Critical Design Review:

Silicon Wafer Temperature Distribution

Sponsors:
John Mazzocco, Client
Daniel Greenberg, Client

Team Members:
Daniel Gonzalez
Nathan Jones
Dylan Justice

CPGRID

Team Advisor:
Sarah Harding: sthardin@calpoly.edu

1 Contents

2 List of Figures

3 List of Tables

4 Introduction

For the design of heat treatment processes in the silicon and semiconductor industry, engineers need a way to determine transient wafer temperature distribution without finite element analysis (FEA) models and software. For our senior project, Applied Materials (AMAT) has presented us the task of designing and validating a system which will provide their engineers an accurate program to model transient wafer temperature distribution. Our team, CPGRID (**C**al **P**oly **G**raphic **R**epresentation of **I**ntegrated Circuit Temperature **D**istribution), consists of teammates Nathan Jones, Daniel Gonzalez, and Dylan Justice. We are currently all undergraduate mechanical engineers working under advisor Professor Sarah Harding of the Mechanical Engineering Department. Our goal is to provide Applied Material's team, more specifically John Mazzocco, Daniel Greenberg and other AMAT engineers with a fast, accurate, and accessible program that analyzes a variety of heat treatment processes.

5 Faculty Resources

5.1 Kim A. Shollenberger, Ph. D.

Dr. Shollenberger has been an associate professor at California Polytechnic State University in the Mechanical Engineering department since 2002. Her areas of interest include computational heat transfer, diagnostic development and experimental design for thermal/fluid systems. Dr. Shollenberger has been advising our analysis methods and assumptions for each case as well as helped us in our hand derivations of each thermal element in the problem. We have also received direction from her for graduate level computational heat transfer texts to consult to further our understandings.

5.2 Richard Savage, Ph. D.

Dr. Richard Savage of the Cal Poly MATE department has been a great resource to us throughout our project. Having worked in Silicon Valley during his time in industry, he is familiar with the challenges associated with silicon processes. Approachable and knowledgeable, he graciously allowed us the use of a vacuum annealing chamber in his clean room lab. He has also helped us out with the support of a MATE department technician, Christine Ghent. She will train us to use the lab equipment and assist in testing.

6 Background

Applied Materials is a global company that has been manufacturing and supplying equipment and providing services utilized for silicon wafers, display, energy and environmental systems since 1967. Their headquarters are based out of Santa Clara, California but they operate in 18 other countries including China, Germany, Switzerland, and numerous others ("Applied Materials" 5).

Our project focuses on only one branch of their business: the silicon systems group. This section of the company specializes on several proprietary processes including atomic layer deposition (ALD), chemical vapor deposition (CVD), physical vapor deposition (PVD), electrochemical plating (EP), and others. Of which, our analysis hones in on the various types of CVD and PVD, as well as their pre- and post-processes.

Chemical Vapor Distribution (CVD) is a subset of semiconductor device fabrication. Integrated circuits, found in everyday electronic devices, go through extensive chemical processing to reach their desired functionality. Electronic circuits are gradually created on a pure semi-conducting wafer (usually silicon) using multiple-step processes. One of the processes is CVD.

In a typical CVD process, the wafer is placed in a chamber, called a reactor. The semiconductor is then exposed to a precursor (an exotic gas) which will react with the substrate surface. This has two main consequences: (1) a high-performance deposit left on the surface of the wafer and (2) other by-products are removed by the flowing gas in the chamber.

Our analysis will be based upon a 'shower head' reactor design. These reactors have a distinct planar surface covered with small perforations that look and function much like a shower head. The porous surface is used to dispense reactant gasses uniformly onto the wafer surface. Other geometric features include the heater and chamber walls. All three of these elements can occupy a range of temperatures.

Typically, only the surface heating the wafer needs to be at the correct temperature for the process to occur. The shower head and the chamber wall are generally colder. This forces the gasses to fall onto the wafer and flow off the edges to an exhaust port. It is also common for the temperature of the chamber walls to be slightly above the ambient temperature to avoid condensation on the walls and in the pores.

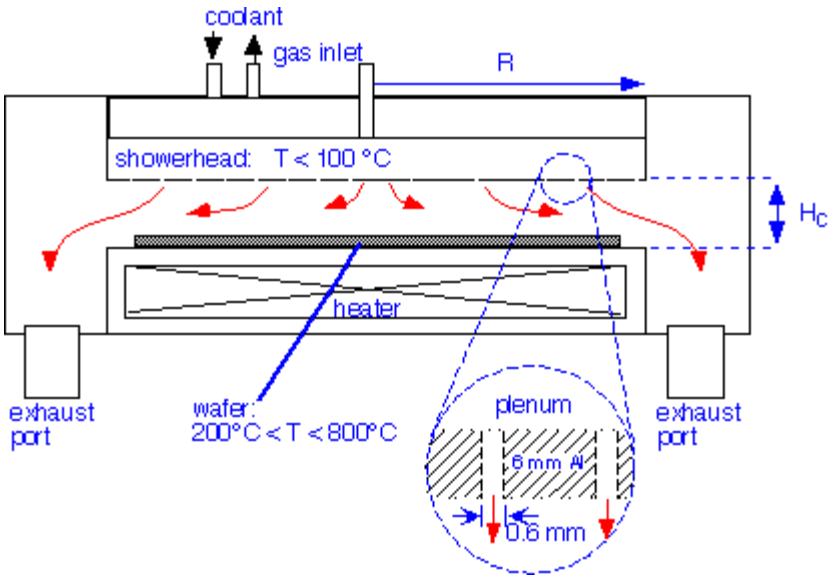


Figure 1. Critical distances when analyzing the CVD process.

One of the most important parameters is the distance between the shower head and the wafer. This is called the ceiling height or electron gap and is noted in Figure 1 as H_C. When this gap is increased, the chamber volume is also increased. This changes the surface-to-volume ratio in a way that will increase the radial flow of gas across the wafer, and in doing so, affect the rate of heat transfer and time to complete each process.

At steady state conditions, heat flowing into the silicon wafer must be balanced by heat flowing out. Our challenge is to model temperature during the transition from its initial temperature to steady state with the chamber.

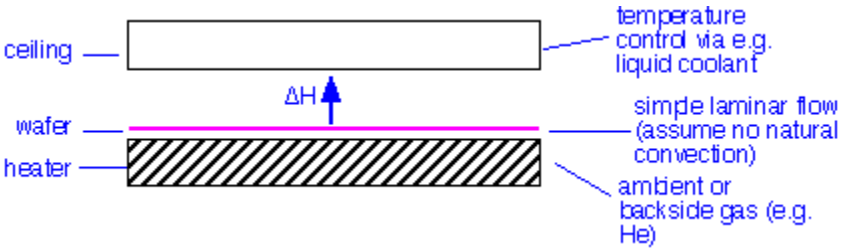


Figure 2. Heat transfer of a thin wafer.

Analysis of the extremely thin wafer can be simplified to heat flow from floor to ceiling, with 1D temperature variation in the radial direction ("Fundamentals of Chemical Vapor Deposition").

Even for relatively flat silicon chips (flatness on the order of microns), contact with the heating pedestal beneath is only made in a few spots. We have also determined that most Applied Materials processes leave the wafer suspended on standoffs above the heating pedestal. As a result, the bulk of heat transfer to the wafer is due to radiation, conduction through the small gap of gas on the back side of the wafer, and, in some cases, convection through the gas on the more spacious front side. Main modes of

heat loss from the wafer will depend on input conditions. After advising and background research we have determined some general guidelines to keep in mind as we analyze this problem. Convection is significant only for high gas flow rates, conduction will be important if the shower head is significantly close to the wafer, and radiation is negligible until the conditions reach relatively high temperatures. We will begin building our program around the most basic case of heat transfer: a transient cooling of a wafer in which radiation and convection will be neglected. Our program will later expand to cover all the necessary modes of heat transfer. We plan on consulting with Applied Materials throughout the analysis in order to represent their processes as accurately as possible.

7 Objectives

Our overall goal is to design and build a program which solves for the transient temperature distribution of a silicon wafer for a variety of thermal processes.

7.1 Requirements and Specifications

We have prepared a list of realistic requirements and specifications to aid us in our program design. Successful completion of each of these will help ensure the reliability of our program.

7.2 Deliverable Requirements

- This system must be fast, accessible, and incorporate plotting functionality. Our sponsor has strongly suggested we use Microsoft Excel to accomplish this. Current FEA systems are tedious, time intensive, and expensive. The use of Microsoft Excel would allow for a familiar user environment which is available on most company computers.
- The system must be accurate. Our sponsor has requested a temperature resolution to one decimal place (xxx.x °C). When actual silicon wafer temperatures deviate from design temperatures, the wafer functionality can potentially fail, either by plastic deformation or unacceptable resulting properties.
- The system will require validation. We will pursue this on two fronts:
    1. Existing FEA or FD tools, such as Abaqus and MATLAB
    2. Experimental data (see next)
- Experimental data must be used to confirm our program. Our team will either, design and build a test fixture to simulate environmental conditions, or use an existing fixture in a Cal Poly facility. When further validation is required, Applied Materials will supply the equipment to reach higher temperature and lower pressures (~500 °C, 1×10^{-09}torr).

7.3 Testing Requirements

- Testing apparatus and procedures must be applicable in confirming results of our analytical heat transfer analysis.
- All required hardware and software purchases must be with our allocated budget of $2500.
- Testing results must have the capability to reach at least the same temperature resolution that our program outputs to confirm temperature uncertainties.

7.4 Levels of Risk

To achieve all of our specifications for the project, we plan on conforming to an iterative design and test process. The final revisions of the program should meet the following specifications.

Table 1. Specifications for the program.

Spec #	Parameter Description	Requirement or Target	Tolerance	Risk	Compliance

1	Accuracy	xxx.x	+/- 0.1 °C	H	A, T
2	Time to Execute	5 seconds	--	L	I, S
3	Time to Learn	15 minutes	--	M	I,S,T
4	Time to Setup	15 sec	--	L	I,T

In Table 1 above, there are three levels of risk and four categories of compliance:

Risk

- High
- Medium
- Low

Compliance

- Analysis
- Testing
- Inspection
- Comparison to similar designs

These levels are distinguished by their influence on the final value of our product. The higher risk specification represents a combination of the most difficult tasks and the most valued deliverable requirements to the client. The paramount specification is to build an accurate program. Applied Materials has requested accuracy to one decimal place. Reaching the desired accuracy will require detailed heat transfer analysis. This is also the main interest of Applied Materials and the reason for the project. We will be able to test and determine the feasibility of this resolution through both analysis and testing procedures. Specifications 2, 3 and 4 are less restricting as we are willing to sacrifice time to execute to help ensure greater accuracy with our calculations.

The time to execute, time to learn and time to setup are less critical. The current timeframe for Applied Materials to get results from their current heat transfer analysis workflow takes weeks. Our program will execute orders of magnitude faster, so the specific times are less important.

8 Method of Approach

In order to deliver a finished solution that functions as well as possible, we began by developing a thorough understanding of our client's requirements. Our project differs from a traditional design project in that we are modeling a process rather than creating a novel design. This means that the bulk of our effort has been spent researching, modeling, and validating to ensure that our solution provides accurate results.

8.1 First Quarter

Our first challenge was to understand and model the heat transfer problem, based on the process schematic and list of variables provided to us by our sponsor. Bi-weekly meetings with our sponsor have given us an opportunity to update them on our models, and ensure we are staying on track toward a solution they will be satisfied with. We also ask Professor Kim Shollenberger at Cal Poly for regular help and guidance while working on the heat transfer analysis. Proprietary information regulations have limited our resources and cut off specific details concerning processes and geometric models (CAD). This has also forced us to use simplified geometry for analysis which may lead to inaccuracies.

8.2 Second Quarter

After we are confident in our heat transfer analysis, we can proceed to working on the deliverable: a simple computer model that can handle all required variables. As we work on the simple model that we will present to our sponsor, we can validate with existing finite element programs. We fully anticipate these processes to be a continuous iterative cycle of analysis, design, and testing.

This iterative process is shown graphically in our Project Workflow diagram shown in Figure 3. This diagram describes how the main problem statement was split up into two, partially independent, branches that make up the design phases. **Branch 1**directly follows tasks related to the primary objective of the project: to develop heat transfer models and the Excel VBA program.

Branch 2 displays the validation methods which can be divided into two sub branches. **Branch 2A** represents validation using existing finite element modeling codes, while **Branch 2B** represents physical testing, data acquisition, and data analysis methods. **Branches 1 and 2** are executed simultaneously throughout our project so that we can work on a continuous cycle of refinement, and spend our time as efficiently as possible. By the time our detailed analysis and model is complete we need to have test results available for comparison.

Branch 1 will meet with one part of the second segment (**Branch 2A**) when we check our Excel results against FEA or FD software results (**Node C**). After we have achieved satisfactory agreement between our Excel program and existing FEA codes, we will proceed to **NodeD** of the flow diagram. This point represents the final phase of our project.

8.3 Third quarter

The final phase of our project will be to validate our model with physical experiment. We expect to simulate simple cases here at Cal Poly. **Branch 2B** represents tasks associated with physical testing, and intersects with **Branch 1** at **Node D.NodeD** represents comparison between the results of our Excel program and the results of our physical testing. This step is essential, we need to know that the simplifying assumptions we made while modeling are valid. If we don't see agreement at this stage it should be a warning that either our model is over simplified, or that our test setup is inadequate to represent the situation. We will use discretion to improve one or the other until we see agreement between modeling and physical test.

We also plan to test more extreme cases (high temp, low pressure, exotic gases) at existing Applied Materials testing facilities. The results of these tests will allow us to refine our model so that it produces accurate results for any scenario.

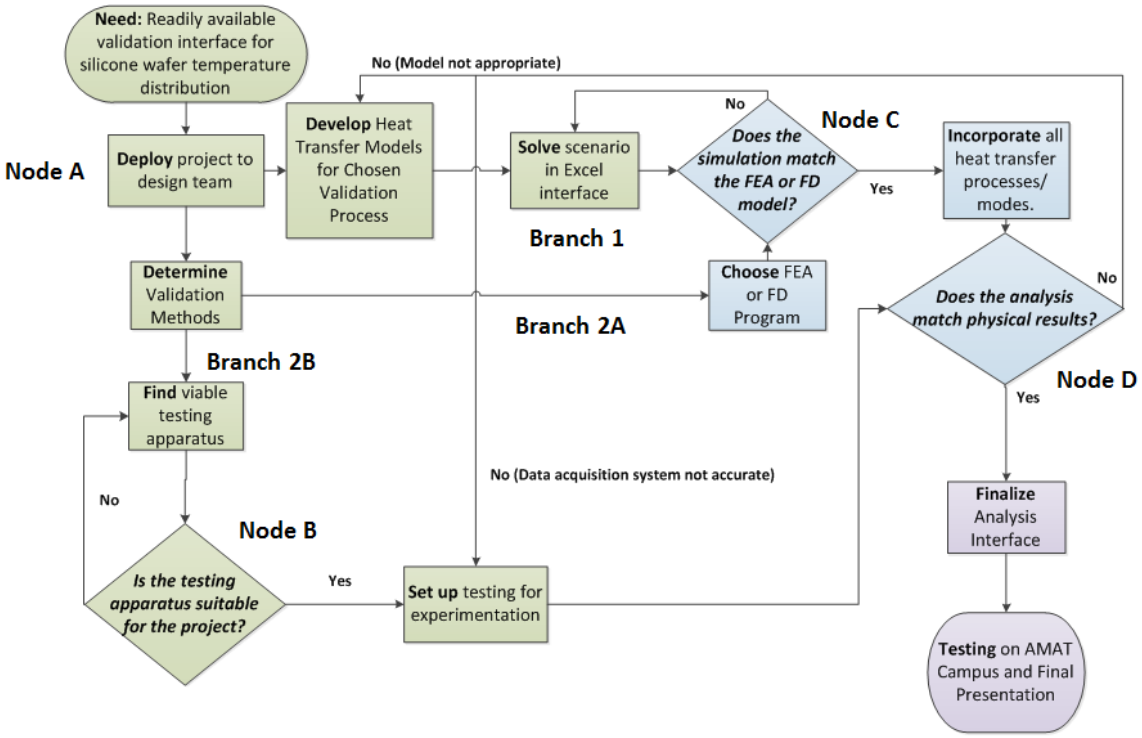


Figure 3. Workflow diagram.

9 Management Plan

In the initial stages of our project, we worked closely as a team to understand the requirements and problem definition. We also worked together to gather background information, communicate with our sponsor, and set up the flow for the rest of our project. This has been beneficial for keeping the team on the same page, focused on the same goal as we move forward with the project.

As we move on to creating the deliverable we have opted to divide the labor based on team member strengths. Nathan has previous experience with Excel VBA Programming so he will most likely create the architecture for our computer model. Daniel and Dylan are currently taking heat transfer, and so will be well prepared for the analysis side of our project. As we translate our analysis into a computer model we hope to work together so that all team members can develop a complete understanding of the solution.

At this point in our project, each of our team members has a detailed understanding of the necessary background information, problem definition and how to move forward. We have established bi-weekly meetings with our sponsor where we present our recent progress and our next steps. All the necessary materials and resources have been arranged for us to begin taking experimental data as soon as our parts arrive. It is most efficient to work on different aspects and simultaneously collaborate as we proceed since our project emphasis is dependent on testing rather than design. This also gives each of us an area of expertise in the project.

The main deliverable is an excel VBA driven program. We have successfully demonstrated the capabilities of the program as a proof of concept to compute the transient solution for a one dimensional scenario. By inspection, our results appear to be realistic. However, we made a few assumptions that may or may not be applicable, one of which was the emissivity of silicon. This is a very difficult value to calculate and at high temperature it has a large effect on the radiation heat flux. To investigate this we adjusted the emissivity value by $\pm$ 10 percent and plotted the results along with the nominal. As expected, the values deviated a substantial amount. We are anticipating similar discrepancies throughout the project.

To help eliminate these errors, we will be testing numerical solutions with hand calculations, finite element analysis, collected data that we take and data from our sponsors. Combining the results from each of these, we will be able to develop an accurate program.

10 Heat Transfer Analysis

The processes that Applied Materials puts their silicon wafers through varied in several domains including: geometry, pressure, temperature, and material properties. For our team, this meant that we had to develop a dynamic program that could differentiate and decide which case it was solving. For this to occur, we have to develop the dividing lines between those cases.

10.1 Program Capabilities

Since we were developing the software from the ground up, there was no practical way to ensure that every known case of heat transfer was covered by our solver. With this in mind, Table 2 summarized the overall assumptions that should have been met to use the heat transfer analysis tool and expect accurate results.

Table 2. Program Capabilities and Model Assumptions.

Program Capabilities and Model Assumptions	
Heat Transfer Modes	Conduction Convection Radiation
Classification of Differential Equation	Wafer: 1-Dimensional, Transient (Partial Differential Equation); Explicit and Implicit Solving: Temperature iteration and forward time steps Gas: 0-Dimensional, Transient (Lumped Capacitance, Ordinary Differential Equation); Explicit Solving Method: Forward time steps with current heat parameters Walls: Quasistatic radiation network; Implicit solving method
Geometry	Thin Cylindrical Disk and Surrounding Cylindrical Volumes
Physical Properties	Constant Pressure Boundary conditions constrained to program inputs Assumed open system in vacuum environment of quiescent air above and below wafer Convection coefficients are estimated according to empirical observations Global convection coefficients, $\bar{h}$, are taken as local convection coefficients, h Linear interpolation of properties Estimated emissivities for surfaces within the chamber Losses to chamber environment are approximated by steady state analysis

The heat transfer modes included conduction within the gas and solids. Free convection covered the state of the environment where no appreciable external gas flow was driving the internal flow dynamics and the only source of fluid movement was density gradients due to the existing temperature gradients. Forced convection varies as a function of both the type of input flow and how the system interacts with that flow. This component of convection was a less reliable number because approximation of the type of flow of the incoming gas had significant uncertainty. Radiation was approximated by finding appropriate models for each component of our system (i.e. gray body characteristics, reradiating surfaces, black bodies, and each respective view factor).

The types of differential equations that could be solved by our system include the following: zero dimensional, transient, first order, ordinary differential equations and one dimensional, transient, partial differential ordinary differential equation. The former had much narrower applicability because it oversimplified the actual physical problem and it only solved a forward explicit time step problem. The latter was able to iterate to solve all the types of problems we encounter but it took more computing power, thus more time to converge to a solution.

The geometries that were approximated by our models were the silicon wafer and the chambers that they interacted with are both cylinders. Therefore, any volume that existed within our model was cylindrical and was only applied to forms that closely resemble cylinders.

We made physical property assumptions that should also be taken into consideration when the program will be utilized. The pressure changes within the chamber were negligible. Our program also constrained the user inputs to be constants. Since the convection coefficient tends to be the most unreliable characteristic of our system, we will approximate the local convection coefficient, h, as the global convection , $\bar{h}$. Lastly, the thermophysical properties will be approximated as linearly interpolated values retrieved from published property tables.

For the walls, we assumed the bodies radiating in a quasistatic state from time to time step. There are no discontinuities in the temperature profile so this assumption caused no foreseeable problems.

10.2 Solving the Simplest Case

The zero-dimensional case where the silicon wafer was lumped into a point mass or capacitance only has temperature varying as a function of time. This solved the physical problem of a cylindrical disk as a wafer which had convection on both sides of the wafer (2), conduction from the gas in contact with the wafer (3), and radiation from two surfaces (4,5) all equal to the time rate change of energy storage within the wafer.

(1) (2) (3) (4) (5)

$$\rho z c_p \frac{\delta T_w}{\delta t} = -\left[h(T_w - T_\infty) + k_{gas}\left(\frac{T_w - T_\infty}{H_1}\right) + \varepsilon\sigma\left({T_w}^4 - {T_{s,1}}^4\right) + \varepsilon\sigma\left({T_w}^4 - {T_{s,2}}^4\right)\right] \quad \text{(Equation 1)}$$

Despite Equation 1 being a linear, first order differential equation, there existed an inherent nonlinearity in sections 1 through 5 this equation that was associated with the thermophysical properties of the silicon wafer and the surrounding gases. Also, segment 5 had a fourth order polynomial in T. For this reason, an analytical solution was not possible and a numerical method was utlized. We chose the Euler Method to solve our problem.

Before we could implement the Euler Method in our program to solve for an unknown result, we had to test our solver with a known solution. We did so by solving the following ordinary differential equation in our Excel VBA program.

Find $x(t)$ if $\left.\frac{dT}{dt}\right|_{t=0} = 0$ and $T|_{t=0} = 1$ for the following ordinary differential equation:

$$\frac{dT}{dt} + T = 0 \qquad \text{(Equation 2)}$$

Analysis:

1. If solution is assumed of the form: $T(t) = \sum_{i=1}^{n} C_i e^{\lambda_i t}$
2. The characteristic equation of the differential equation turns out to be the following: $\lambda + 1 = 0$
3. The eigenvalue of the differential equation is the following: $\lambda = -1$.
4. This implies that the system's time constant, τ = 1 second.
5. The solution to the differential equation is: $T(t) = e^{-t}$.
6. At time $t = \tau = 1\ second$, the $T(\tau) = \frac{1}{e} = 0.376$.

Now, we test Euler's Method on this differential equation displayed in Figure 4.

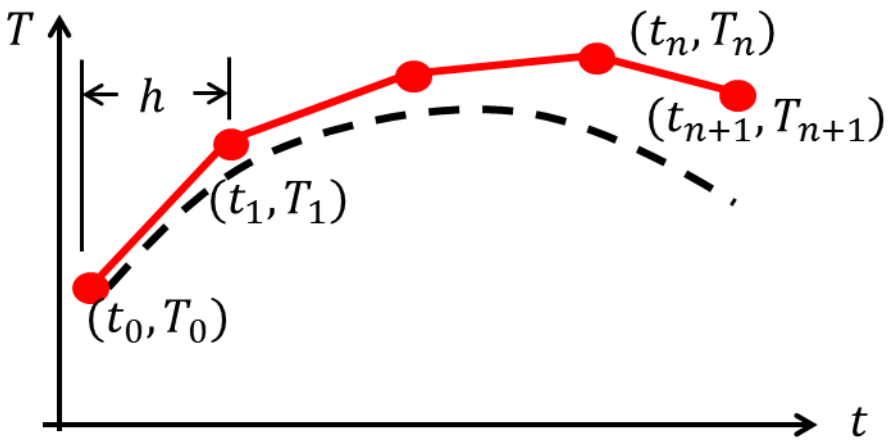
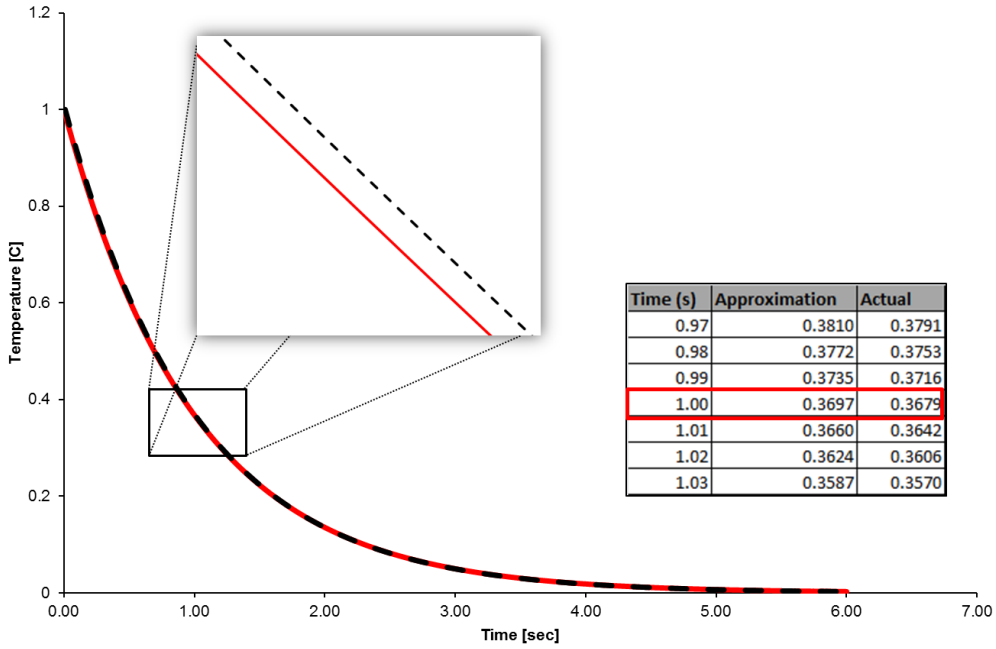


Figure 4. Euler method variable illustration.

Euler's method is dictated by the following algorithm: $T_{n+1} = T_n + hf(t_n, T_n)$ where n denotes the present value for time and in our case temperature and $n + 1$ denotes the next discretecoordinate pair. Both T and t carry these subscripts in order to fully define a coordinate in (t, T) space. The horizontal scaling factor h denotes the length of each discrete time step. For testing purposes, we

chose h to be 0.01 seconds. The function $f(t_n, T_n)$ was dictated by the slope described in the original differential equation. In the case above, $f(t_n, T_n) = \frac{dT}{dt} = -T$. When all of this information was gathered, an explicit loop was programmed into our program to cycle through $t = t_0$, the initial time until $t = t_{end}$, the final time: this was the domain of our system. The number of time increments is a function of the difference between these times and the time step size: $N = \frac{t_{end} - t_0}{h}$. Having a smaller time step allowed for more accuracy but takes more computing time.

The results of the implemented algorithm are reported in Figure 5.



Time (s)	Approximation	Actual
0.97	0.3810	0.3791
0.98	0.3772	0.3753
0.99	0.3735	0.3716
1.00	0.3697	0.3679
1.01	0.3660	0.3642
1.02	0.3624	0.3606
1.03	0.3587	0.3570

Figure 5. Results of Euler Method for sample differential equation

The percent deviation between the actual value and the approximation was 0.48 percent. The method was proven to give accurate solutions and could be taken to the next step. A full proof test would be to test method with a nonlinear ordinary differential equation, but we decided to compare our results with actual experimental data from Applied Materials. These results can be seen in the programming planning procedures section.

This method proved to work with simple cases where there pressure was at less than 5 torr but began to vary significantly against results at higher pressures. This meant that the physical model, thus the mathematical model had to be revamped in order to capture energy distribution that was not encompassed all in one point; a system must be created that captures the energy distribution and a spatial temperature gradient.

10.3 Solving the Comprehensive Case

To take into account a spatial gradient of temperature, a different approximation other than the lumped capacitance method was implemented. Taking away this lumped mass assumption led to a need of a

heat diffusion equation (HDE). This equation can be derived with respect to any type of coordinate system. Two coordinate systems that were logical candidates for options were Cartesian (Figure 6) and Cylindrical (Figure 7). Since our geometry was best approximated as a cylinder, the cylindrical coordinate system was our choice.

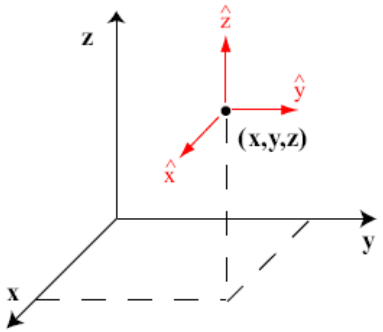


Figure 6. Cartesian coordinate system.

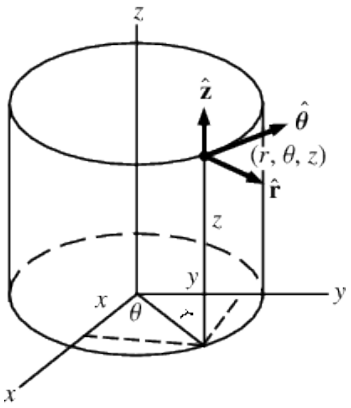


Figure 7. Cylindrical coordinate system.

1 2 3 4 5

$$\rho c_p \frac{\partial T_w}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T_w}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \theta}\left(k\frac{\partial T_w}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T_w}{\partial z}\right) + q \qquad \text{(Equation 3)}$$

The HDE shown in Equation 3 is a second order, nonlinear, partial differential equation with variable coefficients. For our system, we assumed that changes in temperature in the angular direction, θ, or in

the height, z was neglected. This meant that term 3 and 4 drop out of the equation 3 and a less complex equation 4 is derived.

$$\rho c_p \frac{\partial T_w}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(kr\frac{\partial T_w}{\partial t}\right) + q \qquad \text{(Equation 4)}$$

Equation 4 shows the added benefit of using a cylindrical coordinate system compared to a cartesian coordinate system. Instead of having two variables x and y, we have only one variable, r. This ensures that we are optimizing the computing power by eliminating any degrees of freedom that are not necessary.

Another assumption that was made is that the temperature in our system will be symmetrical within the radial domain. This assumption allowed for the reduction of the domain from $-r_{disk} \leq r \leq r_{disk}$ to $0 \leq r \leq r_{disk}$. This reduced the computation error by two. Another consequence of this assumption was that it makes the center of our geometry, or $r = 0$, an adiabatic boundary for our system.

The final HDE that we are left with can be solved analytically for a limited number of cases with constant coefficients and the right boundary condictions. Otherwise, numerical methods was utilized. There were two main methods of solving differential equations: finite element analysis and finite differencing. Since the finite difference approach must be used for the time domain, we also chose to use it for the spatial domain.

The first step in applying this method is to deconstruct our geometry into discrete pieces; this is shown in Figure 8.

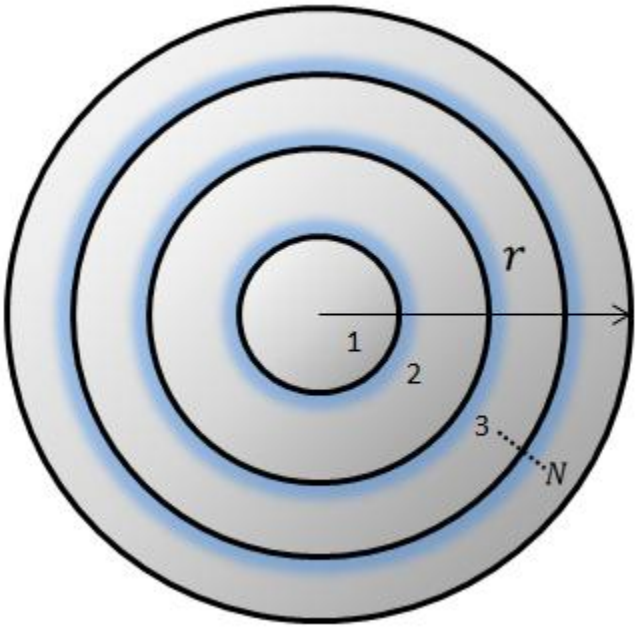


Figure 8. Discretization of the cylinder geometry.

The cylinder was split into Npieces. This integer, N, dictated the mesh size and it was dictated by the user when implemented into the final program. The larger the N value, the more accurate our approximation will be to the actual value.

The accuracy of the method can also be improved by decreasing the size of the time step, Δt. Of course, increasing N and decreasing Δt has a limit that will have to be derived from finite differencing stability criterions. Apart from the stability criterion, these values are limited variably depending on the computational power of the computer that this calculation will be running on.

Every piece created by the discretization is termed an annulus. An energy balance is applied to each annulus and the following system of equations is created at each step in time.

$$\begin{aligned} a_{11}T_1 + a_{12}T_2 + a_{13}T_3 + \cdots + a_{1N}T_N &= C_1 \\ a_{21}T_1 + a_{22}T_2 + a_{23}T_3 + \cdots + a_{2N}T_N &= C_2 \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots & \\ a_{N1}T_1 + a_{N2}T_2 + a_{N3}T_3 + \cdots + a_{NN}T_N &= C_N \end{aligned}$$

$$A \equiv \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}, \quad T \equiv \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}, \quad C \equiv \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} \Rightarrow [A][T] = [C]$$

Figure 9. System of equations derived by analyzing each annulus.

The radiation network can be analyzed the same way an electric circuit is analyzed. The circuit nodes are the energies of each component as a blackbody and as an approximation to the real body; the through variable represents the heat flux. These nodes are separated by the equivalent of circuit resistors that have as values the effective emissivity of each component. Another resistive element that exists between the real bodies is the view factors from body to body. Figure 10 illustrates an example geometry that we might consider. Figure 11 below is a representation of a sample radiation network derived from the system in Figure 10. This network would be used to find the heat flux due to radiation at each annulus of our finite difference model when given a guess temperature of the radiating surface.

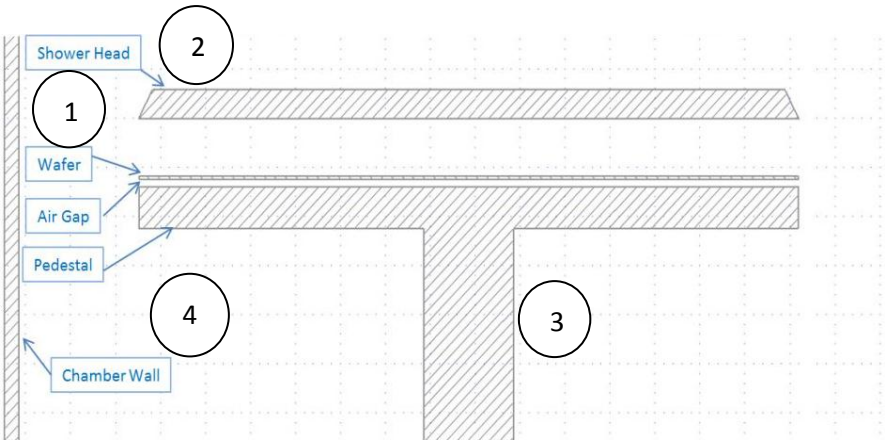


Figure 10. Sample surrounding geometry.

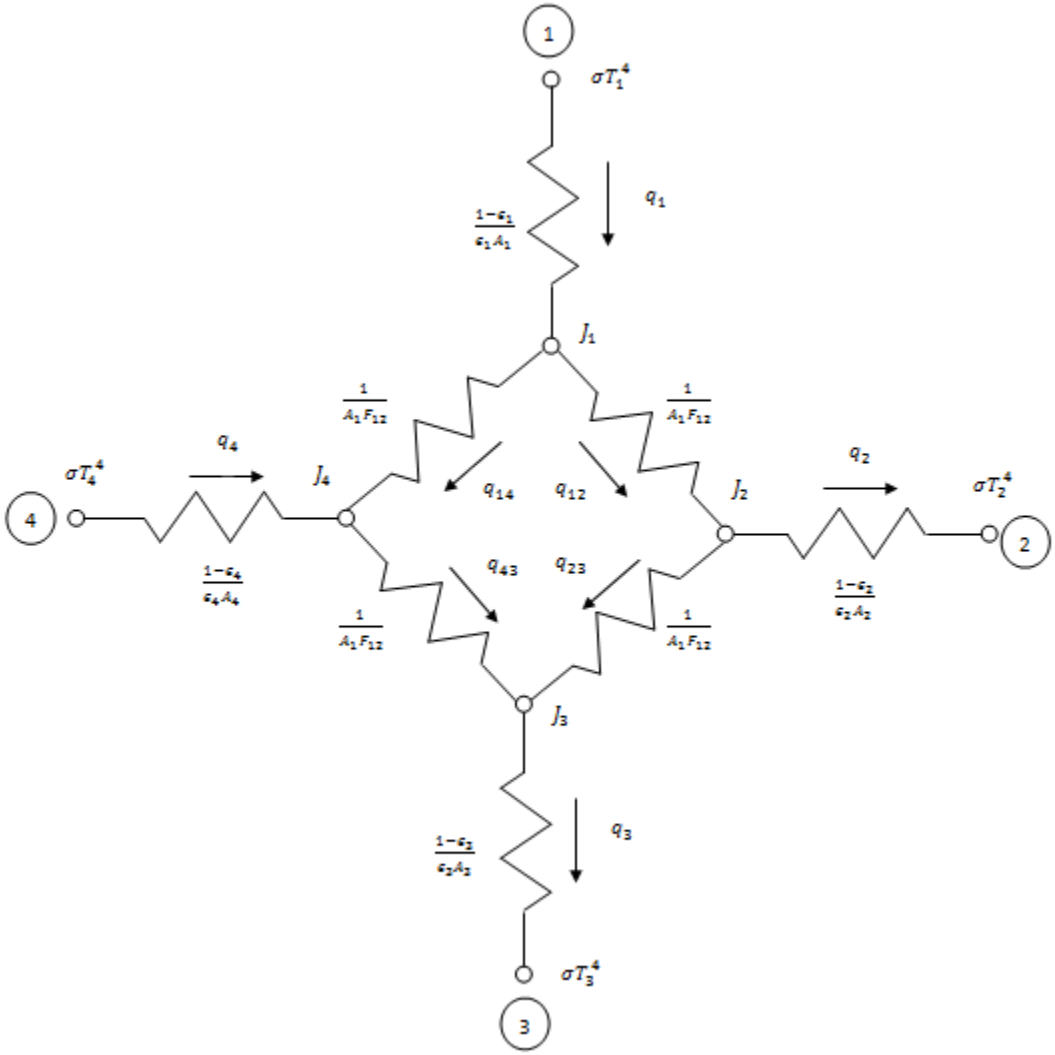


Figure 11. Radiation network derived from sample geometry.

In Figure 11, Body 1 is the N^{th} annulus of the silicon wafer, Body 2 is the shower head, body 3 is the pedestal, and body 4 is the chamber wall.

Radiation equations derived from this network become part of the system of equations to be solved. When a program is developed to iteratively solve these systems of equations, they will be compared to independently computed solutions that come from a separate FD or FEA program. Before all of this is done, the iterative PDE solver will have to be tested by comparing our solution to an analytically solvable problem. This is done to make sure that we are developing the finite difference models correctly. After this step is complete, the iterative solver will then go on to interface with the user.

10.4 Procedure to Solve Computationally

The following list outlines the solving process for our system. The radiation problem is solved in the EES code, which is in Appendix B. The rest of the problem is outlined in Appendix D.

1. Set initial temperature of each body in system
2. Set fixed boundary conditions
3. Use radiation model to give unknown heat fluxes at each time step
4. Use the finite difference model for wafer to solve for the unknown temperature of each body.
5. Use the finite difference model for the gas to solve for the increase in temperature
6. March forward in time.
7. Repeat steps 3 to 6 until final time step is reached.

11 Programming Plan and Procedures

Our programming model requires that we approach the design from two points of view. The first is from the viewpoint of the user and the second is from the viewpoint of the designing engineer. The end value of our deliverable is based on the following criteria: accuracy and user-friendliness. Since these are the paramount objectives, we have organized the programming model as shown in Figure 13.

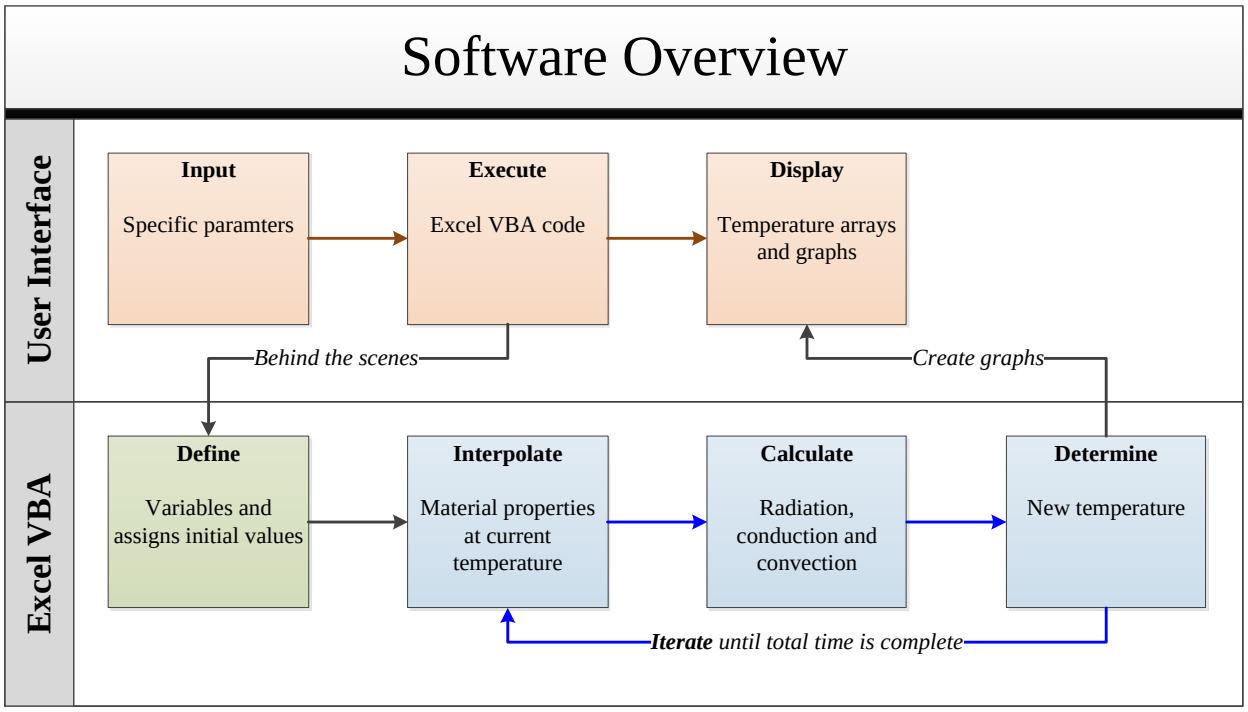


Figure 12. Programming model.

11.1 User Friendliness

From the user's point of view, the program needs to be simple. The existing workflow for Applied Materials requires too much time due to the iterative nature of designing transient thermal silicon processes. Typically, only a few parameters will be changed at a time before executing consecutive iterations. This method would be more effective if the iterations could be conducted rapidly. Our project is to create the iteration tool so that only the final revisions need to be confirmed with Applied Material's heat transfer analysis group. Our programs quick execution time and ease of use will supplement more time consuming methods well, and cut down on time spent iterating.

The user interface requires only three steps. This will ensure a fast execution, a quick learning curve and ultimately encourage the use of our deliverable. These three steps are:

1. **Input** specific parameters
2. **Execute** the calculations
3. **Display** results

11.1.1 Input Specific Parameters

The input parameters specified by Applied Materials can be seen in Figure 13. The user specified values describe all the parameters needed to proceed with heat transfer analysis. These parameters can be lumped into the following categories: Wafer, Pedestal, Showerhead, Chamber, Geometric Spacing and Time. Depending on the specific inputs, the program will be able to determine whether the analysis is describing a heating, cooling or other process for wafer.

Input		Value	Type	Limits / Choices	Accuracy	Units
Wafer	Wafer Starting Temperature	--	input	0 - 999	x.xx	°C
	Wafer Thickness	--	input	0.1 - 2	x.xxx	mm
	Wafer Top Surface Material	--	list	Si, SiO_2, Al, Cu		N/A
	Wafer Bottom Surface Material	--	list	Si, SiO_2 (polished)		N/A
	Backside of Wafer Static Pressure	--	input	1 x 10^{-9} - 900	2 sig fig	Torr
	Backside of Wafer Gas	--	list	N2, Ar, He, H_2		N/A
Pedestal	Pedestal Temperature	--	input	0 - 900	x.x	°C
	Pedestal Material	--	list	Al, Al_2O_3, SS, AlN		N/A
Showerhead	Shower Head Temperature	--	input	0 - 900	x.x	°C
	Shower Head Material	--	list	Al, Al_2O_3, SS		N/A
Chamber	Chamber Wall Temperature	--	input	0 - 900	x.x	°C
	Chamber Wall Material	--	list	Al, Al_2O_3, SS		N/A
	Chamber Static Pressure	--	input	1 x 10^{-9} - 900	2 sig fig	Torr
	Gas in Chamber	--	list	N2, Ar, He, H_2		N/A
Spacing	Wafer to Pedestal Spacing	--	input	0 - 50	x.x	mm
	Wafer to Shower Head Spacing	--	input	0 - 100	x.x	mm
	Wafer Edge to Chamber Wall	--	input	0 - 50	x.x	mm
Time	Total Time	--	input	0 - 600	x.xx	sec
	Time Interval	--	input	0.01 - 10	x.xx	sec

Figure 13. Input parameters for excel program.

11.1.2 Execute the Calculations

The execute operation currently utilizes a Microsoft Excel customizable *Macro* button as shown in Figure 14. We have programmed the macro so that after the calculations are complete, the 'inputs' sheet will be replaced with the 'output' sheets. This way the user will not have to go searching for the results; instead they will appear by default in front of the user.

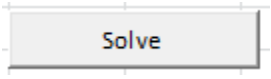


Figure 14. This macro will execute the heat transfer analysis.

We chose this method for its inherent simplicity. This approach presents one command that solves the problem regardless of which parameters were specified. Other methods that we chose not to implement include writing an excel file for each type of thermal process ranging from simpler to more complicated scenarios. However, having just one program as a deliverable is much more streamlined.

11.1.3 Display Results

The results will be compiled into a matrix with columns for time and wafer temperature. Applied Materials is interested in both the wafer temperature at the center of the wafer and also at the extreme edge. The final revisions of our program will include the functionality to select the point of interest, or it will solve and display the results for both points. The temperature difference between these two will result in a gradient that can be solved for in detail using Applied Materials' proprietary finite element heat transfer tools. Our current model solves for the temperature assuming a lumped capacitance and therefore only outputs one average uniform temperature for the wafer as seen in Figure 15.

Time (s)	Wafer Temp (°C)
1	300.00
2	263.23
3	234.55
4	211.33
5	192.04
6	175.70
7	161.65
8	149.42
9	138.66
10	129.13
11	120.63
12	112.98
13	106.05

Figure 15. Resulting matrix from heat transfer analysis.

The macro will also produce an x-y plot, showing the relationship between the wafer temperature and time. This plot is used to confirm the anticipated results by visual inspection. Detailed analysis on specific temperatures in time should be looked up in the output temperature array. Figure 16, shows the corresponding graph for the entire process modeled in the same run as Figure 15 above. Figure 16 below makes it easy to see that then resulting temperatures seems possible.

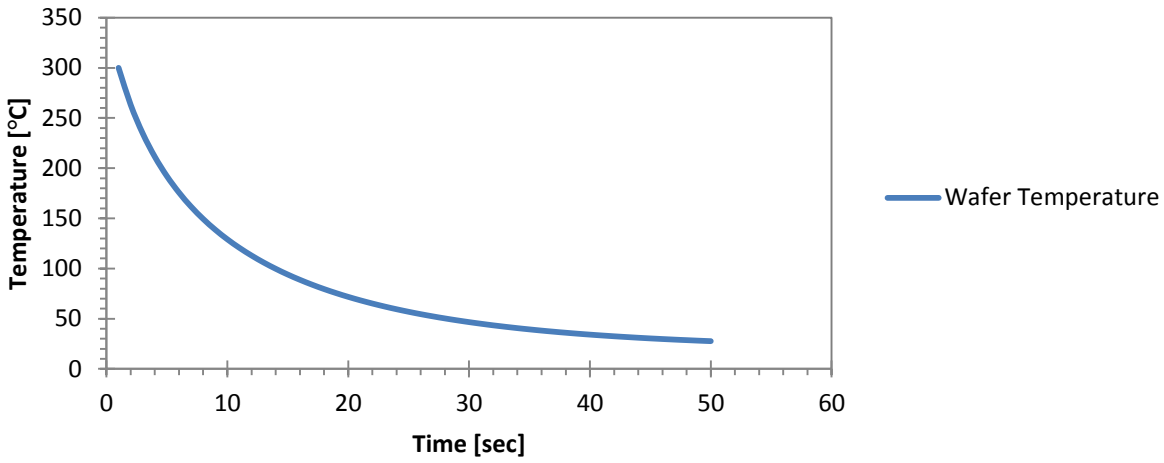


Figure 16. Graphical results of heat transfer analysis.

11.2 Visual Basic for Applications (VBA)

VBA is derived from Microsoft's Visual Basic programming language and is integrated into Microsoft Office applications such as Microsoft Excel. It is an event driven language which means that it is cued by actions such as selection changes or in our case, the use of a macro button. VBA has allowed us to create a series of user-defined functions that cooperate to solve for specific values and ultimately solve our heat transfer problem.

11.3 Solving the Math

VBA is an object-oriented language. This means it is compiled and executed in a similar way that common languages such as C++ are complied. Code can be written in multiple files to avoid having to write thousands of line of code executed from top to bottom. It is also much neater and easier to read, write and comprehend. In our case, these are all desirable things.

11.3.1 Function Calls

The design of our program is based on testing and implementing small tasks. These smaller tasks are called functions, each one with a specific purpose. For instance, the first function we wrote has one job: to calculate the instantaneous conduction heat flux given an array of parameters.

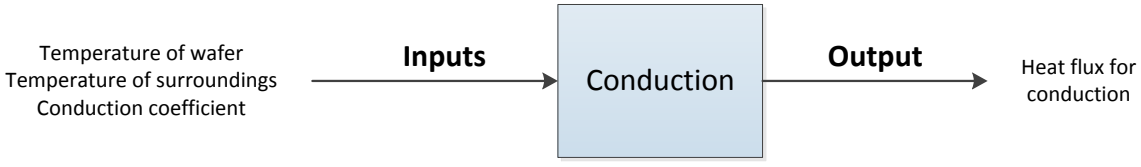


Figure 17. Block diagram of a conduction calculation function.

Similar functions exist for calculating convection and radiation. The sum of all these results will give us the instantaneous heat flux, a necessary part of determining transient temperatures. When the final program runs, it will have a tested function for every value that it needs to calculate. Figure 18 shows the programming overview for calling functions. When the program is executed, we begin with initializations and declaring bit sizes for each variable. Figure 12 shows that the program then spends most of its time in an iterative loop, interpolating values, calculating heat fluxes and finding the next temperature in a time array. The iterative loop can also been seen below where the main program has the ability to demand values from different function whenever it needs them.

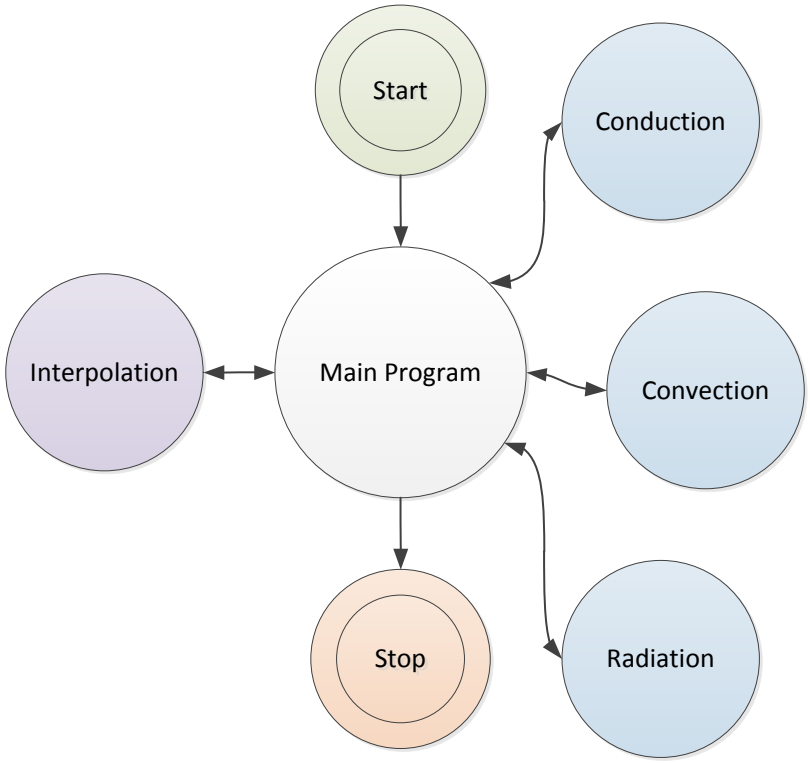


Figure 18. The main program handles data and calls other functions**.**

The advantage of using functions is that we can individually test smaller pieces of our program. After confirming that smaller parts of our calculations are correct, we can proceed with more confidence that our solution is the correct one. This is also a useful tool to compare data with known solutions. Most heat transfer solutions available to us ask for instantaneous heat fluxes, transient problems are inherently more difficult.

11.3.2 Interpolation

Interpolating properties is a crucial part of our project because it needs to happen at every step along our time array. Since almost all physical properties of our system are changing over time, we wrote a generic interpolation function to accommodate the demand. The function requires three input parameters from the main program:

1. The table to interpolate

2. The temperature to interpolate at
3. The parameter to return

We have defined each property table as an individual “Sheet” in Microsoft Excel. Our macro will search through all the hidden sheets to find the one specified in our interpolation function call. With this table open, our code will interpolate the parameter at a certain temperature. This calculated value will be returned to the main program.

11.4 Material Properties

One of the unique challenges embedded into our project is the distinct non-linearity due to changing properties. This is one of the reasons that we cannot derive a linear differential equation and compare results to those of ODE solvers such as MATLAB. Using just the lumped capacitance method, we have simplified the problem to the following differential equation:

$$\rho z c_p \frac{\delta T_w}{\delta t} = -\left[h(T_w - T_\infty) + k_{gas}\left(\frac{T_w - T_\infty}{H_1}\right) + \varepsilon\sigma\left({T_w}^4 - {T_{s,1}}^4\right) + \varepsilon\sigma\left({T_w}^4 - {T_{s,2}}^4\right)\right]$$

Each of the variables listed below are dependent on both temperature and pressure. As stated earlier, one of our over-arching assumptions is that pressure does not vary with time. Temperatures, however, do and have a large effect on most material properties. The combination of every property acting as a function of temperature and pressure defines the non-linearity present in the problem.

- **Gas**
  - ρ - Density
  - k - Thermal Conductivity
  - Pr - Prandtl Number
  - h - Conduction Coefficient
  - c_p - Specific Heat Capacity
  - α – Thermal Diffusivity
- **Wafer**
  - ρ – Density
  - k – Thermal Conductivity
  - c_p - Specific Heat Capacity
  - ε – Emissivity
- **Geometry**
  - ρ - Density
  - ε– Emissivity

To solve the problem of changing parameters, we utilized Engineering Equation Solver (EES), a simultaneous equation solver capable of incorporating many mechanical and fluid concepts such as material properties. EES has national standards data tables built into in. We exported these tables and brought them into our Excel program. We then wrote an interpolation function which grabs any of the above parameters from any of the given tables at a specified temperature.

The ideal gas law was used to find properties at temperatures other than atmospheric. The variable k represents a material property. Subscript 1 represents the property at atmospheric pressure and

subscript 2 represents the property at another pressure. The material properties are proportional to the ratio of pressures.

$$k_2 = \left(\frac{P_1}{P_2}\right) \cdot k_1$$

11.5 Non-Linearity

Non-linearity plays a huge part in our analysis and is prevalent even in the most general cases. Using our program, we have demonstrated this by simulating a cooling process for a silicon wafer with varying pressure. The wafer's initial temperature is 300°C and the surrounding geometry is held at room temperature. To simplify things a little more, we decided to only look at the heat flux from conduction.

11.5.1 The Effects of Pressure

Atmospheric pressure is equal to 760 torr (mmHg). This pressure is too high for most silicon thermal processes which require smaller changes in temperatures over a given amount of time. Lowering the pressure in an annealing chamber and conducting this experiment as various pressure makes this very clear. Figure 20 shows the difference in conduction heat flux with changing pressure.

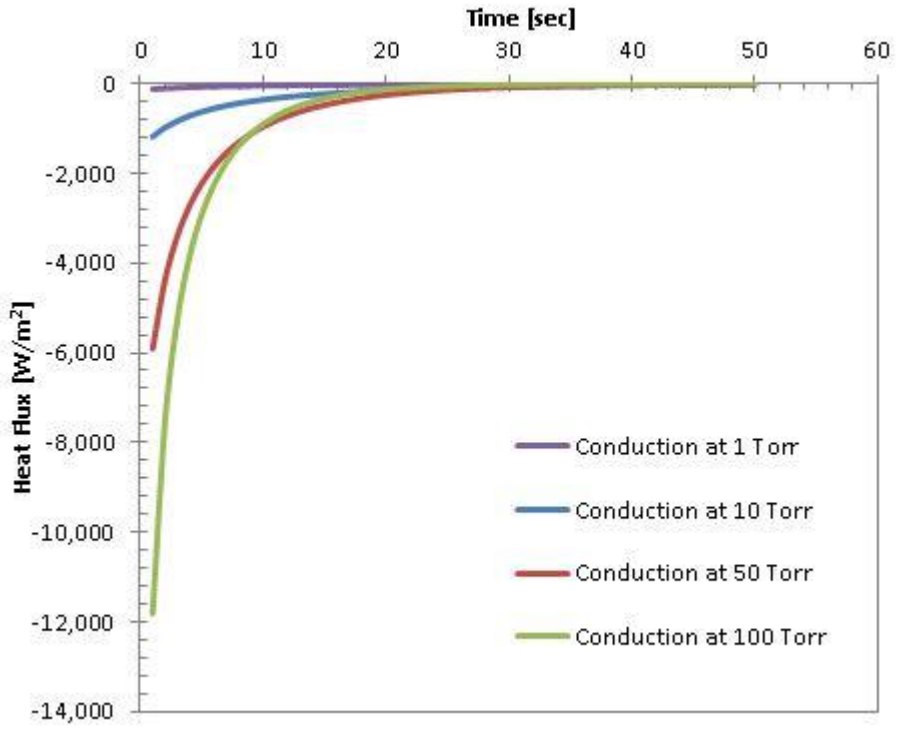


Figure 19. The effect of pressure on the conduction heat flux.

At 100 torr, conduction has a large effect on the heat flux, especially when the wafer temperature is near 300°C. This value tapers off quickly because at such high pressures it doesn't take long for the wafer to reach the room temperature surroundings. Such abrupt and rapid temperature changes cause

deformation due to temperature gradients in the wafer and can also result in cracking and undesirable semiconductor properties.

As the pressure is decreased, conduction plays less and less of a significant role in the heat treatment process. This also allows the wafer to cool down at a slower rate as shown in Figure 21 below.

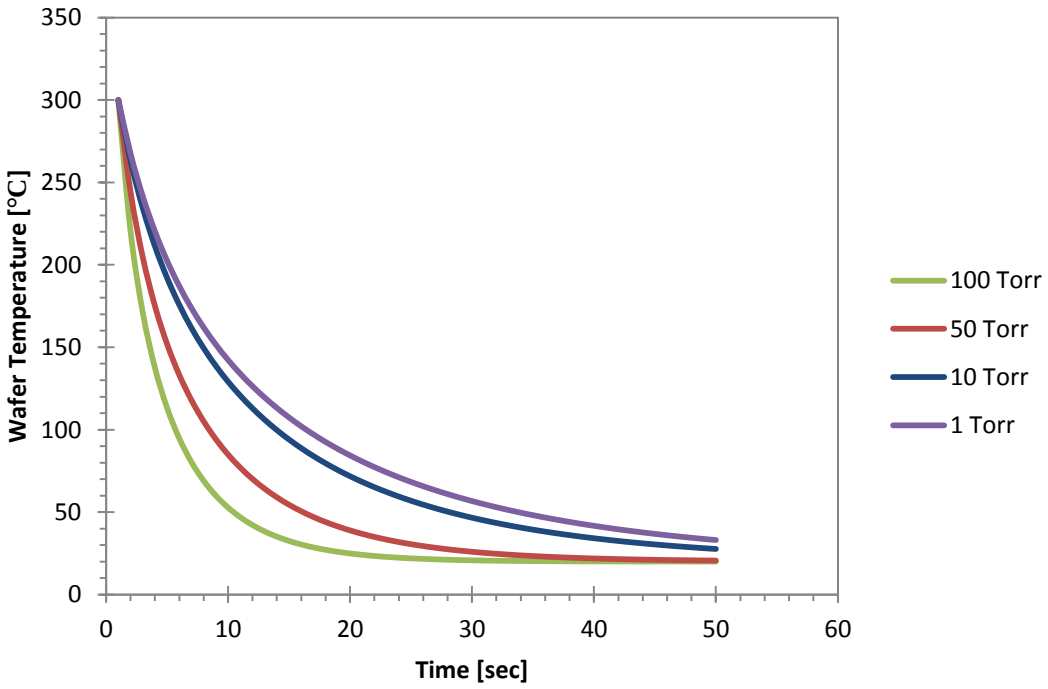


Figure 20. The effect of pressure on the thermal cooling process.

The non-linear characteristics are results from all the material properties that change with temperature and pressure. The two graphs above, Figure 20 and Figure 21, it is clear that the change in material properties due to pressure makes a large difference on how quickly the wafer decreases in temperature.

11.5.2 The Effects of Temperature

Temperature is the largest contributor to the changes in material properties. Typically properties only vary a little with temperature gradients. However, combining each of these small differences together yields a very different resulting temperature array. This presents a problem with utilizing ODE solvers such as MATLAB to compare our data to. Other solvers do not have the engineering capability to accommodate the change in material properties and solve differential equations. An example of this non-linearity can be seen in Figures 22 and 23.

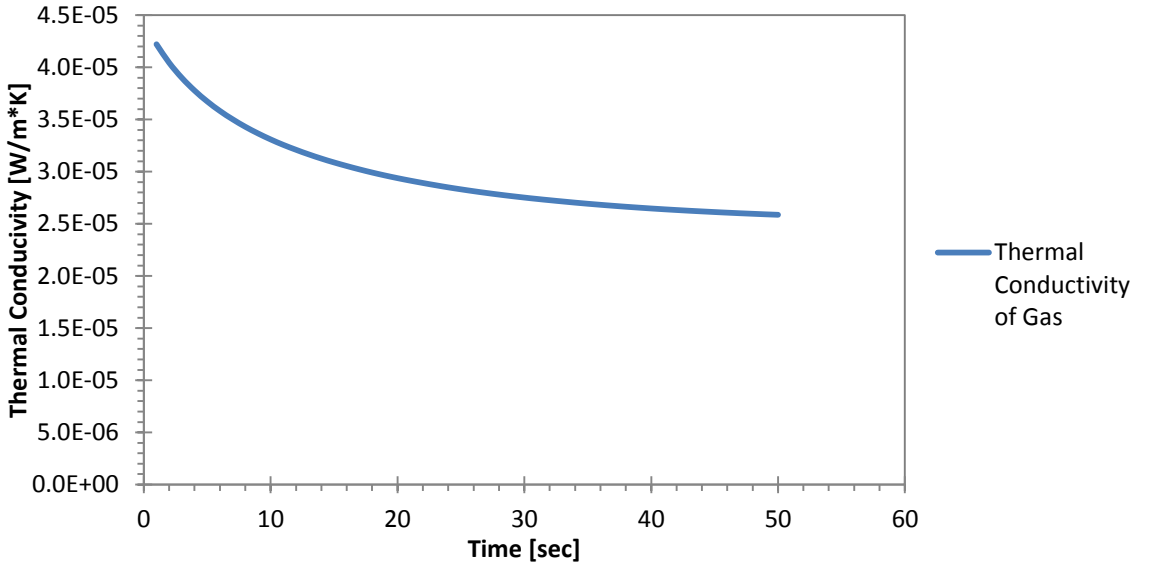


Figure 21. The change in thermal conductivity of the surrounding.

These plots correspond to the same scenario that was modeled in Figures 15 and 16. The ambient and wafer temperatures decrease over the 50 second time frame and approach the steady state value of room temperature.

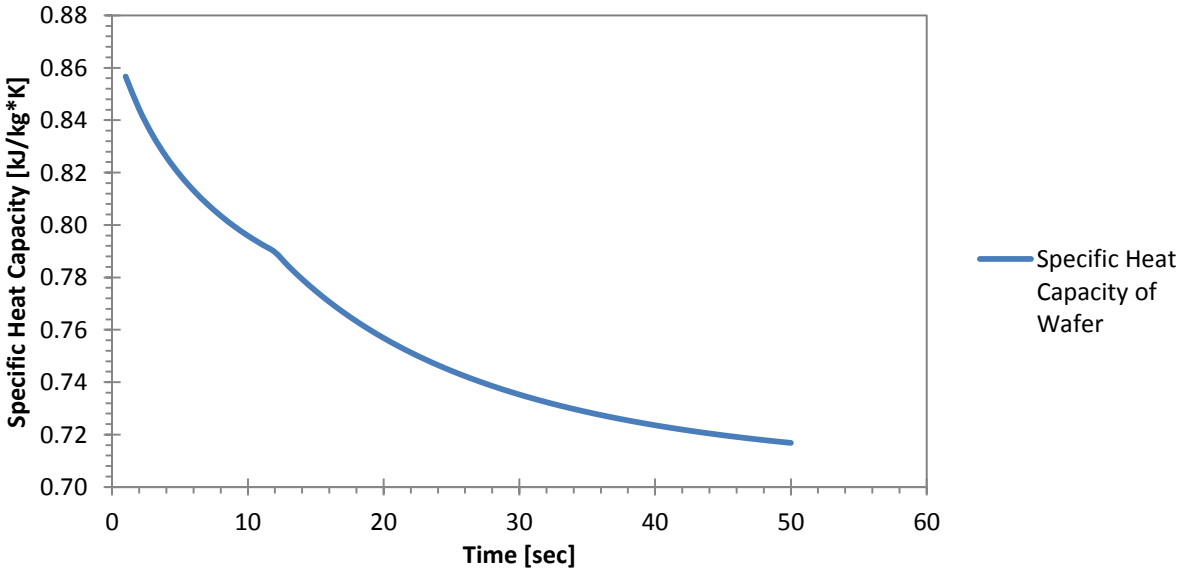


Figure 22. The change in the specific heat capacity of silicon.

11.6 Current Progress

Our current results are limited by the lumped capacitance method of analysis. This is an oversimplification that we chose to incorporate first to get smaller piece of the code working

cooperatively. Applied Materials has supplied us with data for a wafer cooling scenario that we should have the full capabilities to model. Figure 24 shows the outputs of our program plotted against AMAT data.

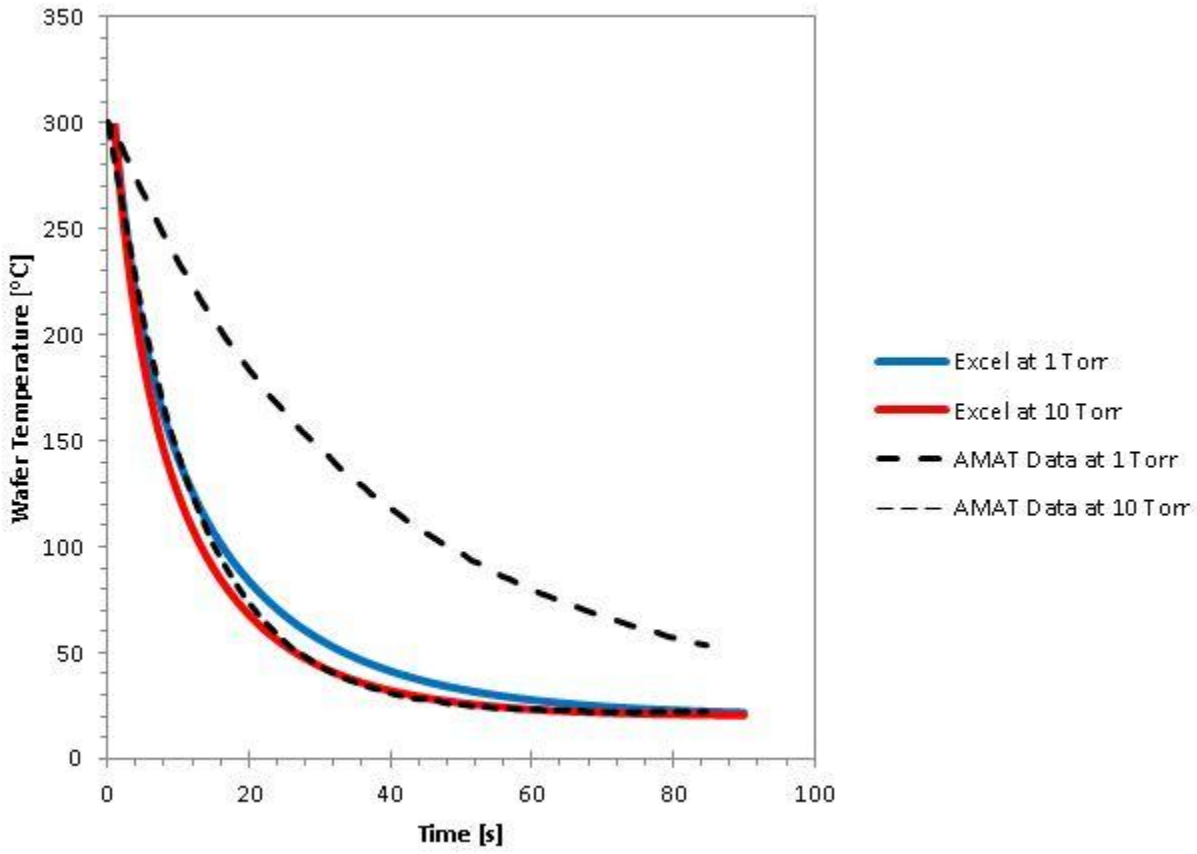


Figure 23. Wafer cooling processes modeled by CPGRID and AMAT.

Figure24 above shows two runs, one with the pressure at 1 torr and another at 10 torr. The transient results from our program are graphed in red and blue. In this figure it is obvious that the results are not lining up as they should. It appears that our analysis is a good approximation at 10 torr. However, there is little difference between the results from our analysis at the two different pressures.

Over the course of the remainder of our project, we will attempt to find the sources of error and correct them. We will continue to receive data from Applied Materials as well as conduct our own transient silicon wafer experiments and use these as further comparisons to pinpoint the inaccuracies in our analysis.

We can associate some of the discrepancies in Figure 24 to some known causes of error. The most significant of these is the lumped capacitance analysis method. In itself, it is unrealistic and implemented here as a proof of concept. We are encouraged that some of the results above have lined

up as well as they did. After coding the new analytical method, we anticipate much more consistent results.

12 Test Setup

After comparing the results of our heat transfer analysis against data from a few test scenarios provided by Applied Materials, there was a need to expand testing functionality and begin verifying additional cases. Applied provided access to their labs & equipment, however - the distance between San Luis Obispo and AMAT facilities in San Jose was prohibitive for more than a few trips. A better solution was to construct a test setup at Cal Poly that approximated AMAT process chambers as closely as possible. We wanted the minimum functionality of the chamber to include the ability to pump a vacuum, supply heat to the wafer, and obviously to take temperature readings at points of interest during testing. Fortunately, the Cal Poly Materials engineering department has a chamber purposed for annealing that fit the project needs relatively well. Using this chamber, a wide variety of cases were tested for the purpose of validating our analysis code.

12.1 The Cal Poly Annealing Chamber

The MATE department chamber (pictured below in Figure 24) is sized to fit a four inch silicon wafer and has an integrated heating chuck as well as the ability to maintain vacuum pressures as low as 1.5 Torr. While not capable of achieving pressures as low as those modeled with the analysis code, it sufficed for validation of the heat transfer principles used. Introduction of exotic gases was not possible due to safety considerations. The vacuum pump for this chamber vents to an adjacent closet, which could have create a potentially dangerous situation.

Fortunately, the chamber has several available feed through ports that can be used to route temperature sensors from the low vacuum pressure inside the chamber to our DAQ system outside. Since a high pressure differential is present between the external atmosphere and the inside of the chamber, specialized fittings and a potted feed through are required for routing wires in and out of the chamber.

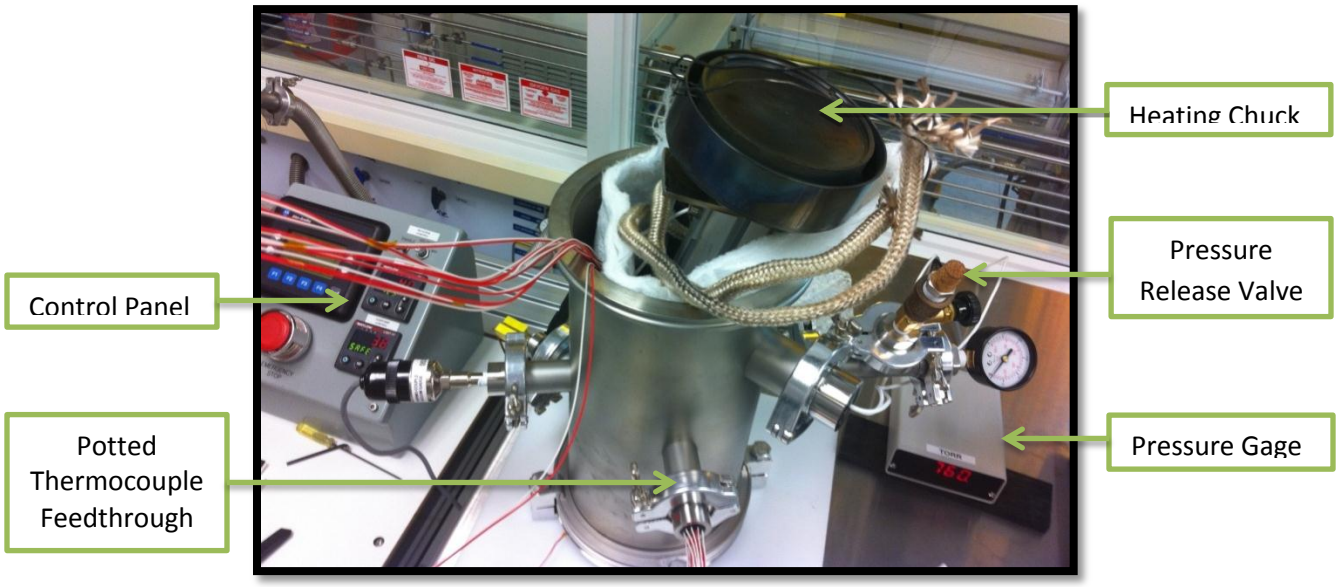


Figure 24 : Model of Cal Poly MATE department annealing chamber.

12.2 Differences Between Test Setup and AMAT System

One of the most challenging aspects of this project was to build a thorough understanding of the process chambers to be modeled. With this understanding, it was possible to evaluate how well the test chamber approximated the real process chambers. It is imperative to be conscious of all differences so that they can be accurately accounted for when comparing results of the analysis to physical test results. Figure 25 illustrates notable differences between the two.

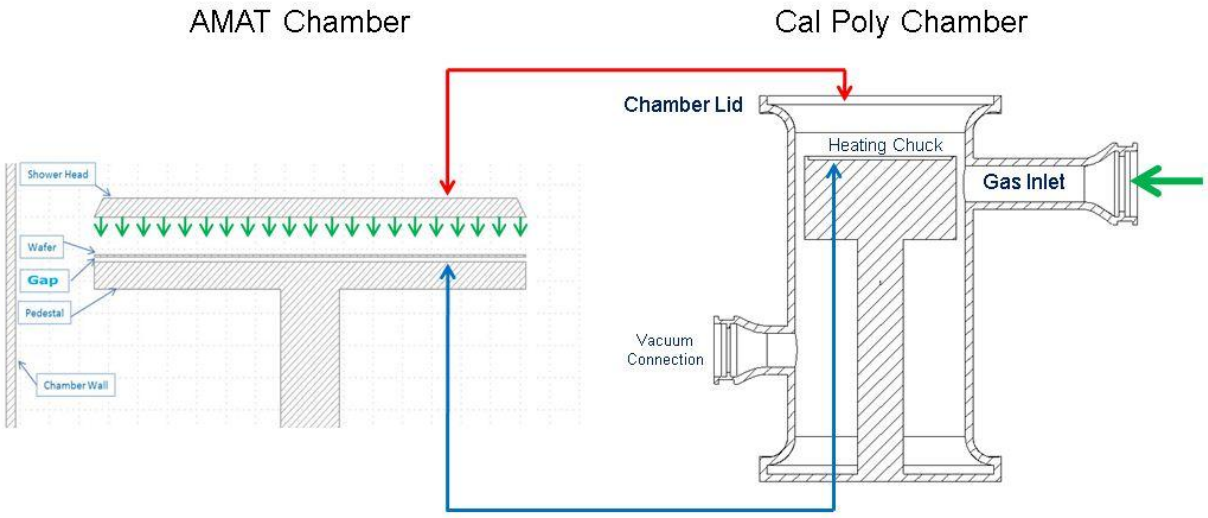


Figure 25. Cross sectional view of AMAT chamber and Cal Poly chamber.

As mentioned in the background information, the heat transfer model is based on a CVD chamber using a 'shower head' reactor design. Features of interest are depicted by the diagram on the left side of figure 25.

The pedestal supplies heat to the wafer in the AMAT system, in the test chamber heat was similarly supplied by a chuck beneath the wafer (indicated by blue arrow). Recall from our discussion of vapor deposition processes that the shower head (see red arrow) is located directly above the wafer and dispenses gases in a uniform manner (see green arrow), while maintaining a stable temperature. The test chamber is less sophisticated – the lid simply completes the chamber enclosure and does not offer means for temperature control. Air can potentially leak into the chamber in a non-uniform manner, around the chamber lid and through the pressure release valve. This difference poses the greatest risk for introducing error into analytical results, as the gas flow across the wafer could be significantly different for the two configurations.

In order to examine the potential effects of this difference, thermocouples were placed at the inlet and at the outlet (vacuum connection) and the increase in air temperature between the two points was

monitored during several tests (see Figure 26). The bulk of air leaking into the chamber came from the pressure release valve, especially when testing at intermediate pressures when the valve was used to regulate pressure. The path of this flow did not cross the wafer, as both the inlet and outlet are located below the elevated heating chuck, with no significant obstruction in between. The observed temperature rise between the inlet and exit was no more than 1°C for any of the data sets, indicating that the air flow does not interact much with the heated chuck and wafer above. This difference was addressed by using an adjusted convection coefficient when comparing physical tests to analytical results. Determining convection coefficient correlations is a field of study to itself, and outside the scope of this project. Using engineering judgment, an appropriate convection coefficient was estimated for each scenario. To further explore the effects of convection, the analysis was run with and without convection, with little change in result. This indicates that other modes of heat transfer are dominant, and a slight inaccuracy in the estimated convection coefficient will not have a significant impact on overall results.

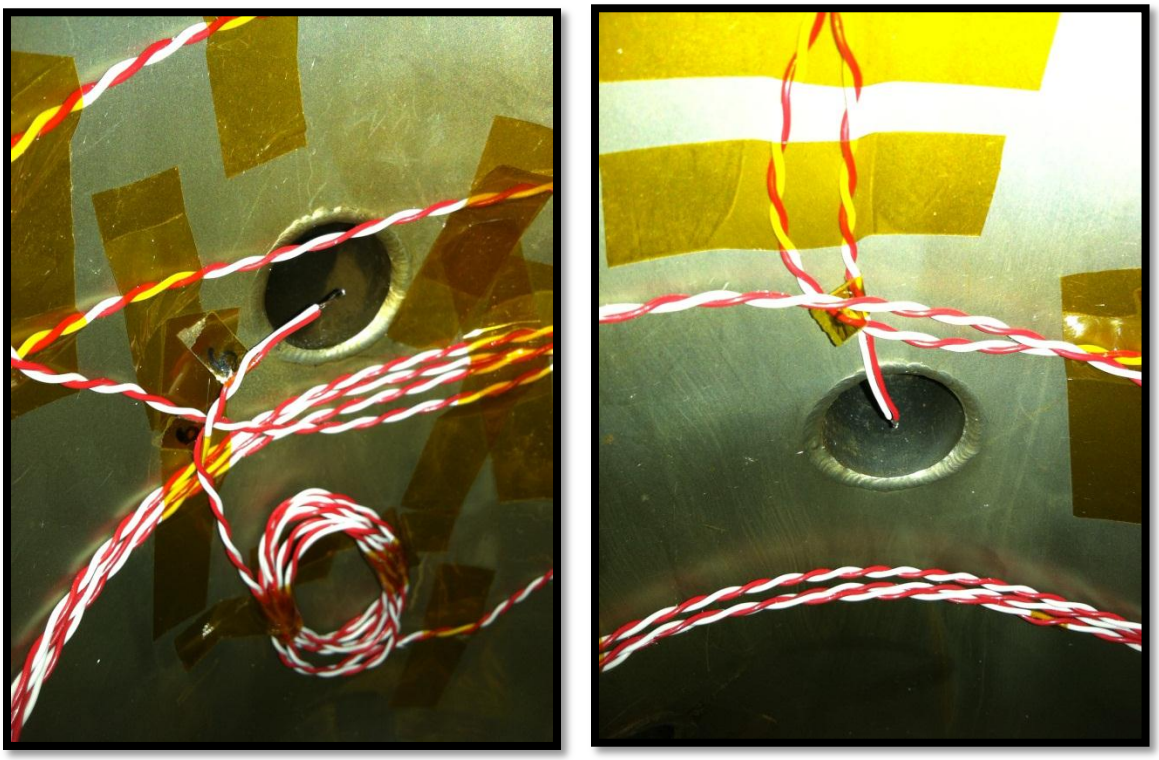


Figure 26: Thermocouples at flow inlet and exit

A simple capability difference also exists between the two: the project aims to achieve accurate analysis results up to 900°C and down to nano-torr pressure range, while the test chamber was capable of only 250°C and down to the single digit torr pressure range.

The most significant differences between the test chamber and the AMAT process chamber are summarized below, as well as how each difference was addressed when comparing results.

Table 3. Summary of Hardware Differences

Summary of Hardware Differences	
Difference	**How to address**
Gas flow through the chamber comes from a showerhead above the wafer in the AMAT chamber, and from a pressure release valve below & to the side of the wafer in the test chamber.	Estimate an appropriate convection coefficient for each scenario.
Chamber walls and shower head are maintained at a carefully controlled temperature in the AMAT chamber. In the test chamber the walls are passive & heat up when the chuck is turned on.	Substitute a lookup table with physical test data into analysis in place of a set temperature for walls, lid, chuck
AMAT chambers can achieve temperatures up to 900^{o}C and pressures down to the nano-torr range. The test chamber is limited to temperatures of 250^{o}C and single digit torr pressure range.	If range of cases are tested and validated with the analysis, it should be safe to extrapolate to higher temperatures and lower pressures

12.3 Testing Capabilities

After observing these differences, it can be noted that the test chamber is not capable of all tests that need to be run. Capabilities are summarized in Table 3. The analysis code is fully validated on the limited range of test data collected from our chamber. Other scenarios were either validated using data provided by Applied, or extrapolated using heat transfer principles.

Table 4: Summary of local testing capabilities and AMAT lab testing capabilities.

Summary of Testing Capabilities			
Test	**Parameter**	**Location**	
		Cal Poly	**AMAT Facilities**
Heating	Atmospheric pressure	X	X
	Low pressure range	X	X
	Extremely low pressure range		X
	Exotic gases present	P	X
	Extremely high temps		X
Cooling	Atmospheric pressure	P	X
	Low pressure	P	X
	Extremely low pressure		X
	Exotic gases present		X

X = Capable P = Possibly capable, will wait to see if results are accurate

12.4 Integrating Test Equipment into the Chamber

While the annealing chamber provided us with a great starting point, modifications were necessary in order to create a suitable testing setup. Our primary objective was to integrate temperature sensing

and data acquisition capabilities into the chamber. As mentioned above, the chamber is equipped with several feedthrough ports. Working with an available NW-25 flanged port, we ordered an epoxy potted feed through from LDS vacuum – a trusted supplier recommended to us by Dr. Savage.

Working with a custom supplier enabled us to create a feed through that perfectly suited our needs. The custom order process allowed us to choose the flange fitting compatible with our chamber, as well as the wire type, length, and quantity. After specifying the fitting type for our port, we turned our attention to choosing instrumentation.

LDS offered several choices for thermocouple type, but we eventually decided on one of the most common. J-type thermocouples are rated for use with temperatures from 0^{o} to 760^{o}C, vacuum pressure, and have the added benefit of being compatible with older DAQ equipment. We wanted to keep the capability for quick measurements using older handheld devices, so J-type fit our needs better than any other. Anticipating that we might need to re-weld the thermocouples if we ran into problems later, we ordered extra length on the inner side of the flange connector. We welded all thermocouple beads ourselves, for the sake of learning more about the process.

Another major element of this test setup is our data acquisition system. We wanted the capability to collect data from eight thermocouples simultaneously, with a resolution at least equal to the resolution our analysis will provide. As we are on a short time frame, reliability is another major concern. With these three criteria in mind, we selected the Omega DAQ USB 2401. This module is compact, capable of sample rates well beyond our requirements, and carries a replacement warranty so our project will not be hampered by faulty equipment.

After acquiring all our testing equipment, we were ready to begin setting up the chamber. This involved placing our thermocouples, placing our sample wafer, and establishing the dimensions of the chamber so we can use the data to compare with our analysis. In order to ensure a reliable mechanical and thermal connection between our thermocouple beads and the components they measured, we used a highly conductive thermal compound. Pictured below is our process of getting the chamber ready for testing.

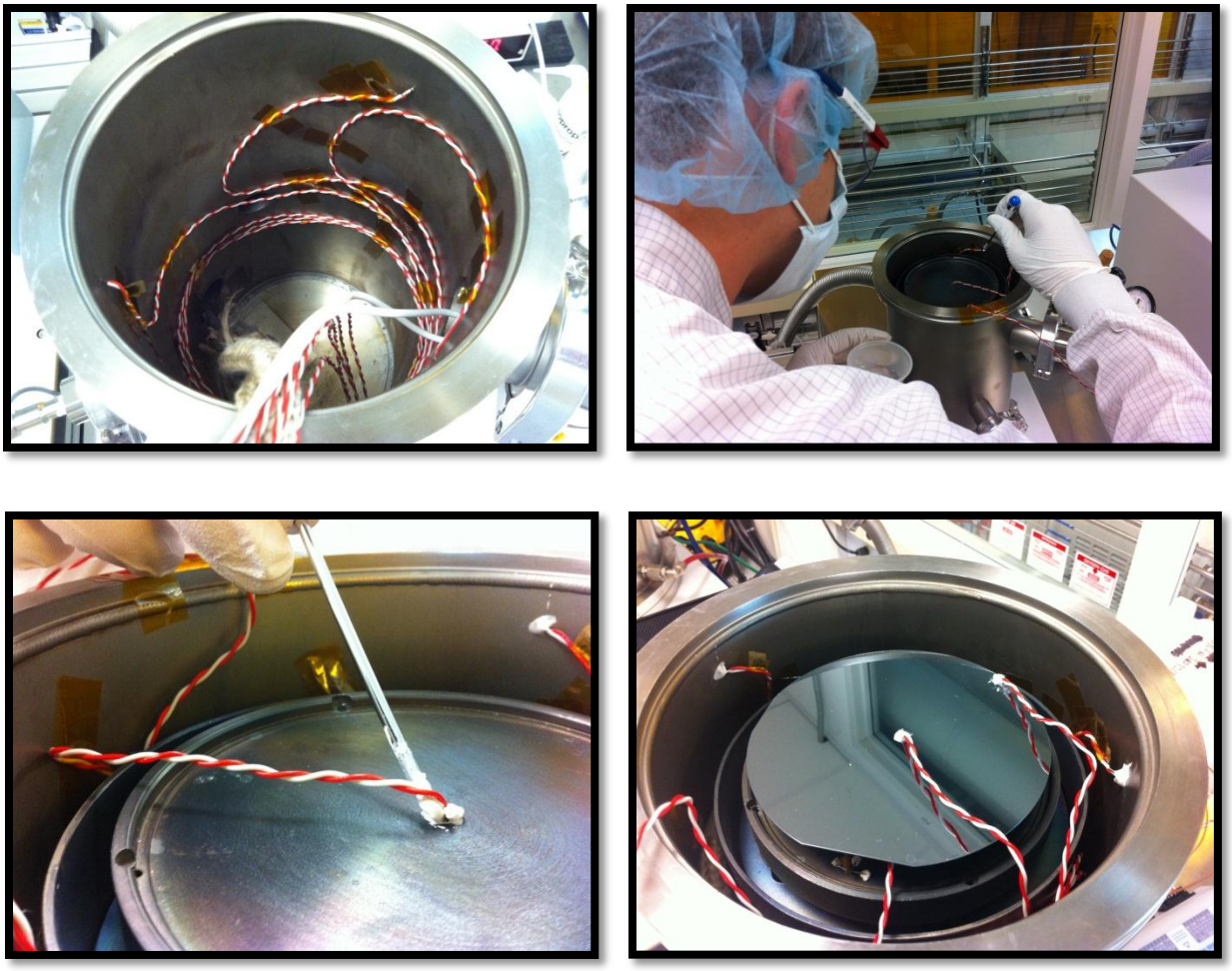


Figure 25: Clockwise from upper left – routing thermocouple wires inside chamber, applying thermal compound to wall thermocouples, applying thermal compound to chuck thermocouple, sample silicon wafer with all thermocouples in place

12.5 Test Plan

Fortunately, a thorough standard operating procedure exists for the vacuum test chamber (included in Appendix A). This procedure, along with guidance from Christine Ghent, the MATE department technician, was used to familiarize the team with operation of the chamber.

Basic scenarios were tested first, followed by the more complicated. Based on results provided by Applied in the early stages of this project, greater variation in results is to be expected between pressures on the lower end of the spectrum, below 10 Torr (see Figure 29). As a consequence, it will be more informative to conduct the majority of testing at pressures below 10 Torr.

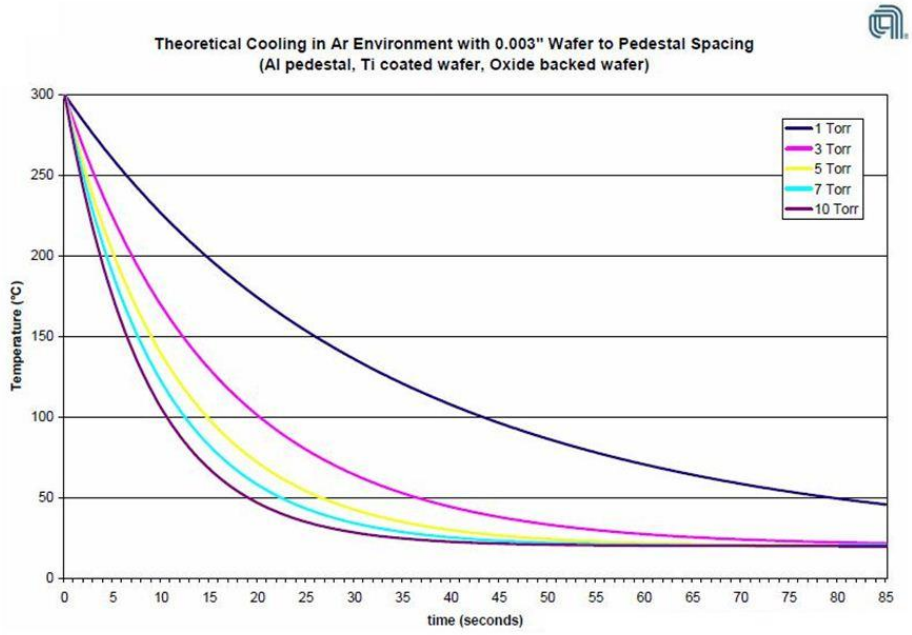


Figure 26: Applied Materials Wafer Cooling Theoretical Data.

The overall goal for this project is to establish a model that will reveal the effects of parameter changes on the time response of the wafer, and temperature differential across the wafer. The testing plan reflects these goals – test scenarios were carefully chosen such that data can be used to find relationships between each design parameter and our metrics of interest. Below is a list of test cases, the results of which will be presented in the next section.

Table 5. Summary of Testing Data

Wafer Spacing Above Chuck (mm)	Chuck Temperature (°C)	Chamber Pressure (torr)
12.65	150	1.6
		4.9
		7.2
		10
	200	1.6
		5.1
		7
		10
	225	1.7
		5.2
		10

9.37	200	1.7
	225	1.7
	240	1.5
		5
		7

12.6 Setbacks

During our physical testing of the system we were forced to overcome a handful of unanticipated roadblocks. This forced us to engineer fast and effective solutions in order to keep our project on schedule. Each problem pushed testing back a significant amount of time:

1. Thermocouple damage requires re-welding the individual beads (2hrs)
2. Any separation of thermocouples to surfaces requires removing and re-applying the cement
3. The two part cement requires an accurate weight measurement system and a 24 hour curing time
4. The heating chuck takes around 3 hours to cool back to room temperature

A summary of the major testing issues can be found in Table XXX. The following sections will talk briefly about some of the important setbacks.

Table 6. Summary of testing setbacks

Problem	Solution
Periodic noise from heating chuck	Lower sampling frequency
Heating chuck controller unreliability	Keep temperatures lower than 250℃
Damaged DAQ channels	Record some temperatures by hand
Thermal epoxy dried up	Use of thermal paste instead

12.7 Thermocouple Noise and Interference

The data acquisition system used in this project was an 8-channel DAQ from OMEGA. Each of these channels was tested in the spring quarter to ensure that fall testing would go more smoothly. The channels were tested against a known temperature such as an ice bath (32°F) and boiling water (212°F). The response time was also evaluated by immersing the thermocouple in each environment. The system was fast and accurate enough for our purposes.

However, during our fall testing we noticed a significant noise pattern in our data readings when the thermocouples were placed on the heater. The pattern represents phenomena in the data and does not represent the actual periodic change in temperature of the heating chuck.

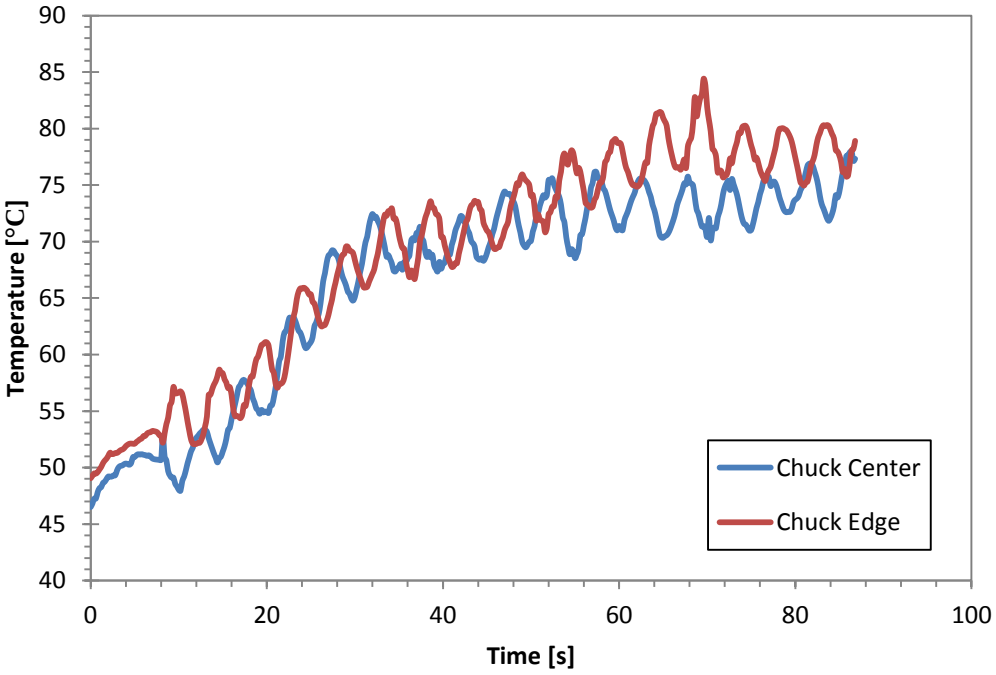


Figure 27 Data points taken at 5 Hz from two different locations on the heating chuck

Figures 27-28 show a distinct oscillatory pattern in the thermocouple data. This anomaly only occurred when measuring temperatures directly on or nearby the heating chuck when it was active. For this reason we believed that high wattage electrical resistance heating coils were inducing a voltage on the thermocouple wires. This would explain why the period appears to be consistent.

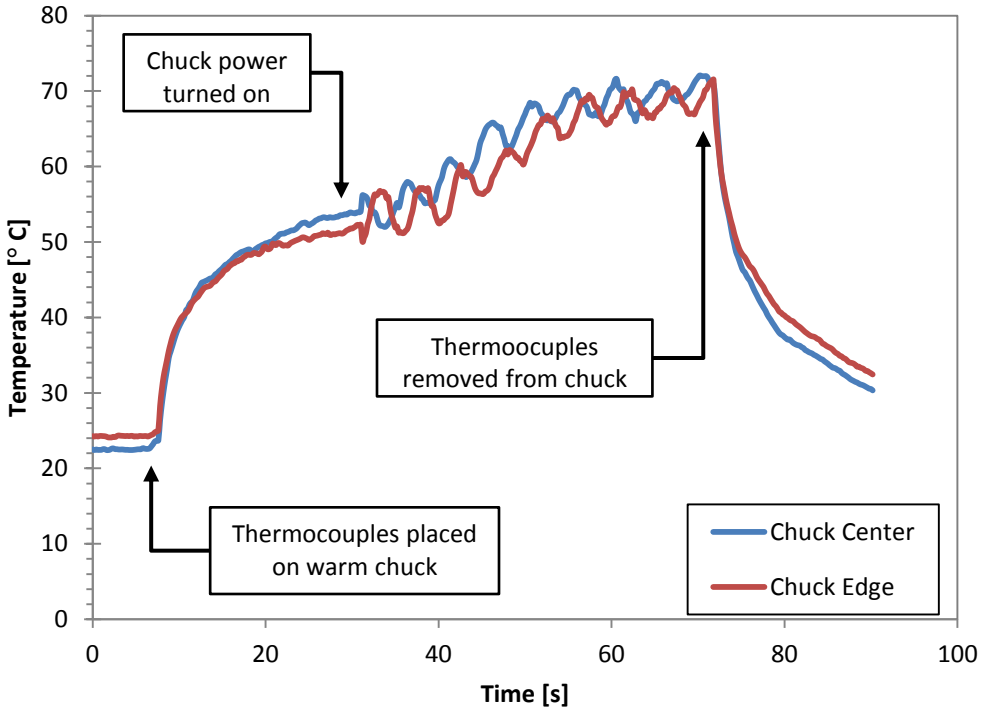
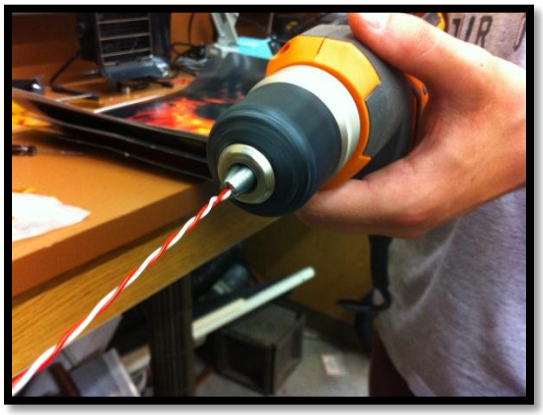
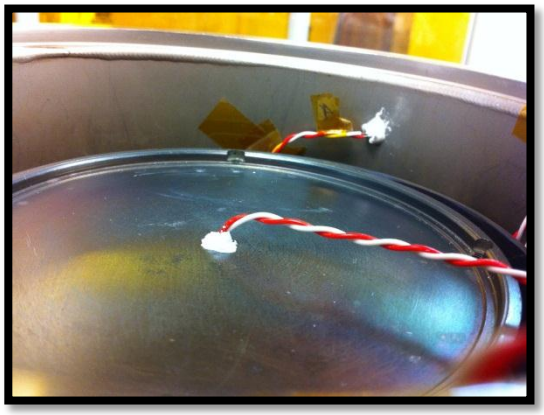


Figure 28. Oscillatory patterns only occur when the heating chuck is turned on

Recommended solutions to this problem included twisting the wires to use the resulting induction as a filter. This method was recommended by Professor John Ridgely of the Mechanical Engineering Department at Cal Poly. According to Dr. Ridgely, similar inductance and noise problems are common in mechatronics applications. Twisting the wires or placing ferrite clamps around the wires can help reduce and eliminate the noise. The recommended method for a uniform twist is to use a handheld drill and slowly walk towards the other end of the wires as the wires are spun together.





This solution had a small effect in our case. We found that reducing the sample rate to 1 Hz produced smooth data points. This was an acceptable solution because each test runs for around 10 minutes. Data at once per second is acceptable. Other possible ways to investigate the problem would be to continually increase the sampling rate and observe how the data changes. Unfortunately we did not have enough time in the lab to look further into this problem.

12.8 Heating Chuck Controller Unreliability

One of the largest setbacks for our testing schedule was the result of the heating chuck controller. The controller simply allows the user to set the desired temperature above room temperature and turn the heater on or off. Turning the chuck on will result in a first order response with the steady state temperature being just under the set point temperature. During our training on the system, the heating chuck in this chamber was declared to reach a maximum temperature of 250°C. As radiation has a significantly larger effect on heat transfer at higher temperatures due to increased values in emissivity, it was to our advantage to set the heating chuck as high as possible. Figure 29 shows the result of setting the temperature to 250°C.

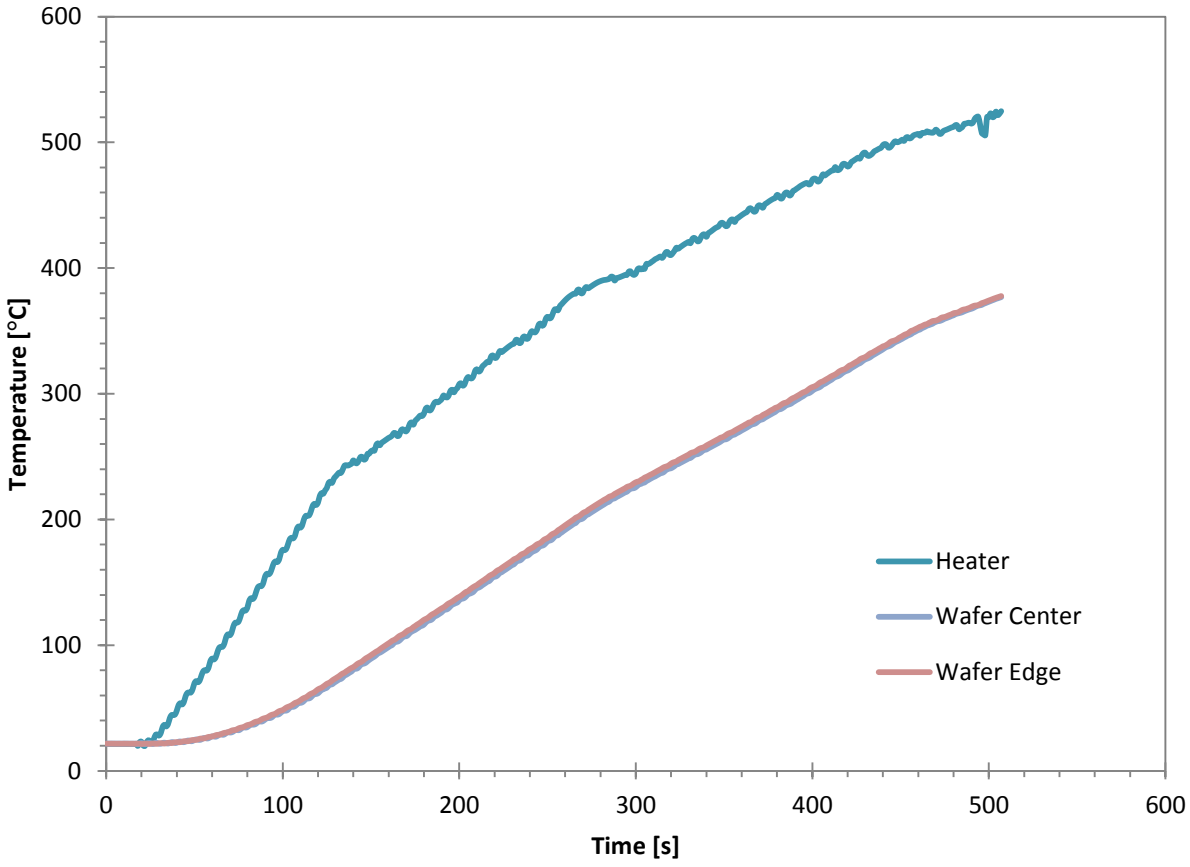


Figure 29. Heating chuck temperature increasing without limit

We were forced to manually turn off the power to the heating chuck when it reached temperatures significantly higher than we anticipated. After allowing the system to cool down we found that the

plastic casing around the thermocouple wires had melted. This caused the wires to short, giving us inaccurate temperature readings and forcing us to re-weld some of the thermocouple beads. Unfortunately we were unable to handle anything inside the chamber until it cooled to a reasonable temperature.

12.9 Damaged DAQ Channels

Four out of the eight channels in our DAQ were unusable by the end of our testing. The loss of these four channels is a result of having to re-weld the thermocouples as quickly as possible. In doing so we failed to remove the opposite ends of the wire from the DAQ. When we attempted to weld the beads, the large voltage differences damaged some of the internal wiring within the DAQ hardware. It took a long time to come to this conclusion.

13 Results

Our primary objective for this project was to create a tool to predict the effects of chamber design parameters on thermal behavior of the wafer. Most importantly we look at the wafer time constant, and the temperature differential across the wafer. We used finite difference heat transfer analysis to create a tool to predict this behavior, and the testing hardware described in Section 12 to collect data for the purpose of validating our tool.

Some parameters used in our analysis are difficult to calculate on a purely theoretical basis, rather – they should be based on empirical calculations. Convection and emissivity presented us with the most challenges. Our test data helped us adjust these parameters in our analysis until our results nicely matched our physical test data. Shown below are plots demonstrating our analysis compared to data provided to us by Applied, and compared to data we collected with our own hardware.

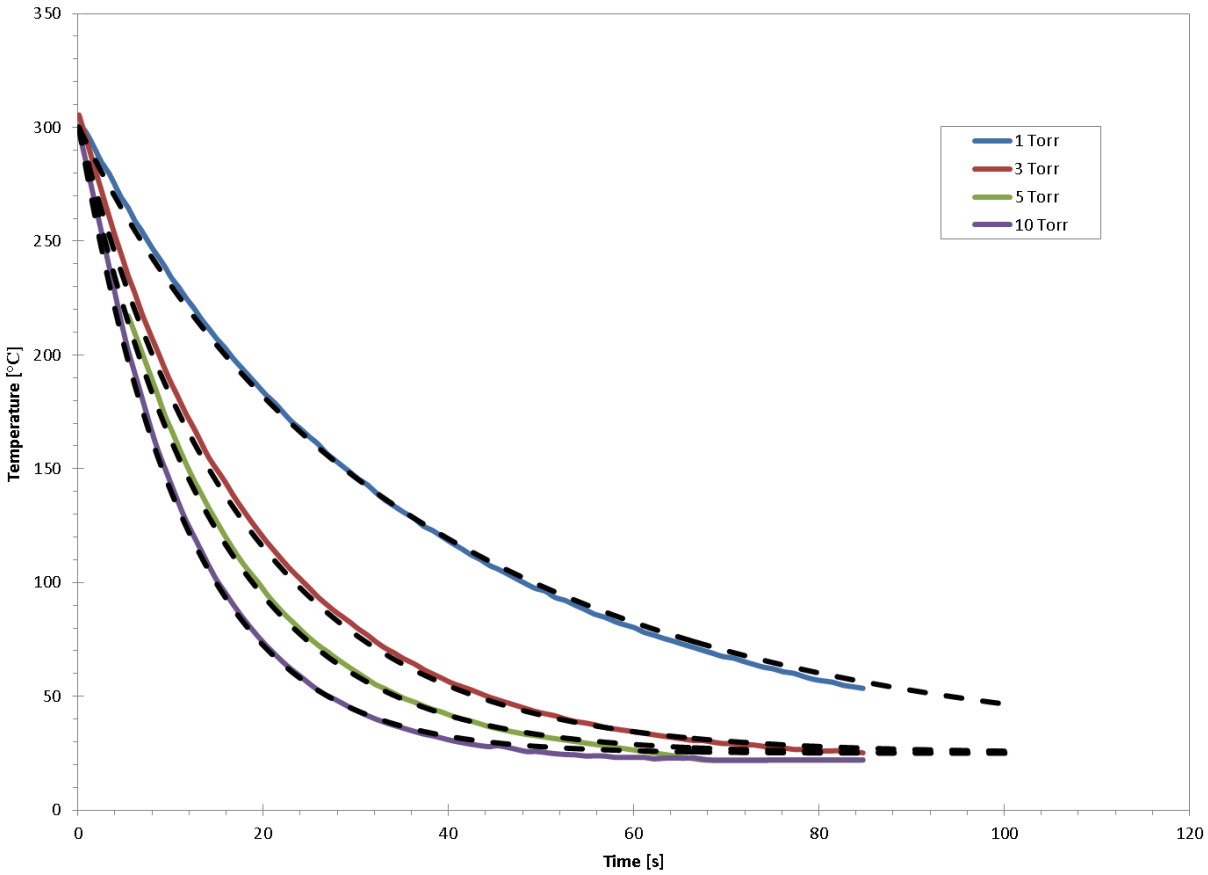


Figure 30. CPGRID Analysis compared to Applied Materials' theoretical cooling data for a twelve inch wafer with a 0.003" pedestal spacing. CPGRID results shown in dotted lines

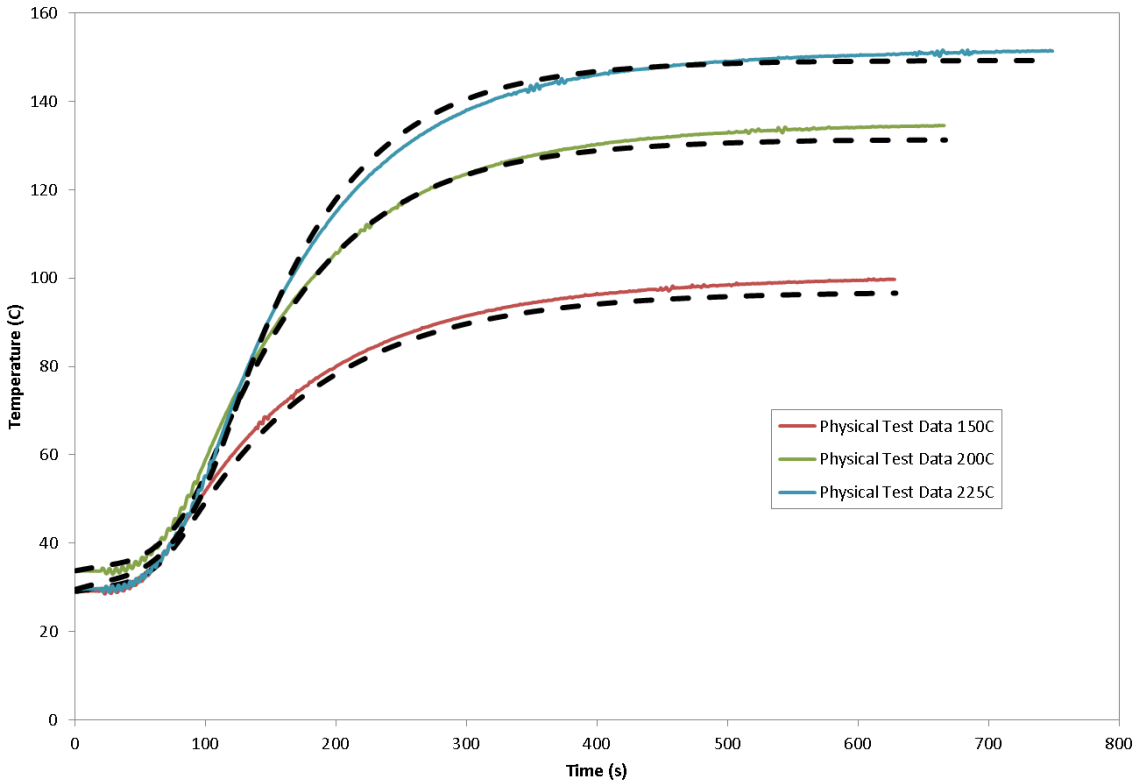


Figure 31. CPGRID Analysis compared to physical testing data for a four inch wafer. CPGRID results are shown in dotted.

14 Conclusion

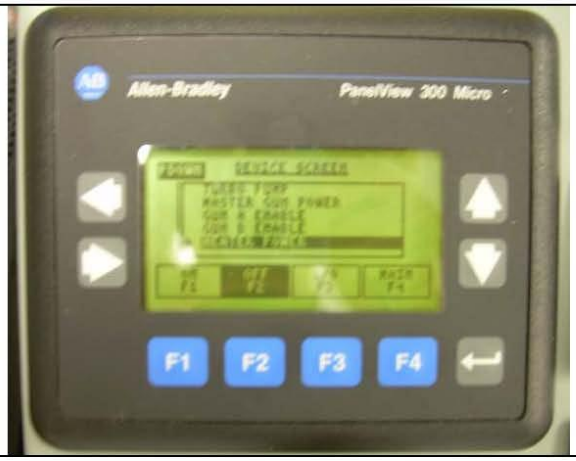
A successful working model of transient temperature distribution in a silicon wafer was created and validated using physical and theoretical testing data. The model incorporates all the user defined inputs to the program as part of the heat transfer analysis. Using a structured project flow and efficient management planning, we were able to complete all the objectives defined in the project proposal. The close correlation between the model predictions and the physical data indicate that all assumptions are acceptable and that the resulting outputs are sufficient to be of value in Applied Material's chamber analysis iterations.

Standard Operating Procedures:
Vacuum Annealing Chamber

1. Open chamber and place lid on clean wipe.	
2. Place wafer with thin film facing up. Ensure heating chuck does not contact the chamber directly. If so, place a glass slide to prevent direct contact.	

3. Replace lid and ensure that vent valve is fully closed.	
4. Turn on vacuum pump located in the chase. The pump should be directly behind the vacuum annealing chamber and between the two Rohwedder chillers.	
5. Check vacuum sensor to ensure that the vacuum pump is pumping correctly. Allow vacuum pump to pump for about 30 minutes before any heat treatment.	

6. Double check desired temperature by looking at the settings on the temperature controller monitor. Adjust by using the up and down arrows on the top thermocouple reader labeled "Heater Control."	
7. Turn on heating chuck from the Rohwedder controller. • F4 – Main • F3 – Devices • Scroll down to "Heater Power" • F1 – On	
8. Begin timer for process and monitor chamber during vacuum annealing process.	

9. After process timer is done, shut off heater power on Rohwedder controller. • F4 – Main • F3 – Devices • Scroll down to "Heater Power" • F2 – Off Allow chamber to cool to room temperature before venting the system.	
10. Once chamber has reached room temperature, shut off the vacuum pump in the chase. Open vent valve, remove wafer and replace lid.	

16 Appendix B

EES Code

```
"Heat transfer due to radiation in CPGRID Testing Chamber at Steady State"
"===========================================================================
================"
"Inputs"
"===========================================================================
================"
sigma = 5.670373*10^(-8)                      "Stefan-Boltzmann Constant W/((m^2)*(K^4))"
N = 5                                         "Set the number of nodes"
Np = 3                                        "Number of time steps"
DELTAt = 0.1                                  "Time step"
k = k_('Silicon', T_T[1,3])                   "W/(m K) 105.8"
c_p= c_('Silicon', T_T[1,3])                  "J/(kg*K)"
rho = rho_('Silicon', T_T[1,3])               "kg/m^3"
alpha = k/(rho*c_p)                           "m^2/s"
duplicate p = 1,Np
    t[p] = (p-1)*DELTAt
end
h_TOP = 2                                     "W/m^2K"
h_BOT = 2                                     "W/m^2K"
w = 0.021*Convert(in,m)                       "Wafer thickness in m"
R_chamber = 8                                 "Units of Thermal Resistance"

"===========================================================================
================"
"Temperatures"
"===========================================================================
================"
"Initial Conditions"
duplicate i = 1, N
    T_T[1, 2+ i] = ConvertTemp(C,K, 25.5)     "Initial Wafer Temperatures"
end
duplicate i = 1, N
    T_B[1, 2+ i] = ConvertTemp(C,K, 25.5)     "Initial Wafer Temperatures"
end

duplicate p = 1,Np
    T_T[p,1] = ConvertTemp(C,K, 45.8) {ConvertTemp(C, K, Lookup('KnownTemps', p,'Column1')) }
    T_T[p,2] = ConvertTemp(C,K, 49.1) {ConvertTemp(C, K, Lookup('KnownTemps', p,'Column3')) }
    T_B[p,1] = ConvertTemp(C,K, 226) { ConvertTemp(C, K, Lookup('KnownTemps', p,'Column2'))}
    T_B[p,2] = ConvertTemp(C,K, 49.1) {ConvertTemp(C, K, Lookup('KnownTemps', p,'Column3'))}
end

{q_T[1,1] = 0                                 "Initial adiabatic lid"
q_T[1,2] = 0                                  "Initial adiabatic top walls"
q_B[1,1] = 10                                 "Constant heat flux through heat chuck"
q_B[1,2] = 0                                  "Intial adiabatic bottom walls"}

T_infinity_T[1] = ConvertTemp(C,K, 45.8)      "Initial temperature of gas on top"
T_infinity_B[1] = ConvertTemp(C,K,45.8)       "Initial temperature of gas in the bottom"
T_infinity_Lab = ConvertTemp(C,K,20)          "Constant temperature of the lab"

"RAMP TEMPERATURES"
```

```
{duplicate p = 2,Np
    T_T[p,1] = ConvertTemp(C,K, 200)
    T_T[p,2] = ConvertTemp(C,K, 200)
end}

"Top Blackbody Fluxes in W/m^2"
duplicate p = 1,Np
E_b_T[p,1] = sigma*T_T[p,1]^4                         "Blackbody energy of lid"
E_b_T[p,2] = sigma*T_T[p,2]^4                         "Blackbody energy of top wall"
duplicate i= 1,N
    E_b_T[p,2+i] = sigma*T_T[p,2+i]^4                 "Blackbody energy of top of wafer"
end
end

"Bottom Blackbody Fluxes in W/m^2"
duplicate p= 1,Np
E_b_B[p,1] = sigma*T_B[p,1]^4
E_b_B[p,2] = sigma*T_B[p,2]^4
duplicate i= 1,N
    E_b_B[p,2+i] = sigma*T_B[p,2+i]^4
end
end
"================================================================
================"
"Geometry"
"================================================================
================"
"Dimensions"
h_T = 1.005*Convert(in,m)
h_B = 0.396*Convert(in,m)
d_wafer = 4.005*Convert(in,m)
d_chuck = 4.005*Convert(in,m)
d_chamber = 4.005*Convert(in,m)
DELTAr = d_wafer/(2*(N-0.5))
duplicate i= 1,2
    r_m[i] = 0 "Midpoint is zero of these bodies"
    r_o[i] = d_wafer/2 "Outer radius of the two bodies"
end
duplicate i= 1,N
    r_m[2+i]= (i-1)*DELTAr                            {DELTAr/2 "Variable to store radius to the
center of each node"}
    r_o[2+i] = r_m[2+i]+DELTAr/2                      "Outer radius in m"
end

"Top Areas in m^2"
A_T[1] = pi*(d_wafer/2)^2                             "Area of lid in m^2"
A_T[2] = pi*(d_chamber)*h_T                           "Area of top wall in m^2"
A_T[3] = pi*r_o[3]^2                                  "Area of center node in m^2"
duplicate i= 2,N                                      "Area of each annulus in m^2"
    A_T[2+i]= 2*pi*r_m[2+i]*DELTAr
end

"Bottom Areas in m^2"
A_B[1] = pi*(d_wafer/2)^2                             "Area of heating chuck in m^2"
A_B[2] = pi*(d_chamber)*h_B                           "Area of bottom wall in m^2"
A_B[3] = pi*r_o[3]^2                                  "Area of center node in m^2"
```

```
duplicate i= 2,N                                          "Area of each annulus in m^2"
    A_B[2+ i]=  2*pi*r_m[2+ i]*DELTAr
end
"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
"View Factors"
"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
"Top Enclosure View Factors Relationships"
F_T[1,1]= 0                                               "Lid to Cylinder"
R_T[1,1]= 0
R_T[1,2]= 1
X_T[1,1] =  (L_T^4+ 2*L_T^2*(1+ R_T[1,1]^2)+ (1-R_T[1,1]^2)^2)^(1/2)
X_T[1,2] =  (L_T^4+ 2*L_T^2*(1+ R_T[1,2]^2)+ (1-R_T[1,2]^2)^2)^(1/2)
F_T[2,1] =  1/(4*L_T)*(X_T[1,2]-X_T[1,1]+ R_T[1,2]^2-R_T[1,1]^2)
A_T[1]*F_T[1,2]= A_T[2]*F_T[2,1]

2*F_T[2,1]+ F_T[2,2]= 1                                   "Cylinder to everything else; Doubling the first
view factor accounts for the lid and the wafer viewing the cylinder"

duplicate i= 1,2
    R_T[2,i] =  0
    X_T[2,i] =  0
end


L_T =  h_T/(d_wafer/2)                                    "View factor of the wafer"

R_T[3,1]= (r_m[3])/(d_wafer/2)
R_T[3,2]= (r_o[3])/(d_wafer/2)
X_T[3,1] =  (L_T^4+ 2*L_T^2*(1+ R_T[3,1]^2)+ (1-R_T[3,1]^2)^2)^(1/2)
X_T[3,2] =  (L_T^4+ 2*L_T^2*(1+ R_T[3,2]^2)+ (1-R_T[3,2]^2)^2)^(1/2)
F_T[2,3] =  1/(4*L_T)*(X_T[3,2]-X_T[3,1]+ R_T[3,2]^2-R_T[3,1]^2)     "Annulus to cylinder walls"
A_T[1]*F_T[1,3]= A_T[3]*F_T[3,1]
A_T[2]*F_T[2,3]= A_T[3]*F_T[3,2]
duplicate j= 1,N
    F_T[3,2+ j]= 0
end
F_T[3,1]+ F_T[3,2]+ F_T[3,3] = 1

duplicate i= 2,N
    R_T[2+ i,1]= (r_m[2+ i]-DELTAr/2)/(d_wafer/2)
    R_T[2+ i,2]= (r_m[2+ i]+ DELTAr/2)/(d_wafer/2)
    X_T[2+ i,1] =  (L_T^4+ 2*L_T^2*(1+ R_T[2+ i,1]^2)+ (1-R_T[2+ i,1]^2)^2)^(1/2)
    X_T[2+ i,2] =  (L_T^4+ 2*L_T^2*(1+ R_T[2+ i,2]^2)+ (1-R_T[2+ i,2]^2)^2)^(1/2)
    F_T[2,2+ i] =  1/(4*L_T)*(X_T[2+ i,2]-X_T[2+ i,1]+ R_T[2+ i,2]^2-R_T[2+ i,1]^2)    "Annulus to cylinder
walls"
    A_T[1]*F_T[1,2+ i]= A_T[2+ i]*F_T[2+ i,1]
    A_T[2]*F_T[2,2+ i]= A_T[2+ i]*F_T[2+ i,2]
    duplicate j= 1,N
        F_T[2+ i,2+ j]= 0
    end
    F_T[2+ i,1]+ F_T[2+ i,2]+ F_T[2+ i,2+ i] = 1
end
```

```
"Bottom Enclosure View Factors Relationships"
F_B[1,1]= 0                                                    "Heating chuck to Cylinder"
R_B[1,1]= 0
R_B[1,2]= 1
X_B[1,1] =  (L_B^4+ 2*L_B^2*(1+ R_B[1,1]^2)+ (1-R_B[1,1]^2)^2)^(1/2)
X_B[1,2] =  (L_B^4+ 2*L_B^2*(1+ R_B[1,2]^2)+ (1-R_B[1,2]^2)^2)^(1/2)
F_B[2,1] =  1/(4*L_B)*(X_B[1,2]-X_B[1,1]+ R_B[1,2]^2-R_B[1,1]^2)
A_B[1]*F_B[1,2]= A_B[2]*F_B[2,1]

2*F_B[2,1]+ F_B[2,2]= 1                                        "Cylinder to everything else; Doubling the first
view factor accounts for the lid and the wafer viewing the cylinder"

duplicate i= 1,2
    R_B[2,i] =  0
    X_B[2,i] =  0
end


L_B =  h_B/(d_wafer/2)                                         "View factor of the wafer"

R_B[3,1]= (r_m[3])/(d_wafer/2)
R_B[3,2]= (r_o[3])/(d_wafer/2)
X_B[3,1] =  (L_B^4+ 2*L_B^2*(1+ R_B[3,1]^2)+ (1-R_B[3,1]^2)^2)^(1/2)
X_B[3,2] =  (L_B^4+ 2*L_B^2*(1+ R_B[3,2]^2)+ (1-R_B[3,2]^2)^2)^(1/2)
F_B[2,3] =  1/(4*L_B)*(X_B[3,2]-X_B[3,1]+ R_B[3,2]^2-R_B[3,1]^2)   "Annulus to cylinder walls"
A_B[1]*F_B[1,3]= A_B[3]*F_B[3,1]
A_B[2]*F_B[2,3]= A_B[3]*F_B[3,2]
duplicate j= 1,N
    F_B[3,2+ j]= 0
end
F_B[3,1]+ F_B[3,2]+ F_B[3,3] = 1



duplicate i= 2,N
    R_B[2+ i,1]= (r_m[2+ i]-DELTAr/2)/(d_wafer/2)
    R_B[2+ i,2]= (r_m[2+ i]+ DELTAr/2)/(d_wafer/2)
    X_B[2+ i,1] =  (L_B^4+ 2*L_B^2*(1+ R_B[2+ i,1]^2)+ (1-R_B[2+ i,1]^2)^2)^(1/2)
    X_B[2+ i,2] =  (L_B^4+ 2*L_B^2*(1+ R_B[2+ i,2]^2)+ (1-R_B[2+ i,2]^2)^2)^(1/2)
    F_B[2,2+ i] =  1/(4*L_B)*(X_B[2+ i,2]-X_B[2+ i,1]+ R_B[2+ i,2]^2-R_B[2+ i,1]^2)  "Annulus to cylinder
walls"
    A_B[1]*F_B[1,2+ i]= A_B[2+ i]*F_B[2+ i,1]
    A_B[2]*F_B[2,2+ i]= A_B[2+ i]*F_B[2+ i,2]
    duplicate j= 1,N
        F_B[2+ i,2+ j]= 0
    end
    F_B[2+ i,1]+ F_B[2+ i,2]+ F_B[2+ i,2+ i] = 1
end


"Top Emissivities"
epsilon_T[1]=  0.6 {.55}
epsilon_T[2]=  0.6 {.55}
duplicate i= 1,N
    epsilon_T[2+ i]=  .60
```

```
end

"Top Emissivities"
epsilon_B[1]=  0.60 {.75}
epsilon_B[2]=  0.60 {.55}
duplicate i= 1,N
    epsilon_B[2+ i]=  .60
end

"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
"Radiation Analysis at Steady State"
"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
"Top"
{duplicate p= 1,Np
    duplicate i= 1,2
         q_T[p+ 1,i] =  0* (T_T[p,1]-T_infinity_Lab)/R_Chamber
    end
    q_B[p+ 1,2] =   0* (T_T[p,2]-T_infinity_Lab)/R_Chamber
    q_B[p+ 1,1] =  10                                          "Power of chamber"
end
}

duplicate p= 1, Np
duplicate i =  1, N
    q_T[p,2+ i,1]=   (J_T[p,2+ i]-J_T[p,1])* (A_T[2+ i]* F_T[2+ i,1])
end
end

duplicate p= 1, Np
duplicate i =  1, N
    q_T[p,2+ i,2]=   (J_T[p,2+ i]-J_T[p,2])* (A_T[2+ i]* F_T[2+ i,2])
end
end

duplicate p= 1, Np
duplicate i =  1,N
    q_T[p,2+ i] =  SUM(q_T[p,2+ i,k],k= 1,2)
end
end

duplicate p= 1, Np
q_T[p,1,2]= (J_T[p,1]-J_T[p,2])* (A_T[1]* F_T[1,2])

duplicate i =  1,N+ 2
    q_T[p,i] =  (E_b_T[p,i]-J_T[p,i])* (epsilon_T[i]* A_T[i])/(1-epsilon_T[i])
end
end

duplicate p= 1, Np
q_T[p,1] =  q_T[p,1,2]-SUM(q_T[p,2+ i,1],i= 1,N)
q_T[p,2] =  -q_T[p,1,2]-SUM(q_T[p,2+ i,2],i= 1,N)
end
```

```
"Coupling"
duplicate p= 2, Np
duplicate i= 1,N
    E_b_T[p,2+ i]= E_b_B[p,2+ i]
end
end


"Bottom"
duplicate p= 1, Np
duplicate i =  1, N
    q_B[p,2+ i,1]=   (J_B[p,2+ i]-J_B[p,1])* (A_B[2+ i]* F_B[2+ i,1])
end
end

duplicate p= 1, Np
duplicate i =  1, N
    q_B[p,2+ i,2]=   (J_B[p,2+ i]-J_B[p,2])* (A_B[2+ i]* F_B[2+ i,2])
end
end

duplicate p= 1, Np
duplicate i =  1,N
    q_B[p,2+ i] =  SUM(q_B[p,2+ i,k],k= 1,2)
end
end

duplicate p= 1, Np
q_B[p,1,2]= (J_B[p,1]-J_B[p,2])* (A_B[1]* F_B[1,2])
end

duplicate p= 1, Np
duplicate i =  1,N+ 2
    q_B[p,i] =  (E_b_B[p,i]-J_B[p,i])* (epsilon_B[i]* A_B[i])/(1-epsilon_B[i])
end
end

duplicate p= 1, Np
q_B[p,1] =  q_B[p,1,2]-SUM(q_B[p,2+ i,1],i= 1,N)
q_B[p,2] =  -q_B[p,1,2]-SUM(q_B[p,2+ i,2],i= 1,N)
end
"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
"Convection and Radiation Analysis for Wafer"
"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
F_0 =  (alpha* DELTAt)/(DELTAr)^2                          "Fourier number for system"
Bi_T =   h_TOP* DELTAr/k                                   "Biot number for the top"
Bi_B =  h_BOT* DELTAr/k                                    "Biot number for the bottom"

"Inner Disk (Body 3)"
duplicate p =  1, Np
    T_T[p+ 1,3] =  F_0* (4* T_T[p,4]+ ((DELTAr)/w)* (Bi_T* T_infinity_T[p]+ Bi_B* T_infinity_B[p])-
(DELTAr^2/(k* w))* (q_T[p,3]/A_T[3]+ q_B[p,3]/A_B[3]))+ (1-4* F_0-(DELTAr/w)* (Bi_T+ Bi_B)* F_0)* (T_T[p,3])
end
```

```
"Check for disk stability"
CheckDisk= 1-4*F_0-(DELTAr/w)*(Bi_T+Bi_B)*F_0

"Interior Nodes (Body 4-> N-1)"
duplicate p = 1, Np
    duplicate j = 4,N+1
        T_T[p+1,j] = F_0 *((1-DELTAr/(2*r_m[j]))*T_T[p,j-
1]+(1+DELTAr/(2*r_m[j]))*T_T[p,j+1]+(DELTAr/w)*(Bi_T*T_infinity_T[p]+Bi_B*T_infinity_B[p])-
(DELTAr^2/(k*w))*(q_T[p,j]/A_T[j]+q_B[p,j]/A_B[j]))+(1-2*F_0-(DELTAr/w)*(Bi_T+Bi_B)*F_0)*T_T[p,j]
    end
end

"Check for Inner Stability"
CheckInner = 1-2*F_0-(DELTAr/w)*(Bi_T+Bi_B)*F_0

"Outermost Node"
duplicate p = 1, Np
    T_T[p+1,N+2] = F_0*(((1-
DELTAr/(2*r_m[N+2]))*T_T[p,N+1])+(DELTAr/w)*(Bi_T*T_infinity_T[p]+Bi_B*T_infinity_B[p])-
(DELTAr^2/(k*w))*(q_T[p,N+2]/A_T[N+2]+q_B[p,N+2]/A_B[N+2]))+(1-(1-DELTAr/(2*r_m[N+2]))*F_0-
(DELTAr/w)*(Bi_T+Bi_B)*F_0)*T_T[p,N+2]
end

"Check for Outer Stability"
CheckOuter = 1-(1-DELTAr/(2*r_m[N+2]))*F_0-(DELTAr/w)*(Bi_T+Bi_B)*F_0

"==========================================================
================"
"Convection and Radiation Analysis for Gas"
"==========================================================
================"
alpha_a_T= k_a_T/(rho_a_T*c_p_a_T)                    "Thermal diffusivity of the gas"

k_a_T= 40.7*10^(-3)                                   "W/m K"
rho_a_T= 0.6964                                       "kg/m^3"
c_p_a_T= 1030                                         "J/kgK"

F_0_a_T= alpha_a_T*DELTAt/h_T^2                       "Fourier number for the Gas"

Bi_T_1= h_TOP*h_T/k_a_T
Bi_T_2= h_TOP*h_T/k_a_T
Bi_T_3= h_TOP*h_T/k_a_T

duplicate p = 1,Np
    T_infinity_T[p+1] =
F_0_a_T*(Bi_T_1*T_T[p,1]+2*Bi_T_2*(h_T/r_o[N+2])*T_T[p,2]+2*Bi_T_3*(DELTAr/r_o[N+2])*SUM((r_o[k+
2]/r_o[N+2])*T_T[p,k+2],k=1,N))+(1-
(Bi_T_1+2*Bi_T_2*(h_T/r_o[N+2])+2*Bi_T_3*(DELTAr/r_o[N+2])*SUM(r_o[k+2]/r_o[N+2],k=1,N))*F_0_a_
T)*T_infinity_T[p]
end

CheckGasTop = (1-
(Bi_T_1+2*Bi_T_2*(h_T/r_o[N+2])+2*Bi_T_3*(DELTAr/r_o[N+2])*SUM(r_o[k+2]/r_o[N+2],k=1,N))*F_0_a_
T)
```

```
alpha_a_B = k_a_B/(rho_a_B*c_p_a_B)                    "Thermal diffusivity of the gas"

k_a_B = 40.7*10^(-3)                                   "W/mK"
rho_a_B = 0.6964                                       "kg/m^3"
c_p_a_B = 1030                                         "J/kgK"

F_0_a_B = alpha_a_B*DELTAt/h_B^2                       "Fourier number for the Gas"

Bi_B_1 = h_BOT*h_B/k_a_B
Bi_B_2 = h_BOT*h_B/k_a_B
Bi_B_3 = h_BOT*h_B/k_a_B

duplicate p = 1,Np
   T_infinity_B[p+1] =
F_0_a_B*(Bi_B_1*T_B[p,1]+2*Bi_B_2*(h_B/r_o[N+2])*T_B[p,2]+2*Bi_B_3*(DELTAr/r_o[N+2])*SUM((r_o[k
+2]/r_o[N+2])*T_B[p,k+2],k=1,N))+(1-
(Bi_B_1+2*Bi_B_2*(h_B/r_o[N+2])+2*Bi_B_3*(DELTAr/r_o[N+2])*SUM(r_o[k+2]/r_o[N+2],k=1,N))*F_0_a
_B)*T_infinity_B[p]
end

CheckGasBot = (1-
(Bi_B_1+2*Bi_B_2*(h_B/r_o[N+2])+2*Bi_B_3*(DELTAr/r_o[N+2])*SUM(r_o[k+2]/r_o[N+2],k=1,N))*F_0_a
_B)

"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "

"= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
= = = = = = = = = = = = = = = "
{Duplicate i=1,N+2
c[i,i]=1/epsilon_T[i]-(1-epsilon_T[i])/epsilon_T[i]*F_T[i,i]
Duplicate j=1,N+2
{c[i,j]=-(1-epsilon_T[i])/epsilon_T[i]*F_T[i,j]}
end
end}
```

17 Appendix C
Excel Code

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'CPGRID Mechanical Engineering Senior Project Jan - Dec 2012
'Sponsor - Appled Materials
'
'This code predicts the silicon wafer transient temperature
'response and distribution based on user defined input parameters.
'This code should be used to help in the iteration process of
'the design of CVD and other silicon heat treatment chambers.
'The results of this program are an estimation of the physical
'world. Assumptions are made to help with analysis.
'
'This code was last updated by Nathan Jones on 11/29/12
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'Declare a boolean variable which determines if the main functino can run
Dim CanIRun As Boolean

'This is the main function of the program and is executed when the 'solve'
buttom is pressed
Private Sub CommandButton1_Click()

'check CanIRun
'If it is false then exit the function here and display an error message
If CanIRun = False Then
    disp = MsgBox("Cannot Solve - Check Errors", vbOKOnly, "Error")
    Exit Sub
Else
End If

'If the constant DEBUG_ is true then display debugging info in the immediate
window
#Const DEBUG_ = False

'Don't update the screen
Application.ScreenUpdating = False

'Define different worksheets within ths workbook to reference later in the
function
Set sFDWafer = ThisWorkbook.Worksheets("Finite Difference")
Set sSolution = ThisWorkbook.Worksheets("Solution")
Set sResponse = ThisWorkbook.Charts("Wafer Temperature Response")

'Constants
Dim sigma, h_B, h_T, w, d_wafer, r_wafer, d_chamber, r_chamber, d_chuck,
r_chuck, delta_r, time, h_TOP, h_BOTTOM As Single
Dim timeCells As Integer

'Sigma [W/m^2*K^4]
sigma = 5.670373 * 10 ^ (-8)

'Convection Coefficients are dependent on the pressures
```

```
'NOTE that if the pressures are extremely low the values are forced to 0
'instead of the LN(x) limit of -infinity
h_TOP = 12.432 * Application.WorksheetFunction.Ln(Range("C14").Value) +
7.8646
h_BOTTOM = 12.432 * Application.WorksheetFunction.Ln(Range("C6").Value) +
7.8646
If h_TOP < 0 Then h_TOP = 0
If h_BOTTOM < 0 Then h_BOTTOM = 0

'Define Pi to be used in area calculations
Pi = 4 * Atn(1)

'Distance [mm->m]
h_B = Range("C19").Value / 1000
h_T = Range("C18").Value / 1000

'Dimensions
w = Range("C3").Value / 1000            'Wafer Thickness
d_wafer = Range("C17").Value / 1000
r_wafer = d_wafer / 2
d_chamber = Range("C16").Value / 1000
r_chamber = d_chamber / 2
d_chuck = d_wafer
r_chuck = d_chuck / 2

'Define time parameters
time = Range("C20").Value
delta_t = Range("C21").Value
t_print = (delta_t ^ -1)
printme = 1
timeCells = (delta_t ^ -1) * time

'Force the chamber gas to be Argon regardless of user input
'This done because the supplied comparative data is an Argon only environment
gasTop = "Ar"
gasBot = "Ar"
wafTop = Range("C4").Value
wafBot = Range("C5").Value

'Define nodes
Dim n As Integer
n = Range("C22").Value

'Define delta_r
delta_r = d_wafer / ((2 * n) - 1)

'Define all arrays to be used in Analysis
'Arrays can be 2D and are depented on the number of nodes
Dim T() As Single           'Wafer Temperature
ReDim T(3 To (n + 2))
Dim r() As Single           'Anular Radius Array
ReDim r(3 To (n + 2))
Dim q_T() As Single         'Radiation Flux Top
ReDim q_T(3 To (n + 2))
Dim q_B() As Single         'Radiation Flux Bottom
ReDim q_B(3 To (n + 2))
Dim T_inf() As Single       'T inifity gas
```

```
ReDim T_inf(1 To 2)          '(1) = TOP, (2) = BOTTOM
Dim AirSum1() As Single      'Sigma Summation term in FD for gas
ReDim AirSum1(3 To (n + 2))
Dim AirSum2() As Single
ReDim AirSum2(3 To (n + 2))
Dim E_B() As Single
ReDim E_B(1 To (n + 2), 1 To 1)
Dim E_T() As Single
ReDim E_T(1 To (n + 2), 1 To 1)
Dim C_B() As Single
ReDim C_B(1 To (n + 2), 1 To (n + 2))
Dim C_T() As Single
ReDim C_T(1 To (n + 2), 1 To (n + 2))
Dim emiss() As Single
ReDim emiss(1 To (n + 2))
Dim emiss_b() As Single
ReDim emiss_b(1 To (n + 2))
Dim F_T() As Single
ReDim F_T(1 To (n + 2), 1 To (n + 2))
Dim F_B() As Single
ReDim F_B(1 To (n + 2), 1 To (n + 2))
Dim A_T() As Single
ReDim A_T(1 To (n + 2))
Dim A_B() As Single
ReDim A_B(1 To (n + 2))
Dim J_B() As Variant
Dim J_T() As Variant

'Convert Temperatures to [K]
T_T1 = Range("C10").Value + 273.15      'Lid
T_T2 = Range("C12").Value + 273.15      'TopWall
T_T3 = Range("C2").Value + 273.15       'TopWafer
T_B1 = Range("C8").Value + 273.15       'Chuck
T_B2 = Range("C12").Value + 273.15      'BottomWall
T_B3 = Range("C2").Value + 273.15       'BottomWafer

'Define viewfactor calculation variables
Dim valueR1, valueR2, valueR3, valueX, valueX2, valueL, valueF, valueA,
valueH As Single

'Calculate the Area array
'A_T are the top areas, A_B are the bottom areas
A_T(1) = Pi * r_chamber ^ 2
A_T(2) = 2 * Pi * r_chamber * h_T
A_T(3) = Pi * (delta_r / 2) ^ 2
For i = 4 To n + 2
    A_T(i) = Pi * ((delta_r * ((i - 3) + 0.5)) ^ 2 - ((delta_r * ((i - 4) +
0.5)) ^ 2))
Next i
A_B(1) = Pi * r_chuck ^ 2
A_B(2) = 2 * Pi * r_chamber * h_B
A_B(3) = Pi * (delta_r / 2) ^ 2
For i = 4 To n + 2
    A_B(i) = A_T(i)
Next i
```

```
'F_T Matrix-------------------------------------------------------------------
------
'Zeros on Diagonal
For i = 1 To n + 2
    F_T(i, i) = 0
Next i

'Lid View Factors
valueR1 = r_chamber / h_T
valueR2 = (delta_r / 2) / h_T
valueX = 1 + (1 + valueR2 ^ 2) / (valueR1 ^ 2)
F_T(1, 3) = 0.5 * (valueX - (valueX ^ 2 - 4 * (valueR2 / valueR1) ^ 2) ^ 0.5)
For i = 4 To n + 2
    valueR1 = h_T / r_chamber
    valueR2 = (delta_r * ((i - 4) + 0.5)) / r_chamber
    valueR3 = (delta_r * ((i - 3) + 0.5)) / r_chamber
    F_T(1, i) = 0.5 * (valueR3 ^ 2 - valueR2 ^ 2 - ((1 + valueR3 ^ 2 +
valueR1 ^ 2) ^ 2 _
                - 4 * (valueR3 ^ 2)) ^ 0.5 + ((1 + valueR2 ^ 2 + valueR1 ^ 2)
^ 2 - 4 * valueR2 ^ 2) ^ 0.5)
Next i
valueH = h_T / 2 / r_chamber
F_T(1, 2) = 2 * valueH * ((1 + valueH ^ 2) ^ (1 / 2) - valueH)

'Chamber View Factors
F_T(2, 1) = F_T(1, 2) * A_T(1) / A_T(2)
valueH = h_T / 2 / r_chamber
F_T(2, 2) = (1 + valueH) - (1 + valueH ^ 2) ^ (1 / 2)
valueR1 = h_T / (delta_r / 2)  'H
valueR2 = r_chamber / (delta_r / 2)   'R
F_T(3, 2) = 0.5 * (1 - valueR2 ^ 2 - valueR1 ^ 2 + ((1 + valueR2 ^ 2 +
valueR1 ^ 2) ^ 2 - 4 * (valueR2) ^ 2) ^ 0.5)
F_T(2, 3) = F_T(3, 2) * A_T(3) / A_T(2)
For i = 4 To n + 2
    valueR1 = (delta_r * ((i - 4) + 0.5)) / r_chamber 'R1
    valueR2 = (delta_r * ((i - 3) + 0.5)) / r_chamber  'R2
    valueL = h_T / r_chamber 'L
    valueX = (valueL ^ 4 + ((2 * valueL ^ 2) * (1 + valueR1 ^ 2)) + (1 -
valueR1 ^ 2) ^ 2) ^ 0.5
    valueX2 = (valueL ^ 4 + ((2 * valueL ^ 2) * (1 + valueR2 ^ 2)) + (1 -
valueR2 ^ 2) ^ 2) ^ 0.5
    F_T(2, i) = (1 / (4 * valueL)) * (valueX2 - valueX + valueR2 ^ 2 -
valueR1 ^ 2)
Next i

'Wafer View Factors
F_T(3, 1) = F_T(1, 3) * A_T(1) / A_T(3)
For i = 3 To n + 2
    F_T(3, i) = 0
Next i
For i = 4 To n + 2
    F_T(i, 1) = F_T(1, i) * A_T(1) / A_T(i)
    F_T(i, 2) = F_T(2, i) * A_T(2) / A_T(i)
Next i

'F_B Matrix-------------------------------------------------------------------
------
```

```
'Zeros on Diagonal
For i = 1 To n + 2
    F_B(i, i) = 0
Next i

'Lid View Factors
valueR1 = r_chamber / h_B
valueR2 = (delta_r / 2) / h_B
valueX = 1 + (1 + valueR2 ^ 2) / (valueR1 ^ 2)
F_B(1, 3) = 0.5 * (valueX - (valueX ^ 2 - 4 * (valueR2 / valueR1) ^ 2) ^ 0.5)
For i = 4 To n + 2
    valueR1 = h_B / r_chamber
    valueR2 = (delta_r * ((i - 4) + 0.5)) / r_chamber
    valueR3 = (delta_r * ((i - 3) + 0.5)) / r_chamber
    F_B(1, i) = 0.5 * (valueR3 ^ 2 - valueR2 ^ 2 - ((1 + valueR3 ^ 2 +
valueR1 ^ 2) ^ 2 _
                - 4 * (valueR3 ^ 2)) ^ 0.5 + ((1 + valueR2 ^ 2 + valueR1 ^ 2)
^ 2 - 4 * valueR2 ^ 2) ^ 0.5)
Next i
valueH = h_B / 2 / r_chamber
F_B(1, 2) = 2 * valueH * ((1 + valueH ^ 2) ^ (1 / 2) - valueH)

'Chamber View Factors
F_B(2, 1) = F_B(1, 2) * A_B(1) / A_B(2)
valueH = h_B / 2 / r_chamber
F_B(2, 2) = (1 + valueH) - (1 + valueH ^ 2) ^ (1 / 2)
valueR1 = h_B / (delta_r / 2)  'H
valueR2 = r_chamber / (delta_r / 2)   'R
F_B(3, 2) = 0.5 * (1 - valueR2 ^ 2 - valueR1 ^ 2 + ((1 + valueR2 ^ 2 +
valueR1 ^ 2) ^ 2 - 4 * (valueR2) ^ 2) ^ 0.5)
F_B(2, 3) = F_B(3, 2) * A_B(3) / A_B(2)
For i = 4 To n + 2
    valueR1 = (delta_r * ((i - 4) + 0.5)) / r_chamber 'R1
    valueR2 = (delta_r * ((i - 3) + 0.5)) / r_chamber  'R2
    valueL = h_B / r_chamber 'L
    valueX = (valueL ^ 4 + ((2 * valueL ^ 2) * (1 + valueR1 ^ 2)) + (1 -
valueR1 ^ 2) ^ 2) ^ 0.5
    valueX2 = (valueL ^ 4 + ((2 * valueL ^ 2) * (1 + valueR2 ^ 2)) + (1 -
valueR2 ^ 2) ^ 2) ^ 0.5
    F_B(2, i) = (1 / (4 * valueL)) * (valueX2 - valueX + valueR2 ^ 2 -
valueR1 ^ 2)
Next i

'Wafer View Factors
F_B(3, 1) = F_B(1, 3) * A_B(1) / A_B(3)
For i = 3 To n + 2
    F_B(3, i) = 0
Next i
For i = 4 To n + 2
    F_B(i, 1) = F_B(1, i) * A_B(1) / A_B(i)
    F_B(i, 2) = F_B(2, i) * A_B(2) / A_B(i)
Next i

'Assumes temperature inputs are equal to the showerhead and chuck
respectively
T_inf(1) = T_T1
T_inf(2) = T_B1
```

```
For i = 3 To (n + 2)
    'Define initial heatfluxes
    q_T(i) = 0
    q_B(i) = 0
Next i

'clear sheets
sFDWafer.Cells.Clear
sSolution.Range("A2:C65535").Clear

'input inital temperatures into spreadsheet
For i = 3 To (n + 2)
    sFDWafer.Cells(1, i).Value = T_T3
    T(i) = sFDWafer.Cells(1, i).Value
Next i

'solve for all r_i
For i = 3 To (n + 2)
    r(i) = (i - 3) * delta_r
Next i

'emissivity
For i = 3 To n + 2
    emiss(i) = Module1.dataTableValue("" & wafTop & "", T(3), 5)
    emiss_b(i) = Module1.dataTableValue("" & wafBot & "", T(3), 5)
Next i
emiss(1) = Module1.dataTableValue("" & Range("C11").Value & "", T_T1, 5)
emiss(2) = Module1.dataTableValue("" & Range("C13").Value & "", T_T2, 5)
emiss_b(1) = Module1.dataTableValue("" & Range("C9").Value & "", T_B1, 5)
emiss_b(2) = Module1.dataTableValue("" & Range("C13").Value & "", T_B2, 5)

'Blackbody fluxes in [W/m^2]
E_T(1, 1) = sigma * T_T1 ^ 4
E_T(2, 1) = sigma * T_T2 ^ 4
E_B(1, 1) = sigma * T_B1 ^ 4
E_B(2, 1) = sigma * T_B2 ^ 4

'print inital temperatures in output cells
sSolution.Cells(2, 1).Value = 0
sSolution.Cells(2, 2).Value = T_T3 - 273.15
sSolution.Cells(2, 3).Value = T_T3 - 273.15

'Solve for T() array
For p = 1 To timeCells

    'Finite Difference for Wafer
    'Fourier Number - Silicon
    'look up physical values using dataTableValue module
    rho = Module1.dataTableValue("" & wafTop & "", T(3), 2)
    k = Module1.dataTableValue("" & wafTop & "", T(3), 3)
    c_p = Module1.dataTableValue("" & wafTop & "", T(3), 4) * 1000
    alpha = k / (rho * c_p)
    Fo = alpha * delta_t / (delta_r) ^ 2
    rho_B = Module1.dataTableValue("" & wafBot & "", T(3), 2)
    k_b = Module1.dataTableValue("" & wafBot & "", T(3), 3)
    c_p_B = Module1.dataTableValue("" & wafBot & "", T(3), 4) * 1000
    alpha_b = k / (rho * c_p)
```

```
    Fo_b = alpha * delta_t / (delta_r) ^ 2
    Bi_B = h_BOTTOM * delta_r / k_b
    Bi_T = h_TOP * delta_r / k

    'Fourier Number - Gas
    rho_a = Module1.dataTableValue("" & gasTop & "", T_inf(1), 2)
    k_a = Module1.dataTableValue("" & gasTop & "", T_inf(1), 4)
    c_p_a = Module1.dataTableValue("" & gasTop & "", T_inf(1), 3) * 1000
    alpha_a = k_a / (rho_a * c_p_a)
    rho_a_b = Module1.dataTableValue("" & gasBot & "", T_inf(2), 2)
    k_a_b = Module1.dataTableValue("" & gasBot & "", T_inf(2), 4)
    c_p_a_b = Module1.dataTableValue("" & gasBot & "", T_inf(2), 3) * 1000
    alpha_a_b = k_a_b / (rho_a_b * c_p_a_b)
    Fo_a_b = alpha_a_b * delta_t / (delta_r) ^ 2
    Bi_a_B = h_BOTTOM * h_B / k_a_b
    Bi_a_T = h_TOP * h_T / k_a

    'Radiation Matrices
    For i = 3 To n + 2
        E_T(i, 1) = sigma * T(i) ^ 4
        E_B(i, 1) = sigma * T(i) ^ 4
    Next i

    'C_T & C_B Matrix-------------------------------------------------------
------------
    For i = 1 To n + 2
        For J = 1 To n + 2
            C_T(i, J) = -((1 - emiss(i)) / emiss(i)) * F_T(i, J)
            C_B(i, J) = -((1 - emiss_b(i)) / emiss_b(i)) * F_B(i, J)
        Next J
    Next i
    For i = 1 To n + 2
        C_T(i, i) = (1 / emiss(i)) - ((1 - emiss(i)) / emiss(i)) * F_T(i, i)
        C_B(i, i) = (1 / emiss_b(i)) - ((1 - emiss_b(i)) / emiss_b(i)) *
F_B(i, i)
    Next i

    'Invert and multiply to solve for the J matrices
    J_T = Application.MMult((Application.MInverse(C_T)), E_T)
    J_B = Application.MMult((Application.MInverse(C_B)), E_B)

    'Solve for the resulting q values
    For i = 3 To n + 2
        q_T(i) = ((E_T(i, 1) - J_T(i, 1)) / ((1 - emiss(i)) / (A_T(i) *
emiss(i)))) / A_T(i)
        q_B(i) = ((E_B(i, 1) - J_B(i, 1)) / ((1 - emiss_b(i)) / (A_B(i) *
emiss_b(i)))) / A_B(i)
    Next i

    'Center Node
    sFDWafer.Cells(p + 1, 3).Value = Fo * ((4 * T(4)) + ((delta_r / w) *
(Bi_T * T_inf(1) + Bi_B * T_inf(2))) _
                                        - (delta_r ^ 2 / (k * w) * (q_T(3) +
q_B(3)))) _
                                        + (1 - 4 * Fo - (delta_r / w) * (Bi_T
+ Bi_B) * Fo) * T(3)
    sFDWafer.Cells(p + 1, 7).Value = q_T(3)
```

```
    sFDWafer.Cells(p + 1, 8).Value = q_B(3)

    'Middle Nodes
    For i = 4 To (n + 1)
        sFDWafer.Cells(p + 1, i).Value = Fo * ((1 - (delta_r / 2 / r(i))) *
T(i - 1) _
                                             + (1 + (delta_r / 2 / r(i))) *
T(i + 1) _
                                             + ((delta_r / w) * (Bi_T *
T_inf(1) + Bi_B * T_inf(2))) _
                                             - ((delta_r ^ 2 / k / w) *
(q_T(i) + q_B(i))) _
                                             + (1 - 2 * Fo - (delta_r / w) *
(Bi_T + Bi_B) * Fo) * T(i)
    Next i

    'Edge Node
    sFDWafer.Cells(p + 1, n + 2).Value = Fo * ((1 - (delta_r / 2 / r(n + 2)))
* T(n + 1) _
                                             + ((delta_r / w) * (Bi_T *
T_inf(1) + Bi_B * T_inf(2))) _
                                             - ((delta_r ^ 2 / k / w) * (q_T(n
+ 2) + q_B(n + 2))) _
                                             + (1 - (1 - delta_r / 2 / r(n +
2)) * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo) * T(n + 2)

    'Air Temperature
    For i = 3 To (n + 2)
        AirSum1(i) = (r(i) / r_wafer) * T(i)
        AirSum2(i) = (r(i) / r_wafer)
    Next i
    sFDWafer.Cells(p + 1, 1).Value = Fo_a * (Bi_a_T * T_T1 + 2 * Bi_a_T *
(h_T / r_wafer) * T_T2 _
                                        + 2 * Bi_a_T * (delta_r / r_wafer) *
Application.WorksheetFunction.Sum(AirSum1)) _
                                        + (1 - (Bi_a_T + 2 * Bi_a_T * (h_T /
r_wafer) + 2 * Bi_a_T * (delta_r / r_wafer) * _

Application.WorksheetFunction.Sum(AirSum2)) * Fo_a) * T_inf(1)
    sFDWafer.Cells(p + 1, 2).Value = Fo_a_b * (Bi_a_B * T_B1 + 2 * Bi_a_B *
(h_B / r_wafer) * T_B2 _
                                        + 2 * Bi_a_B * (delta_r / r_wafer) *
Application.WorksheetFunction.Sum(AirSum1)) _
                                        + (1 - (Bi_a_B + 2 * Bi_a_B * (h_B /
r_wafer) + 2 * Bi_a_B * (delta_r / r_wafer) * _

Application.WorksheetFunction.Sum(AirSum2)) * Fo_a_b) * T_inf(2)

    #If DEBUG_ = True Then
    For i = 3 To (n + 2)
        Debug.Print "r_" & i & " "; r(i)
    Next i
    Debug.Print "Center "; (1 - 4 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo)
    Debug.Print "Middle "; (1 - 2 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo)
    Debug.Print "Edge "; (1 - (1 - delta_r / 2 / r(n + 2)) * Fo - (delta_r /
w) * (Bi_T + Bi_B) * Fo)
    #End If
```

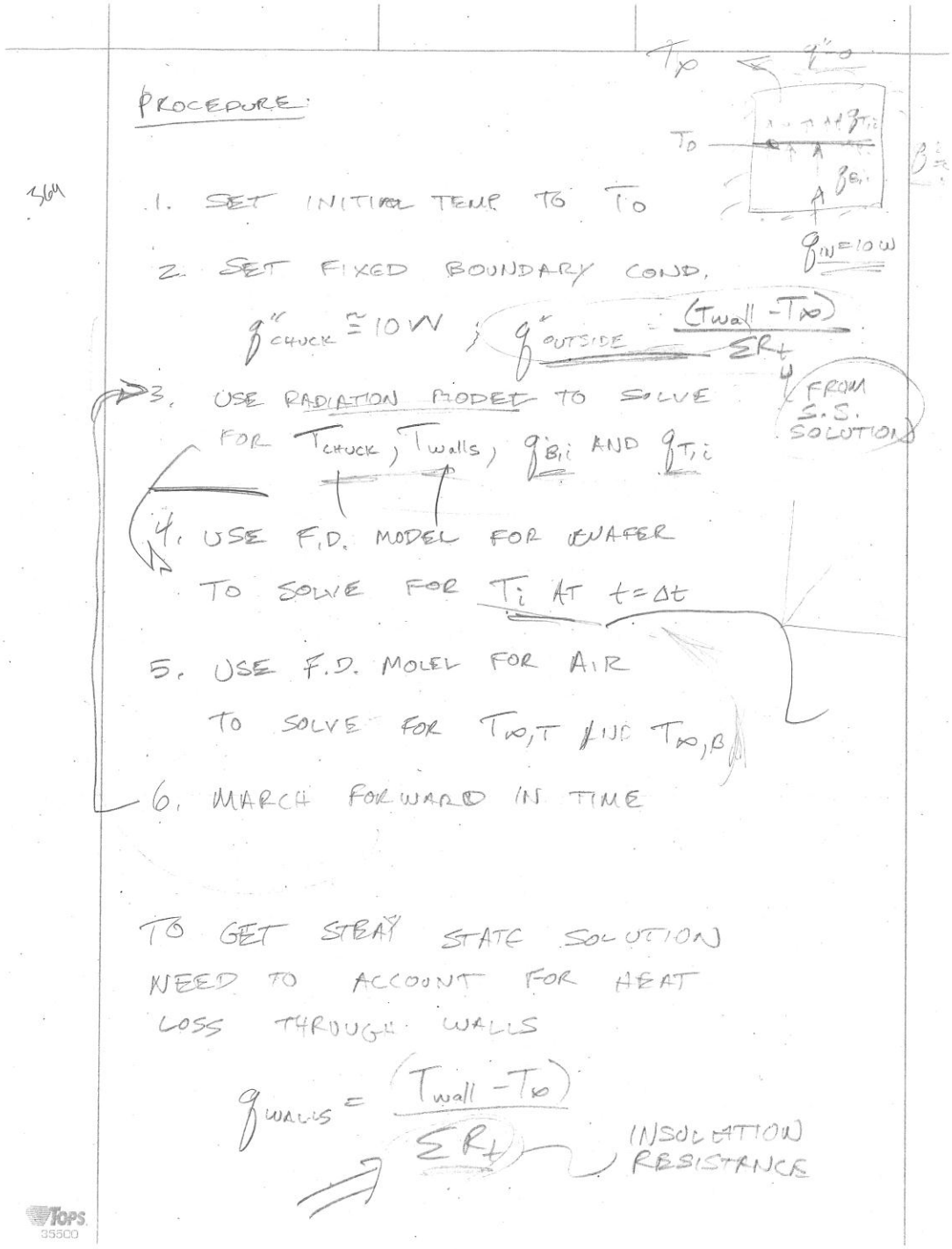
```
    If (1 - 4 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo) <= 0 Then
        disp = MsgBox("Inner Node Unstable: " & vbCrLf & _
                      (1 - 4 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo) &
"< 0" & vbCrLf & _
                      "Recommend Decreasing the Time Step", vbOKOnly,
"Error")
        Exit Sub
    Else
    End If

    If (1 - 2 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo) <= 0 Then
        disp = MsgBox("Intermediate Nodes Unstable: " & vbCrLf & _
                      (1 - 2 * Fo - (delta_r / w) * (Bi_T + Bi_B) * Fo) &
"< 0" & vbCrLf & _
                      "Recommend Decreasing the Time Step", vbOKOnly,
"Error")
        Exit Sub
    Else
    End If

    If (1 - (1 - delta_r / 2 / r(n + 2)) * Fo - (delta_r / w) * (Bi_T + Bi_B)
* Fo) <= 0 Then
        disp = MsgBox("Outer Node Unstable: " & vbCrLf & _
                      (1 - (1 - delta_r / 2 / r(n + 2)) * Fo - (delta_r /
w) * (Bi_T + Bi_B) * Fo) & "< 0" & vbCrLf & _
                      "Recommend Decreasing the Time Step", vbOKOnly,
"Error")
        Exit Sub
    Else
    End If

    If p / t_print = printme Then
        sSolution.Cells(printme + 2, 1).Value = delta_t * p
        sSolution.Cells(printme + 2, 2).Value = sFDWafer.Cells(p + 1,
3).Value - 273.15
        sSolution.Cells(printme + 2, 3).Value = sFDWafer.Cells(p + 1, n +
2).Value - 273.15
        printme = printme + 1
    End If

    'Put values into T()array for next iteration
    'T_inf(1) = sFDWafer.Cells(p + 1, 1).Value
    'T_inf(2) = sFDWafer.Cells(p + 1, 2).Value
    For i = 3 To (n + 2)
        T(i) = sFDWafer.Cells(p + 1, i).Value
    Next i

    For i = 3 To (n + 2)
        emiss(i) = Module1.dataTableValue("" & wafTop & "", T(3), 5)
        emiss_b(i) = Module1.dataTableValue("" & wafBot & "", T(3), 5)
    Next i
Next p

sResponse.Activate

End Sub
```

```
'This Function runs every time a difference cell is selected or when a value
is entered into a cell
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
'Disable screen updating to speed up execution
Application.ScreenUpdating = False

'This boolean varaible is referenced to check if the code should be executed
'CanIRun value is set here and checked in the main function
CanIRun = True

'This set of "IF" statements checks if the entered values are within the
defined bounds
'If the entered values fall outsde the bounds, then the string "Error" is
displayed in that cell
If Range("C2").Value < 0 Or Range("C2").Value > 999 Then Range("C2").Value =
"Error"
If Range("C3").Value < 0.1 Or Range("C3").Value > 2 Then Range("C3").Value =
"Error"
If Range("C6").Value < 0.000000009 Or Range("C6").Value > 900 Then
Range("C6").Value = "Error"
If Range("C8").Value < 0 Or Range("C8").Value > 900 Then Range("C8").Value =
"Error"
If Range("C10").Value < 0 Or Range("C10").Value > 900 Then Range("C10").Value
= "Error"
If Range("C12").Value < 0 Or Range("C12").Value > 999 Then Range("C12").Value
= "Error"
If Range("C14").Value < 0.000000009 Or Range("C14").Value > 900 Then
Range("C14").Value = "Error"
If Range("C17").Value < 0 Or Range("C17").Value > 500 Then Range("C17").Value
= "Error"
If Range("C18").Value < 0 Or Range("C18").Value > 100 Then Range("C18").Value
= "Error"
If Range("C19").Value < 0 Or Range("C19").Value > 50 Then Range("C19").Value
= "Error"
If Range("C20").Value < 0 Or Range("C20").Value > 1000 Then
Range("C20").Value = "Error"

'Check all the value column cells for the string Error
'If Error then turn the cell red
'If not then format the cell normally
For i = 2 To 20
    checkError = Cells(i, 3).Value
    If checkError = "Error" Then
        Cells(i, 3).Style = "Bad"
        'If there is an error than set CanIRun to false
        CanIRun = False
    Else
        Cells(i, 3).Style = "Normal"
        Cells(i, 3).HorizontalAlignment = xlCenter
    End If
Next i

'check CanIRun, If false then stop function here
If CanIRun = False Then
    GoTo LEAVE
Else
End If
```

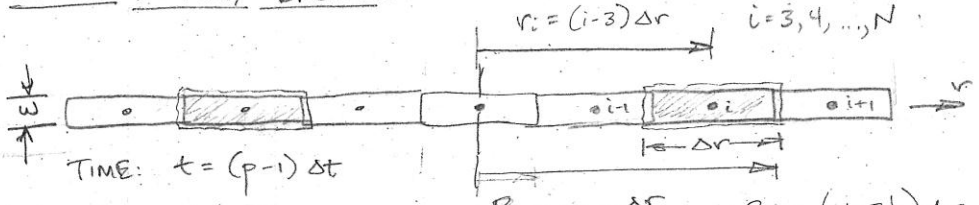
```
'Check diameter of wafer and declare an appropriate number of nodes based on
that diameter
d_wafer = Range("C17").Value
Select Case d_wafer
    Case Is < 200
        Range("C22").Value = 3
    Case Is < 250
        Range("C22").Value = 4
    Case Is < 300
        Range("C22").Value = 5
    Case Is < 350
        Range("C22").Value = 6
    Case Is < 300
        Range("C22").Value = 7
    Case Is < 350
        Range("C22").Value = 8
    Case Is < 400
        Range("C22").Value = 9
    Case Else
        Range("C22").Value = 10
End Select

'Reformat all boarders that were changed during the above formatting
LEAVE:
Range("C7").Borders(xlEdgeBottom).LineStyle = xlContinuous
Range("C7").Borders(xlEdgeBottom).Weight = xlMedium
Range("C9").Borders(xlEdgeBottom).LineStyle = xlContinuous
Range("C9").Borders(xlEdgeBottom).Weight = xlMedium
Range("C11").Borders(xlEdgeBottom).LineStyle = xlContinuous
Range("C11").Borders(xlEdgeBottom).Weight = xlMedium
Range("C15").Borders(xlEdgeBottom).LineStyle = xlContinuous
Range("C15").Borders(xlEdgeBottom).Weight = xlMedium
Range("C19").Borders(xlEdgeBottom).LineStyle = xlContinuous
Range("C19").Borders(xlEdgeBottom).Weight = xlMedium

'Ensure that the chamber diameter is equal to the wafer diameter
Range("C16").Value = Range("C17").Value
End Sub

'This function interpolates a data table
Function dataTableValue(table As String, temperature As Variant, parameter As
Variant)

Dim lookUpTable As String
Dim lookUpTemperature As Single
Dim lookUpParameter As Integer
Dim sheetTable As Worksheet

Dim cellOne As Integer
Dim cellTwo As Integer

Dim temperatureOne As Single
Dim temperatureTwo As Single
Dim parameterOne As Single
Dim parameterTwo As Single
```

```
lookUpTable = table
lookUpTemperature = temperature
lookUpParameter = parameter

'Set parameter table
Set sheetTable = ThisWorkbook.Worksheets("" & lookUpTable & "")
'Set rows corresponding to temperatures
cellOne = Application.Match(lookUpTemperature, sheetTable.Range("A2:A38"), 1)
cellTwo = cellOne + 1

'Set temperatures
temperatureOne = sheetTable.Cells(1 + cellOne, 1).Value
temperatureTwo = sheetTable.Cells(1 + cellTwo, 1).Value

'Find parameters according to input column
parameterOne = Application.VLookup(temperatureOne,
sheetTable.Range("A2:H38"), lookUpParameter, True)
parameterTwo = Application.VLookup(temperatureTwo,
sheetTable.Range("A2:H38"), lookUpParameter, True)

'Debug
'sheetTable.Cells(39, 2).Value = temperatureOne
'sheetTable.Cells(40, 2).Value = temperatureTwo
'sheetTable.Cells(41, 2).Value = parameterOne
'sheetTable.Cells(42, 2).Value = parameterTwo

'Interpolate
dataTableValue = (((parameterTwo - parameterOne) / (temperatureTwo -
temperatureOne)) _
                  * (temperature - temperatureOne)) + parameterOne
End Function
```

18 Appendix D



WAFER ENERGY BALANCE:



TIME: $t = (p-1)\Delta t$

$\alpha = k/(\rho c)$

$R_i = r_i + \frac{\Delta r}{2}$; $R_N = \left(N - 2\tfrac{1}{2}\right)\Delta r$

INTERIOR NODES: $(4 \le i \le N-1)$ $\qquad \Delta r = \frac{R_N}{N-2.5}$

$$\dot{E}_{st} = \dot{E}_{IN} - \dot{E}_{OUT} + \cancel{\dot{E}_g} \quad \text{(ASSUME: CONDUCTION AND CONV. IN)}$$

$$\dot{E}_{st} = \rho c \forall_i \frac{\partial T}{\partial t} \cong \frac{k}{\alpha}\left(2\pi r_i \cdot \Delta r \cdot w\right)\cdot\left(\frac{T_i^{p+1} - T_i^p}{\Delta t}\right)$$

$$\dot{E}_{IN,COND} = \left(-kA\nabla T\right)_{i-\frac{1}{2}} + \left(kA\nabla T\right)_{i+\frac{1}{2}} \quad \text{(FOURIER'S LAW)}$$

$$\cong k \cdot w \cdot 2\pi\left[\left(r_i - \frac{\Delta r}{2}\right)\frac{\left(T_{i-1}^p - T_i^p\right)}{\Delta r} + \left(r_i + \frac{\Delta r}{2}\right)\frac{\left(T_{i+1}^p - T_i^p\right)}{\Delta r}\right]$$

$$\cong \frac{2\pi k w r_i}{\Delta r}\left[\left(1 - \frac{\Delta r}{2r_i}\right)T_{i-1}^p + \left(1 + \frac{\Delta r}{2r_i}\right)T_{i+1}^p - 2T_i^p\right]$$

$$\dot{E}_{IN,CONV} = h_B A_{s,i}\left(T_{B,\infty}^p - T_i^p\right) + h_T A_{s,i}\left(T_{T,\infty} - T_i^p\right)$$

$$= 2\pi r_i \Delta r\left[h_B T_{B,\infty}^p + h_T T_{T,\infty}^T - \left(h_B + h_T\right)T_i^p\right]$$

$$\dot{E}_{OUT,RAD} = A_{s,i}\left(q''_{T,i} + q''_{B,i}\right) \quad \text{(FOR RADIATION LEAVING (i))}$$

COMBINE AND MULTIPLY BY: $\frac{\Delta r}{2\pi k w r_i}$

$$\left(\frac{\Delta r^2}{\alpha \Delta t}\right)\left(T_i^{p+1} - T_i^p\right) = \left(1 - \frac{\Delta r}{2r_i}\right)T_{i-1}^p + \left(1 + \frac{\Delta r}{2r_i}\right)T_{i+1}^p - 2T_i^p$$

$$+ \frac{\Delta r^2}{kw}\left[h_B T_{B,\infty}^p + h_T T_{T,\infty}^p - \left(h_B + h_T\right)T_i^p - \left(q''_{T,i} + q''_{B,i}\right)\right]$$

LET: $F_o = \frac{\alpha \cdot \Delta t}{\Delta r^2}$ FOURIER NUMBER

$Bi_B = \frac{h_B \cdot \Delta r}{k}$ BIOT NUMBER, TOP

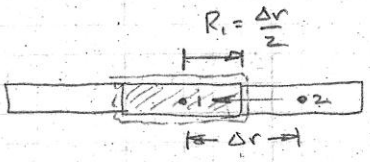
$Bi_T = \frac{h_T \Delta r}{k}$ BIOT NUMBER BOTTOM

$$T_i^{P+1} = F_o \left\{ \left(1 - \frac{\Delta r}{2r_i}\right) T_{i-1}^P + \left(1 + \frac{\Delta r}{2r_i}\right) T_{i+1}^P + \right.$$
$$\left. + \left(\frac{\Delta r}{\omega}\right)\left(Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P\right) - \left(\frac{\Delta r^2}{k\omega}\right)\left(q''_{T,i} + q''_{B,i}\right) \right\}$$
$$+ \left[1 - 2F_o - \left(\frac{\Delta r}{\omega}\right)\left(Bi_T + Bi_B\right) F_o \right] T_i^P$$

STABILITY CRITERION:

$$\left[1 - 2F_o - \left(\frac{\Delta r}{\omega}\right)\left(Bi_B + Bi_T\right) F_o \right] \geq 0$$

$$\Rightarrow F_o \leq \frac{1}{2 + \left(\frac{\Delta r}{\omega}\right)\left(Bi_B + Bi_T\right)}$$



INNER DISK: $(i = 3)$

$$\frac{k}{\alpha}\left(\frac{\pi}{4}\Delta r^2 \cdot \omega\right)\left(\frac{T_i^{P+1} - T_i^P}{\Delta t}\right) = k \cancel{2\pi}\left(\frac{\Delta r}{2}\right)\omega \frac{\left(T_2^P - T_i^P\right)}{\Delta r}$$
$$+ \left(\frac{\pi}{4}\Delta r^2\right)\left[h_B T_{B,\infty}^P + h_T T_{T,\infty}^P - \left(h_B + h_T\right) T_i^P - \left(q''_{T,i} + q''_{B,i}\right)\right]$$

$$\frac{1}{F_o}\left(T_i^{P+1} - T_i^P\right) = 4\left(T_2^P - T_i^P\right) + \left(\frac{\Delta r}{\omega}\right)\left[Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P \right.$$
$$\left. - \left(Bi_T + Bi_B\right) T_i^P \right] - \frac{\Delta r^2}{k\omega}\left(q''_{T,i} + q''_{B,i}\right)$$

$$T_1^{P+1} = F_o \left\{ 4T_2^P + \left(\frac{\Delta r}{\omega}\right)\left(Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P\right) - \left(\frac{\Delta r^2}{k\omega}\right)\left(q''_{T,i} + q''_{B,i}\right) \right\}$$
$$+ \left[1 - 4F_o - \left(\frac{\Delta r}{\omega}\right)\left(Bi_T + Bi_B\right) F_o \right] T_1^P$$

NOTE: FOR N=3 (SINGE DISC + NO CONDUCTION)

$$T_1^{P+1} = F_o\left[\left(\frac{\Delta r}{\omega}\right)\left(Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P\right) - \left(\frac{\Delta r^2}{k\omega}\right)\left(q''_{T,1} + q''_{B,1}\right)\right] + \left[1 - \left(\frac{\Delta r}{\omega}\right)\left(Bi_T + Bi_B\right)\right] T_1^P$$

$q'' = 0$

OUTER DISK ($i = N$)

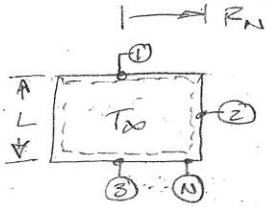
$$\frac{k}{\alpha}\left(2\pi r_N \cdot \Delta r \omega\right)\left(\frac{T_N^{P+1} - T_N^P}{\Delta t}\right) = 2\pi' k\omega \cdot \left(r_N - \frac{\Delta r}{2}\right)\frac{\left(T_{N-1}^P - T_N^P\right)}{\Delta r} + 2\pi r_i \Delta r\left[h_B T_{B,\infty}^P + h_T T_{T,\infty}^P - \left(h_B + h_T\right) T_N^P - \left(q''_{T,N} + q''_{B,N}\right)\right]$$

$$\frac{1}{F_o}\left(T_N^{P+1} - T_N^P\right) = \left(1 - \frac{\Delta r}{2r_N}\right)\left(T_{N-1}^P - T_N^P\right) + \left(\frac{\Delta r}{\omega}\right)\left[Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P - \left(Bi_T + Bi_B\right) T_N^P\right] - \frac{\Delta r^2}{k\omega}\left(q''_{T,N} + q''_{B,N}\right)$$

$$T_N^{P+1} = F_o\left[\left(1 - \frac{\Delta r}{2r_N}\right) T_{N-1}^P + \left(\frac{\Delta r}{\omega}\right)\left(Bi_T T_{T,\infty}^P + Bi_B T_{B,\infty}^P\right) - \left(\frac{\Delta r^2}{k\omega}\right)\left(q''_{T,N} + q''_{B,N}\right)\right] + \left[1 - \left(1 - \frac{\Delta r}{2r_N}\right)F_o - \left(\frac{\Delta r}{\omega}\right)\left(Bi_T + Bi_B\right)F_o\right] T_N^P$$

NOTE: FOR NO CONVECTION $Bi_T = Bi_B = 0$

AIR ENERGY BALANCE:



(USE FOR TOP AND BOTTOM)

$$\dot{E}_{st} = \dot{E}_{in} - \cancel{\dot{E}_{out}}^{0} + \cancel{\dot{E}_g}^{0}$$

$$\rho_a C_{p,a} \forall_a \frac{\partial T_\infty}{\partial t} = h_1 A_1 (T_1^P - T_\infty^P) + h_2 A_2 (T_2^P - T_\infty^P) + h_3 \sum_{i=3}^{N} A_i (T_i^P - T_\infty^P)$$

$$\frac{k_a}{\alpha_a} (\pi R_N^2 L) \frac{(T_\infty^{P+1} - T_\infty^P)}{\Delta t} = h_1 (\pi R_N^2)(T_1^P - T_\infty^P) + h_2 (2\pi R_N L)(T_2^P - T_\infty^P) + h_3 \sum_{i=1}^{N} 2\pi r_i \Delta r (T_i^P - T_\infty^P)$$

$\frac{L}{k_a R_N^2}$

$$\left(\frac{L^2}{\alpha_a \Delta t}\right)(T_\infty^{P+1} - T_\infty^P) = \left(\frac{h_1 L}{k_a}\right)(T_1^P - T_\infty^P) + \left(\frac{h_2 L}{k_a}\right)\left(\frac{2L}{R_N}\right)(T_2^P - T_\infty^P) + 2\left(\frac{h_3 L}{k_a}\right)\left(\frac{\Delta r}{R_N}\right) \sum_{i=3}^{N} \left(\frac{r_i}{R_N}\right)(T_i^P - T_\infty^P)$$

LET: $F_{o_a} = \frac{\alpha_a \Delta t}{L^2}$; $Bi_1 = \frac{h_1 L}{k_a}$; $Bi_2 = \frac{h_2 L}{k_a}$; $Bi_3 = \frac{h_3 L}{k_a}$

$$T_\infty^{P+1} = F_{o_a}\left[Bi_1 T_1^P + 2Bi_2\left(\frac{L}{R_N}\right)T_2^P + 2Bi_3\left(\frac{\Delta r}{R_N}\right)\sum_{i=3}^{N}\left(\frac{r_i}{R_N}\right)T_i^P\right] + \left\{1 - \left[Bi_1 + 2Bi_2\left(\frac{L}{R_N}\right) + 2Bi_3\left(\frac{\Delta r}{R_N}\right)\sum_{i=3}^{N}\left(\frac{r_i}{R_N}\right)\right]F_o\right\}T_\infty^P$$

NOTE: FOR $N=3$ $A_3 = A_1 = \pi R_N^2$

$$T_\infty^{P+1} = F_{o_a}\left[Bi_1 T_1^P + 2Bi_2\left(\frac{L}{R_N}\right)T_2^P + Bi_3 T_3^P\right] + \left\{1 - \left[Bi_1 + 2Bi_2\left(\frac{L}{R_N}\right) + Bi_3\right]F_o\right\}T_\infty^P$$

19 References

Fundamentals of Chemical Vapor Deposition.TimeDomain CVD, Inc., 8 July 2008. Web. 3 Feb. 2012. <http://www.timedomaincvd.com/CVD_Fundamentals/reactors/reactorTOC.html>

Applied Materials."Applied Materials Cal Poly Senior Project Winter 2012." 4 Jan. 2012. PDF file.