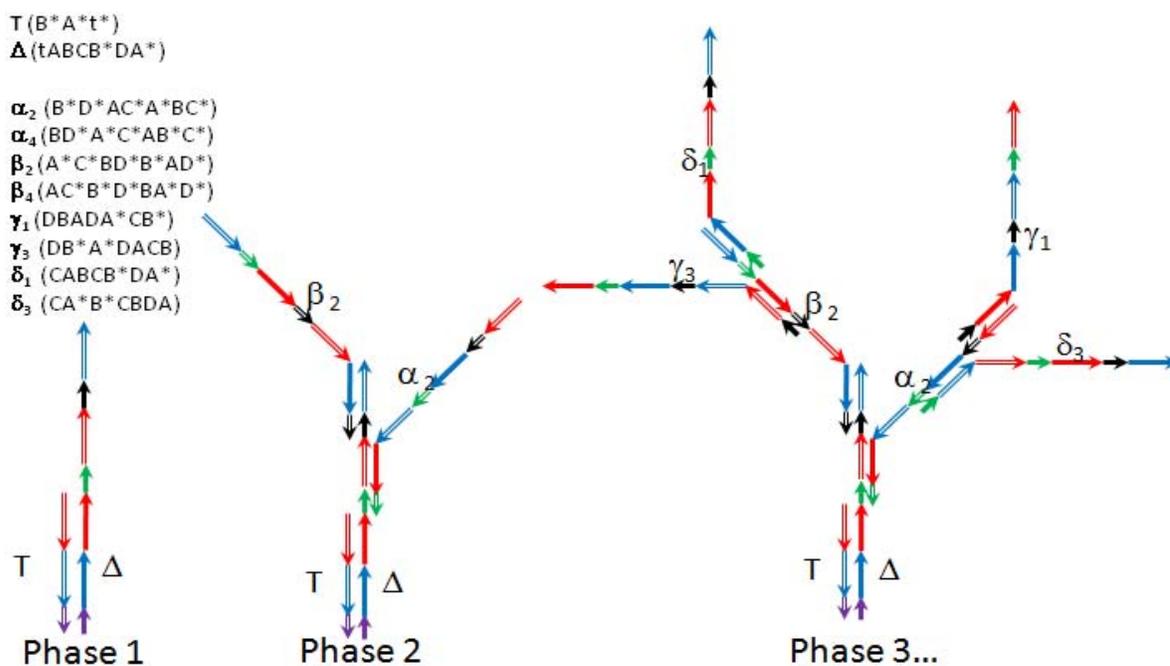


Figure 18: Exponential growth HCR process

In Figure 18, phase one shows the Δ double hairpin binding to the target and unfolding exposing two binding sites. In phase 2, α_2 and β_2 hybridize with this target delta exposing for a total of four binding sites. Phase 3 shows these four binding sites overtaken with four more double hairpins exposing a total of eight binding sites.

4.3.2.2 HCR Probe Finder

The exponential growth HCR presents an obvious advantage over linear growth HCR. Since this method of HCR is new, no computer-based method exists yet to take advantage of this technique as a means for target identification and amplification. HCR Probe Finder fills that gap. Key to the development of the computer program was breaking the problem down into two major stages. The first stage looked for optimum places to choose your initiator strands, by allowing the user to set constraints such as the length of the B, A, T subsequences, and their respective free energy strengths. It also allows the user to constrain the internal cross-hybridization between the probes A, B & T subsequences which in effect increases the signal-to-noise ratio. The second stage of HCR Probe Finder deals with maximizing the number of uniquely identifiable targets. This is accomplished by choosing probes out of Stage 1 that are the most unique from each other.

Stage 1

The first stage of HCR Probe Finder is meant as a filter to find the most optimal initiator strands in each target. Since each of the nine double hairpins used to identify the target are built from the subsequences of the initiator strand, it's important that strict constraints are placed on these initiators so that the dendritic trees grow properly and the potential for false positives reduced. The three main sets of filters were determined to be important to ensure proper initiator strands: thermodynamic constraints, internal cross-hybridization constraints, and unique "barcode" constraints. Descriptions of these constraints and their importance are presented below after the length constraints are explained. The numbers in the brackets represent the default values that were used for this project.

Length Constraints

Length constraints allow the user to control the length of the subsequences used to build the nine double hairpins used in each of the probes. What is important in choosing lengths is that the A and B subsequence lengths (-a_length and -b_length, respectively) are significantly longer than the T subsequence length (-t_length). This allows for the neck of the double hairpins to be long and stable enough to offset the unbounded neck and tail pieces. Here is a summary of the parameters:

```
-t_length    <6>      (C and D subsequences will be set to this length in Stage 2)
-a_length    <14>
-b_length    <14>
```

Thermodynamic Constraints

Thermodynamic constraints allows the user to set a minimum and maximum WC free energy for each of the B, A, and T subsequences. It also allows the user to set thermodynamic constraints on the strength of BAT sequence hybridizing to the target. Finally, it allows the user to set a maximum difference between the A subsequence strength and the B subsequence strength. Below is a summary of the parameters:

```
-t_wc_max    <-3.6>,    T subsequence thermodynamic maximum
-t_wc_min    <-6.3>,    T subsequence thermodynamic minimum
-a_wc_max    <-18.7>,   A subsequence thermodynamic maximum
-a_wc_min    <-14.3>,   A subsequence thermodynamic minimum
-b_wc_max    <-18.7>,   B subsequence thermodynamic maximum
-b_wc_min    <-14.3>,   B subsequence thermodynamic minimum
-bat_wc_max  <-49>,     BAT sequence maximum thermodynamic threshold
-bat_wc_min  <-42>,     BAT sequence minimum thermodynamic threshold
-delta_ab    <0.5>,     Maximum difference between thermodynamics
```

The purpose of the thermodynamic strengths is to ensure a minimum “signal” level strength of wanted hybridizations. It is also desired for the signal sequences to have similar thermodynamic characteristics, so an upper bound on thermodynamic constraints must be set. This helps ensure that binding sites are utilized evenly, and helps minimize abnormal growth.

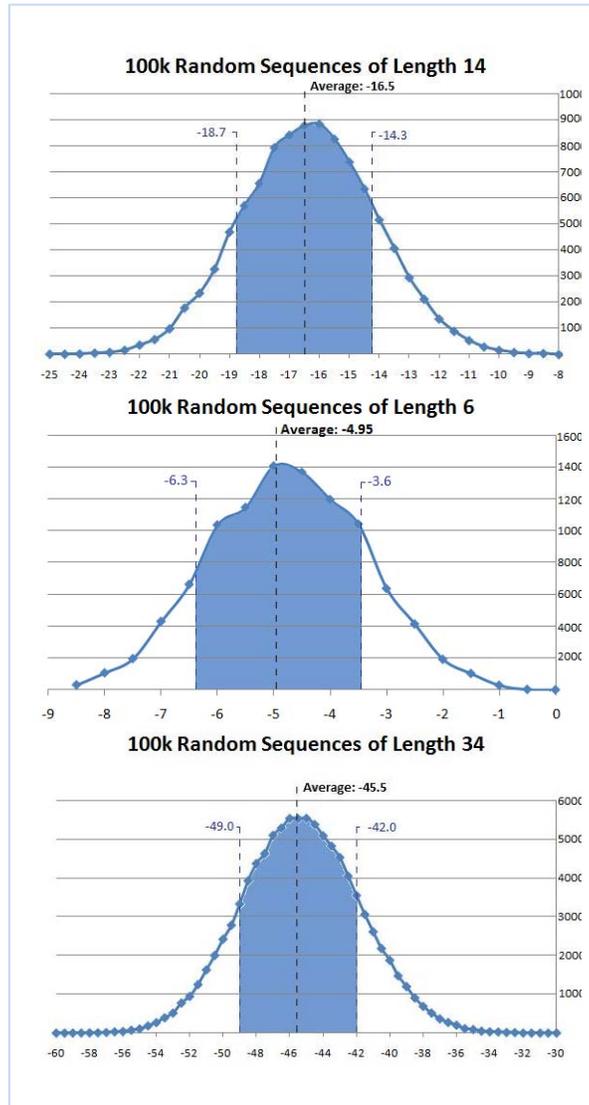


Figure 19: Free energy distributions for various DNA sequence lengths

Figure 19 presents the histograms of 100,000 (100k) randomly generated sequences and their respective free energy distributions. These distributions were used to calculate default constraints to have for this stage of the program. As one can see in the Figure 19, the distributions approximate to a normal distribution. Choosing values within one standard deviation of the average gives us the upper and lower bounds to set for our free energies. It's important to note that these distributions and thresholds only apply if the user does not vary the length of any of the strands. If the user varies the strand length, these thresholds no longer apply and it is up to the user determine the new thresholds on which to base new boundaries.

Cross-Hybridization Constraints

The next important constraint to set for the probes is the internal cross-hybridization constraint. The purpose of this constraint is to prevent various double hairpins of the same probe's type from interacting with each other, and that the secondary structures that are predicted actually do form. An example of what the internal cross-hybridization check examines can be seen in Figure 20. The parameter is:

`-ch_neck_internal <-4>` Internal cross-hybridization threshold

	A	B	A*	B*		T
A	-9.6	-3.5	-16.7	-5.4	T*	-5.6
B		-4.3	-5.4	-17	T	-1.5
A*			-9.6	-3.5		
B*				-4.3		

Figure 20: Internal cross-hybridization matrix

Figure 20 shows the internal cross-hybridization matrix for a given probe. Highlighted in blue are the WC free energies that can be seen as our signal. Subsequence A hybridizing to subsequence A*, B to B*, and T to T* are hybridizations needed to build the double hairpins and our dendritic tree. Entries highlighted in red can be seen as the noise, which are all cross-hybridizations. It's only necessary to check half of them because, for example A hybridizing with B has the same free energy as A* hybridization with B*. If the user were to set a threshold for these cross-hybridizations to -6 then this probe would obviously fail. `-ch_neck_internal` allows the user to set the maximum allowed red square in the matrix above.

Unique Barcode Check

The final test for the HCR probe finder to complete in the first stage is what is called the unique barcode check. The purpose of this test is to ensure that the BAT sequence located on the target delta (which initiates tree growth) only hybridizes well with the target it was sourced from and no others. An example of this unique barcode check matrix can be seen in Figure 21. Its parameter is:

-barcode_threshold <0> Barcode cross-hybridization threshold, use 0 to skip

	T1P1	T2P2	T2P3
Target 1	-45	-13.1	-20.7
Target 2	-12.3	-43.1	-45.3
Target 3	-15.2	-9.8	-34.8
Target 4	-5.2	-12.3	-40

Figure 21: Unique barcode test matrix

Figure 21 shows an example of various probes and how they hybridize with the target set. The matrix is set up the same way as the one in Figure 20. The desired signal is highlighted in blue while the noise is highlighted in red. Probe 1, for example, sourced from Target 1 hybridizes very well with that particular target and to a lesser extent with the other targets. While Probe 3 sourced from Target 2 hybridizes well to Target 2, it also hybridizes relatively well to Targets 3 and 4, so it would likely be rejected. By using the -barcode threshold argument, the user is allowed to set constraints on the maximum value for red to allow a probe to pass. A value of 0 passed (default) will leave the barcode check off.

Stage 1 Output

Once a probe passes the thermodynamic, internal cross-hybridization, and barcode constraints the probe is an acceptable solution for identifying a specific target. The probe is put into a possible useful probe database and stored for further analysis. Stage 1 continues searching the targets for possible probes until all the possible probe sequences are exhausted. An example of Stage 1 output is shown below into Figure 22.

```

...
T8-P18: a: -14.6 b: -14.6 t: -6.0 <CH>ab: -1.7 ab': -3.8 aa: -2.5 bb: -3.7</CH> delta_ab: 0.0 Sequence:
TGGTAGACAATTTGCTCAGATTCGTAGACCGAGT
T8-P19: a: -15.5 b: -15.4 t: -5.6 <CH>ab: -2.6 ab': -2.5 aa: -2.9 bb: -2.1</CH> delta_ab: 0.1 Sequence:
GTTACAAGATGACGGCAGTTATCGCATACTTCGG
T8-P20: a: -15.5 b: -15.5 t: -5.1 <CH>ab: -2.5 ab': -2.4 aa: -2.1 bb: -2.9</CH> delta_ab: 0.0 Sequence:
CGATTTACCGAACTTTCTTCTTCCAACGGAACGA
T9-P21: a: -17.9 b: -17.8 t: -5.0 <CH>ab: -2.7 ab': -3.4 aa: -3.5 bb: -2.2</CH> delta_ab: 0.1 Sequence:
CCGCCTTTAGTGGTCCTGGAGACATCCGCTGTTG
T9-P22: a: -16.3 b: -16.4 t: -4.1 <CH>ab: -3.8 ab': -1.9 aa: -3.3 bb: -1.7</CH> delta_ab: 0.1 Sequence:
TATCTTTCACCGCCTTTAGTGGTCCTGGAGACAT
*****Stage 1 Complete*****
Probes:
-Total Probes Checked: 10082
-Failed: 10060
- 1: 3015 (29.90%) Name: T-Thermo
- 2: 2037 (28.82%) Name: A-Thermo
- 3: 1441 (28.65%) Name: B-Thermo
- 4: 3414 (95.12%) Name: Delta AB
- 5: 29 (16.57%) Name: Overall MFE
- 6: 63 (43.15%) Name: CH A-A
- 7: 40 (48.19%) Name: CH B-B
- 8: 10 (23.26%) Name: CH A-B'
- 9: 11 (33.33%) Name: CH AB

```

Figure 22: HCR Probe Finder Stage 1 output

Figure 22 shows the actual output of the HCR Probe Finder program. At the top are the last few acceptable probes that the software encountered. Seen are their thermodynamic properties, cross-hybridization matrices, and their respective BAT sequences. Important statistics about Stage 1 are then presented.

Seen in the statistics is the overall number of probes that were checked, in this case ~10,000. This is completely dependent on the target file supplied; in this case the target file was ten ~1,000 length sequences hence 10,000 sub probes checked. Next in the statistics is the number of probes that failed followed by the breakdown of how many and what percentage of probes failed each test. For example 3,015 probes failed the T-Thermo test, leaving 7,045 probes passed along to the A-Thermo test. This represents an approximate failure rate of 30%. Then the 7,045 remaining probes are passed along to the second filter, the A-Thermo test, of which 2,037 failed. This represents another failure rate of approximately 30%. This seems to make sense based upon the process for selecting the thresholds for the thermo test. Selecting within one standard deviation in either direction of the mean would allow approximately a 70% pass rate, which aligns with exactly what the statistics say. Continuing down through the statistics are the other thermodynamic checks, cross-hybridization checks, and barcode checks. Delta AB was set very restrictive for no other reason than creating a very small pool of probes.

Stage 2

The second stage of HCR Probe Finder takes the probe pool from Stage 1 and looks at the similarity between these probes. The purpose of this is to create the largest possible subset of these probes that will not interact with each other, because if they do interact, false positives can occur. An example of this problem on a very small scale is outlined in Figure 23.

-ch_neck_pool <-4> CH threshold in Stage 2 for A-B subsequences of different probes
-ch_tail_pool <-2.5> CH threshold in Stage 2 for T-C-D subsequences of different probes
-time_limit <5> Time limit in minutes to search for probe solution set in Stage 2

	Elimination Vectors				
Blue	Blue	Green	Purple	Turquoise	Orange
Red	Red				
Green	Green	Blue	Turquoise	Orange	
Purple	Purple	Blue			
Turquoise	Turquoise	Blue	Green		
Orange	Orange	Blue	Green		

Figure 23: Elimination vector matrix for a 6 probe set

Figure 23 provides an example of the problem that Stage 2 of HCR Probe Finder is meant to solve. Figure 23 shows six probes assigned to different colors and their respective elimination vectors. The first step of Stage 2 is to build these elimination vectors. To build these, we take each probe from Stage 1, and use something similar to the cross-hybridization filters of Stage 1 to check the probe against all others in the probe pool. Probes similar to each other by cross-hybridization are added to each other's elimination vector. Probes that identify the same target are also added to each other's elimination vector. The reason for this is that it is not necessary or really advantageous to have more than one probe identifying the same target. Finally, probes are added to their own elimination vector because we do not want to make a selection using the same probe more than once. The result of this filtering is an elimination vector matrix such as that shown in Figure 23. This elimination vector matrix tells us for example if our solution pool contains the blue probe, the only other probe we can add to the pool is red. But adding the red probe to our solution pool allows us the choices of blue, green, purple, turquoise, and orange. In other words, choosing the "blue" probe allows one only one choice, while choosing the "red" probe avails one many more possible moves. What arises out of a process like this is a tree data structure as seen in Figure 24.

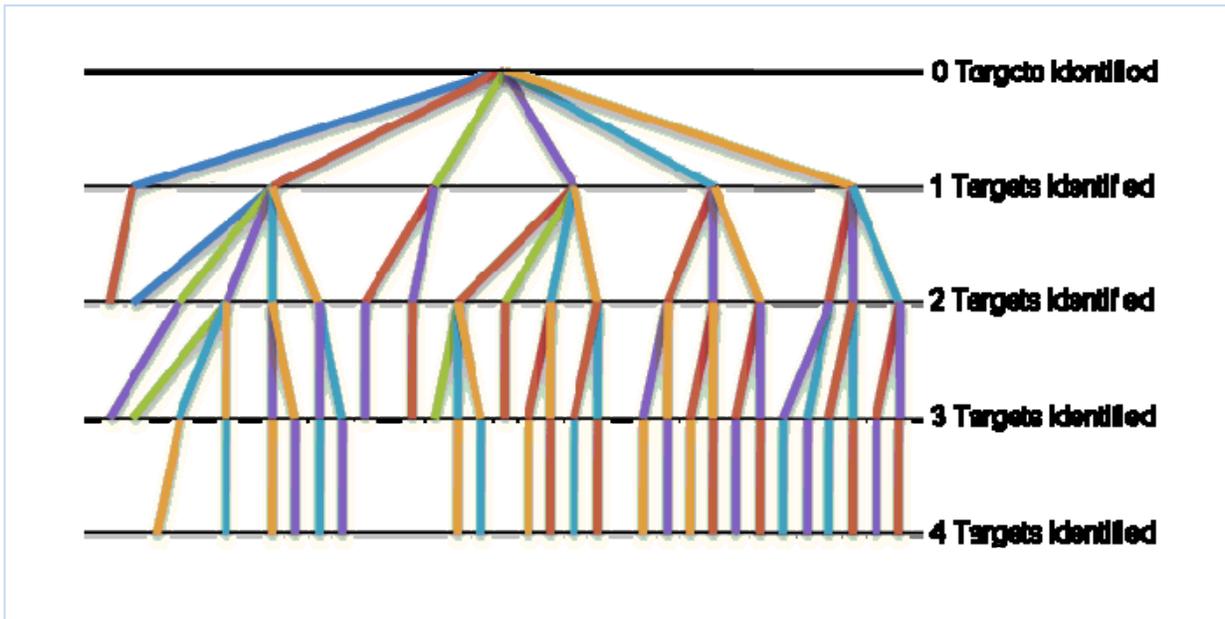


Figure 24: Tree structure for maximizing target identification

Figure 24 shows the tree structure that “stems” out of Figure 23’s elimination vector matrix. The goal now is to traverse the tree in such a way as to visit nodes at the maximum depth of the tree. Visiting the largest node in this tree ensures you have the largest subset of probes and will therefore be able to identify the most targets. On a small scale, with tens or hundreds of probes this is a relatively easy problem. Software could just be written to exhaustively search the tree and come up with the correct answer, but at larger scale this becomes impossible. The data structure grows so big that it becomes impossible to search more than a very small fraction of it. Imagine if there are 500 probes, the number of branches on this data structure could potentially be 500 factorial, which is 1,124 digits long.

The problem of finding the largest subset of workable probes in a given environment required a considerable portion of the development time for the initial version of HCR Probe Finder. Several different algorithms were developed, and a metric against seven different probe/target situations was checked to determine the most effective algorithm. Below in Figure 25 are the different test cases the algorithms were tested against.

Test	Targets	Probes	Similarity Measure
Test 1	10	117	95 (81.19%)
Test 2	10	117	42 (35.90%)
Test 3	20	232	125 (53.87%)
Test 4	20	232	75 (32.33%)
Test 5	50	613	315 (51.39%)
Test 6	50	613	180 (29.36%)
Test 7	50	613	96 (15.66%)

Figure 25: Algorithm test cases

The tests in Figure 25 were designed to represent very different tree complexities and topologies to make sure the algorithms would work across a variety of situations they may encounter. Below, in chronological order, are brief descriptions of the algorithm approaches that were pursued.

- The Filtering Algorithm involved starting with the entire probe set that comes out of Stage 1 and scoring them based on the number of other probes that would have to be eliminated if chosen. The probe with the highest score would be eliminated, and the remaining probes would be re-scored. This process would be repeated until the probe set contains no probes with a score greater than zero which means a valid solution set has been found and the filtering is complete. Initially, it was believed that this approach would be an effective way of choosing a large probe solution set. Once the tree data structure was conceived as a solution to this problem, however, this was not the case. This approach only filters down one possible solution path. Once tree data structure was conceived, the large size of the problem became apparent revealing how little of the solution space this method actually searched.
- An alternative approach was an Exhaustive Search of the entire tree data structure to search for all possible nodes. At first the algorithm proved to be very effective searching probe solution sets for identifying five and ten targets against which the program was initially being tested. As the target and probe sets became larger, this algorithm proved to be very ineffective. The trees grew so large that in test runs, varying from a few minutes to hours in length, the program would barely search even 0.1% of the tree, and the portion it was searching was far from location of the best solution space. This led to the next solution of weighing the nodes to be able to search more relevant portions of the tree.

- An Old Random Search approach to searching the tree involved weighting the branches based on the number of probes that would still remain at the next level if they were chosen. A certain percentage of these branches would then be randomly chosen based on the weights, and the subnodes would be examined. These subnodes would then be weighted and randomly chosen as well. This process would continue until a certain depth threshold was reached and a mini-exhaustive search would then begin for the nodes deeper than the current node. The purpose of this was to avoid the time it takes to weight the nodes and avoid re-visiting nodes previously visited which occurs in the random search. It was felt that through weighting and then randomly choosing branches to search, a greater depth of the tree would be searched allowing one to find a better solution more quickly. Although this method of searching the tree was actually the most complicated/powerful search developed, it proved to be difficult to pick the parameters that would match the tree topology. The end user would have a difficult time knowing how to set certain search thresholds to maximize this method.
- Since the previous approach seemed overly complicated, it was decided to use a New Random Weighted Search which weighted branches as in the previous approach, then randomly choose weighted branches. Travel to next level would then occur performing the same steps on the remaining branches until a dead end is reached. The process would then start over from the root. Early on it was believed that this method would not work as well as the previous approach, but it actually performed better under the given test cases. This was found to be the most effective algorithm developed so far for the HCR Probe Finder to search the tree for the maximum subset.

Figure 26, shows the overall effectiveness of the algorithms across the seven different test cases under a two minute time constraint.

After HCR Probe Finder reaches its limit, and is done finding the best solution set in Stage 2, the next step is to generate the C and D subsequences. This consists of generating random sequences and passing them through the Stage 1 constraints until we have enough C-D subsequences to assign to each probe in the solution set. Once that is done these C-D subsequences are paired up with the probes. This is done in such a way as to average out binding site strengths. If the C subsequence is strong it gets paired up with a weaker A subsequence. Likewise, if C is weak it gets paired with a strong A. Once these sequences are paired up Stage 2 is done.

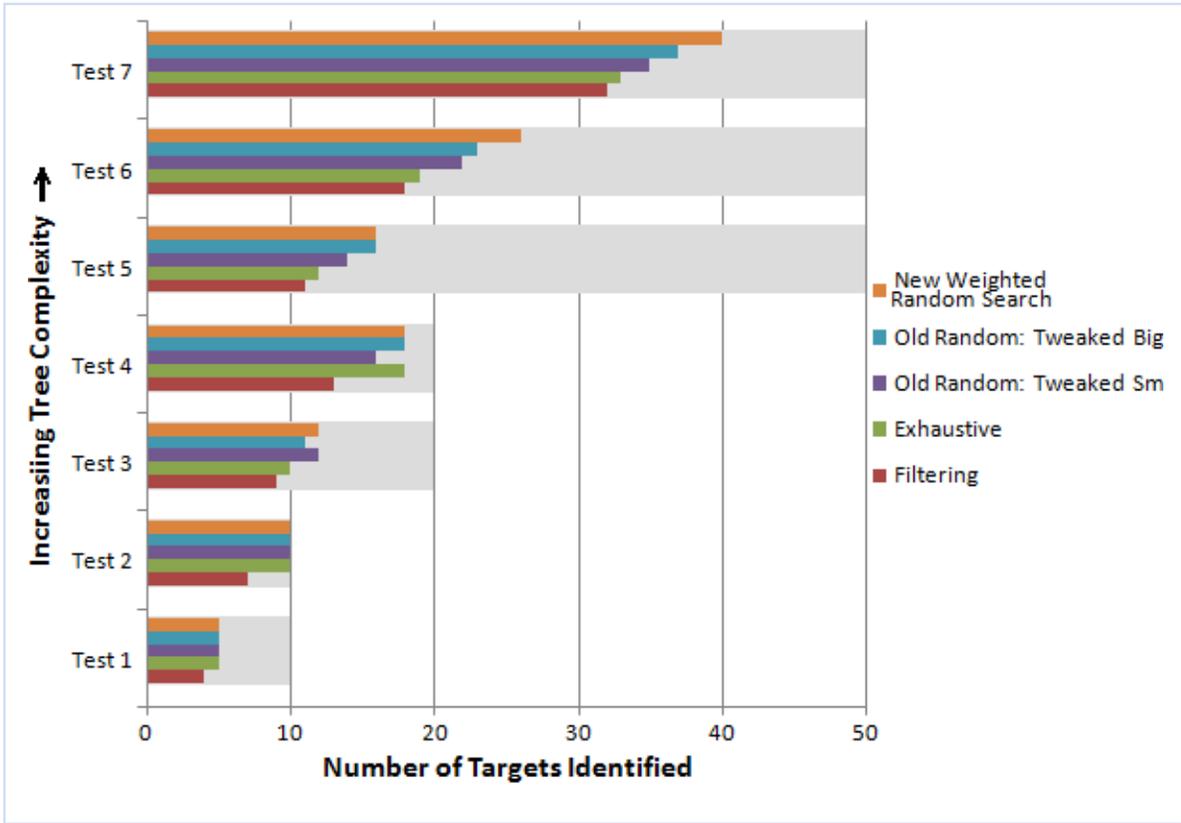


Figure 26: Algorithm effectiveness for finding maximum possible subset

Stage 2 Output

Once C-D subsequences have gotten paired up with A-B subsequences, HCR Probe Finder is basically done. The only thing left to do is providing the data to the user. There are two main methods for providing the user the final results.

The first method of providing results for the user is shown in Figure 27 which represents the double hairpin output of Stage 2. For each probe in the solution set of Stage 2, the T, A, B, C, and D subsequences used to build the double hairpins are listed. It also shows what target in the target file these double hairpins will identify along with the actual sequences of the specific double hairpins to be used.

```

Probe #0: - Identifies Target 7
-t: CAAAGG
-a: CTCAGTTCAGAGCG
-b: GTACCCACCTCTTG
-c: AGTTCA
-d: GCATAC
Hairpin #0-1: CAAAGGCTCAGTTCAGAGCGGTACCCACCTCTTGAGTTCACAAGAGGTGGGTACGCATACCCTCTGAACTGAGT
(tABCb*DA*)
Hairpin #0-2: CAAGAGGTGGGTACGTATGCCGCTCTGAACTGAGTGAAGTTCGCTCTGAACTGAGGTACCCACCTCTTGAACT
(B*D*A*C*A*BC*)
Hairpin #0-3: GTACCCACCTCTTGATGCCGCTCTGAACTGAGTGAAGTTCAGTTCAGAGCGCAAGAGGTGGGTACTGAACT
(BD*A*C*AB*C*)
Hairpin #0-4: CGCTCTGAACTGAGTGAAGTGTACCCACCTCTTGATGCCAAGAGGTGGGTACCTCAGTTCAGAGCGGTATGC
(A*C*BD*B*AD*)
Hairpin #0-5: CTCAGTTCAGAGCGTGAAGTCAAGAGGTGGGTACGTATGCCGTACCCACCTCTTGCGCTCTGAACTGAGGTATGC
(AC*B*D*BA*D*)
Hairpin #0-6: GCATACGTACCCACCTCTTGCTCAGTTCAGAGCGGCATACCGCTCTGAACTGAGAGTTCACAAGAGGTGGGTAC
(DBADA*CB*)
Hairpin #0-7: GCATACCAAGAGGTGGGTACCGCTCTGAACTGAGGCATACCTCAGTTCAGAGCGAGTTCAGTACCCACCTCTTG
(DB*A*D*ACB)

```

Figure 27: Stage 2 double hairpin output

The other mode of output can be seen below in Figure 28. This consists of outputting the binding site thermodynamic lookup table for the cross-hybridization between the various binding sites of all the double hairpins. This is what the stochastic simulation work explained in the next section of this report uses to simulate the tree growth for a given hairpin. To activate this output mode use `-build_thermo_table 1`.

`-build_thermo_table <0>` Builds the thermodynamics tables for stochastic simulation

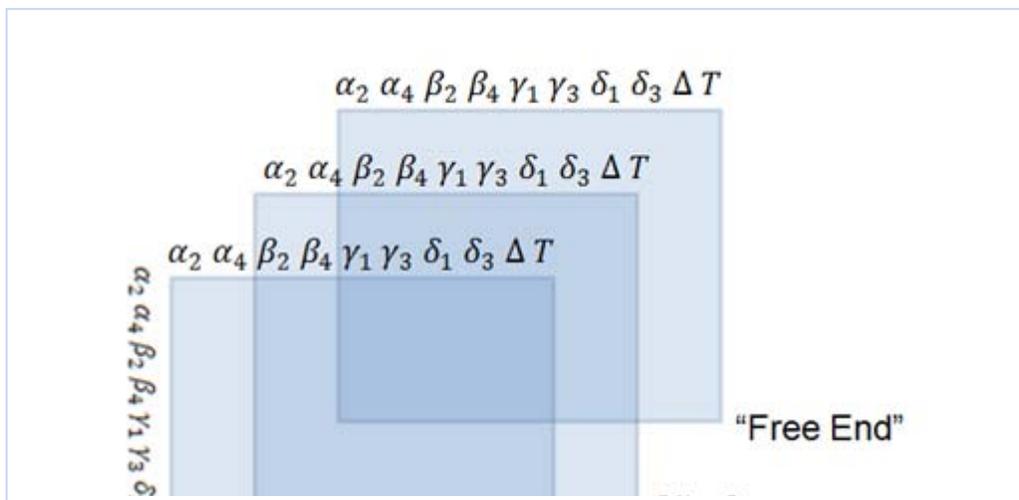


Figure 28: Binding site thermo lookup table output

4.3.3 Stochastic Simulation of DNA Hairpin Kinetics [30]

Recall that HCR is the process by which stable DNA monomers combine to form dendritic tree structures upon detection of a target DNA strand, exponentially amplifying the signal. The reactions involved in forming the dendritic tree structures can be modeled as a chemical kinetics system, with the set of DNA hairpins as reactants and tree structures as products. Chemical kinetics systems can be described mathematically by a set of ordinary differential equations whose values represent species concentrations and vary continuously and deterministically with time. For large systems with many species, discrete simulation of positions, velocities, and interactions between all molecules is too computationally expensive. As an alternative, stochastic simulations can provide exact solutions based on a few assumptions about the system. Using a MATLAB implementation of Gillespie's Algorithm, a stochastic simulation of DNA hairpin kinetics was developed as a virtual verification of the hairpin design process. Statistics such as number, average size, and composition of trees formed, along with information on noise caused by false-positive, spontaneous tree growth, can be used as an indication of the quality of hairpins designed. Further optimization of tree growth can be obtained at this stage by varying parameters in the simulation such as initial concentrations of various species, delay between addition of different species to solution, and the temperature of the environment. By making use of the simulation results as feedback, the hairpin design process can be fine-tuned to optimize the growth of dendritic tree structures by HCR.

4.3.3.1 Background for Stochastic Simulation Task

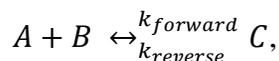
Designing sets of DNA hairpins that can be used during HCR is a difficult problem. The set of hairpins must reliably detect the presence of target DNA, triggering growth of the dendritic tree, while minimizing unintended interaction between hairpins in the absence of target DNA. This design process can be well optimized theoretically by imposing thermodynamic constraints, but this by itself does not necessarily optimize growth of the resulting dendritic trees.

Interactions in solution between DNA hairpins from a large set (nine hairpins in this case) are not well understood. Various factors inherent to chemically reacting systems involving DNA, including competitive hybridization, potential for cross-hybridization, species concentrations, and temperature dependence, have an effect on growth of dendritic trees during HCR. Creating an accurate simulation of DNA hairpin interaction as a chemical kinetics system is a crucial first step in developing an understanding of dendritic tree self-assembly. This simulation can then be leveraged to further optimize tree growth for false positive reduction and signal amplification, as well as to study the effects of varying experimental conditions.

The DNA hairpin HCR system can be first be generalized as a simple chemical kinetics system and then described mathematically. Description of the system has been partially adapted from Chapter 3 of [31], which examines the case of bound DNA probes and diffusing targets (gene chip or microarray).

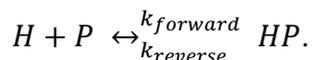
Chemical Kinetics

In general, chemical kinetics is the study of the rate at which reactant species combine to form products during reversible chemical reactions. A simple bimolecular chemical reaction between reactants A and B producing product C takes the form,

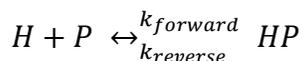


where $k_{forward}$ and $k_{reverse}$ are the forward and reverse reaction rates, respectively.

An interaction between two DNA hairpins can be generalized as a bimolecular reaction. Given unfolded hairpin H and folded hairpin P , the bimolecular reaction takes the form,



Interactions involving more than two DNA hairpins can be described as a series of consecutive bimolecular reactions. A schematic of this reaction can be seen in Figure 29. Given unfolded hairpin H and two folded hairpins P , the product HPP is formed by,



and,

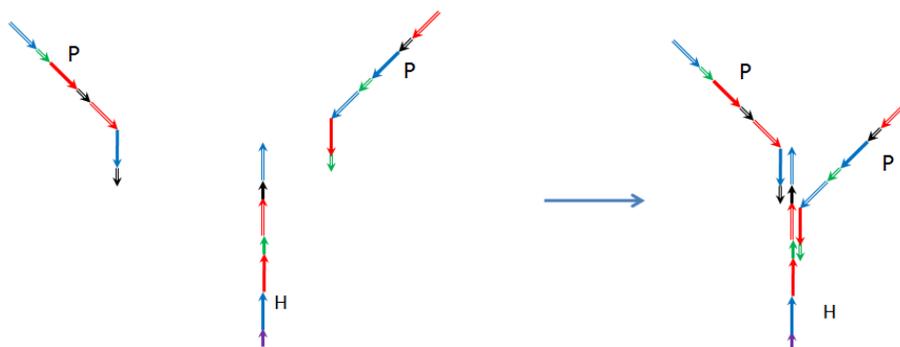
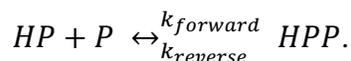


Figure 29: Consecutive bimolecular reactions

Each time a hairpin unfolds during exponential HCR, two free binding sites are exposed. Hairpin H has two free binding sites, H^1 and H^2 , with subsequences perfectly complementary to the single free binding sites on two folded hairpins in solution, P_1^1 and P_1^2 , respectively. Other folded hairpins in solution have free binding sites P_0^1 and P_0^2 that are not perfectly complementary.

We may observe hairpin H in one of nine possible states, represented in the form HP_i^k , where

$$i = \begin{cases} 1 & \text{for perfectly complementary hairpin subsequences} \\ 0 & \text{for unmatched hairpin subsequences} \end{cases} \quad (13)$$

and,

$$k \in \{1, 2\} \text{ is the binding site.} \quad (14)$$

Hairpin H is unbound

$$1. H \quad (15)$$

“Specific Hybridization” – Hairpin H is properly bound to at least one other hairpin at the proper binding site.

$$\begin{aligned} &2. HP_1^1P_1^2 \text{ – intended hybridization} \\ &3. HP_1^1P_0^2 \\ &4. HP_0^1P_1^2 \\ &5. HP_1^1 \\ &6. HP_1^2 \end{aligned} \quad (16)$$

“Non-specific Hybridization” – Hairpin H is improperly bound to at least one other hairpin.

$$\begin{aligned} &7. HP_0^1 \\ &8. HP_0^2 \\ &9. HP_0^1P_0^2 \end{aligned} \quad (17)$$

Transition between the unbound state and the eight bound states (products) can be described the following set of reversible chemical reactions, with forward and reverse reaction rates $k_{i,j}$ and $k_{j,i}$, respectively:



Although these reactions are completely reversible in theory, a truly reversible reaction does not exist in practice. For this reason, reaction rates $k_{1,2}$ and $k_{2,1}$, for example, are not equivalent and cannot be reduced.

Modeling and Simulating Chemical Reactions

Systems of chemical reactions can be modeled in several different ways. The first approach is to use the Chemical Master Equation (CME). The CME is a set of n linear Ordinary Differential Equations (ODEs), where n is the number of possible states for the system. The solution at time t to any differential equation in the CME set is the probability of the system being in that particular state at time t . The dimension of any ODE in the set is given by the total number of possible states for the system, and the number of states is dependent on the total number of molecules. [34] In the original example of a small system consisting of only one unfolded hairpin H with two free binding sites, two folded hairpins P and P , and the product HPP , there were nine possible states for the system, so any differential equation in the CME set would have dimension nine. In the DNA hairpin system to be modeled for this project, there are ten different reactants (nine hairpins and target DNA) combining to form products, and there are large concentrations of each. An estimate of the number of states for this system follows.

Hairpin H is unbound

$$1. H \quad (H[\text{none}][\text{none}]) \quad \binom{10}{1} = 10 \quad (19)$$

“Specific Hybridization” – Hairpin H is properly bound to at least one other hairpin at the proper binding site.

$$\begin{aligned} 2. HP_1^1 P_1^2 & \quad \text{Intended Hybridization} & \quad \binom{10}{1} * 1 * 1 = 10 \\ 3. HP_1^1 P_0^2 & \quad (HP_1^1[\text{any}]) & \quad \left(\binom{10}{1} * 1 * \binom{10}{1} \right) - \binom{10}{1} = 90 \\ 4. HP_0^1 P_1^2 & \quad (H[\text{any}]P_1^2) & \quad \left(\binom{10}{1} * \binom{10}{1} * 1 \right) - \binom{10}{1} = 90 \\ 5. HP_1^1 & \quad (HP_1^1[\text{none}]) & \quad \binom{10}{1} * 1 = 10 \\ 6. HP_1^2 & \quad (H[\text{none}]P_1^2) & \quad \binom{10}{1} * 1 = 10 \end{aligned} \quad (20)$$

“Non-specific Hybridization” – Hairpin H is improperly bound to at least one other hairpin.

$$\begin{aligned} 7. HP_0^1 & \quad (H[\text{any}][\text{none}]) & \quad \binom{10}{1} * \binom{10}{1} = 100 \\ 8. HP_0^2 & \quad (H[\text{none}][\text{any}]) & \quad \binom{10}{1} * \binom{10}{1} = 100 \\ 9. HP_0^1 P_0^2 & \quad (H[\text{any}][\text{any}]) & \quad \left(\binom{10}{1} * \binom{10}{1} * \binom{10}{1} \right) - 2 * \left(\binom{10}{1} * \binom{10}{1} \right) = 800 \end{aligned} \quad (21)$$

This sums to 1,220 possible states if each species has a concentration of 1. With hairpin concentrations possibly on the order of thousands, the number of possible states is extremely large, and only grows larger as the simulation progresses since the number of free binding sites increases exponentially. Modeling with the Chemical Master Equation is not feasible.

Another approach to modeling a system of chemical reactions is to use the Reaction Rate Equation (RRE), which is a set of differential equations derived from the law of mass action. This gives a set of equations that describe the rate of change of the concentrations of each species and the proportional rate of change of the products. The solution at time t to any differential equation in the RRE set is the concentration of a particular species at time t , meaning a separate equation is needed for each species in the system. [32]

For the small sample system consisting of only one unfolded hairpin H with two free binding sites, two folded hairpins P and P , and the product HPP , there are nine possible species. The Reaction Rate Equation has nine differential equations as follows.

$$\begin{aligned}
\frac{d[HP_1^1P_1^2]}{dt} &= k_{1,2}[H][P_1^1P_1^2] - k_{2,1}[HP_1^1P_1^2] \\
\frac{d[HP_1^1P_0^2]}{dt} &= k_{1,3}[H][P_1^1P_0^2] - k_{3,1}[HP_1^1P_0^2] \\
\frac{d[HP_0^1P_1^2]}{dt} &= k_{1,4}[H][P_0^1P_1^2] - k_{4,1}[HP_0^1P_1^2] \\
\frac{d[HP_1^1]}{dt} &= k_{1,5}[H][P_1^1] - k_{5,1}[HP_1^1] \\
\frac{d[HP_1^2]}{dt} &= k_{1,6}[H][P_1^2] - k_{6,1}[HP_1^2] \\
\frac{d[HP_0^1]}{dt} &= k_{1,7}[H][P_0^1] - k_{7,1}[HP_0^1] \\
\frac{d[HP_0^2]}{dt} &= k_{1,8}[H][P_0^2] - k_{8,1}[HP_0^2] \\
\frac{d[HP_0^1P_0^2]}{dt} &= k_{1,9}[H][P_0^1P_0^2] - k_{9,1}[HP_0^1P_0^2] \\
\frac{d[H]}{dt} &= k_{2,1}[HP_1^1P_1^2] + k_{3,1}[HP_1^1P_0^2] + k_{4,1}[HP_0^1P_1^2] + k_{5,1}[HP_1^1] + k_{6,1}[HP_1^2] \\
&\quad + k_{7,1}[HP_0^1] + k_{8,1}[HP_0^2] + k_{9,1}[HP_0^1P_0^2] - k_{1,2}[H][P_1^1P_1^2] \\
&\quad - k_{1,3}[H][P_1^1P_0^2] - k_{1,4}[H][P_0^1P_1^2] - k_{1,5}[H][P_1^1] - k_{1,6}[H][P_1^2] \\
&\quad - k_{1,7}[H][P_0^1] - k_{1,8}[H][P_0^2] - k_{1,9}[H][P_0^1P_0^2]
\end{aligned} \tag{22}$$

Like the Chemical Master Equation, the RRE becomes very large when applied to our whole DNA hairpin system and continues to grow as the simulation progresses. Although smaller than the CME, the RRE is still too large to be a feasible method for modeling the system.

A third method for modeling a system of chemical reactions is to use the Stochastic Simulation Algorithm (SSA), also known as Gillespie's Algorithm. This method allows us to compute indirectly with the Chemical Master Equation by stepping in time the reactions that change the state of the system. We can then solve only the differential equations relevant to a particular state rather than calculating the entire probability distribution for the system. [33] The basis for the SSA is the single major assumption that the system is well-stirred.

In a well-stirred system, there may be thousands of collisions between molecules at any given time, but the vast majority of those collisions are non-reactive. More often than not, molecules collide but do not react. Instead, nonreactive molecules bounce off of one another, randomly altering their velocities and trajectories. This helps to maintain a uniform distribution of the positions and velocities of the molecules in the system and allows us to ignore nonreactive collisions and worry only about those that change the state of the system. [33]

The state of the system at time t is defined by the concentration of each type of species in a vector $X(t)$. Each reactive collision R_j will change the species populations in a different way, described by a state-change vector v_j , and will occur after a certain length of time τ . The state of the system is then updated by $X(t + \tau) = x + v_j$. In other words the system starts in state x and jumps to state $x + v_j$ after one reaction. The reaction that occurs and the time at which it occurs are characterized by the overall probability of a reaction occurring within the system in a given time interval. [33] The Stochastic Simulation Algorithm and its implementation are well described in *Stochastic Simulation of Chemical Kinetics*. [33]

4.3.3.2 Method of Implementing the Stochastic Simulation Algorithm

Simulation of the DNA hairpin system was performed using a MATLAB implementation of the Stochastic Simulation Algorithm (hp_kinetics.m). An overview of the .m file can be seen in Figure 30.

Initial Conditions

At the beginning of the simulation, initial conditions are set, large data structures are constructed, and necessary data about the DNA hairpin set is loaded. The initial conditions of the system include the starting concentrations for each type of hairpin and the target DNA, temperature of the system, and the length of the delay time before addition of the target DNA to the system. By varying these parameters at the beginning of the simulation, the user is able to investigate the effect of each change on the dendritic tree structures that result.

hp_kinetics.m

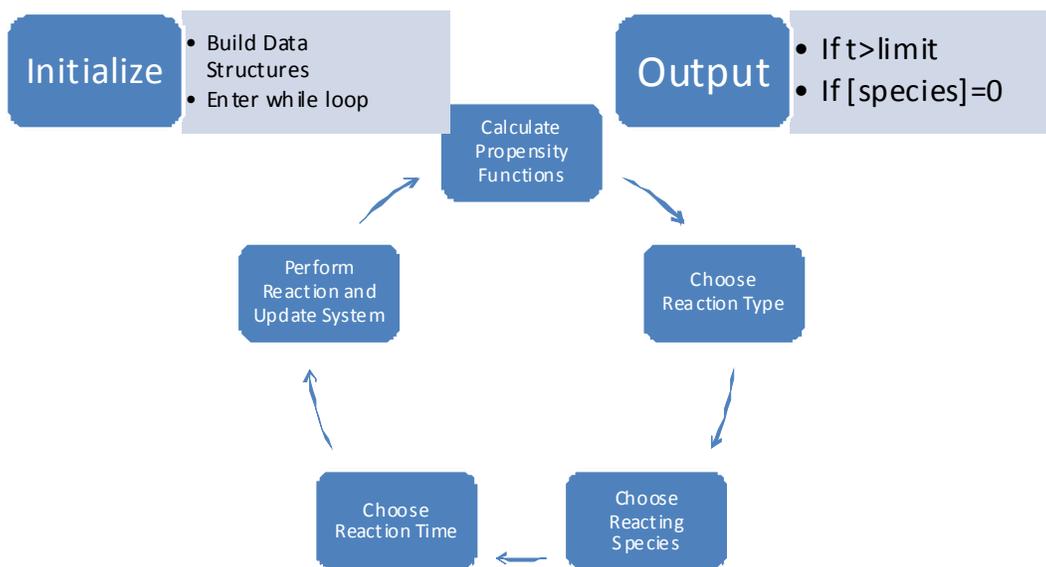


Figure 30: Overview of MATLAB Implementation

Data structures that are expected to be very large are initialized in advance for speed. These include a structure called “additions_to_struct”, used to track the composition of each tree and the order in which hairpins are added, as well a structure called “struct_fbs”, which tracks the free binding sites on each tree as the reaction progresses. The total number of free binding sites across all dendritic trees gives an indication at any given time of the likelihood that the next reaction to occur will be one in which an unbound hairpin adds to an existing tree.

Finally, enthalpy and entropy data in the form of a .csv file is loaded from the working directory. This data is generated at the end of the hairpin design process as a multidimensional combinatorial array. For every possible combination of hairpin to binding site, the enthalpy and entropy are calculated. It is assumed that a folded hairpin has a free binding site only at its “free end”, an unfolded hairpin has two free binding sites, and a target strand always has one free binding site at the “free end”. These configurations can be seen in Figure 31 and the data structure can be seen in Figure 32.

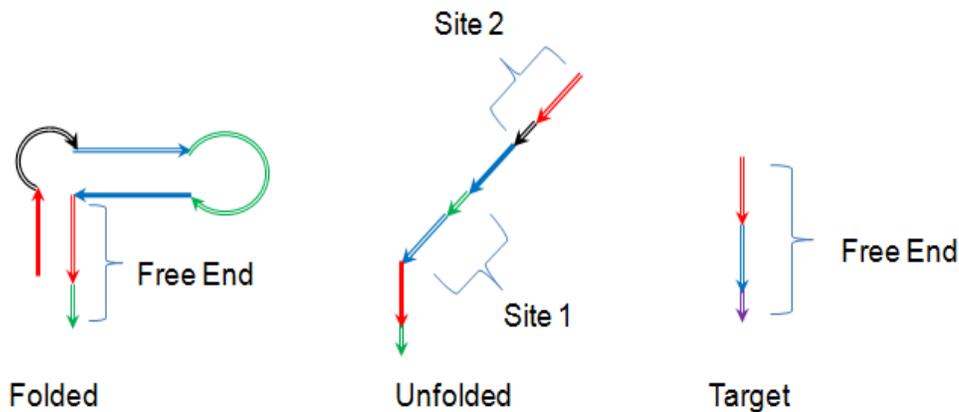


Figure 31: Assumed binding site configurations

Enthalpy and entropy can be used to calculate the Gibbs free energy of a particular conformation by the equation $\Delta G = \Delta H - T\Delta S$, where ΔH is enthalpy, T is temperature in Kelvin, and ΔS is entropy. The lower the free energy of a particular conformation, the stronger the bond is between the two strands. A perfect Watson-Crick hybridization will have very low free energy, whereas a cross-hybridization will have a higher free energy. For each possible combination, enthalpy and entropy are only calculated for perfect alignments of hybridizing strands. The enthalpies and entropies associated with shifting one strand relative to another can be ignored because a perfect alignment represents a “worst case scenario”. In other words, an unfavorable hybridization would only be made more unfavorable by an improper alignment.

Calculate Propensities and Choose Reaction Type

There are four different types of reactions that are assumed to be possible: a hairpin binding to another hairpin, a hairpin binding to a branch of an existing tree, dehybridization of an existing branch, and an “overtake”, where a hairpin takes the place of one that is weakly bound to a tree. For each type of reaction, every possible hairpin combination can occur, each with a certain probability based on free energies.

In order for a hairpin to unfold, it must bind to a free binding site on another hairpin. Hybridization will only occur, however, if the free energy of the new configuration is lower than the free energy of the hairpin in its stable folded state. For each type of reaction and for every possible combination, the difference between the free energy of the current state was subtracted from the free energy of the potential new state. If this difference is very negative, the hybridization is very likely to occur (or if already hybridized, very likely to remain hybridized), and if the difference is very positive, the hybridization is unlikely (or if already hybridized, very likely to dehybridize).

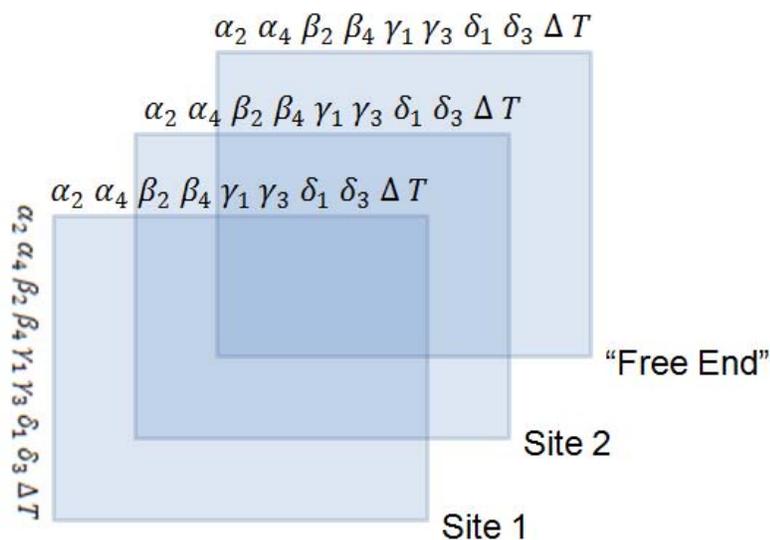


Figure 32: Combinatorial enthalpy and entropy array

The melting temperature for a hybridized pair of strands is the temperature at which there is a 50% chance that the pair would remain hybridized, and a 50% chance that the pair would dehybridize. This occurs when the free energy of a particular combination equals 0. [34]

Since we know that the probability of hybridization is 50% when $\Delta G = 0$, we can apply a Gaussian distribution centered around 0 with an amplitude of 0.5 and standard deviation of 1. This value will be the probability of hybridization as a function of ΔG . The Gaussian distribution takes the form,

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad (23)$$

where a is the amplitude, b is the position of the center, and c is the width of the “bell.”

The histogram in Figure 33 shows the distribution of free energies of all possible “hairpin free end” to “hairpin free end” combinations. As one would hope, most hairpin:hairpin interactions have a very high free energy, and only two combinations (Target: Δ and Δ :Target) have a very low free energy. This means that there should be very little interaction between most hairpins, and strong interaction between the target DNA and the Δ hairpin. The Gaussian probability distribution is overlaid to show the number of combinations that have a particular probability of hybridization. Since most combinations have a high free energy, most combinations have less than a 1% chance of hybridization. The Target: Δ and Δ :Target combinations that have a very low free energy have a greater than 99% chance of hybridization.

Once every possible combination for each type of reaction has been assigned a probability, each probability is compared against a random number to decide whether or not the reaction will occur at this particular iteration. All combinations that “pass” are still eligible to be the next reaction that occurs within the system, so now it is time to choose among them.

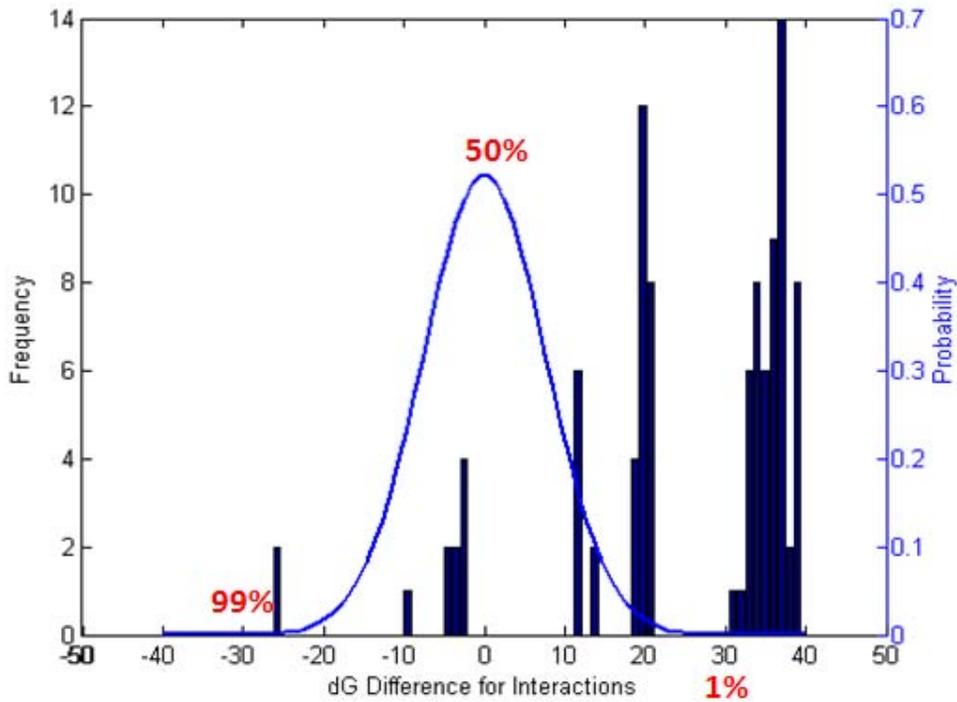


Figure 33: Distribution of free energies and probabilities for hairpin:hairpin combinations

Each passing combination is given a weight or propensity based on its probability of hybridization and the concentrations of each species. If the system is in state x , the propensity for reaction $R_{i,j}$ is given by the function,

$$a_{i,j}(x) = c_{i,j}x_i x_j \quad (24)$$

for a reaction involving two different species, and

$$a_{i,j}(x) = c_{i,j} \frac{1}{2} x_i (x_i - 1) \quad (25)$$

for a reaction involving two of the same species, where c is the probability and x_i and x_j are the concentrations of species i and species j , respectively. [33]

Propensities for all combinations of all four types are stored in four different arrays: $a_{hp/hp}$, $a_{hp/tree}$, a_{dehyb} , and $a_{overtake}$. The average value from each array is calculated and stored in a vector, then a weighted random selection is performed to choose the reaction type. This array has the following form, where n is the number of values in the array:

$$reaction\ type = \left[\frac{1}{n} \sum_{i=1}^n a_{hp/hp} \quad \frac{1}{n} \sum_{i=1}^n a_{hp/tree} \quad \frac{1}{n} \sum_{i=1}^n a_{dehyb} \quad \frac{1}{n} \sum_{i=1}^n a_{overtake} \right] \quad (26)$$

As the system evolves over time and species concentrations change, the most favorable type of reaction also changes. At the beginning of the simulation when all hairpins have been added to solution, target DNA is present, and no trees have formed, hairpin to hairpin binding will be strongly favored as the most favorable reaction. At the end of the simulation when there are many trees formed in solution and the concentration of unbound hairpins is near zero, dehybridization or “overtake” will be most favorable.

Choose Reacting Species and Reaction Time

Once the reaction type has been chosen, the reacting species must be selected. Using the propensity array for the chosen reaction type, weighted random selection is performed to select an entry. Since the propensity array is a combinatorial matrix $([\alpha_2 \alpha_4 \beta_2 \beta_4 \gamma_1 \gamma_3 \delta_1 \delta_3 \Delta T] \times [\alpha_2 \alpha_4 \beta_2 \beta_4 \gamma_1 \gamma_3 \delta_1 \delta_3 \Delta T])$, the indices of the selected entry are the two reacting species. For hairpin to hairpin binding, for example, a weighted random selection from values in $a_{hp/hp}$ may be the entry $a_{hp/hp}(2,4)$, indicating that Hairpin α_4 binding to Hairpin β_4 will be the next reaction to occur.

Hairpin to hairpin binding is the most straightforward reaction, since only the two reacting species must be selected. For the other three reaction types, however, the tree on which the reaction will occur must also be selected. In these cases, we must look through the trees that have formed and again use random weighted selection to choose one. For the reaction $a_{hp/tree}(2,4)$, we randomly choose among all trees that have a β_4 branch with an open binding site to which an α_4 hairpin can attach. Similarly, for the reaction $a_{dehyb}(2,4)$, we randomly choose among all trees that have an α_4 hairpin that could dehybridize from a β_4 branch.

The time at which this next reaction occurs is determined by an exponentially distributed random variable τ , using the formula,

$$\tau = \frac{1}{a_0(x)} \ln \frac{1}{r}, \quad (27)$$

where $a_0(x) = \sum a(x)$ and r is a normally distributed random number. [33] At the beginning of the simulation when interactions between molecules are highly probable, the reaction occurs very quickly and therefore the time step calculated by τ will be very small. At the end of the simulation when most of the reactants have been used up, interactions between molecules are less likely to occur so the time step increases. This relationship can be seen in Figure 34, which shows the length of the time step as a function of time for an actual simulation run.

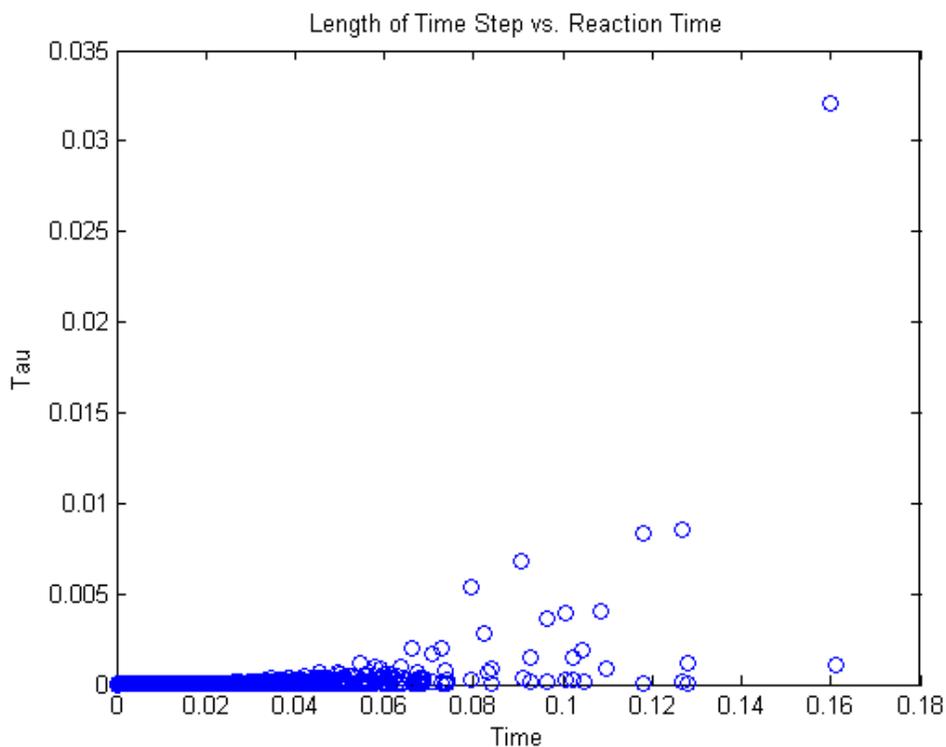


Figure 34: Length of time step vs. simulation time

Perform Reaction and Update System

The final step in the simulation is to actually perform the reaction and update the appropriate data structures to reflect the new system state. Since the goal of this simulation is to better understand the growth of dendritic tree structures, the most important system characteristics to track are the size, composition, and total number of trees. As mentioned previously, a data structure called “additions_to_struct” is used to keep track of all tree structures that have formed. This data structure has size 4 rows x j columns x k layers, where j is the number of additions to any tree, and k is the total number of trees that have formed.

Four rows are required to fully describe the state of a tree k at any time that its structure changes, with a new column for each addition to the tree. The entries in rows 1 and 2 refer to the two reacting species (row 2 is the hairpin that was added to the tree, and row 1 is the hairpin (branch) to which it was added), row 3 refers to the binding site where the addition occurred, and row 4 is used to describe whether or not a particular branch is a “terminal” branch or an interior branch. This fourth row becomes important when tracking dehybridizations because dehybridization at an interior branch will result in a cleaved section that becomes a separate tree ($tree\ total = k + 1$). Dehybridization at a terminal branch, on the other hand, means that only one hairpin will be removed from a tree, leaving the total number of trees unchanged. Figure 35 shows two examples of small tree structures that may form during the HCR reaction and will be used to demonstrate the way the data structure is filled.

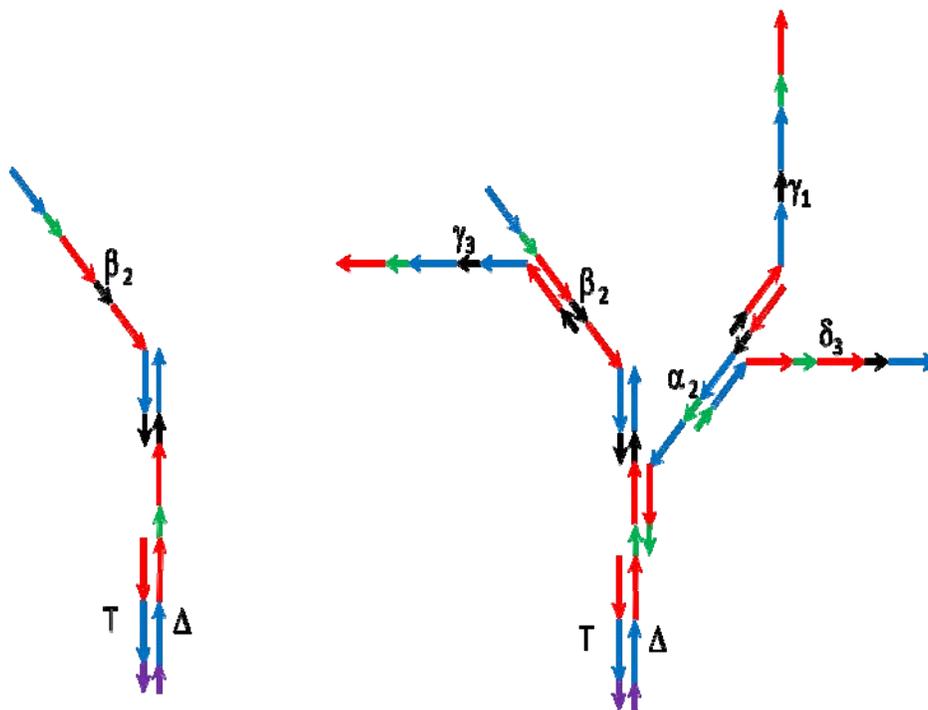


Figure 35: Two example trees formed

If the small tree on the left is tree number k_1 , then the k_1^{th} layer of additions_to_struct would read,

$$\begin{bmatrix} 10 & 9 \\ 9 & 2 \\ 3 & 2 \\ 0 & 1 \end{bmatrix}.$$

By using this syntax, we know that $\Delta(9)$ bound to the target (10) at the target's site 3, then β_2 bound to $\Delta(9)$ at Δ 's site 2. We also know that the β_2 branch is a terminal branch. Similarly, if the small tree on the right is tree number k_2 , then the k_2^{th} layer of additions_to_struct would read,

$$\begin{bmatrix} 10 & 9 & 1 & 1 & 9 & 2 \\ 9 & 1 & 8 & 5 & 2 & 6 \\ 3 & 1 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Using this format to track dendritic trees is very convenient because it allows for simple manipulation to simulate various types of dehybridizations and also allows for straightforward analysis of the size of all trees by looking at the lengths of each vector.

4.3.3.3 Results of Simulation Validation

Various aspects of the simulation can be investigated separately to ensure that the result obtained agrees with the result expected. The next few subsections of the report will provide an overview of the early validation work.

Exponential Increase in Binding Sites

The most important validation to perform is to show that there is in fact exponential signal amplification as the reaction progresses. To do this, one can keep track of the total number of free binding sites available at any point during the simulation. The total number of free binding sites across all trees is an indication of the amplification of the signal because each free binding site can accept one more hairpin with a fluorescent particle or gold nanoparticle that would be visible to the naked eye.

The simulation was run with uniform concentrations of 500 hairpins of each species at 20 deg. C. Figure 36 shows a plot of the total number of free binding sites as a function of reaction time. From Figure 36 we can see that the total number of free binding sites grows quickly and levels off as the reaction progresses, but it does not look like the exponential curve expected. Recall that the time step τ is an exponentially distributed number, so a semi-log plot should show the exponential behavior in a form that is familiar to us (Figure 37). The small tail at the end of the exponential appears as all reactant species near zero, so the number of free binding sites eventually begins to level off. Finally, to verify that this curve is in fact an exponential function, a log-log plot is shown in Figure 38. Since the curve is linear on a log-log plot, the increase in free binding sites during the process is exponential.

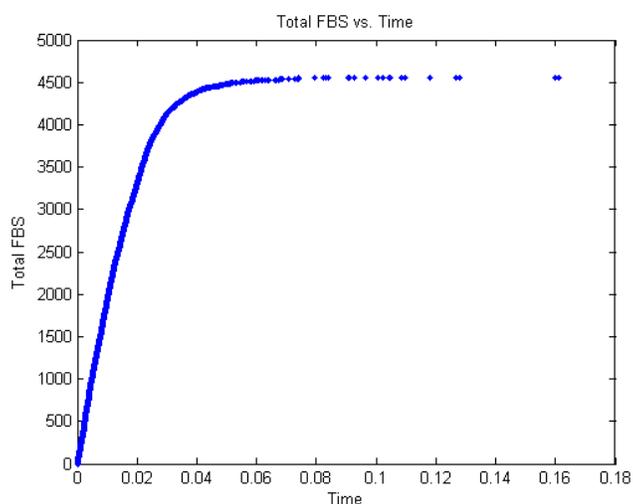


Figure 36: Free binding sites as a function of reaction time

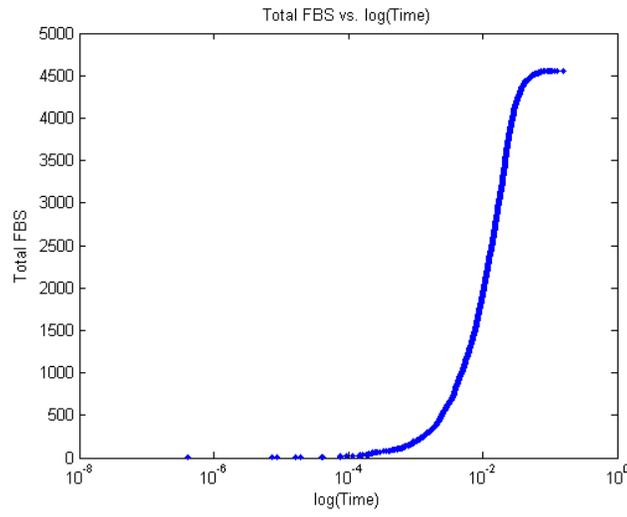


Figure 37: Semi-log plot

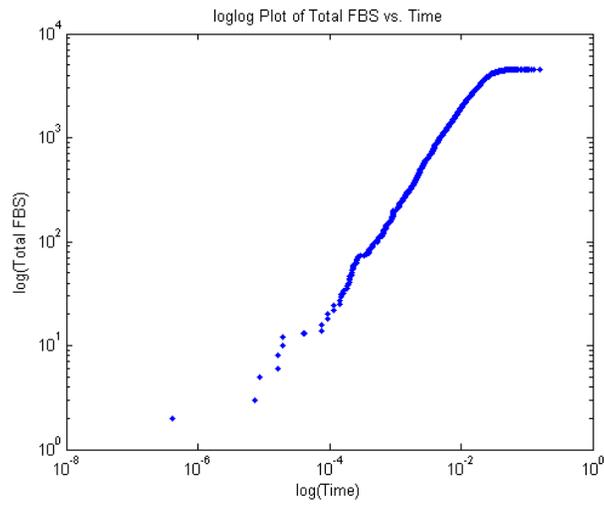


Figure 38: log-log plot

Average Tree Size vs. Target Concentration

The average molecular weight of HCR products should vary inversely with target concentration. [19] For our system, the HCR products are the dendritic trees and the target DNA acts to initiate the reaction. An inverse relationship is observed between the initial concentration of target DNA and the average size of dendritic trees formed.

Figure 39 shows the change in average and maximum size of dendritic trees as the total number of Target strands in the system is varied. As expected, as the total number of Target strands increases, the maximum and average size of dendritic trees decreases. These results were obtained using a uniform concentration of 250 hairpins of each type and were averaged over three runs.

Although not necessarily a means of validating the simulation, another interesting result was obtained while investigating the relationship between tree size and target concentration. While the primary focus is on how this relationship effects the end result of an HCR process, the average size of the trees throughout the process is worthy of mention. Figure 40 shows the average size of trees as a function of time for varying concentrations of Target DNA.

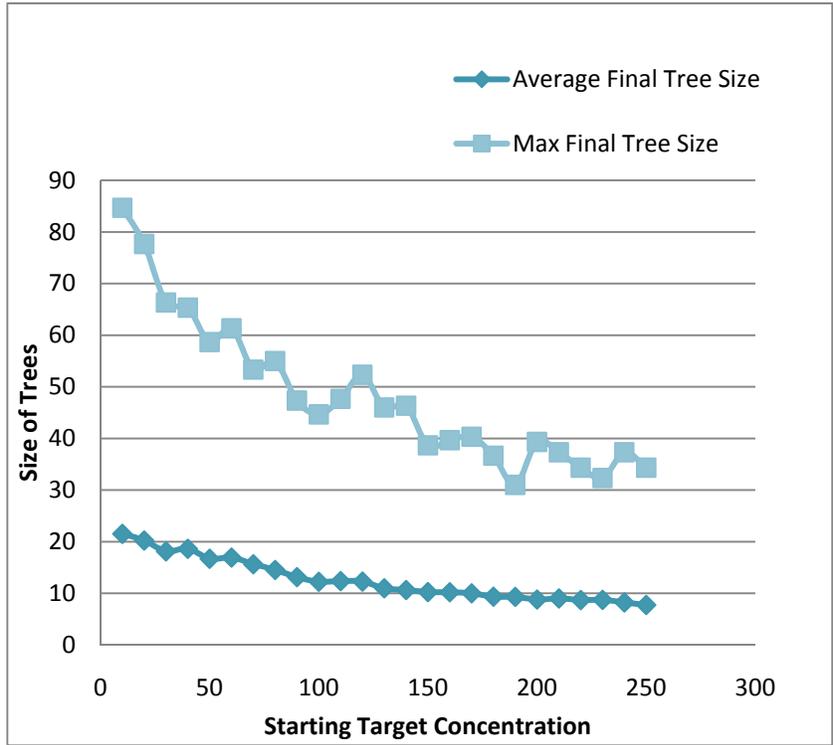


Figure 39: Tree size vs. target concentration

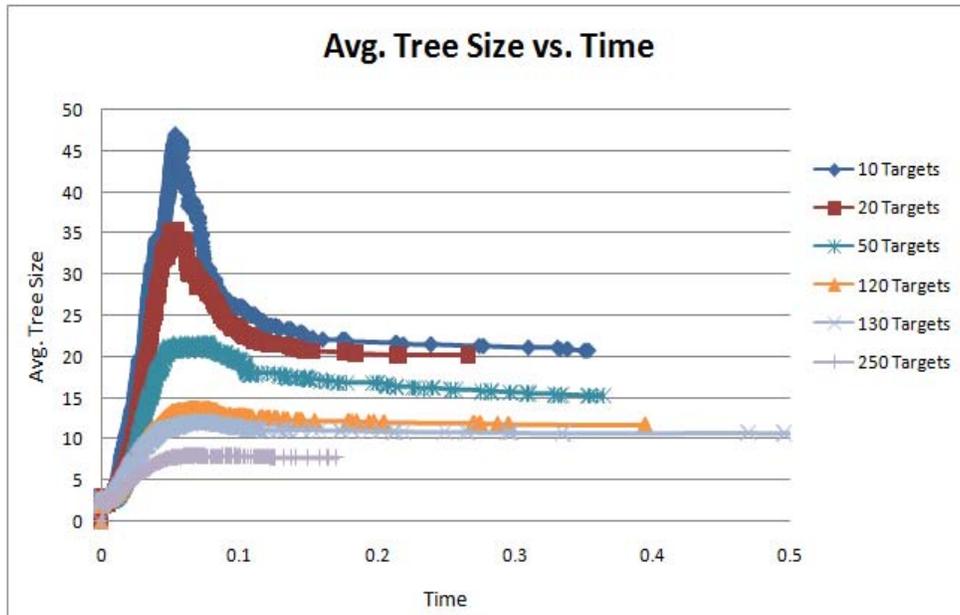


Figure 40: Average tree size vs. time

This result shows the effect of dehybridization reactions on growing trees. Occasional dehybridization causes very large trees to break apart into several smaller trees, quickly reducing the average size. For relatively low concentrations of Target DNA, a small number of very large trees form and then break apart. Dehybridization reactions play a major role in the outcome. For high concentrations of Target DNA, a large number of small trees form and continue to grow.

Comparison with Experimental Results

In *Triggered Amplification by Hybridization Chain Reaction* [19], linear HCR was verified experimentally using two DNA hairpins and a strand of target DNA. They performed three tests to verify that HCR was actually occurring during the reaction observed using hairpins H_1 and H_2 , along with target strand I. The results of the tests can be seen in Figure 41.

In the first test, equal concentrations of both hairpin species and a lower concentration of target produced the red curve shown in Figure 41. Before adding the target at $t=2000s$, no reaction occurs. After adding the target, the reaction occurs quickly and fluorescence quickly levels off.

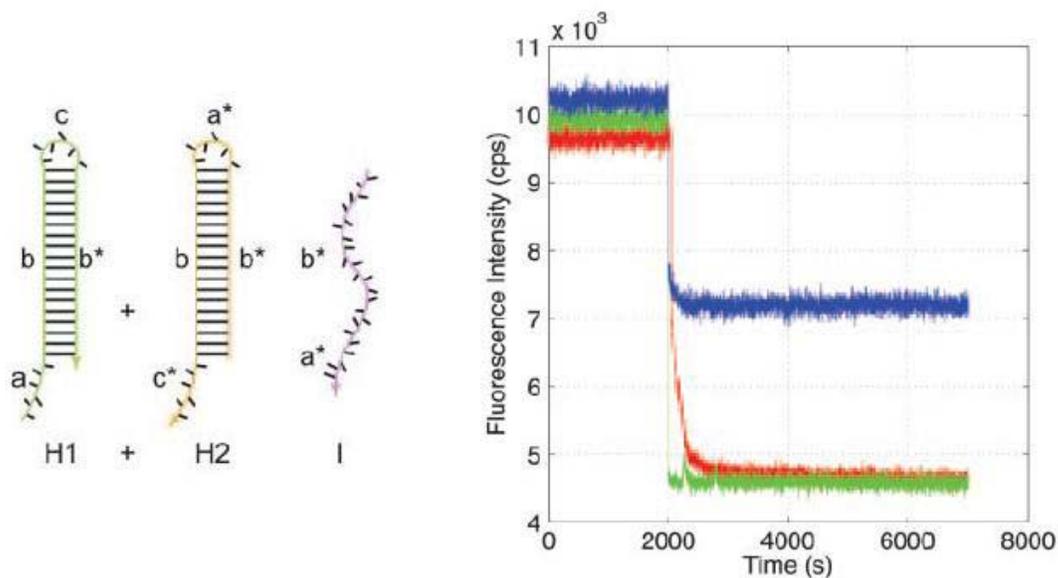


Figure 41: Testing by Dirks and Pierce [19], 2004 Copyright by The National Academy of Sciences of the USA

In the second test, only the H_1 strand was used, and the target was added in excess. The same fluorescence is observed (green curve). In the final test, only the H_1 strand was used, and the target was added in the same concentration as test 1. This time the H_1 strand was only partially depleted (blue curve), indicating that HCR must be responsible for completely depleting the H_1 strand, not just hybridization with the target strand. [19]

A similar set of tests was performed using the `hp_kinetics` simulation. Although Dirks and Pierce measured fluorescence to examine the behavior of the system, the system behavior for this project can be evaluated using change in concentration of tree structures since these two measures are related. In the first test, equal concentrations of α_2 and Δ hairpins were added (250 in number), along with 50 target strands. The result is shown in Figure 42 as a baseline. Note that the total number of tree structures continues to increase long after depletion of the target strands.

In the second test (Figure 43), the Δ hairpin was added at a concentration of 250 molecules, then the system was flooded with target strands (1000 molecules). Since the target strands are in such large excess and the Δ hairpins are depleted long before the target strands, we can infer that the depletion of the Δ hairpins was caused entirely by binding with target strands. The total number of structures continues to rise until the target strands are finally depleted.

In the final test (Figure 44), the Δ hairpin was added at a concentration of 250 molecules and the target strands were added at a concentration of 50 molecules. In this case, the total number of structures climbs only until the target strands are depleted, then levels off. Recall that in the first test, the total number of structures continued to climb even after depletion of the target strands. This suggests that HCR must be responsible for continued tree growth after depletion of the target strands.

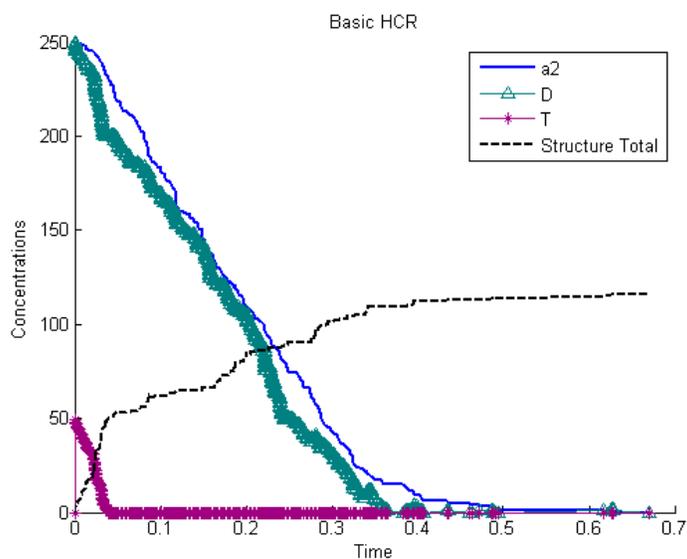


Figure 42: Baseline results

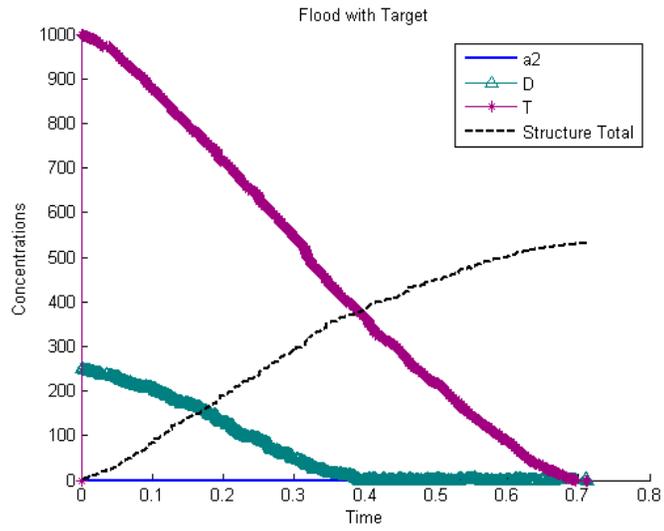


Figure 43: Delta hairpin with large number of target strands

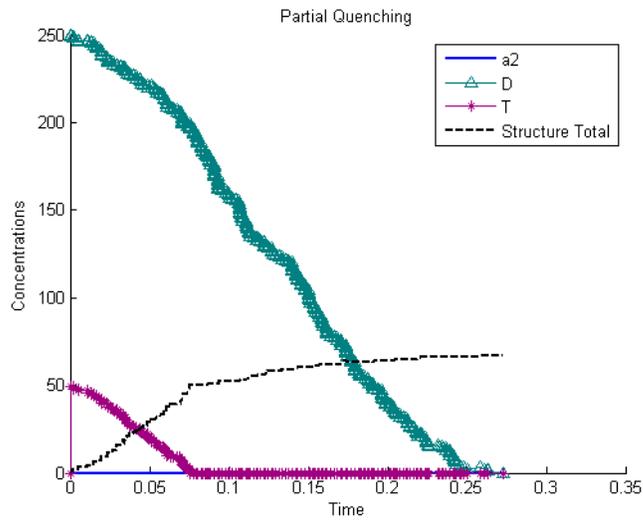


Figure 44: Delta hairpin with small number of target strands

Delayed Addition of Target DNA to System

Since the goal of the exponential HCR system is to provide maximal signal amplification with minimal false positives, the most important result to consider is the effect of delaying the addition of target DNA to the system. When there is no target DNA present in the system, a good hairpin set will have little to no interaction between hairpins, but when the target DNA is added, the reaction should proceed quickly.

Figure 45 shows the result of a simulation in which uniform concentrations of 500 hairpins of each species were added to the solution at the same time (20 °C). The reaction does proceed quickly, as expected, but there is no way to tell from this figure whether or not tree growth was in fact triggered by the presence of target DNA, or whether the trees are false positives.

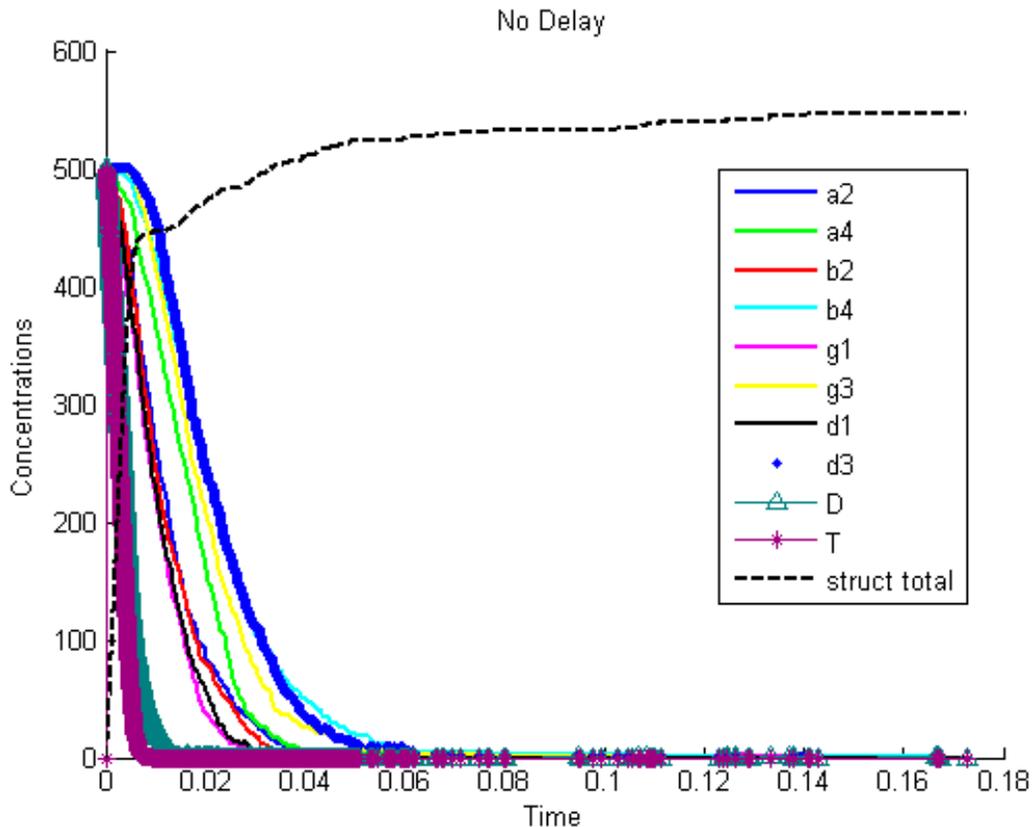


Figure 45: No delay in addition of target DNA strands to the system

Figure 46 shows the result of a simulation with uniform concentrations of 500 hairpins of each species except the target DNA. The target was added after $t=0.04$, at which time the reaction proceeds quickly. Prior to addition of the target, no reaction should take place. This means all tree formation was triggered by the addition of the target DNA, not by unintended hybridization.

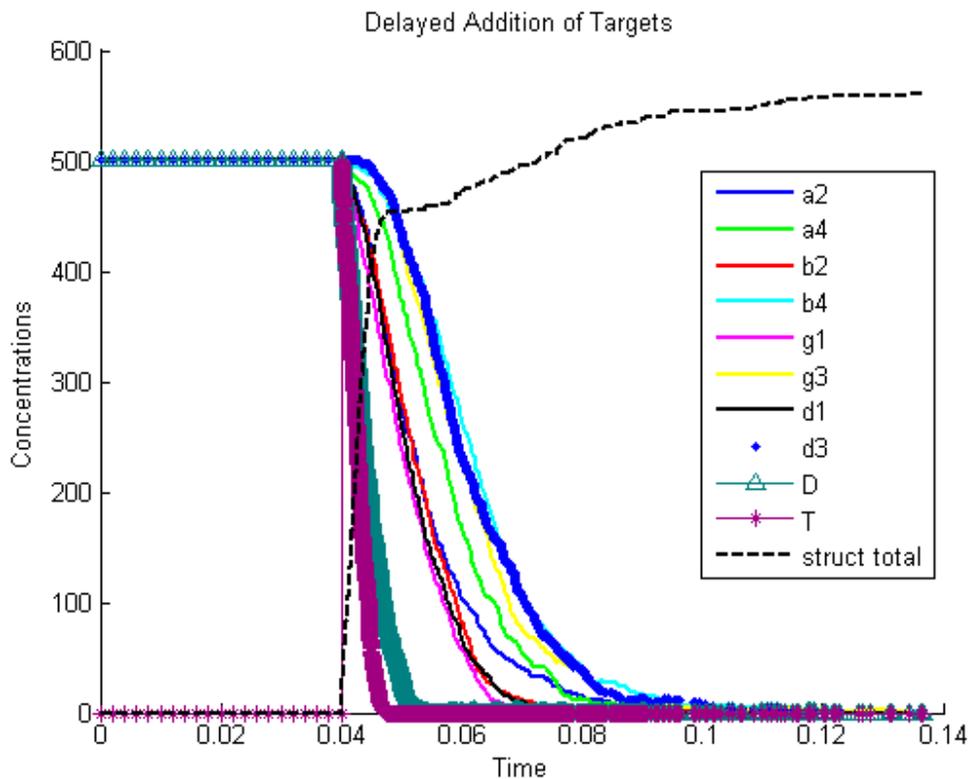


Figure 46: Delayed addition of target DNA to the system

Hairpin Sets with Varying Constraints

As mentioned previously, an important application of the hairpin kinetics simulation is to act as feedback for the hairpin set design process. The simulation developed is capable of taking enthalpy and entropy information from a hairpin set, loading the necessary information, and beginning the reaction. In order to demonstrate the effect of hairpin design, two simulation runs were performed. For the first simulation, a set of hairpins was designed with very stringent thermodynamic constraints and should perform very well. There should be no interaction between hairpins before the target DNA is added, and the number of structures should increase quickly then level off as the average tree size increases. The second simulation was performed using a hairpin set that was purposely designed to perform poorly, with a weak A:A* neck and a strong B:B* neck. This result should be significantly different from the first. The results of the first simulation can be seen in Figure 45 and the result of the second simulation can be seen in Figure 47.

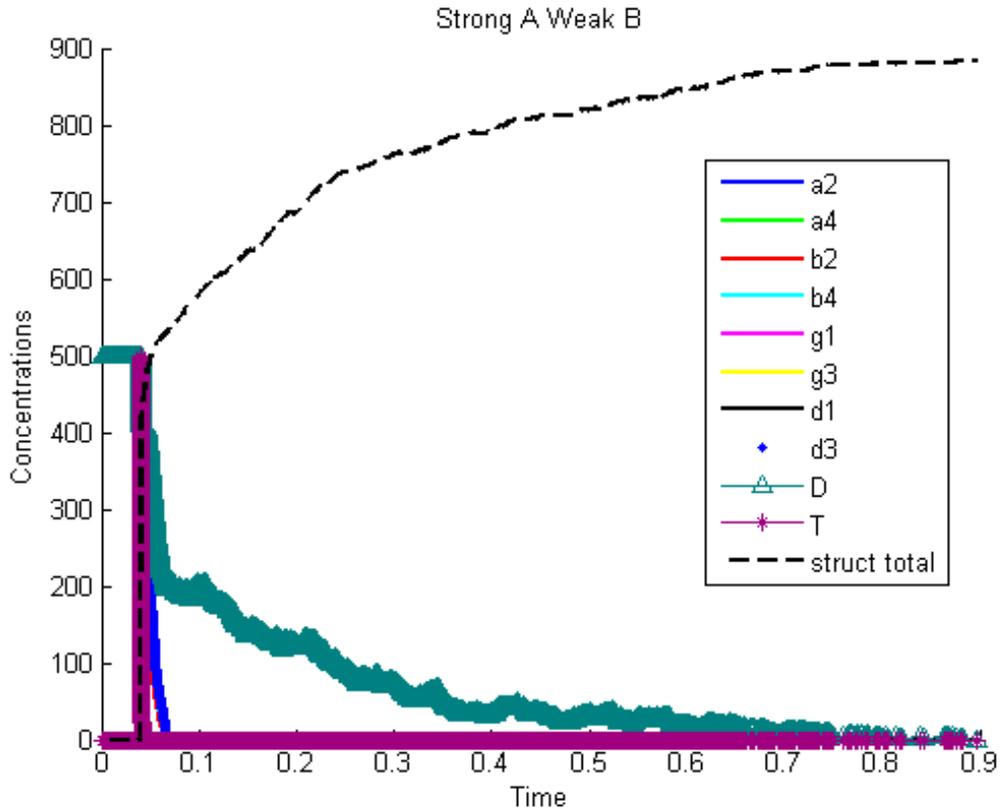


Figure 47: Experimental hairpin set designed to work poorly

In Figure 45, one can see that the hairpin set performed well. There was no interaction between hairpins before the target DNA was added, and the structure total increases quickly then begins to level off as soon as the target strands are depleted. When the total number of structures levels off, this indicates that all subsequent hybridizations occur as hairpins adding to existing structures rather than forming new structures. Additionally, since the “trunks” of the trees should begin with a target strand, the total number of structures should be relatively close to the initial concentration of target strands.

In contrast, we see in Figure 47 that this hairpin set did not perform as well. While there was no interaction between hairpins before the addition of the target strands, the rest of the reaction did not proceed as desired. The Δ hairpin is depleted very slowly, which reflects the discrepancy between A:A* strength and B:B* strength. Since the target strand has sections T*A*B* and the Δ hairpin has binding site BAT, they should still hybridize frequently due to the strength of the A:A* hybridization. Other hairpins with A sections in their binding sites, however, will also hybridize strongly with the target strand. As soon as all target strands have been depleted, hybridization with the Δ hairpin is thermodynamically unfavorable, while hybridization with all other hairpins remains favorable. Further, by zooming in on Figure 47 at the beginning of the reaction, we can see that the $\alpha_2, \alpha_4, \gamma_1$ and γ_3 hairpins, with A:A* necks, are depleted more

quickly than the $\beta_2, \beta_4, \delta_1,$ and δ_3 hairpins, with B:B* necks, as expected. This detail can be seen in Figure 48.

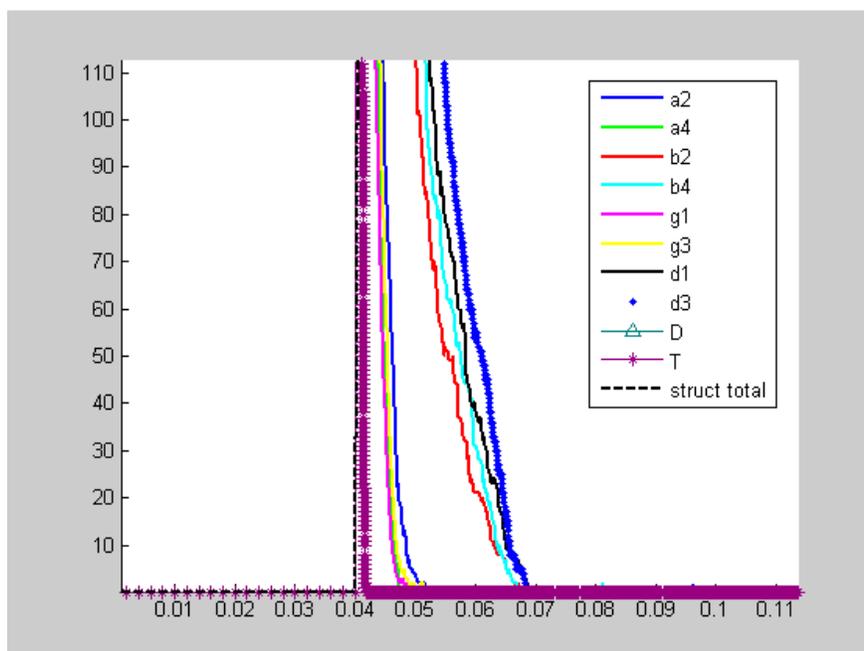


Figure 48: Close-up showing detail of depleting hairpins

4.3.4 Modeling Strand Hybridization [35]

Currently, the graphical representation of a dendritic tree, made up of DNA hairpins ‘unzipping’ due to HCR, as shown by Figure 49, is only created manually and in two dimensions. This is not an accurate representation of either the actual structure of DNA or the dendritic tree. It does, however, provide a good starting point. This task focused on establishing a foundation for creating a realistic three-dimensional representation of dendritic structure formation that will allow researchers to see how proposed hairpins will interact with each other when combined in solution prior to actually conducting an experiment. This representation will allow for virtual verification of the structure of the tree as well as allow non-experts to better understand its creation. This work will increase the odds of obtaining experimental success and better exploit HCR. The ultimate vision for the computer-based tool being pursued in this task is to automatically, rather than manually, generate a three dimensional rendering of DNA as it forms into a dendritic tree. Before the creation of a three-dimensional tree is possible, however, the structure of DNA in three dimensions must first be determined and rendered.

4.3.4.1 Method of Approach

The structure of DNA was determined through research, comparison of resources, and verification through mathematical calculations. After the initial research, a program was written in MATLAB to render the determined coordinates for the nucleotide bases, as well as the backbone. From there, an estimated single strand of DNA was rendered and followed by a more accurate model of double stranded DNA. The program created in MATLAB was transferred into Visual Studio, using C++ and OpenGL for cross-platform possibilities as well as being a better choice for graphical representation. Once that was done the bending of DNA was researched along with the structure of hairpins and single stranded DNA.

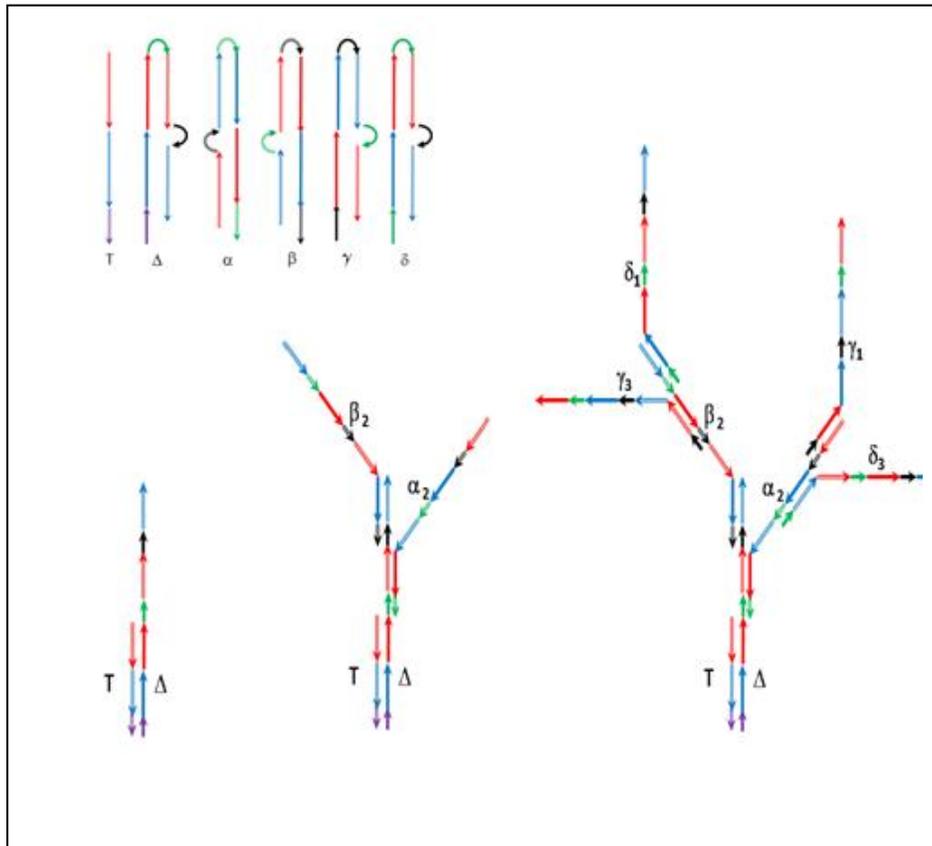


Figure 49: A progressive two-dimensional manual build of dendritic tree formed from the hairpins depicted in the upper left hand corner of the figure

4.3.4.2 Basic DNA Structure

The basic structure of DNA consists of the four nucleotide bases adenine, cytosine, guanine, and thymine, as well as the deoxyribose sugar and the phosphodiester backbone linkage. The sugar and backbone were previously referred to as Sugar-Phosphate backbone and for this report will be referred to as such, or as the backbone.

The structure of each base is made up of atoms, which include carbon, nitrogen, and oxygen, bonded together. Base-pair structure consists of two bases, one a purine while the other a pyrimidine, bonded together through hydrogen bonds. The adenine and thymine base-pair contains two such bonds, while the guanine and cytosine base-pair contains three such bonds. The base-pairs contain a center to which the bases rotate about to form a double helix.

The double helical structure of DNA consists of two strands that run in opposite directions. The strand running in the 5'-3' direction, ends in a terminal hydroxyl, while the strand running in the 3'-5' direction ends in a terminal phosphate group. It should be noted that 5' and 3' represent the standard convention by which to refer to the two distinct ends of the structure of a polynucleotide. [36] The structure also consists of base-pair parameters, which determine orientation of the double helix and will be discussed in more detail later. Secondary structures of DNA, often referred to as hairpins, consist of a combination single-stranded and double-stranded sections. The single-stranded sections make up what is known as the loop of the hairpin.

Bases

For the structure of each base, two sets of Cartesian coordinates [37, 38] along with bond lengths and bond angles [39] were discovered. Each set of coordinates varied from the other in more than one aspect. Due to the variance in the coordinates, each set was graphed and compared to the bond lengths and angles found in [39] and found to be correct in that aspect, falling into the standard deviations. One of the sets of coordinates [38], however, remained in the XY-plane with the Z-axis as the vertical axis, while the other set [37] did not remain in the XY-plane and did not use the Z-axis as the vertical axis. Also, a rotation such as propeller twist, discussed later, seemed to be included in the second set. So, the first set [38] was chosen due to remaining in the XY-plane, the use of the Z-axis as the vertical axis, and the exclusion of the rotation found in the second set. An overhead view of the chosen set of coordinates can be seen in Figure 50.

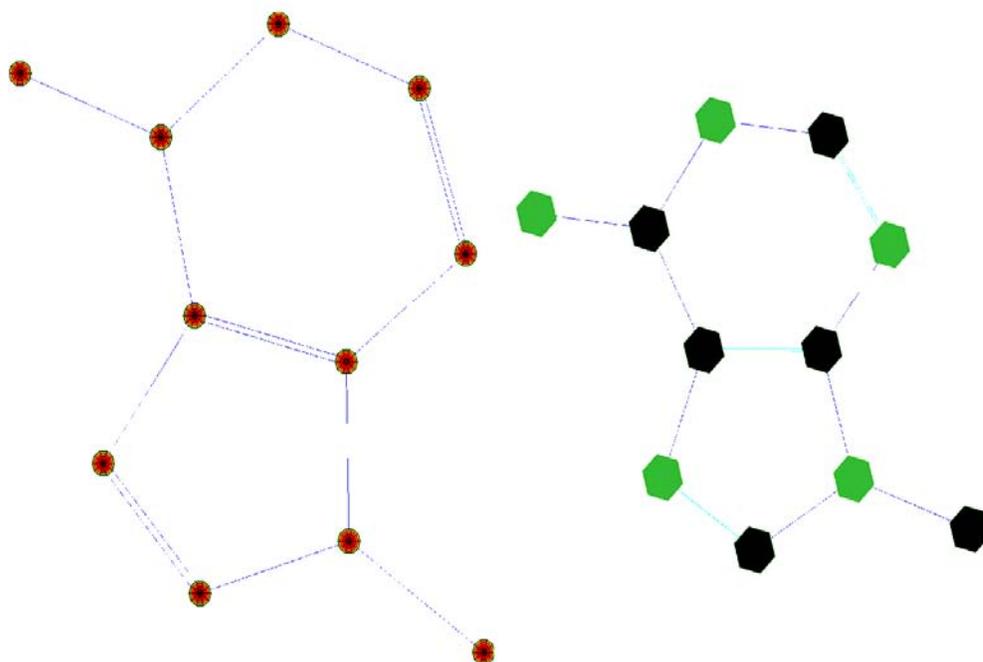


Figure 50: Adenine graphed using MATLAB (left) and OpenGL (right)

Base-pair

After determining which coordinates to use for the bases, the geometry for base-pairs [36, 38] was determined and calculated. The set of coordinates chosen, referred to in [38] allowed for just a reflection across the Y-axis to align the bases and provide the correct distances and angles to form base-pairs according to [38] and are shown in Figure 51. The reflected base is used to form the 3'-5' strand while the original, untransformed base is used to form the 5'-3' strand. The lower bold line represents the angle at which the bases are aligned with each other. The center of the upper bold line represents the center of the helix. The line is drawn from the C6 in pyrimidines to the C8 in purines. The two more fine lines represent the distances between the two bases. The atoms connected by the fine lines are bonded with each other through hydrogen bonds. The base-pair represented only has two bonds while the guanine and cytosine base-pair has three hydrogen bonds.

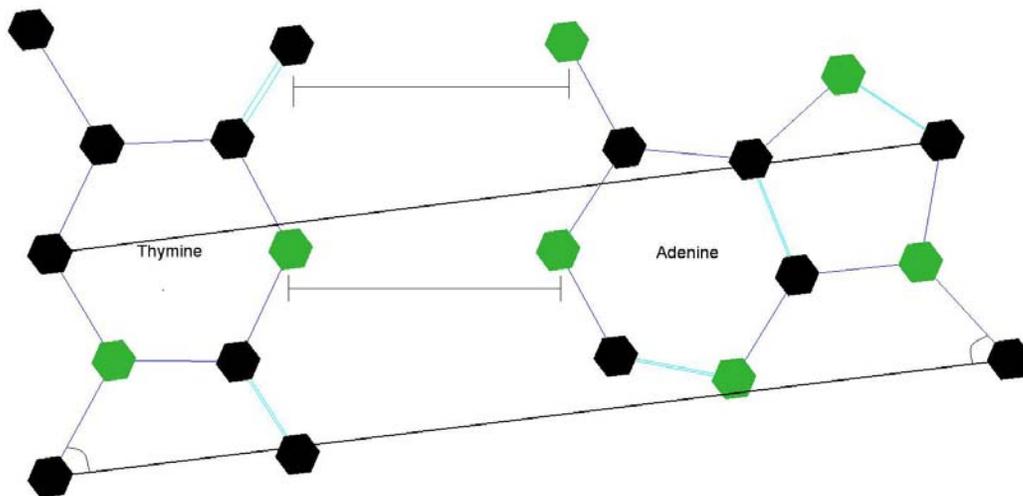


Figure 51: An overhead view of a thymine and adenine base-pair in OpenGL

Hydrogen Bond Length [40]

Early in this task the question came up regarding the appropriate values to use for the hydrogen bond lengths. Such lengths are needed for the modeling and simulation of self-assembled structures of DNA strands into dendritic structures. Even though, when originally asked, the question seemed quite simple, the simplicity was lost as the issue was examined more closely. It was assumed that with all of the previous research activity around DNA that the geometry of the double helix structure was well understood and well defined, which was far from the truth. In fact, when reviewing one of the literature search results, it was found that others have encountered this same dilemma. One article in particular clearly stated up front that, “The apparent simplicity of double-helical DNA, the icon of molecular biology, is deceiving.” [41] Research into the appropriate value for hydrogen bond lengths reveals a discrepancy of values that requires an understanding of when and under which circumstances certain values are to be used. The study for this task revolved around a literature search of published research regarding the bond length of the H and N molecules between the DNA base pairs adenine-thymine, and cytosine-guanine. The literature search process evolved into two phases, an initial search and a refined search. Both searches examined published works in commercial databases, the Defense Technical Information Center database and Google Scholar. After the articles and reports obtained from the literature searches were reviewed, the findings were discussed amongst the AFRL team members. It was found that there were numerous values for the bond lengths listed in the publications as shown in Table 3. As a result of the literature reviews and discussions it was determined that the values of the lengths deviated due to different measuring methods and different environments. For the modeling and simulation being pursued by this task of the project, values obtained through X-ray diffraction were determined to be the best ones to use because the method measures the molecules that are in the “bulk” of the structure. An alternative method using a Scanning Tunneling Microscope (STM), would measure the surface molecule;

something that was felt to be undesirable when modeling structures that will enlarge as the self-assembled dendritic structures grow.

Table 3: Bond lengths found during research

Source, Article, Book, or Paper	NH.....O (AT)	NH.....N (AT)	NH (T)	NH.....O (CG)	NH.....N (CG)
Sinden [36]	2.82Å	2.91Å		2.84Å	2.92Å
Nakamoto (Uracil) [42]	2.84Å				
Faulkner [43]		2.878Å	1.014Å		
Riek [44]			0.98Å		
Khaikin [45]			1.028Å 1.012Å		
Meng [46]	2.951Å	2.879Å		2.934Å 2.817Å	2.951Å
Hobza [47]	3.086Å	2.988Å			
Lee [48]	2.90Å	2.89Å		2.78Å 2.93Å	2.93Å

Backbone

The next step in modeling DNA was determining the correct geometry of the sugar-phosphate backbone. One set of coordinates, referred to in [37], was found [49], graphed, as can be seen in Figure 52, and the bond lengths and angles were calculated. These measurements were then compared to those found in [49-51] and were found to fall within acceptable ranges.

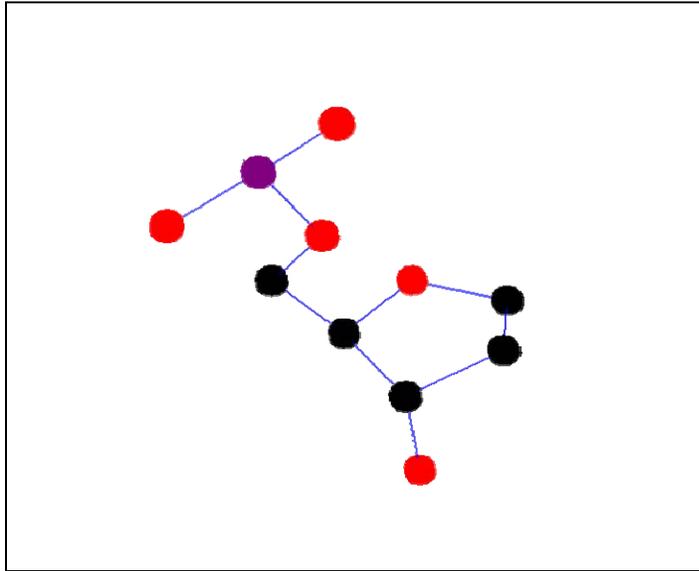


Figure 52: Illustration of the Sugar-Phosphate backbone in OpenGL

Backbone Torsion Angles

The backbone coordinates corresponded to the base coordinates that were not chosen. However, the backbone coordinates and the corresponding base coordinates did not seem to be properly aligned. The backbone also was not in the proper alignment with the chosen base coordinates. In order to obtain a proper alignment with the chosen base coordinates the backbone had to be translated and rotated. However, there was a problem in calculating and achieving the correct torsion angles [37, 49-51] between the backbone and bases as well as between adjacent backbones. The torsion angles remain an issue.

Base-pair parameters

The next step was to create a double strand of DNA by applying base-pair parameters [38, 52] to the bases as well as the backbone. The parameters consist of complementary base-pair parameters and base-pair step parameters. Base-pair step parameters or the transformations between adjacent bases were introduced first. Twist, the rotation about the Z-axis, and rise, the translation in the Z-axis, were used to create the basic helical structure. However, only a single strand was being created at this point because there was no complementary strand. In order to obtain a complementary strand the backbone was reflected over the Y and Z axes, while the base, as described earlier was reflected over the Y-axis. The rest of the base-pair parameters were added in after obtaining the basic double helical structure.

4.3.4.3 Single Strand Structure

The structure of single stranded DNAs was thought of as being highly flexible. According to [53] the rigidity of short single stranded DNAs is dependent on the strands sequence and is not as flexible as first thought. To start with for the implementation of the program, the single strand structure was drawn as a straight, rigid object. This can be edited later to obtain the proper single strand structure.

4.3.4.4 Hairpin Structure

A hairpin structure, as seen in Figure 53, is a single strand of DNA whose bases bond together to form a looped structure that resembles a hairpin. Secondary structures have HCR performed on them, which is how the tree grows exponentially. The structure of the loop of a hairpin is not known. The angles at which the bases in the loop bend and twist have yet to be determined and remain as issues.

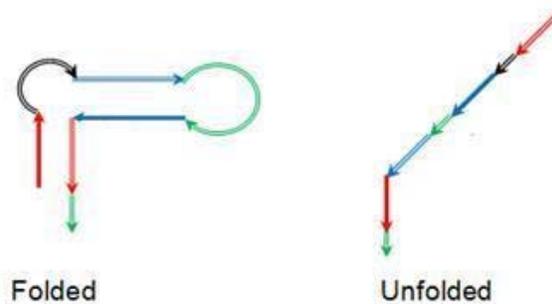


Figure 53: Hairpin Loop

4.3.4.5 Dendritic Tree

The interactions of DNA strands to form a dendritic tree were not included in the program and were only slightly touched upon. Following is a short, quick example of how one can approach the creation of the tree. The tree can be formed by comparing one strand to a second strand at each phase or level of the tree, to determine whether they have complementary sequences. This comparison will continue until each strand is used up or until there are no more complementary sequences to be paired.

4.3.4.6 Initial Software Program Overview

The program was separated into multiple functions. The main function, `DNA_Modeling`, sets up the window in which OpenGL renders the model. All other functions are called by `DNA_Modeling` and all functionality, be it key presses or mouse interaction, takes place in this function. A function exists for each base, `drawAdenine()`, `drawCytosine()`, `drawGuanine()`, and `drawThymine()`, and the backbone, `drawBackbone()`. In order to obtain the antiparallel orientation that exists in the double helix an additional function was created for each base `drawAdenineLeft()`, `drawCytosineLeft()`, `drawGuanineLeft()`, and `drawThymineLeft()`, and for the backbone, `drawBackboneLeft()`. The antiparallel orientation could also be obtained by passing a Boolean value to the original base functions to determine whether it should be drawn left or right. This method was implemented to start, but was changed afterward. The reasons the functions were changed and more were included were one, to prevent the passing of another parameter, two, create more readability for following programmers, and three, to allow the different parameters for each strand, 5'-3' and 3'-5' to be separated as well. There is a function, `drawBase()` and `drawBaseLeft()`, which determine which base function to call according to an array of characters, which represents a DNA strand. Then there is another function, `drawStrand()`, which will draw a strand, single, or double according to a Boolean value. There is also a `drawHairpin()` function which attempts to draw a hairpin, but is incorrect. The `drawStrand()` function calls the `drawBase()` or the `drawBaseLeft()` function, which in turn calls the corresponding base functions.

4.3.5 Structural DNA Nanotechnology Concluding Remarks

Work in this area began through collaboration between research teams at Duke University and the AFRL. While changes in organizational priorities within AFRL interrupted this collaboration and prevented the structural DNA nanotechnology area from being developed to its full potential, it is felt that some significant progress was made. The HCR Probe Finder has established a methodology for selecting the proper sequences to exploit HCR's for target identification. The results gathered through the initial stages of testing and validations indicate that the stochastic simulation provides an accurate portrayal of the interaction between DNA hairpins in solution. Work on both tasks show promising approaches for computer-based tools to promote development of a robust exponential HCR system for signal amplification. The task to develop three-dimensional modeling and simulation capability of strand hybridization during the creation of dendritic structures revealed that a considerable amount of work is still needed. When the stochastic simulation approach can be coupled with accurate information regarding the true shape, size, and geometry of dendritic trees, the stochastic simulation will become significantly more valuable than it is by itself. For example, an accurate model of the dendritic tree structure in 3-dimensions may show that steric effects play a major role in the growth of dendritic trees. The simulation parameters and hairpin design constraints can then be adjusted to see if varying the temperature, concentrations, or sequences alleviates the steric effects and improves signal

amplification. Additional work, however, is necessary to expand the framework that was established to its full potential.

HCR Probe Finder represents a leap forward in the application of using exponential HCR's for target identification. When linked with the stochastic simulation software further work can be done on optimizing subsequence lengths, and double hairpin subsequence structure. Upon reflection there are a few ways HCR Probe Finder could be improved upon. The most important of which would be a proper in-depth study of the data structure that arises out of Stage 2, focusing on the most effective data structures to find largest solution set. Although the current tree structure algorithm is the product of several revisions and approaches, it is still not perfect. Another thing that could be improved upon is distributing the tree navigation across multiple CPU cores or computers. On larger probe pools this could represent a massive improvement in performance and function. Work could also be done on the interface side of the program. A GUI frontend for the software would make using this program much easier and allow much better feedback.

When envisioning the ultimate computer-based tools set for exploiting HCR, the stochastic simulation of DNA hairpin kinetics is a work in progress. The most important addition to the simulation is the ability to model interactions of multiple hairpin sets in the presence of multiple DNA targets. This change is necessary in order to take full advantage of the hairpin set design process, which is currently able to optimize not only hairpins within a single set, but also optimize hairpin sets across multiple targets. Optimization of hairpins within a single set is a relatively simple process, so the simulation cannot currently provide all of the necessary feedback to improve the design process. In order to simulate interactions across multiple targets, the data structures as currently designed will grow very large, so more efficient programming techniques should likely be employed. The combinatorial matrices are size $10 \times 10 \times 3$ elements to identify one target. To identify two targets with a set of nine hairpins each, the array would have to grow to $20 \times 20 \times 9$ elements in order account for all possible combinations. Therefore, to identify n targets, the combinatorial array must have size $10n \times 10n \times 3n$.

Another important change to improve the stochastic simulation is to gradually remove assumptions and add greater detail to the simulation. Spontaneous unfolding of hairpins, for example, is assumed never to occur because the hairpins in their folded conformation are so thermodynamically favorable that it is very unlikely. Although unlikely, this possibility should be included in the simulation and is relatively straightforward to implement. Trees are assumed to diffuse through solution with the same velocity and Brownian motion as a folded hairpin, but a larger molecule should diffuse more slowly and travel smaller distances than a small molecule. Steric hindrance caused by "branches" in close proximity should also be taken into account when choosing the most probable place for a hairpin to attach to a tree.

Finally, the stochastic simulation could be more elegantly coded. MATLAB is a nice environment to develop a first attempt simulation, but other programming languages would likely outperform the MATLAB code. For example, the “additions_to_struct” data structure is physically cut and pasted as a way to perform dehybridizations, whereas pointers could be used effectively to track changes to trees.

The task on modeling strand hybridization revealed that much more time needs to be expended to fully develop a three-dimensional rendering of DNA single or double strands. More work is needed to address issues such as: calculating and determining the proper torsion angles between the backbone and bases; creating a single strand of DNA that forms into a hairpin; creating a dendritic tree, at first manually leading up to it being automatically generated; and determining the structure of a single strand of DNA in the dendritic tree structure. While a hairpin structure was attempted, the true structure of the loop of the hairpin was still unknown when this task was pursued, and so the structure is not complete.

4.4 Biotechnology Concluding Remarks

This aspect of the project changed directions a few times as a result of becoming wiser about the limitation of some concepts and organizational changes. Therefore, this hands-on portion of the project shifted focus from DNA and membrane computing to utilizing structural DNA nanotechnology. It explored the optimization of a HCR structure from the perspective of false positive reduction, signal amplification and structural integrity. As a result, three modifications to the original recursive dendritic structures were defined, increasing the number of hairpins required for the HCR from three to nine. These modifications allows one to create a more methodically correct reaction which optimizes proper hybridization while reducing false-positive noise often associated with self-assembly experiments. After the modifications were identified, the development of the proper computer-based design applications began to facilitate creating equally stable hairpins for more predictable, repeatable and quantitative HCR experiments and structures. An initial computer-based capability was established when this project ended. There is plenty of room for more work.

5.0 QUANTUM SCIENCES

Quantum computation is a novel paradigm of information processing that may provide significant advantages over classical computing methods allowing one to solve profound problems that are otherwise unattainable, such as the ability to factor large numbers. [1, 56] The physical implementation of quantum computing architectures will require novel technology and methods that are not easily supported by current design tools. Therefore the objective of this portion of the project is to explore what is needed for a proper M&S environment by examining the flexibility and adaptability of commercial tools.

5.1 Nanomechanical Resonators

One approach to the practical development of a quantum computer involves the use of nanomechanical resonators. Nanometer-scale, resonating beams are being explored as a possible avenue to exhibit quantum behavior such as superposition and decoherence. [55-57] Numerous research efforts are being conducted to simulate the behavior of these nanomechanical devices in order to understand how they can be designed, modeled, and applied to quantum computing. This potential development of quantum computers utilizing nanomechanical based processors relies on achieving an accurate model of the mechanical system. An interesting aspect of this portion of the project is that, over its duration, new versions of the SolidWorks and COMSOL Multiphysics software were installed. Comparisons of selected features will be made below.

5.1.1 Initial Assumptions and Constraints

To develop a model that accurately depicts the behavior of a nanoscale resonating beam, several factors related to building the proper model needed to be examined. In the literature review that was conducted, it was discovered that current nanoscale models ignore molecular details. [58] The object and components of the system are modeled as a continuum, allowing classical mechanics to be used to model a simple beam. Also, experiments have shown that density, temperature, velocity and displacement can be assumed as being smoothly varying. This means that there are no drastic rate-of-change anomalies, or “jumps” in any of these values when a variable is introduced into the system.

Next, body forces are ineffective at deforming structures compared to surface forces on the nanoscale. This is simply a scaling observation and means that the weight force due to gravity is so small with respect to any load force, i.e. an electrical driving force (applied voltage), that it can be considered negligible in that it will not affect the distance the load force will bend the device in the direction of the gravitational weight force.

Material properties are another important aspect of assumed conditions on the nanoscale level. Currently, researchers and scientists are using the same values that have been determined experimentally [59] on the microscale level. [60] However, these material properties assumptions leave out the temperature dependence of the material properties. All experiments have been run at cryogenic temperatures at or around zero Kelvin, which does not allow for a variance in temperature and therefore provides no insight as to how the properties may change during heating or cooling. Since material properties change from the mesoscale to the microscale, it is also expected that they change from the microscale to the nanoscale. This paradox, however, has not been addressed with much detail yet.

Along with the aforementioned assumptions there exist constraints that are imperative in modeling a nanomechanical beam. A maximum voltage boundary exists on the system. Since an electrostatic voltage signal is used to drive the system into resonance, it is important to determine how much voltage can be applied. This maximum voltage value is known as the “snap-in”, or “collapse” voltage because if the voltage exceeds this value, the resonator risks collapsing onto the substrate permanently, disabling the device. [61] A modification must also be made to the simple harmonic motion, or “mass-spring” equation, in order to accurately model a nanoscale device. The fundamental mass-spring equation is shown below. [62] Constants “m”, “c”, and “k”, represent the mass, damping constant and spring constant, respectively. The Term “x” refers to the displacement, while “v” and “a” refer to the velocity and acceleration, respectively.

$$ma + cv + kx = 0 \quad (28)$$

This equation implies that velocity approaches zero over time. Then, since temperature is a measure of motion of molecules and therefore is proportional to the velocity of the molecules, this would imply that the temperature would also approach zero. Since this model is on the nanoscale, and the nanoscale deals with objects at or above molecular size, temperature is directly proportional to velocity. With that, if the device stops oscillating and moving, then temperature would have to be zero. However, this violates the 2nd Law of Thermodynamics where the temperature of a system cannot vary from the temperature of its surroundings. [62] Therefore, a new motion equation that includes a driving function to keep the system in equilibrium is necessary. [62] This equation is:

$$ma + cv + kx = f_n(t) \quad (29)$$

5.1.2 Mechanical Model

In order to understand the characteristics of a nanomechanical system, it was necessary to generate a simple mechanical model. The aforementioned assumptions need to be applied in this aspect of modeling. To simulate a nanostructure that resonates and is fixed at both ends, a simple beam was chosen as the basis for the model. The beam is fixed at both ends and has length L and cross sectional dimensions w and t, as shown in Figure 54.

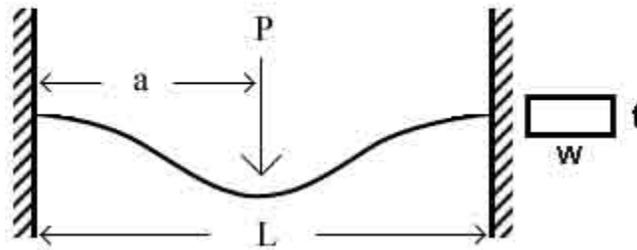


Figure 54: Illustration of basic beam

This simple model allows for the calculation of the resonance frequency, as well as the deflection under an applied load. Figure 55 below shows the coordinate reference system used, as well as the pertinent geometrical parameters for the model.

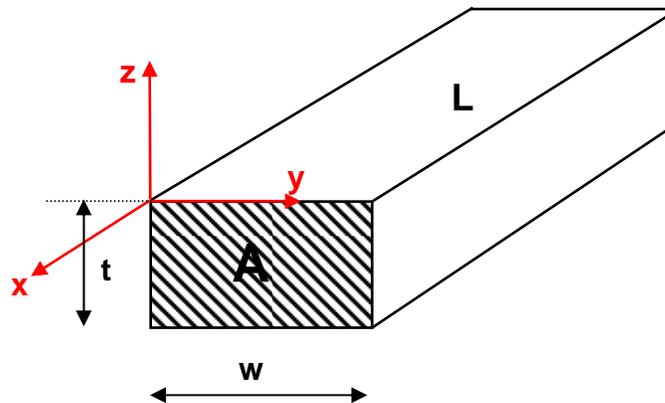


Figure 55: Coordinate reference system for beam

The deflection in a fixed-fixed beam can be calculated using classical mechanics. The ends are assumed to be fixed, but free to rotate, which eliminates any torque or moment on the beam. The characteristic equation for the deflection of a fixed-fixed end beam of this sort is readily available in mechanics of materials books [63] and is:

$$\delta = \frac{PL^3}{48EI} \quad (30)$$

Parameters present in the deflection equation that have yet to be defined include the load, P, E, and the moment of inertia, I. The load is a value that is defined by the applied voltage which is not the focus of this analysis. Young's Modulus is a material constant and can be obtained from experimental values that have been discovered and implemented in other micro and nanoscaled system modeling. [59, 60] There is some concern, however, that the current material property assumptions do not apply on the nanoscale. The moment of inertia is a source for confusion in modeling any mechanical system. In order to accurately portray the moment of inertia in this equation, the direction of oscillation needs to be determined. In this model, the beam oscillates in the z-direction (refer to diagram above). With this, the beam oscillates around the y-axis and therefore the moment of inertia is taken about the y-axis. [64] The characteristic equation for the moment of inertia about the y-axis [63] for this model is:

$$I_y = \frac{1}{12} t w^3 \quad (31)$$

The resonance frequency can also be modeled and is quite useful in a nanomechanical resonator. By knowing the resonance frequency, measurements of frequency can be used to tune the device to sense electronic signals in the beam or changes in frequency. [65] To model the natural frequency, the fundamental equation used is:

$$f_n = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (32)$$

A more useful version of this equation incorporates the fundamental resonance mode (the first mode) into the equation yielding the natural frequency. [62]

$$f_n = 1.03 \frac{t}{L^2} \sqrt{\frac{E}{\rho}} \quad (33)$$

Parameters for this model include geometrical as well as material properties. The geometrical properties are outlined in the model, as well as E. The density, ρ , is another material property that has to be obtained by dividing the mass by the volume of the object.

5.1.3 CAD Drawing and Analysis

The first step was to develop a geometrical model. Figure 56 shows a screenshot of the nanomechanical resonator beam assembly created in SolidWorks 2005 based on dimensions from documented work [66] and material properties from experimental results. [59] It consists of a gold layer 60 nm thick and a silicon layer 185 nm thick. It is 10.7 μm long, and each finger extends 400 nm off of the base section. The ability to include material properties in the three-dimensional model is extremely useful in finding the mechanical characteristics of the structure. This capability also allows the user to determine section properties from the geometry of a model. For example, the required moment of inertia around the y-axis can be determined in the section properties solver, which again pertains to a required parameter in the resonating beam model.

The only difficulty encountered in this model in the SolidWorks 2005 software was the unit limitations. The program is capable of calculating values by utilizing scientific notation and even allows for the assignment of user-chosen units. For example, nanometers were used in this model because they best described the geometry. A size limitation exists, however, in many of the feature tools. In an extrude feature, 100 nm is the smallest value that is allowed for creating a solid. This was a major setback because the 60 nm gold layer is obviously less than 100 nm thick. This did not appear to be an issue with solving for the geometry, but an arbitrary limit set by program developers. This obstacle was overcome by extruding the feature 160 nm, then cut-extruding the feature by 100 nm, leaving 60 nm of material as a solid.

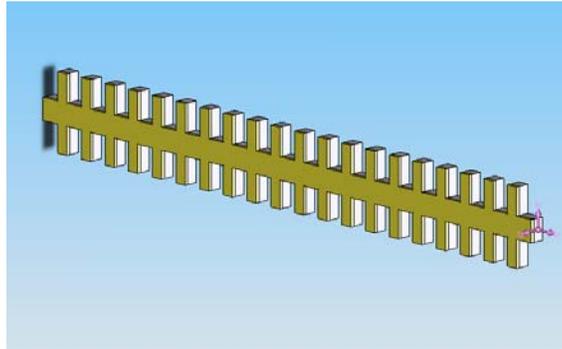


Figure 56: Initial SolidWorks model of nanomechanical beam

5.1.4 Finite Element Modeling and Analysis

Finite Element Analysis, or FEA, is an essential process in mechanical design and modeling in that it looks at the physical properties and characteristics of a system under loading and/or transient conditions. COMSOL Multiphysics 3.2 is an FEA program that has the ability to produce system models from established geometry, loading conditions and constraints. The “Static Analysis” feature allows for the modeling of deflection of a simple beam under loading conditions. The “Eigenfrequency Analysis” offers an avenue to calculate the resonance frequencies under different modes. These two features of the program are pertinent to the nanomechanical resonator model because deflection can be modeled under a certain electrostatic load from an applied voltage, and the resonance frequency can be determined. Stresses and deflections can be determined for the nano beam at resonance. This simulates the device oscillating at high frequencies close to resonance. The modeling software can not only calculate values, but also generate plots and animations of a given model.

One last feature of the software is its “Import CAD” feature. This feature allowed geometry files created in SolidWorks 2005 to be incorporated into geometrical objects in the COMSOL window. This feature was useful because it saved a significant amount of drawing time in COMSOL and allowed for accurate geometry to be portrayed and analyzed. A few difficulties were encountered, however, while importing assembly files from the SolidWorks package into COMSOL. It was later discovered, after a correspondence with a COMSOL Technical Representative, that the current version, version 3.2, does not support assembly files automatically. [67] This means that the program was unable to model assembly files completely as imported from SolidWorks since the transition between CAD files and FEA geometry is not smooth, causing some geometry characteristics to be lost. [67] There were a few possible solutions to this issue. First, a full geometry file could be created using separate geometry files. Basically, this would assign geometrical reference points to two different objects, and where they interact, these reference points would be the same. However, this task proved to be tedious and time-consuming. The second approach involves generating a MATLAB file to construct the geometry with an “.m-file” script. This method could be valuable if modified to allow for user inputs. This would allow easy rebuilding or would allow iterations with different geometry to test the effects of changes on the system and its characteristics. However, the code approach is difficult to implement for novice programmers and therefore was not pursued. The last alternative was to construct a hybrid structure that contained the same geometry as the assembly and only varied in material simulation. Instead of having two materials, this structure would have only one but it would be an effective material. The only value that varies and does not agree with the assembly is E for the two materials. An effective E may be obtained in this application from the following proportion of volumes of the two materials.

$$E_{eff} = \frac{E_{gold}V_{gold} + E_{silicon}V_{silicon}}{V_{total}} \quad (34)$$

With this effective E it was possible to run the modeling analysis. The Eigenfrequency analysis result yielded a resonance frequency similar to that calculated by hand, proving the validity of the Finite Element code. Also, the deflection could be modeled under this resonance mode and the stress could be calculated as well. Figure 57 below shows the stress model for the first mode resonance frequency in Eigenfrequency analysis:

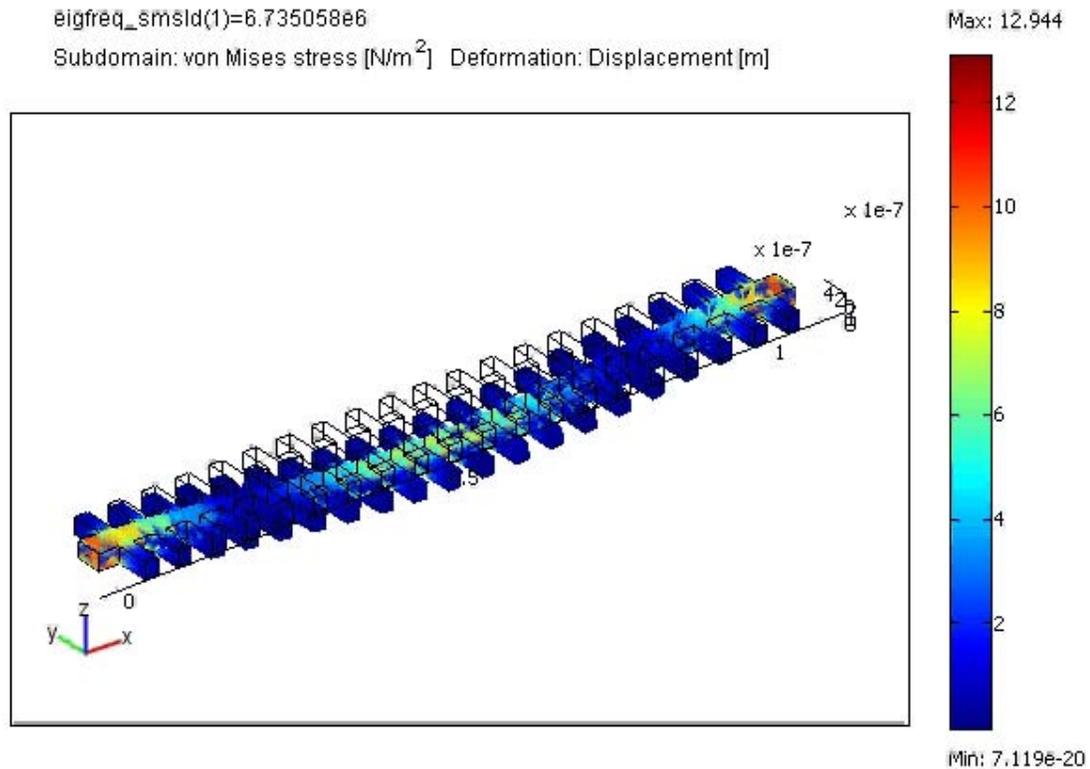


Figure 57: First mode response for beam

As is seen in the above graphic outlining the stress, the Eigenfrequency output “eigfreq_smsld(1)” corresponds to the first mode resonance frequency and is given as 6.735×10^6 Hz. The maximum stress experienced is located at the ends and is shown in red. The typical shape of the first mode resonance frequency is experienced in this model with the one peak oscillating up and down. This analysis tool proved helpful in understanding the characteristics of a nanostructure beam.

COMSOL Multiphysics Version 3.3 brought with it a significant improvement in the capabilities of its troublesome import feature discussed above. Version 3.3 was able to handle complex structures with multi-layered geometries. The improvement of the “Create Composite Object” feature eliminates redundant interior boundaries while still recognizing different components and allowing for characteristic material properties to be applied. Figure 58 highlights the assignment of material properties to the different component subdomains created in SolidWorks and imported in COMSOL.

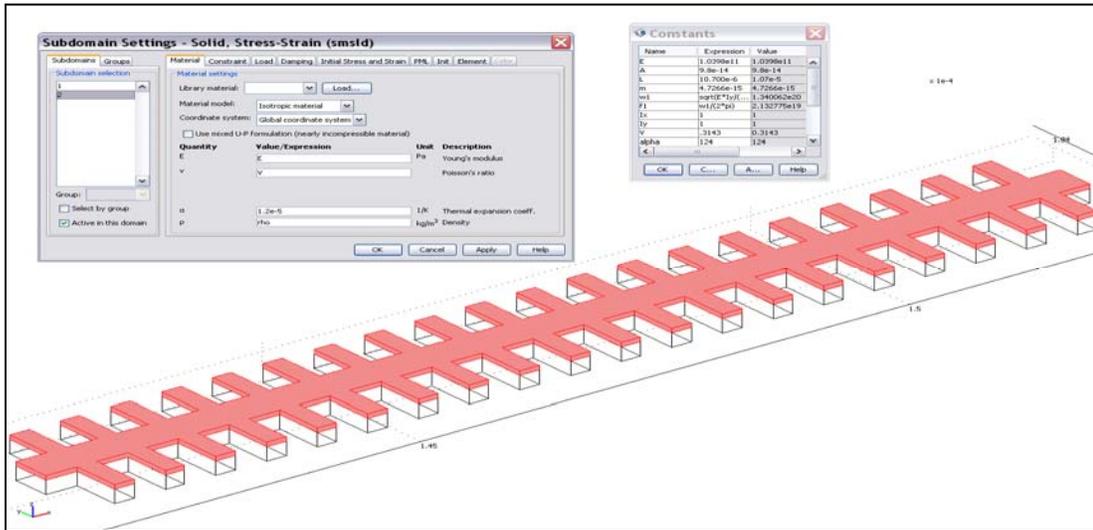


Figure 58: Snapshot of subdomain material definition for beam geometry

Simulations were rerun under COMSOL Version 3.3. Figure 59 below shows the third mode Eigenfrequency response of the beam in COMSOL. The magnitude of the frequency and the mode shape were both shown to have good agreement with the documented results from researchers at Boston University. [66]

5.1.5 Nanomechanical Resonator Concluding Remarks

An introduction to the fundamental details and complexity of modeling a nanomechanical resonator has been provided. Through a literature review it was discovered that little research has been conducted on nanomaterials under both room temperature and cryogenic conditions. A better understanding of material properties on the nanoscale and their impact on quantum mechanical modeling will be needed for future research in order to more accurately model nanodevices and systems. Commercially available software proved to be a very useful in modeling a nanomechanical resonator beam.

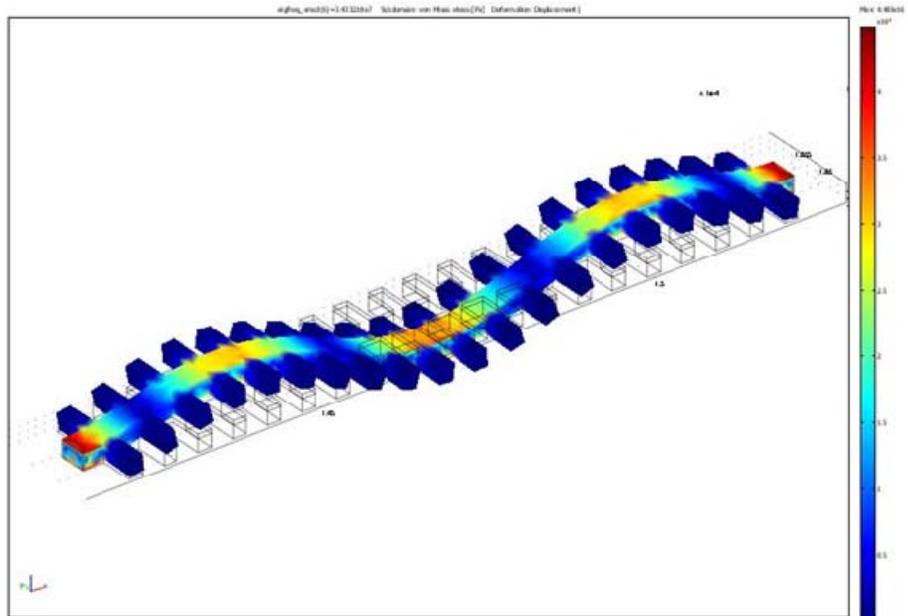


Figure 59: Third mode response for beam

5.2 Ion Trap

A second approach to quantum computing being pursued by some researchers is the use of ion traps. Basically, ion traps encode and process data with a string of ions that are confined in a field. The field depends on the type of ion trap: the Penning Trap makes use of magnetic fields, and the Paul Trap, or Linear Trap, utilizes radio frequency (RF) electric fields. Lasers are used for inputs, as well as for a means of laser cooling ions. [68] The promise of the ion trap lies in its dependence on relatively well-known technology. Fabrication of the proper device geometry for ion trap concepts can leverage a combination of MEMS, CMOS [69] and Gallium Arsenide (GaAs) [69, 70] technology. At the forefront of research into ion traps is a group at the University of Maryland, led by Christopher Monroe (previously at The University of Michigan). [70, 71] Investigation into planar geometry of ion traps by the Maryland group has led to an understanding of the hyperbolic electrode model, as well as the influence of geometrical relationships and aspect ratios on the potential field. [70] They have also developed a concept for a scalable array of ion traps, thus adding more merit to the ion trap becoming a practical means of building a quantum computer. Another group from Imperial College of London has focused on the scalability of ion trap chip design and its implementation with existing fabrication technology, specifically by use of existing silicon-based MEMS processes. The Imperial College group reaffirmed the importance of geometry and aspect ratio on the efficiency of the device, as well as proposed the consideration of resonance frequency and RF heating as modeling parameters. [69] Published models and research by these groups provided a basis for investigation of the applicability of commercially available M&S software for the design and analysis of ion traps. The pertinent parameters that are to be investigated include the electric field for ion confinement [68, 69] and the resonance frequency for mechanical validity. [69] This portion of the project also provided insight into the idiosyncrasies of modeling ion traps.

5.2.1 Ion Trap Model Generation

Ion trap geometry was implemented into SolidWorks according to the existing Imperial College design. [69] The structure consists of, essentially, five layers: a silicon base, two silicon dioxide surrounding layers, and two gold electrode layers. Reducing the structure into components of each layer in an assembly allowed for different material assignment with correct geometric relationships. Embedded properties and appearance for the prescribed materials allowed accuracy in representation and visualization. Figure 60 shows the completed geometry structure from SolidWorks.

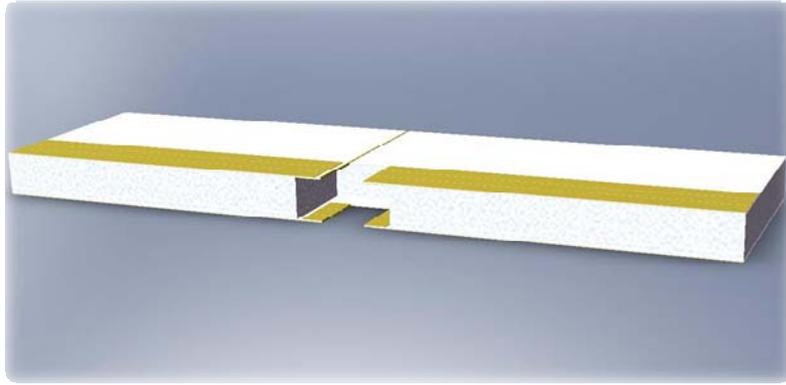


Figure 60: CAD model of individual linear ion trap

5.2.2 Ion Trap Analysis

The initial work with ion traps was interested in the resonance frequency, a mechanical parameter, and the electric field, an electronic phenomenon. COMSOL possesses a built-in geometry generator, which is beneficial for reducing complex CAD models to offer a reduction in model size and ultimately solution time. This feature, along with the MATLAB interface, means that the iterative nature of the modeling and simulation process does not imply starting over from scratch. Previous modeling work can be recycled and reused.

The resonance frequency of the ion trap structure, specifically the electrodes, must be sufficiently far from the RF input frequency to avoid interference. [69] Therefore, it is necessary to investigate the resonance frequency of the electrodes and structure to ensure there will not be interference. A simple approximation of the geometry is found in Equation 35 below for the electrode from the ion trap with length, h , and width, w , modeled as a cantilever. The approximation neglects the gold electrode layer and only uses silicon dioxide and its relevant material properties in Equation 35. With the published dimensions from the design [69], a frequency of 154 kHz from this model was found to be sufficiently far from the drive frequency of about 21 MHz.

$$f = 2\pi \left(0.16 \sqrt{\frac{E}{\rho} \frac{w}{h^2}} \right) \quad (35)$$

To validate this approximation, FEA was performed on the simple structure of exact dimensions and material properties. Agreement was found within 1 percent, adding confidence in modeling procedures as well as the software package. To add a more realistic approach to the model, the gold layer for the electrode was introduced using the prescribed process. A solution of approximately 96 kHz can be found in Figure 61

below. This result was considered reasonable, as adding mass reduces the frequency according to Equation 35. A simple hand calculation of the more complex geometry of this structure, however, was not possible. The number of elements required to obtain a solution was also investigated in this portion of the project. This work established an understanding of the complexity necessary to achieve a certain level of accuracy, providing insight to simplifying complex models for future work. Mesh refinement was inputted manually and the solution and convergence time were recorded for each case. Figure 62 shows the result for the mesh case study. Clearly, solution time increases significantly as mesh refinement increases. However, the accuracy of the solution was not affected by more than 5 percent. This suggests that in future modeling, mesh refinement may be coarse for complex models to still achieve a reasonable degree of accuracy. This will save significant resources, allow for reduction in time to optimize geometry and integrate analytical results into system models.

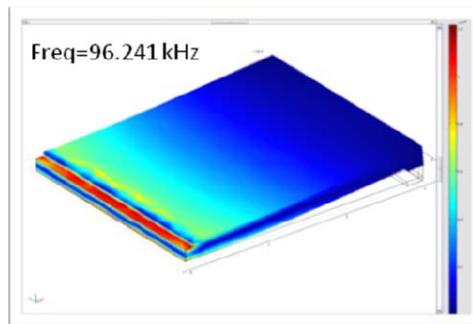


Figure 61: COMSOL resonance frequency of electrode

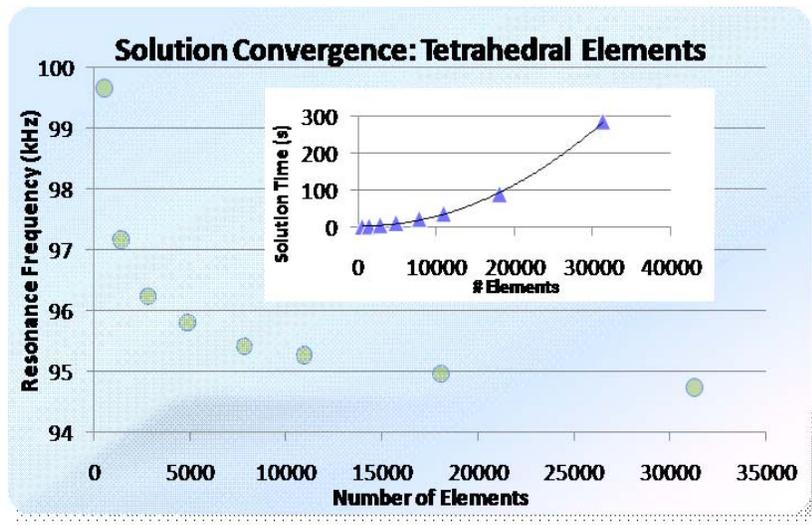


Figure 62: Mesh case study results

The knowledge gained from the case study was applied to the entire geometry structure. Figure 63 shows the geometry mesh case for the ion trap zone and Figure 64 shows the results of the analysis. A solution of about 93 kHz is approximately 3 percent different than the simplified case. The significance of this correlation of results is that it provides an option to performing several small, simplified studies with individual models when a single model can be reused several times.

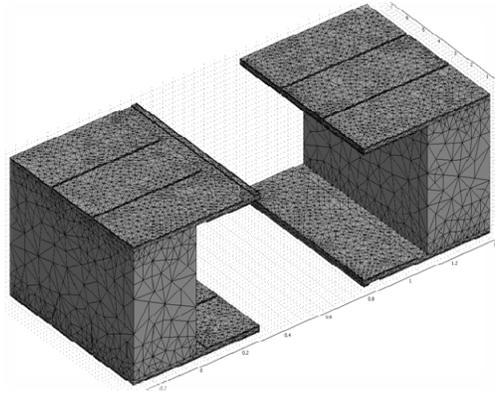


Figure 63: Mesh structure for resonance frequency

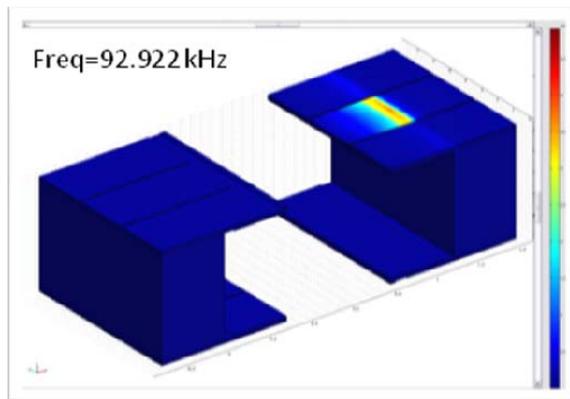


Figure 64: Solution for resonance frequency

The electrical characteristics were also pertinent to our performance output goals of the modeling and simulation process. A stable trapping potential must be maintained to confine ions. [69]

The RF drive voltage, $V_o \cos \Omega_T t$, and DC endcap voltage, V_{dc} , are constant parameters in the model, while the voltages V_{c1} and V_{c2} are variables, applied to the upper center electrodes, that adjust the electric potential field. In modeling the field in the simulation process, these variables can be adjusted to obtain the desired trapping potential. COMSOL and its AC/DC module were used to investigate this model with time-harmonic boundary conditions according to the RF input. For the RF voltage, there must be an outer boundary condition that defines the perimeter. Elements must also be present in the zone (as opposed to open space) in order to have a solution in that zone. Therefore, the ion trap zone structure was enclosed in a box, as was shown in the top of Figure 65, allowing for the definition of the boundaries as well as a medium between electrodes to solve for the potential field. The drawback to this approach, however, is the complexity of the model and the issues encountered with aspect ratio errors. According to the electrode geometry, the outer box is many times larger and thus requires hundreds of thousands of elements to obtain a result. This is not practical, as the solution time is significant. With this, the geometry generation capability of COMSOL allows for modification of the CAD structure to only include the trap zone. Figure 65 illustrates this transformation and shows the reasonable mesh incorporated into the structure. Application of the boundary conditions allows for solving of this structure, and the solution may be found in Figures 66 and 67.

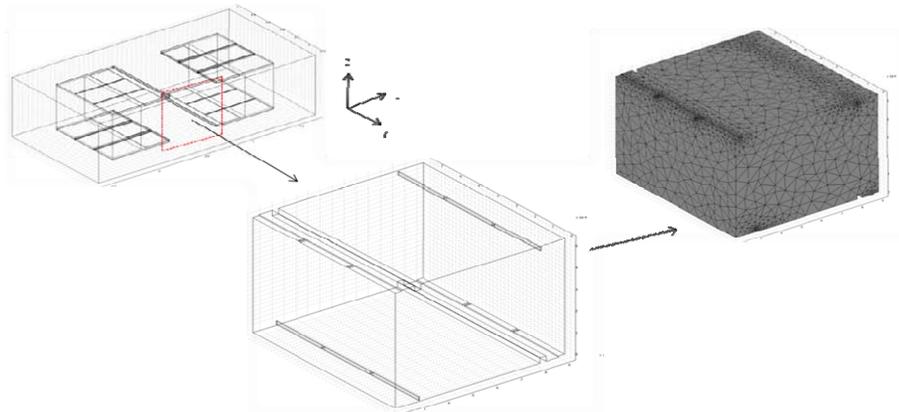


Figure 65: Simplification of ion trap geometry for solution

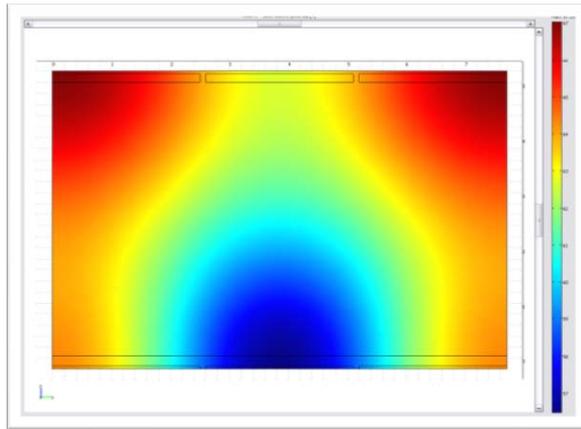


Figure 66: View of potential field in YZ plane

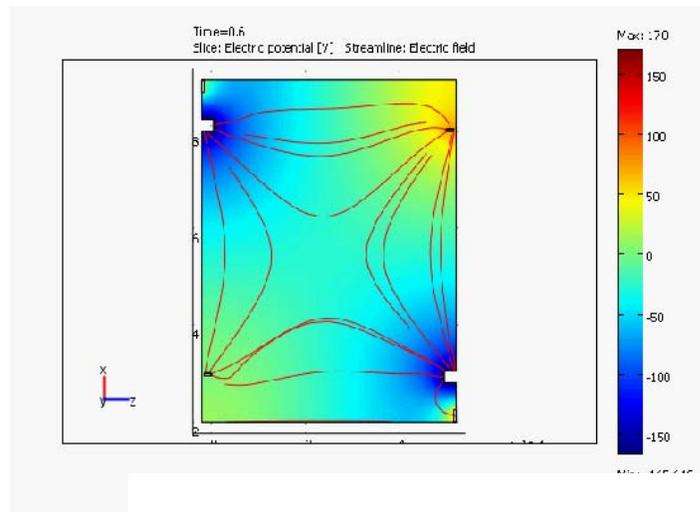
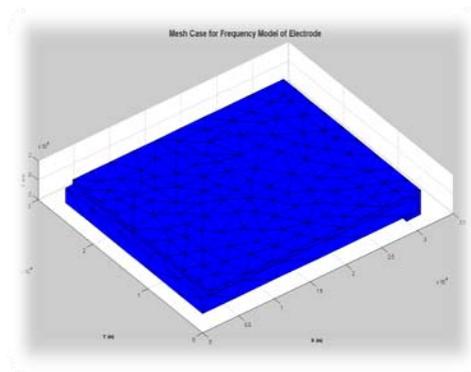


Figure 67: View of solution in XZ plane

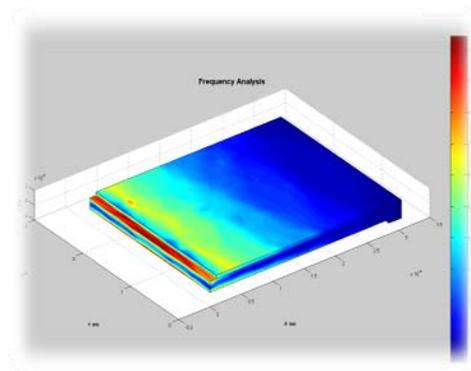
The solution illustrates a hyperbolic electrode geometry similar to and in good agreement with that documented by the University of Maryland group. Modification and refinement of electric potentials V_{c1} and V_{c2} allows for fine-tuning of the trap zone to achieve the desired field. The FEA package in COMSOL Multiphysics allows designers to perform case studies which, though tedious, offer suggestions on the implementation of the entire model.

5.2.3 Scripting and Customization

To allow for more flexibility in models, especially those not supported by COMSOL Multiphysics or other FEA packages, scripting and code generation are necessary elements of the design process hierarchy. COMSOL, as was previously mentioned, included a MATLAB [72] interface as well as its own, stand-alone scripting program. [73] With this, you are able to save models as “.m” extensions to convert to MATLAB format. This allows for integration of any MATLAB function, as well as use of existing COMSOL functions simultaneously running in MATLAB. Another helpful feature is the ability to use the MATLAB command window to execute or research (via help) COMSOL commands. This is applied in debugging, as well as in verifying the status of the solution. A significant benefit of the MATLAB interface is found in the ability to incorporate any mathematical relationship, such as a novel and thus unsupported governing equation, into the COMSOL model. This includes writing programs to use looping to perform a case study. An example of the application of a program loop can be found in the analysis of mesh refinement from Figure 62. Recall that manual input of mesh refinement was necessary, as well as manual recording of the pertinent output values. This tedious process can be eliminated by writing a script to perform the analysis with different mesh cases while recording the solution and plotting. By converting the initial COMSOL model file from the resonance frequency analysis to a MATLAB “.m” file, the boundary conditions and geometry are all preserved and all that is necessary is to modify the code to incorporate the loop. The output parameters are solved for and recalled from the COMSOL functions in MATLAB, and recorded for post-processing so that the user may review the results. Figure 68 below shows a result of an iteration of the case study which is similar to that of Figure 61.



Mesh Performed In Matlab



Solution Computed In Matlab

Figure 68: Response frequency solution from MATLAB

The results in Figure 69 demonstrate a plot with a similar trend to that in Figure 62. It is evident that the solutions are not exactly the same as those found in COMSOL, and this is due to the difference in solving algorithms between the two programs. However, the overall solution accuracy is within 1 percent.

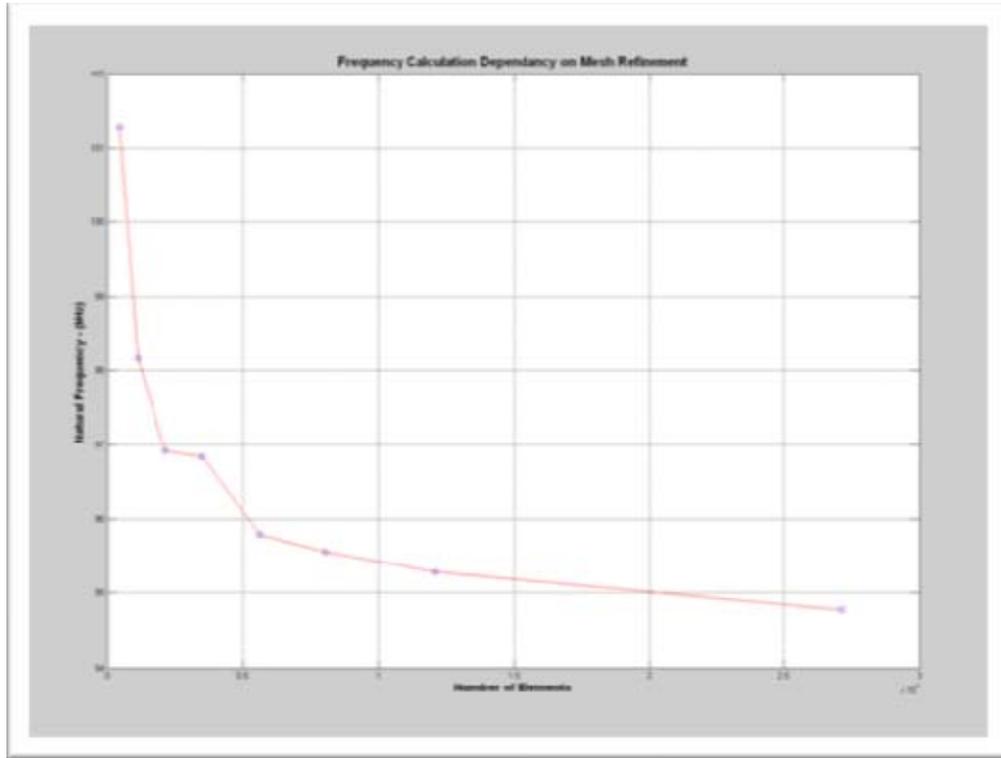


Figure 69: Dependence of solution on mesh refinement with MATLAB

Another application, as was mentioned in the introduction to this section, is incorporating mathematical relationships that are not supported in COMSOL. This is best illustrated in the example of the ideal hyperbolic potential. [71] A model with boundary conditions similar to that of Figure 54 was introduced as a two-dimensional model in COMSOL and converted to MATLAB. The AC/DC model was solved for the potential field. The solution can then be compared to the ideal case, represented as a mathematical equation. [74] Figure 70 shows the model solution and the integration of the ideal hyperbolic geometry for a visual comparison in the MATLAB environment.

Overall, customization in MATLAB scripting allows for flexibility and versatility of COMSOL models. Case studies also prove to verify and instill confidence in models by illustrating the behavior of the device in a wide range of situations. The ability to incorporate new technology into existing models provides the functionality and performance necessary to investigate novel concepts under the proposed process and contributes immensely to the efforts of SCHETCH.

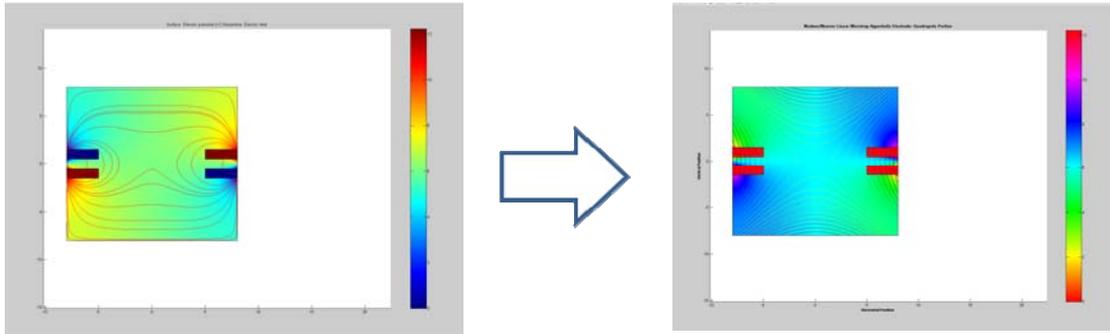


Figure 70: Model comparison of solution to ideal case for hyperbolic potential

5.2.4 Circuit Models

Circuit analysis of models adds a mode of functionality for ultimate implementation into a systems model. COMSOL Multiphysics recognizes circuits via an interface with SPICE. SPICE is a general purpose circuit simulation program that is used in integrated circuits (ICs) to analyze circuit behavior and performance. [75]

COMSOL and its SPICE import allow for definition of circuit elements, such as resistors, capacitors, sources, and nodes. Nodes are then assigned to the respective physical components in the model and incorporated as boundary conditions. This allows for definition of the model as a component of a circuit, and potentially as an element in a larger control circuit or system circuit. An example of a control system is found in the control circuit used in experiments by the Maryland group. [76] A smaller sub-circuit may also be defined and implemented from individual components of the model. Such an example exists in the investigation of RF heating by the London Group. [69]

Though a promising feature and integral step in the modeling and simulation hierarchy, circuit modeling in SPICE needs to be further investigated for its adaptability. More knowledge on the implementation of models in the SPICE environment also needs to be obtained to investigate all of its functionality.

5.2.5 Ion Trap Concluding Remarks

The initial finite element modeling of ion traps in SolidWorks and COMSOL has shown that commercially available M&S software can play a role in the analysis and development of ion trap concepts. Further work is needed to understand additional characteristics of ion traps such as RF heating and “breakdown” voltages. SPICE could also play a role in understanding the control circuitry required for ion traps and aid in comparing the characteristics of different design concepts.

In order to implement a physically-functioning quantum information processor, knowledge of the performance of the entire system including ion trap devices, control circuitry, and support devices is necessary. Even with the initial success, the question remains whether existing M&S tools and techniques such as VHDL can be used to properly model a quantum computer design, or do new M&S paradigms need to be developed? If VHDL can be used to model new alternative computing processes than comparison with standard computing processes is possible. It would also imply that there is an increased likelihood of integrating new alternative computing concepts with the current state-of-the-art in computer technology.

6.0 CLOSING REMARKS

An overview of the SCHETCH project has been presented. While several alternative computing concepts were examined under this project, the hands-on experience was focused on examining carbon nanotubes, quantum computing concepts, and structural DNA nanotechnology. The nanotubes and quantum computing concepts were addressed in the earlier portion of the project, while the structural DNA nanotechnology was concentrated on in the later portion of the project.

From the perspective of the carbon nanotubes and quantum computing concepts, it was found that commercially available three dimensional M&S software tools can be used to analyze concepts for physical components of the technologies. A process for modeling and simulation was defined, but more work is needed to implement SPICE-type models and understand how to conduct system level models of quantum computing concepts. Several gaps related to M&S were defined throughout the report. The biggest gap was the lack of a clear understanding and proper definition of material properties for modeling structures at the nanoscale. Future work with quantum computing concepts will be conducted under other in-house efforts especially those examining cluster state entanglement and solid state approaches to quantum computing.

While the work with biotechnology did not materialize as initially envisioned, progress was made in the area of structural DNA nanotechnology. Initial concepts for computational tools to exploit structural DNA nanotechnology were developed. An initial framework for a design suite was defined that would allow one to define a library of the proper DNA strands, explore the interaction of the strands prior to experimentation and refine the library if necessary, and then visualize through 3-D graphics the growth of the DNA based structure. Unfortunately the project was terminated before the initial concepts could be fully refined and validated so more work remains to be done for the concepts to be fully realized.

7.0 REFERENCES

- [1] Thiem, C., Drager, S., Flynn, C., Renz, T., and Burns, D., "Advanced Computer Technology for Novel Information Processing Paradigms," *Journal of Aerospace Computing, Information and Communication*, **1**, 2004, pp. 308-317.
- [2] Thiem, C., Bhat, S., and Blount, T., *Establishing Tools for Computing Hybrids*, AFRL-IF-RS-TR-2006-296, Air Force Research Laboratory, Rome, NY, October 2006.
- [3] Pelsko, J., and Bernstein, D., Modeling of MEMS and NEMS. Chapman & Hall/CRC. 2003.
- [4] Baglio, S., Castorina, S., and Savalli, N., Scaling Issues and Design of MEMS. John Wiley & Sons. 2007.
- [5] Kurowski, P., Finite Element Analysis for Design Engineers. SAE international. 2004.
- [6] Iijima, S., "Helical Microtubules of Graphitic Carbon", *Nature*, **354**, 1991, pp. 56-58.
- [7] Callister, W., Materials Science and Engineering: An Introduction. New York, John Wiley & Sons, Inc. 2007, pp. 431-433.
- [8] Banerjee, K., and Srivastava, N., "Are Carbon Nanotubes the Future of VLSI Interconnections?", *Proc. of the 43rd Annual Conference on Design Automation*, ACM Press, 2006, pp. 809 – 814.
- [9] Ruoff, Rodney S. et al., "Mechanics of Carbon Nanotubes", *ASME Journal of Appl. Mech Rev.*, **55**, Number 6, 2002, pp. 495-533.
- [10] Lukes, J., and Zhong, H., "Thermal Conductivity of Individual Single-Wall Carbon Nanotubes," *ASME Journal of Heat Transfer*, **129**, 2007, pp. 705-716.
- [11] Ghai, S., Kim, W. T., Escobar, R., Amon, C., and Jhon, M., "A Novel Heat Transfer Model and its Application to Information Storage Systems", *Journal of Applied Physics*, **97**, 2004, pp. 10P703-1 - 10P703-3.
- [12] Yang, X., "Modelling Heat Transfer of Carbon Nanotubes", *Modelling and Simulation in Materials Science and Engineering*, **13**, 2005, pp. 893-902.
- [13] Incropera, F., Dewitt, D., Bergman, T., and Lavine, A., Fundamentals of Heat and Mass Transfer, 6th Edition, Hoboken, John Wiley & Sons, 2007.

- [14] Corne, D., and Fogel, G., "An Introduction to Bioinformatics for Computer Scientists." *Evolutionary Computation in Bioinformatics*. Gary B. Fogel and David W. Corne, Editors. Morgan Kaufmann Publishers/Elsevier Science, San Francisco, California, 2003, pp. 3-18.
- [15] Paun, G., and Paun, R., 2005. Membrane Computing as a Framework for Modeling Economic Processes. <http://ppage.psystems.eu/>. Active as of June 18, 2009.
- [16] Petreska, B. and Teuscher, C. <http://www.teuscher.ch/christof/psystems.php>. Active as of June 18, 2009.
- [17] Dwyer, C. and Lebeck, A., Introduction to DNA Self-Assembled Computer Design. Artech House, Inc. 2008.
- [18] Mullis, K., and Faloona, F., "Specific Synthesis of DNA in Vitro Via a Polymerase-Catalyzed Chain Reaction," *Methods in Enzymology*, **155**, 1987, pp. 335-50.
- [19] Dirks, R. and Pierce, N., "Triggered Amplification by Hybridization Chain Reaction," *Proc. Nat. Acad. Sci., U.S.A.*, **101**, 2004, pp. 15275-15278.
- [20] Green, S., Lubrich, D., and Tuberfield, A., "DNA Hairpins: Fuel for Autonomous DNA Devices," *Biophys. J.*, **91**, 2006, pp. 2966-2975.
- [21] Chemeris, D., Nikonorov, Y., and Vakhitov, V., "Real-Time Hybridization Chain Reaction," *Doklady Biochem. and Biophys.*, **401**, 2008, pp. 53-55.
- [22] Corn, R., *Programmable Self-Assembly of DNA-Dendrimer and DNA-Fullerene Nanostructures*, AFRL-IF-RS-TR-2004-287, University of Wisconsin, Madison, WI, October 2004.
- [23] Smith, D., "Dendritic Supermolecules – Towards Controllable Nanomaterials," *Chem. Commun.*, Roy. Soc. Chem., 2006, pp. 34-44.
- [24] Bishop, M., Thiem C., Renz, T., Schultes, E., Chandran, H., and Reif, J. "Structural Optimization of Dendritic DNA Self-Assembly, abstract, *Foundations of Nanoscience (FNANO) 2009*, Snowbird, Utah, Sciencetechnica, April 2009, pp. 89-90.
- [25] Reif, J. and Schultes, E., *AFSOR Bio-X: Encapsulated DNA-Based Molecular Autonomous Sensing Devices with Photonic I/O*, Duke University, Durham, NC, January 2009.
- [26] Gorczyca, A, *DNA: Optimization of Taggants to Encode and Process Information Reliably*, Air Force Research Laboratory, Rome, NY, 2009 End of Summer Report to be published.

- [27] SantaLucia, J., Hicks, D., “The Thermodynamics of DNA Structural Motifs.” *Annu. Rev. Biophys. Biomol. Struct.* , **33**, 2004, pp. 415–40.
- [28] Zimm, B., “Theory of the Melting Transition of Synthetic Polynucleotides: Evaluation of the Stacking Free Energy.” *J Mol Biol* , 1964, pp. 1-9.
- [29] SantaLucia, J., “A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics.” *Proc. Nat. Acad. Sci., U.S.A.*, **95**, 1998, pp. 1460-1465.
- [30] Flack, A., *Stochastic Simulation of DNA Hairpin Kinetics*, Air Force Research Laboratory, Rome, NY, 2009 End of Summer Report to be published.
- [31] Cherepinsky, V., “On Mathematical Aspects of Genomic Analysis,” Ph. D. Dissertation, Department of Mathematics, New York University, September 2003.
- [32] Gantovnik, V., and Gibas, C., “Modeling of Competitive Kinetics of DNA Hybridization Reactions,” Presentation, Department of Computer Science, University of North Carolina, March 2008.
- [33] Gillespie, D., “Stochastic Simulation of Chemical Kinetics,” *Annu. Rev. Phys. Chem.*, **58**, 2007, pp. 35-55.
- [34] Manthey, J., “mFold, Delta G, and Melting Temperature: What does it mean?” Retrieved August 2009, from Integrated DNA Technologies: http://eu.idtdna.com/Support/Technical/TechnicalBulletinPDF/mFold_Delta_G_and_melting_temperature_explained.pdf , 2005, pp. 1-8.
- [35] Cain, S., *Three Dimensional Graphical Representation of DNA Strand Interaction for a Virtual Verification Environment*, Air Force Research Laboratory, Rome, NY, 2009 End of Summer Report to be published.
- [36] Sinden, R., *DNA Structure and Function*. Oxford: Academic Press Inc., 1994.
- [37] Parkinson, G., Vojtechovsky, J., Clowney, L., Brunger, A., and Berman, H., “New Parameters for the Refinement of Nucleic Acid-Containing Structures,” *Acta Cryst.* **D52**, 1996, pp. 57-64.
- [38] Suzuki, M., Berman, H. M., Olson, W. et al., “A Standard Reference Frame for the Description of Nucleic Acid Base-pair Geometry,” *J. Mol. Biol.* **313**, 2001, pp. 229-237.
- [39] Clowney, L. Jain, S., Srinivasan, A., Westbrook, J., Olson, W., and Berman, H., “Geometric Parameters in Nucleic Acids: Nitrogenous Bases,” *J. Am. Chem. Soc.* **118**, 1996, 509-518.

- [40] McMahon, M., *Study of the Hydrogen Bond Length*, Air Force Research Laboratory, Rome, NY, 2009 End of Summer Report to be published.
- [41] Svozil, D., Kalina, J., Omelka, M., and Schneider, B., "DNA conformations and their sequence preferences" *Nucleic Acids Research*, **36**, No. 11, 2008, pp. 3690-3706.
- [42] Nakamoto, K., Margoshes, M., and Rundle, R., "Stretching Frequencies as a Function of Distances in Hydrogen Bonds" *J. Am. Chem. Soc.*, **77**, No. 24, 1955, pp. 6480-6486.
- [43] Faulkner, C., and Macrae, R., "A DFT Study of Hydrogen Bonding Between Adenine and Thymine," *Proceedings of the 2006 University of Evansville Undergraduate Conference for Science, Engineering and Mathematics*, University of Evansville, March 2006.
- [44] Riek, R., "Characterization of Hydrogen Bond Lengths in Watson-Crick Base Pairs by Cross-Correlated Relaxation" *Journal of Magnetic Resonance*, **149**, 2001, pp. 149-153.
- [45] Vogt, N., Khaikin, L., Grikina, O., Rykov, A., and Vogt, J., "Study of the Thymine Molecule: Equilibrium Structure from Joint Analysis of Gas-Phase Electron Diffraction and Microwave Data and Assignment of Vibrational Spectra Using Results of Ab initio Calculations" *J. Phys. Chem. A*, **112**, 2008, pp. 7662-7670.
- [46] Meng, F., Wang, H., Xu, W., and Liu, C., "Substituent Effect of Large Conjugate Groups on the DNA Base Pair Derivatives: Density Functional Study" *International Journal of Quantum Chemistry*, **104**, 2005, pp. 79-86.
- [47] Hobza, P.; Spöner, J.; "Structure, Energetics, and Dynamics of the Nucleic Acid Base Pairs: Non-empirical Ab Initio Calculations" *Chem. Rev.*, **99**, 1999, pp. 3247-3276.
- [48] Lee, M., and Sankey, O., "Theory of tunneling across hydrogen-bonded base pairs for DNA recognition and sequencing" *Physical Review E*, **79**, 2009, pp. 051911-1--051911-10.
- [49] Packer, M., and Hunter, C., "Sequence-dependent DNA Structure: The Role of the Sugar-phosphate Backbone," *J. Mol. Biol.*, **280**, 1998, pp. 407-420.
- [50] Gelbin, A., Schneider, B., Clowney, L., Hsieh, S., Olson, W., and Berman, H., "Geometric Parameters in Nucleic Acids: Sugar and Phosphate Constituents," *J. Am. Chem. Soc.*, **118**, 1996, pp. 519-529.
- [51] Schneider, B., Neidle, S. and Berman, H. M., "Conformations of the Sugar-phosphate Backbone in Helical DNA Crystal Structures," *Biopolymers*. **42**, 1997, pp. 113-124.

- [52] Dickerson, R. E. et al., "Definitions and Nomenclature of Nucleic Acid Structure Components," *Nucl. Acids Res.* **17**, 1989. pp. 1797-1803.
- [53] Goddard, N., Bonnet, G., Krichevsky, O., and Libchaber, A., "Sequence Dependent Rigidity of Single Stranded DNA," *American Physical Society.* **85**, 2000, pp. 2400-2403.
- [54] Shor, P., "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," *Proc. 35th Annu. Symp. on the Foundations of Computer Science. Los Alamitos*, 1994, pp. 124-134.
- [55] Choi, Charles Q., "Qubit Twist: Bending Nanotubes as Mechanical Quantum Bits", *Scientific American*, April, 2005.
- [56] Gaidarzhy, A., Zolfagharkhani, G., Badzey, R., and Mohanty, P., "Evidence for Quantized Displacement in Macroscopic Nanomechanical Resonators", *Physical Review Letters*, **94**, 2005, pp. 030402-1 - 030402-4.
- [57] Menting, A., "World's Fastest Oscillating Nanomachine Holds Promise for Telecommunications, Quantum Computing: BU Team's Nanomechanical Device Bridges Classic and Quantum Physics", News & Events – News Releases, 2005, Boston University, valid 13 June 2006.
<<http://www.bu.edu/phpbin/news/releases/display.php?id=858>>
- [58] Pelesko, J., and Bernstein, D., Modeling NEMS and MEMS, Boca Raton, Chapman and Hall/CRC, 2003, pp. 17-80.
- [59] Sharpe Jr., W., Hemker, K., and Edwards, R., "Mechanical Properties of MEMS Materials," AFRL-IF-RS-TR-2004-76, Johns Hopkins University, Baltimore, MD, March 2004.
- [60] Teva, J., et. al., "System on Chip Mass Sensor Based on Polysilicon Cantilevers Arrays for Multiple Detection", *Sensors and Actuators A*, April, 2006, p. 8.
- [61] Abadal, Gabriel et. al., "Electromechanical Model of a Resonating Nano-Cantilever-Based Sensor for High-Resolution and High Sensitivity Mass Detection", *Nanotechnology*, April, 2001, pp 100-104.
- [62] Nguyen, Clark T.C "Micromechanical Signal Processors", Ph.D. Dissertation, Dept. of Electrical Engineering and Computer Sciences, University of California at Berkeley, December 1994.
- [63] Beer, F., Johnston Jr., E., and DeWolf, J., Mechanics of Materials, 4th Edition, New York, McGraw-Hill, 2006.

- [64] Teva, J., et. Al., "On the Electromechanical Modeling of a Resonating Nano-Cantilever-Based Transducer", *Science Direct – Ultramicroscopy*, August, 2004, **100**, Issues 3-4, pp 225-232.
- [65] Cho, A., "Researchers Race to Put the Quantum into Mechanics", *Science*, 3 January 2003, **299**, pp. 36-37.
- [66] Gaidarzhy, A., Zolfagharkhani, G., Badzey, R., and Mohanty, P., "Spectral Response of a Gigahertz Range Nanomechanical Oscillator", *Appl. Phys. Lett.* **86**, 17 Jun 2005, pp. 254103-1 - 254103-3.
- [67] Juethner, Konrad, "COMSOL Error Message," COMSOL, Inc. Technical Support e-mail correspondence, 9 Aug. 2006.
- [68] Monroe, C., and Wineland, D., "Quantum Computing with Ions," *Scientific American*. August 2008, pp. 64-71.
- [69] Brownutt, M., et al, "Monolithic Microfabricated Ion Trap Chip Design for Scaleable Quantum Processors," *New Journal of Physics*, **8**, 2006, pp. 232-247.
- [70] Madsen, M., Hensinger, W., Stick, D., Rabchuck, J., Monroe, C., "Planar Ion Trap Geometry for Microfabrication," *Appl. Phys. B*, **78**, 2004, pp. 639-651.
- [71] Monroe, C., et al, "Development of Microfabricated Linear Ion Traps for Scalable Quantum Computing", 2003,
<http://www.iontrap.umd.edu/research/trap/trap_dev_2003_files/frame.htm>.
- [72] The MathWorks, MATLAB R2008, www.mathworks.com/products/matlab
- [73] COMSOL Inc., COMSOL Multiphysics 3.4, <www.comsol.com>.
- [74] Dehmelt, H., Radiofrequency Spectroscopy of Stored Ions I: Storage. *Adv. At. Mol. Phys.*, **3**, 1967, pp. 53-72.
- [75] Quarles, T.; Pederson, R.; Newtown, R.; Sangiovanni-Vincentelli, A.; Wayne, C. Simulation Program with Integrated Circuit Emphasis (SPICE). EECS Department of the University of California at Berkley
<<http://bwrc.eecs.berkeley.edu/Classes/icbook/SPICE/>>
- [76] Madsen, M., "Advanced Ion Trap Development and Ultrafast Laser-Ion Interactions." Thesis Dissertation, University of Michigan, 2006.

LIST OF ACRONYMS

a	acceleration, or distance from fixed end
A	depends on context; cross sectional area of beam (mechanical systems), adenine (base found in DNA), a subsequence in a DNA strand, or a chemical reactant
A*, A'	complement of an A subsequence
AC	alternating current
AFRL	Air Force Research Laboratory
B	depends on context; a subsequence in a DNA strand, or a chemical reactant
B*, B'	complement of an B subsequence
BioCOMP	Bio-Computation Program
BioFlips	Bio-Fluidic Chips Program
Bio-SPICE	Biological Simulation Program for Intra-Cellular Evaluation
c	damping constant
C	depends on context; cytosine, a subsequence in a DNA strand, or the product of a chemical reaction
C*, C'	complement of an C subsequence
CAD	computer-aided design
CME	Chemical Master Equation
CMOS	Complementary Metal Oxide on Silicon
CTC	core technical competency
D	a subsequence in a DNA strand
D*, D'	complement of an D subsequence
DARPA	Defense Advanced Research Projects Agency
DC	direct current
DNA	deoxyribonucleic acid
E	Young's Modulus, or Modulus of Elasticity
ETCH	Establishing Tools for Computing Hybrids Project
FEA	Finite Element Analysis
FEM	Finite Element Method
G	guanine
H	depends on context; convection heat transfer coefficient, or a hairpin
HCR	Hybridization Chain Reaction
I	moment of inertia
IC	integrated circuit
k	thermal conductivity in heat transfer systems; spring constant in mechanical systems
$k_{forward}, k_{i,j}$	forward reaction rate
$k_{reverse}, k_{j,i}$	reverse reaction rate
L	length
m	mass constant
MEMS	microelectromechanical systems
M&S	modeling and simulation

NEMS	nanoelectromechanical systems
NN	nearest neighbor
ODE	ordinary differential equation
P	depends on context; load, or a folded hairpin
PCR	Polymerase Chain Reaction
PDE	partial differential equation
q	heat flux
QuIST	Quantum Information Science and Technology
RF	radio frequency
R&D	research and development
RRE	Reaction Rate Equation
SCHETCH	Simulation Concept – How to Exploit Tools for Computing Project
SIMBIOSYS	Simulation of Biological Systems Program
SPICE	Simulation Program with Integrated Circuit Emphasis
SSA	Stochastic Simulation Algorithm
STM	Scanning Tunneling Microscope
SWCNT	Single Walled Carbon Nanotube
t	depends on context; time, temperature, beam thickness
T	depends on context; temperature, thymine (base found in DNA), or a subsequence in a DNA strand
T*	complement of a subsequence in a DNA strand
v	velocity
V	voltage
VHSIC	Very High Speed Integrated Circuit
VHDL	VHSIC Hardware Description Language
VLSI	Very Large Scale Integration
w	width of beam
WC	Watson Crick
x	displacement
ΔG	Gibbs free energy
ΔH	enthalpy
ΔS	entropy
ϵ	emissivity
f_n	natural frequency
λ	electron/phonon mean free path
ρ	density
σ	Stefan-Boltzmann's Constant

APPENDIX - Publications

Structural Optimization of Dendritic DNA Self-Assembly

Note: This poster paper was presented at the 2009 Foundations of Nanoscience Conference (FNANO), Snowbird, Utah, April 2009, but the proceedings with the full paper have not yet been published.

Structural Optimization of Dendritic DNA Self-Assembly

Morgan A. Bishop¹, Clare D. Thiem¹, Thomas E. Renz¹, Erik Schultes², Harish Chandran², and John Reif²

¹Information Directorate, Air Force Research Laboratory, Rome, New York

²Department of Computer Science, Duke University, Durham, North Carolina

Hybridization Chain Reaction (HCR) is a bottom-up signal amplification technique that has been proposed during the use of DNA target recognition experiments. Its recursive self-assembly reaction is an intriguing alternative to conventional Polymerase Chain Reaction (PCR) based signal amplification. In this paper we present three modifications to the original recursive dendritic structures that were previously used in HCR experiments. We optimize the activating hairpin structure from the point of view of false positive reduction, signal amplification and structural integrity while still maintaining a simplistic recursive reaction. The outcome of our optimization is shown in Figure 1.

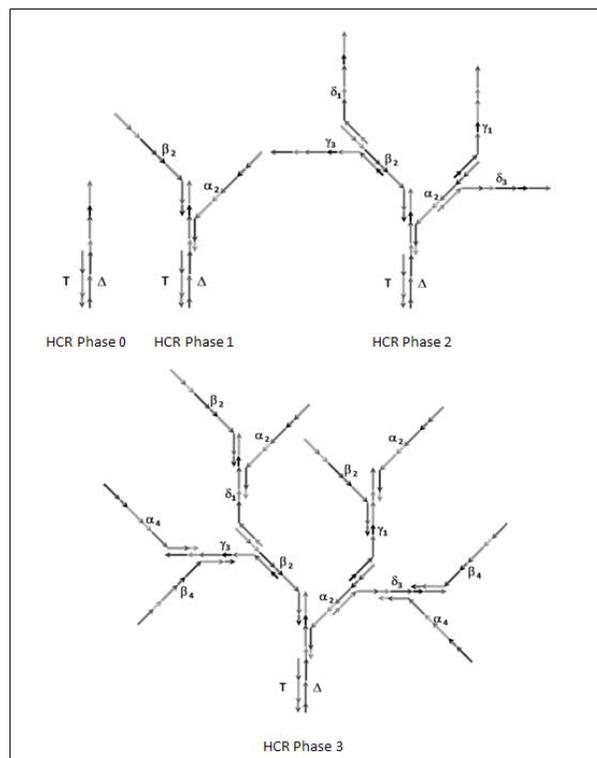


Figure 1 – Example of our Hybridization Chain Reaction employing nine optimized hairpin sequences.

1. Introduction

Reliable recognition and reporting of extremely small numbers of target DNA strands plays an important role in diverse areas from medical diagnosis to molecular taggants for controlled materials. One problem is that molecular recognition events provide low reporting signal per event, requiring significant instrumentation for detection of small amounts of target material. DNA recognition amplification techniques such as the PCR significantly increase the available report signal by increasing the availability of recognition strands and their associated reporting mechanism [1]. Amplification occurs as long as there

is more target material for the increased recognition strands to react with. When the target material is exhausted, amplification ends for recognition amplification. A relatively new technique, HCR exists that can provide an autocatalytic amplification of the reporting material without requiring an amplification of the recognition material or large amounts of target material. The initial HCR technique, involved the use of two complimentary catalyst hairpin strands, one of which was tailored to recognize an initiator or trigger strand [2]. The complimentary hairpin strands then open each other in turn by a chain reaction that is fueled by the energy released as the hairpins open [3]. A large concatenated duplex chain's results can then be isolated by gel electrophoresis. Amplified reporting signal is achieved in the growing chain by attaching an emitter such as a fluorophore to each catalyst strand. Attaching a quenched fluorophore allows the use of real-time detection as the signal strand grows [4].

The chain based HCR amplification rate is limited to a simple linear progression depending on the number of trigger molecules and eventually, bio-processing errors limit the size of the concatenated strand. A way around this is to design the catalyst strands to form dendrimer branches. As each catalyst strand opens, more than one new initiation site can be formed to react with a complimentary strand. The resulting structure grows at an exponential rate and because it is volume limited, can achieve a much larger agglomeration [5, 6]. This paper describes three design improvements to improve HCR yield in dendritic self-assembly systems.

2. HCR Modifications

Our *first modification* reduces the false positive rate by creating a specific external toehold for the initiator hairpin. The analyte to be detected, denoted T, will systematically interact with only this initiator strand, denoted Δ , during the target recognition phase of the reaction. Thus, the toehold length may be optimized to maximize signal-to-noise ratio without negatively affecting the HCR amplification phases. In addition, the concentration of this hairpin will establish a tree-to-branching ratio which is related to the expected number of growing trees within a reaction versus the expected amount of branching within a tree. We can predict the average amplification within a tree using the ratio $\Delta:H$, where Δ and H are the concentration of the initiator strand and all other hairpin strands, respectfully:

$$\text{Average Amplification} = \frac{2H}{\Delta} \quad (1)$$

The *second modification* helps achieve exponential amplification beginning at the first stage of the reaction and creates more equally stable hairpin structures before a HCR begins. Exponential amplification is achieved by creating two amplification binding sites, one which is interior (IBS) and one which is exterior (EBS), immediately following the target recognition trigger (Phase 0 in Figure 2). Each of these sites will then be amplified exponentially throughout the HCR. See Figure 2. More equally stable hairpins, or those with similar melting temperatures, will help ensure that the different types of hairpins will have similar reaction rates regardless of environmental conditions. With these proper hairpins, poor tree growth will occur at the consequence of the entire system as opposed to an individual component. Clearly, systematic failure is a feature which is easier to detect experimentally than single component failure. Thus, error rates associated with environmental conditions are more stable and error recognition can be deciphered more readily.

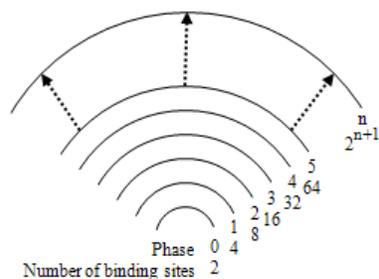


Figure 2 – Dendritic nanostructure complex with phase-specific number of hybridization sites.

The *third modification* alleviates an identified potential structural issue while minimizing the required number of extra distinct hairpins. The prevention of this structure requires four additional hairpins, but they are products of the same DNA sub-strands that exist in the previous modification’s hairpins. If this structure being prevented was to arise merely once in a single dendritic tree it would cut any future potential signal amplification from that branch site in half. The structure that this modification prevents is shown in Figure 3.

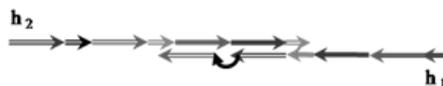


Figure 3 – Hairpin (h_2) binding to both IBS and EBS of another mediated hairpin (h_1).

With all three modifications employed, we can experience a more structurally sound dendritic tree recursive reaction. Table 1 shows the new phase growth with respect to the specific hairpin binding sites. These sites may then be targeted by signal emitters such as flourophores or gold nanoparticles.

Table 1 – Phase growth with respect to number of specific hairpin binding sites

Sites		Phase									
		0	1	2	3	4	5	6	7	8	
bc*	bc*	1		2		8		32		128	
	ad*	1		2		8		32		128	
ca*	ca*		1		4		16		64		
	db		1		4		16		64		
db*	db*		1		4		16		64		
	ca		1		4		16		64		
b*c*	b*c*			2		8		32		128	
	a*d*			2		8		32		128	

3. Future Work

The primary focus of this paper is the optimization of dendritic structures resulting from HCR reactions. This is a manually intensive process requiring considerable expertise to implement correctly. Interaction between the Air Force and Duke research teams has clarified the complexity of defining the proper dendritic structure for HCR experiments. The next step is to bring together the software tools which will capture the fundamentals for sequence design and will result in a robust approach for generating libraries of optimized hairpins. Initial sets of potential libraries consisting of substrands for composing analyte and hairpin sequences will be generated using a tool such as SynDCode [7]. The final library will then be chosen from the set by employing an adapted PairFold algorithm, which is a more complex but provides a more exact calculation of DNA stability, in order to further optimize the hairpins [8]. The structural integrity of the final library will then be checked using folding and hybridization software. Currently we are using Mfold [9], but we would like have the ability to check the interaction/reaction of more than two sequences, so we will be looking at alternatives. We will also pursue concepts for virtual verification simulations of the desired sequence interaction/reaction using the refined sequence library prior experimental verification. One approach that has been suggested for this step is to look at Gillespie-based kinetics.

4. Conclusions

This paper presents a brief look at an effort to optimize a HCR structure resulting from the point of view of false positive reduction, signal amplification and structural integrity. Three modifications to the original recursive dendritic structures were presented increasing the number of hairpins required for the HCR from three to nine. These modifications allow us to create a more methodically correct reaction which optimizes proper hybridization while reducing false-positive noise often associated with self-assembly experiments. The next step is to develop the necessary computer design applications to create these equally stable hairpins which will lead to more predictable, repeatable and quantitative HCR experiments.

5. Citations

- [1] K. Mullis, F. Faloona, "Specific Synthesis of DNA in Vitro Via a Polymerase-Catalyzed Chain Reaction," *Methods in Enzymology*, 1987, **155**, 335-50.
- [2] R. Dirks, N. Pierce, "Triggered Amplification by Hybridization Chain Reaction," *Proc. Nat. Acad. Sci., U.S.A.*, 2004, **101**, 15275-15278.
- [3] S. Green, D. Lubrich, A. Tuberfield, "DNA Hairpins: Fuel for Autonomous DNA Devices," *Biophys. J.*, 2006, **91**, 2966-2975.
- [4] D. Chemeris, Y. Nikonorov, V. Vakhitov, "Real-Time Hybridization Chain Reaction," *Doklady Biochem. and Biophys.*, 2008, **401**, 53-55.
- [5] R. Corn, "Programmable Self-Assembly of DNA-Dendrimer and DNA-Fullerene Nanostructures," *DTIC*, 2004, ADA428157
- [6] D. Smith, "*Dendritic Supermolecules – Towards Controllable Nanomaterials*," *Chem. Commun., Roy. Soc. Chem.*, 2006, 34-44
- [7] M. Bishop, A. Macula, W. Pogozelski, T. Renz, V. Rykov, "SynDCode: Cooperative DNA Code Generating Software," *3rd Annual Conference of Foundations of Nanoscience*, 2006, 227-231.
- [8] M. Andronescu, A. Rosalia, A. Condon, and H. Hoos, "RNAsoft: a suite of RNA secondary structure prediction and design software tools," *Nucleic Acids Research*, 2003, **31 (13)**, 3416-3422. *Software available at: www.RNAsoft.ca.*
- [9] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic Acids Res.*, 2003, **31(13)**, 3406-3415.

Approved for Public Release, 88AWB-2009-0825