12-2021

Thermometry via Diffusion in Ferrous Core-Shell Nanoparticles for Induction Heating Applications

Hayden Carlton
University of Arkansas, Fayetteville

Thermometry via Diffusion in Ferrous Core-Shell Nanoparticles for Induction Heating Applications

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering

by

Hayden Carlton
Arkansas Tech University
Bachelor of Science in Mechanical Engineering, 2016
University of Arkansas
Master of Science in Mechanical Engineering, 2019

December 2021
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

David Huitink, Ph.D.
Dissertation Director

Rick Couvillion, Ph.D.
Committee Member

Jingyi Chen, Ph.D.
Committee Member

Paul Millett, Ph.D.
Committee Member

Lauren Greenlee, Ph.D.
Committee Member

Robert Coridan, Ph.D.
Committee Member

ABSTRACT

Induction heating causes the release of enormous amounts of heat from dispersed magnetic nanoparticles. While the rate of heat transfer can be easily quantified calorimetrically, measuring the temperature of the nanoparticles on the nanoscale presents experimental challenges. Fully characterizing the temperature and thermal output of these magnetic particles is necessary to gauge overall heating efficiency and to provide a more holistic understanding of heat transfer on the nanoscale. Herein, this dissertation seeks to develop a novel nanoparticle thermometry technique, which correlates diffusion behavior in core-shell nanoparticles to local temperature. Initial measurements suggested that heating silica capped ferrous nanoparticles (SCNPs) via induction in a saline environment encouraged the diffusion of dissolved sodium ions into the silica shell. The concentration gradient of sodium ions within the shell underwent an observable transition after only a few seconds of heating, thus implying that the increased core temperature was the driving force behind the diffusion event. Calculating nanoscale temperature required a three-prong approach, which combined experimental and theoretical analyses. First, a computational model of the core-shell system was developed to accurately depict diffusion into the core-shell structure. Experimental X-ray methods then analyzed the relationship between diffusivity and temperature for the material system and also measured nanoscale concentration gradients within physical SCNPs. By comparing the experimental diffusion data to the theoretical model, the estimated nanoscale temperature was able to be extracted. Understanding nanoscale temperature provides insight into more encompassing thermal models for nanoparticle induction heating, which will ultimately lead to advancements in numerous applications.

ACKNOWLEDGEMENTS

Nobody achieves anything alone. The entirety of my doctoral work would not be possible without the help of numerous other faculty, staff, fellow peers, friends, relatives, and loved ones. Thank you all for keeping me sane and helping me when I needed it. I could not have done it without you.

I would first like to start by thanking my graduate advisor Dr. David Huitink (Dr. H) for guiding me along the way towards achieving my PhD and going above and beyond as a graduate advisor. I appreciate you teaching me how to be successful in the world of research. I enjoyed the semesterly group outings, and how you were genuinely invested in the success of the students working for you. I hope you maintain this leadership style throughout your tenure at UARK. In addition, I also want to thank all the graduate and undergraduate members of the EMPIRE lab that I have worked with over the past 5 years. I appreciate your friendship and your invaluable help.

I would also like to thank my Doctoral committee members: Dr. Jingyi Chen, Dr. Robert Coridan, Dr. Rick Couvillion, Dr. Lauren Greenlee, and Dr. Paul Millett. I have gotten the opportunity to know and learn from each of you throughout my graduate study, and I appreciate your willingness to serve on my committee.

I would like to say a big thank you to all the teachers and instructors that I have learned from during my PhD study. Thank you for providing me the knowledge to complete my dissertation research.

Throughout my PhD study, I have needed to use several characterization instruments ranging from electron microscopy to X-ray characterization. I was extremely blessed to have the opportunity to interact with and learn from the members of the Arkansas Nano-Bio Materials Characterization Facility, led by Dr. Mourad Benamara. I am very grateful for Dr. Benamara's

guidance and friendship, and without his help, this dissertation would not have been possible. I would also like to thank other members of the lab, including Dr. Betty Martin, Dr. Andrian Kuchuk, and Dr. Dongdong Yu. In addition, I would also like to thank Dr. Erik Pollock of the Stable Isotope Lab for his help in gathering ICP-MS measurements.

Many would call synthesizing nanoparticles an art form rather than a science. With my background being in mechanical engineering, I found myself in need of a lot of help with making the nanoparticles for my dissertation. I found a tremendous amount of help and support from Dr. Lauren Greenlee's lab. They helped me gather chemicals, utilize equipment, gave me helpful tips and tricks, and were very welcoming to someone with minimal experience in the complex chemistry involved in creating nanoscale materials. I would like to specifically say thank you to Shelby Foster for the many hours of help and walkthroughs.

Many thanks to the MEEG administrative staff as well! Thank you for helping me purchase items for my research, register me for conferences, and pointing me in the right direction when I needed help. Thank you to Ben Fleming and Jason Bailey for all my machine shop needs. I would also like to give a big thank you to the staff at the ENRC, specifically Kyle Cook, Eric denBoer, and Jeff Knox. You three have helped me more times than I can count over the years, and I greatly appreciate it.

Thank you to Mom, Dad, and Robin for always being there and giving me your continued support. Also, thank you to all my grandparents, aunts, uncles, cousins, etc. for your love and support as well. I love you all.

Finally, I would like to say a huge thank you to my wife, Emma. What a blessed life I live knowing I married my best friend. Getting a PhD is hard, but you make it a million times easier with your love and friendship.

DEDICATION

For PawPaw

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Induction heating represents a novel and efficient way to generate enormous amounts of heat in both macro and nanoscale conductive materials. Specifically, on the nanoscale, much of the research over the past few decades has focused on both characterizing the thermal output of the heated nanoparticles, as well as improving overall heating efficiency; essentially, modifying nanoparticle structural properties to release more heat at lower magnetic fields. However, nanoscale heat transfer, particularly the determination of nanoscale temperature, represents a point of contention in this field of study. A key principle of heat transfer involves the transfer of energy as a direct result of a difference in temperature, in this case, the temperature difference between the heated nanoparticles and the surrounding media. The contention exists in the quantification of this temperature difference, i.e., the temperature of the nanoparticles with respect to their surroundings. The presented research in this dissertation investigates the quantification of nanoparticle temperature through analyzing transient diffusion. As an introduction, this chapter gives a brief analysis of induction heating on the nanoscale, a review of the existing methods of quantifying nanoscale temperature, and an overview of the preliminary research.

1.1. Brief Overview of Nanoparticle Induction Heating

Simply put, induction heating is the generation of thermal energy in conductive materials in response to an alternating magnetic field (AMF). When a macroscale conductive material interacts with an AMF, induced electric currents, called eddy currents heat the material through Joule heating, as a function of the material's resistivity [1]. The efficiency of the heating also varies with respect to the position within the RF coil from which the AMF is generated, with the most efficient heating occurring at the geometric center of the coil. Magnetic materials (ferro/ferrimagnetic) receive an added heating bonus as a direct result of hysteresis heating.

Essentially, the orientation of the magnetic domains within the magnetic material will align with the orientation of the AMF. As the AMF oscillates, the domains will reorganize in response to the changing field direction; this process requires the input of work, which manifests as heat [2]. The combination of both Joule and hysteresis heating makes ferromagnetic materials, such as iron, steel, nickel, and cobalt, ideal candidates for induction heating. In the modern-day, macroscale induction heating is utilized in material processing, such as case hardening [3], as well as induction stovetops for food preparation [4].

Nanoscale induction heating requires the usage of different material types than traditional induction heating processes. Ferromagnetic materials offer the highest thermal output due to their high saturation magnetization; however, creating nanoparticles comprised of pure metals proves to be difficult. Pure ferromagnetic metal nanoparticles (iron, nickel, cobalt) lack stability on the nanoscale, are prone to oxidation, and exhibit low biocompatibility [5]. In terms of biomedical applications, pure metal nanoparticles were less than ideal candidates. Most efforts in recent days utilize ferrimagnetic metal oxide nanoparticles, which still possess relatively high magnetic saturation but maintain stability on the nanoscale [6, 7].

As the heated material reduces in size to the nanometer size scale, the method of induction heating via an AMF remains the same, but the physics of heat generation changes. An encompassing review by Dennis *et al.* describes in detail the mechanisms which drive nanoparticle induction heating, as well as some of the challenges faced by researchers [8]. To start, reducing the size of a magnetic material (from here on called a "particle") intrinsically reduces the number of magnetic domains. Once the size of the particle crosses a certain threshold (depending on the material type) the energy cost to maintain multiple domains becomes too great, and the particle transitions to a single domain structure with a uniform magnetic moment, rather than a multi-

domain structure [9]. The magnetic moment in single-domain particles tends to orient parallel or anti-parallel to an axis known colloquially as the “easy axis”, thus giving the particle magnetic anisotropy. In this case, the anisotropic energy is the energy required to reverse the magnetic moment along the easy axis (rotating 180°), which also implies the existence of a minimum magnetic field necessary for moment reversal [8]. Several properties impact particle anisotropy including particle crystal structure, shape, and surface [10, 11]. In terms of nanoscale induction heating, the anisotropic energy can greatly influence the effect of hysteresis heating. In macroscale induction heating, the reorientation of the magnetic moments in response to the AMF drives the thermal output; however, in single domain structures, overcoming the anisotropic energy barrier determines hysteresis losses (release of heat), such as in the study by Noh *et al.*, where the size of the hysteresis curves was changed by modifying the geometry of various spherical and cubic nanoparticles (Figure 1) [12].

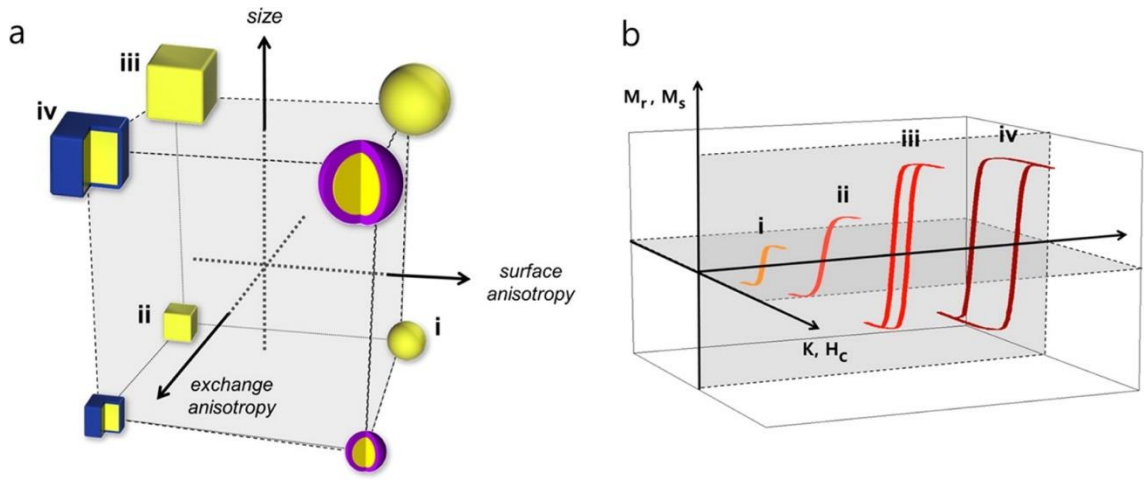
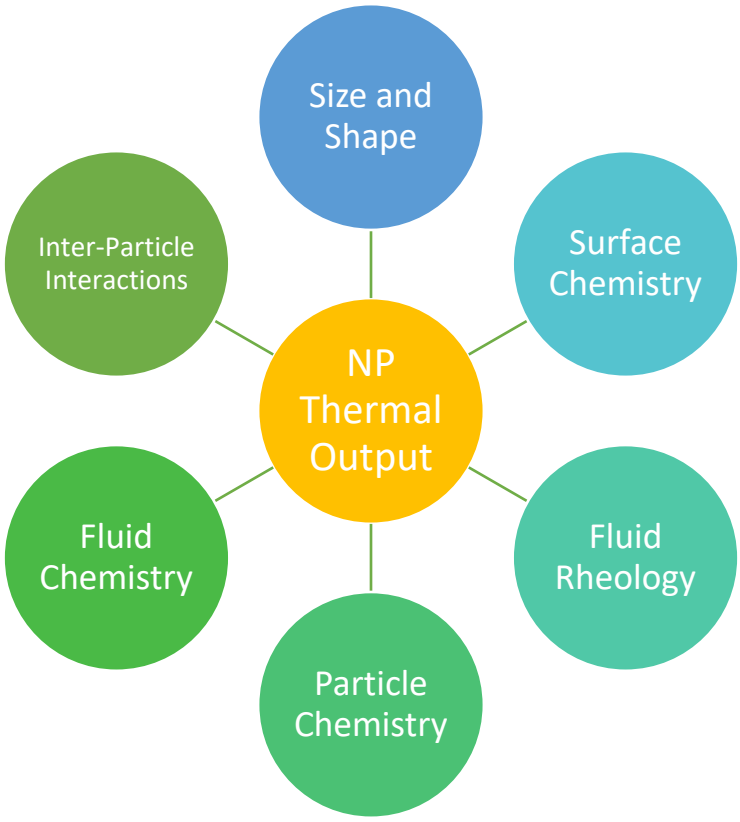


Figure 1: (a) The researchers tested numerous types of particle geometries in two different size regimes including sphere, cube, and core/shell structure; (b) By changing particle geometry, the hysteresis loop is also modified. In this case, the largest cube size with the core-shell structure resulted in the largest hysteresis loop area, thus providing the highest thermal output. (Reprinted with permission from [12]. Copyright 2012 American Chemical Society).

At a sufficiently small size, single-domain particles can become superparamagnetic, where the thermal energy of a system can overcome the anisotropic energy barrier and cause the magnetic moment to fluctuate. Superparamagnetic particles lack coercivity or remanence (inability to remain magnetized after an external field is removed) like paramagnets; however, they possess the high saturation magnetization, characteristic of a ferri/ferromagnetic material, which gives them the name superparamagnetic [13]. Because of their lack of remanent magnetization, losses due to hysteresis are minimized. The heat produced by hysteresis depends highly on the area within the hysteresis loop (Figure 1b), and for a particle system with no remanence, the hysteresis loop area is zero. Particles in this size regime experience losses due to thermal relaxation, whereas the two main types are Brownian and Néel. Brownian relaxation occurs when the moment rotates the entire particle within the fluid suspension [14]; a recent effort made attempts to quantify the Brownian contribution to the total thermal output within various nanoparticle suspensions [15]. Néel relaxation corresponds to the internal moment rotation within the crystal structure [16, 17]. Mechanisms such as thermal relaxation provide insight into the driving force of thermal output in particles within the superparamagnetic regime.

Several aspects of nanoparticle colloids have significant influence over thermal output, including their geometry, chemical composition, capping agents, and the properties of the fluid medium; a notable review highlighting the specific influence of nanoparticle structure can be found by Abenojar *et al.* [18]. Prior to this dissertation, some early work by the candidate focused on the effects of the surrounding environment on particle heating, as well as inter-particle interactions. Particles capped with a nonionic surfactant were found to be heavily influenced by the presence of dissolved ions within an aqueous solution, inducing heavy aggregation and altering heating performance [19]. *In situ* characterization using neutron scattering techniques allowed for the inter-

particle interactions during heating to be observed by quantifying nanoparticle cluster size [20]. Additional work was also performed to correlate nanoparticle thermal output with mean inter-particle distance for monodisperse iron oxide nanoparticles [21]. While an in-depth review of how each parameter affects heat transfer is outside the scope, an illustration of these properties with their associated references can be seen in Figure 2.



Property	Reference(s)
Size and Shape	[22-24]
Surface Chemistry	[25-27]
Fluid Rheology	[15, 28, 29]
Particle Chemistry	[30, 31]
Fluid Chemistry	[19, 32]
Inter-Particle Interactions	[21, 33-35]

Figure 2: Diagram highlighting nanoparticle suspension properties that influence thermal output with associated references.

One of the most researched applications for nanoparticle induction heating continues to be magnetic nanoparticle hyperthermia (MNH). Hyperthermia involves the treatment of cancer cells through increased local temperature; thermal shock induces cancer cell death (apoptosis), due to the cancer cells being more susceptible to changes in temperature than healthy cells. The usage of magnetic nanoparticle induction heating as the heat source for hyperthermia treatments provides a more targeted approach. Ideally, by injecting nanoparticles directly into tumor sites, the small size of the nanoparticles allows them to pass through biological membranes to heat the cancer cells directly using an AMF and limit damage to healthy tissue [36, 37]. Research into MNH as an alternative cancer treatment has also revealed its success when used in conjunction with other treatments. A recent study highlighted the potential of using MNH with a combination of radiotherapy and immunotherapy for increased effectiveness [38]. The same group also observed the triggering of immune responses within the body after nanoparticle exposure [39]. Obviously, MNH represents a novel alternative cancer treatment method, but additional research is necessary to understand the full breadth of nanoparticle interactions within the human body.

As indicated, the optimization of nanoparticle induction heating continues to be a multi-faceted issue. Everything from the particle structure to the properties of the suspended fluid impacts the magnitude of the thermal energy produced. While several references in literature thoroughly quantify the impacts of each of these aspects, much is still unknown regarding this novel heat transfer phenomenon. A full understanding of nanoparticle induction heating requires knowledge of the nanoscale thermal transport occurring between the nanoparticle and surrounding fluid during induction heating, which is a newly developing area.

1.2. Current Methods for Estimating Nanoparticle Temperature

A fundamental difference in temperature is the driving force behind all forms of heat transfer. Calorimetric data indicates that heated nanoparticles release enormous amounts of heat with respect to the amount of material present. While the energy released is easily quantified by measuring the temperature of the fluid over time, the temperature difference between the particle and the fluid is largely unknown and represents a newly evolving area of study in nanoparticle science. Some researchers claim the temperature difference between the nanoparticle and surrounding fluid is negligible; however, novel experimental efforts within the past decade have been successful in measuring the temperature in the immediate vicinity. In addition to thermometry with magnetic nanoparticle heating, efforts in gold nanoparticle thermometry during plasmonic heating will also be addressed.

In terms of theoretical analyses, the existence of a nanoscale temperature difference is met with some skepticism. Specific to magnetic nanoparticle induction heating, a recent review provided an insightful introduction into the work of various researchers and their attempts to define nanoscale temperature and heat transfer [40]; the paper describes both sides of the argument: those who think the temperature difference is significant and those who do not. An early work analyzed the problem and concluded that the temperature difference on the nanoscale between the nanoparticle and surrounding fluid (water in this case) would not exceed 10^{-5}°C at steady state [41]. A few years later, a similar work corroborated these results and added that a nanoscale temperature difference between the surrounding fluid and nanoparticle surface would only be possible for extremely dilute suspensions at an exorbitantly high heating power [42]. Outside of the dissenting opinion, molecular dynamics simulations of heated gold nanoparticles indicate the presence of a nanoscale thermal gradient during heating. One study revealed that the fluid in the

immediate area around the nanoparticles can experience large temperature gradients and heat fluxes, which can cause particle degradation [43]. A more recent simulation effort specifically observed the heat conduction at the particle/fluid interface and the associated thermal gradient; their results indicated the dominant mechanism of heat transfer, within the tested timescale, was conduction rather than convection [44]. At this solid/liquid boundary, other researchers have suggested the existence of nanobubbles created by the vaporization of water at this interface during heating [45-48]. With magnetic nanoparticles, simulation evidence indicates that inter-particle interactions influence nanoscale heat generation as well [49]. As can be seen, increased computational power over the past decade has allowed for these nanoscale interactions to be modeled effectively, which results in improved quantification of nanoscale thermal phenomena.

Recent experimental works have also demonstrated novel methods of measuring temperature in the immediate vicinity of the nanoparticles. As indicated in the article by Piñol *et al.*, two distinct groups arise when discussing nanoscale temperature measurement with magnetic nanoparticles, dual-particle and single-particle methods, however, some examples in the literature indicate a hybrid between the two, as shown in Figure 3 [50]. Dual-particle heating modes arose first through the usage of quantum dots to act as nanoscale thermometers, where the temperature-sensitive fluorescent emissions of the quantum dots provided insight into their temperature [51]. Essentially, the quantum dots were mixed into the nanoparticle suspension, and during heating, their emissions were monitored and correlated with temperature. However, the authors did not observe a difference between particle temperature and the temperature of the bulk fluid. Similarly, quantum dots were also used to measure the efficiency of gold nanorod heating [52]. In addition to quantum dots, nanoscale diamond particles have been utilized as nanoscale thermometers with high accuracy [53]. One group attached gold structures to an $Al_{0.94}Ga_{0.06}N$ thin film with

incorporated Er^{+3} atoms, which acted as a temperature sensor; this sensor measured the thermal gradient around a single nanoparticle during heating [54]. The same group later proposed a different method using Er_2O_3 nanoparticles rather than the thin film; the nanoparticles measured the temperature in the vicinity of several gold nanorods during optical heating [55].

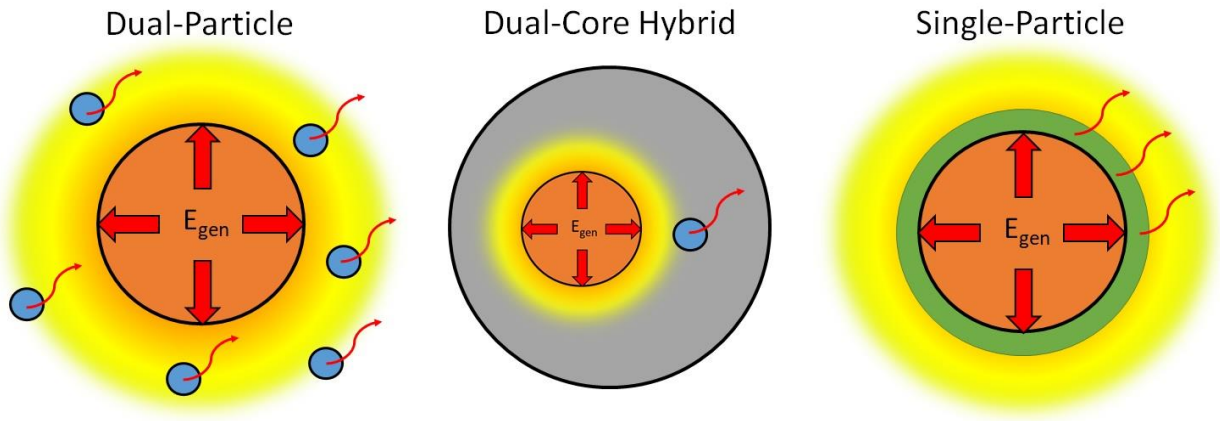


Figure 3: (left) Dual-particle approaches use a system of separate nanoparticles to act as thermometers within the direct vicinity of the heated nanoparticle; (middle) Hybrid dual-core nanoparticle where thermometer and heated particle are contained within the same shell; (right) Single-particle methods implement a thermometer coating which provides a more direct way to measure temperature

While dual-particle methods provide the means for researchers to quantify nanoparticle temperature, they are primarily limited by their proximity to the heated particle, i.e., are the temperature-sensing particles close enough to the heated particles to ensure an accurate temperature measurement? A hybrid dual-particle method attempted to increase accuracy by encapsulating the heated nanoparticle with the temperature-sensing nanoparticle in a single silica shell (dual-core nanoparticle) in order to ensure they were in close proximity to one another [56]. The heating of the dual-core particle provided evidence of a nanoscale thermal gradient that increased with respect to magnetic field and exposure time. The issue of proximity to the heated

particle led to the development of the single-particle measurements, where the coating of the heated nanoparticles or the nanoparticle itself acts as the nanoscale thermometer. Using the coating rather than a separate particle ensured that the location of temperature measurement occurred at the surface of the particle rather than in the adjacent space. One of the first examples of this methodology used particles capped with a temperature-sensitive fluorescent dye; a second dye was also dispersed in the surrounding fluid [57]. The comparison between the two dye locations showed that a 5°C temperature difference between the surface of the nanoparticles and the surrounding fluid. Rather than two dyes, a similar study synthesized core-shell particles with two emission peaks, one from the core and one from the shell [58]. Another study utilized Eu and Tb complexes as nanoparticle coatings and observed a notable difference between the surface of the particle and the surrounding fluid [50]. A few other thermo-sensitive nanoparticle coatings have been used in literature to estimate the temperature of both magnetic and gold nanoparticles [59, 60]. Apart from using a temperature-sensitive coating or dye, unique single-particle methods have utilized anti-Stokes emissions [61], as well as local ionic conductivity [62] to determine local temperature. A particularly interesting study attempted to quantify nanoparticle temperature as a function of distance from the core [63]. In this case, polyethylene glycol (PEG) served as a spacer between the heated particle and the thermo-sensitive dye; by using PEG molecules with varying molecular weight, the dye could be placed at a set distance away from the core during heating, allowing for temperature measurements with improved spatial resolution. All the single-particle methods verified the presence of nanoscale temperature gradients between the nanoparticle and the surrounding fluid, but a possible concern would be the resistive effects of thermo-sensitive coatings. To frame it as a question, does adding additional temperature-sensing molecules to the surface of the heated nanoparticles impact the convective heat transfer between the nanoparticles

and the fluid? Further, would the same temperature gradient be present in the absence of the organic molecules introduced to measure nanoscale temperature?

1.3. Thermometry via Diffusion: Preliminary Research

In a previous study by the Huitink and Greenlee labs, the interactions between silica capped iron nanoparticles (SCNPs) and the surrounding saline suspension during induction heating were analyzed in terms of how the dissolved ions affected nanoparticle structure over time. Silica capping provided the particle with a chemically inert, biocompatible shell; however, elemental analysis of the silica shell after induction heating indicated the diffusion of ions into the core-shell structure (Figure 4). While the project sought to analyze how dissolved ions may degrade the silica shell after prolonged heating, the time-dependent ionic diffusion into the silica shell piqued additional interest regarding the temperature-dependency of the diffusion events. Diffusivity for a material system follows an Arrhenius relationship ($D = D_0 e^{\frac{-E}{RT}}$), where "$D_0$" is the diffusivity at infinite temperature, and "E" is the activation energy for a diffusion event. With knowledge of the diffusivity vs. temperature function, temperature could theoretically be extrapolated by measuring the relative concentration of Na with respect to particle radius before and after heating. Measuring temperature in this way would differ from previous single-particle thermometry techniques because additional capping agents would not be necessary. The temperature would be calculated by observing how a single nanoparticle interacted with the dissolved ions in suspension without changing its structure or composition. The conjecture of diffusion-based nanoscale temperature calculation serves as the underlying hypothesis of this work.

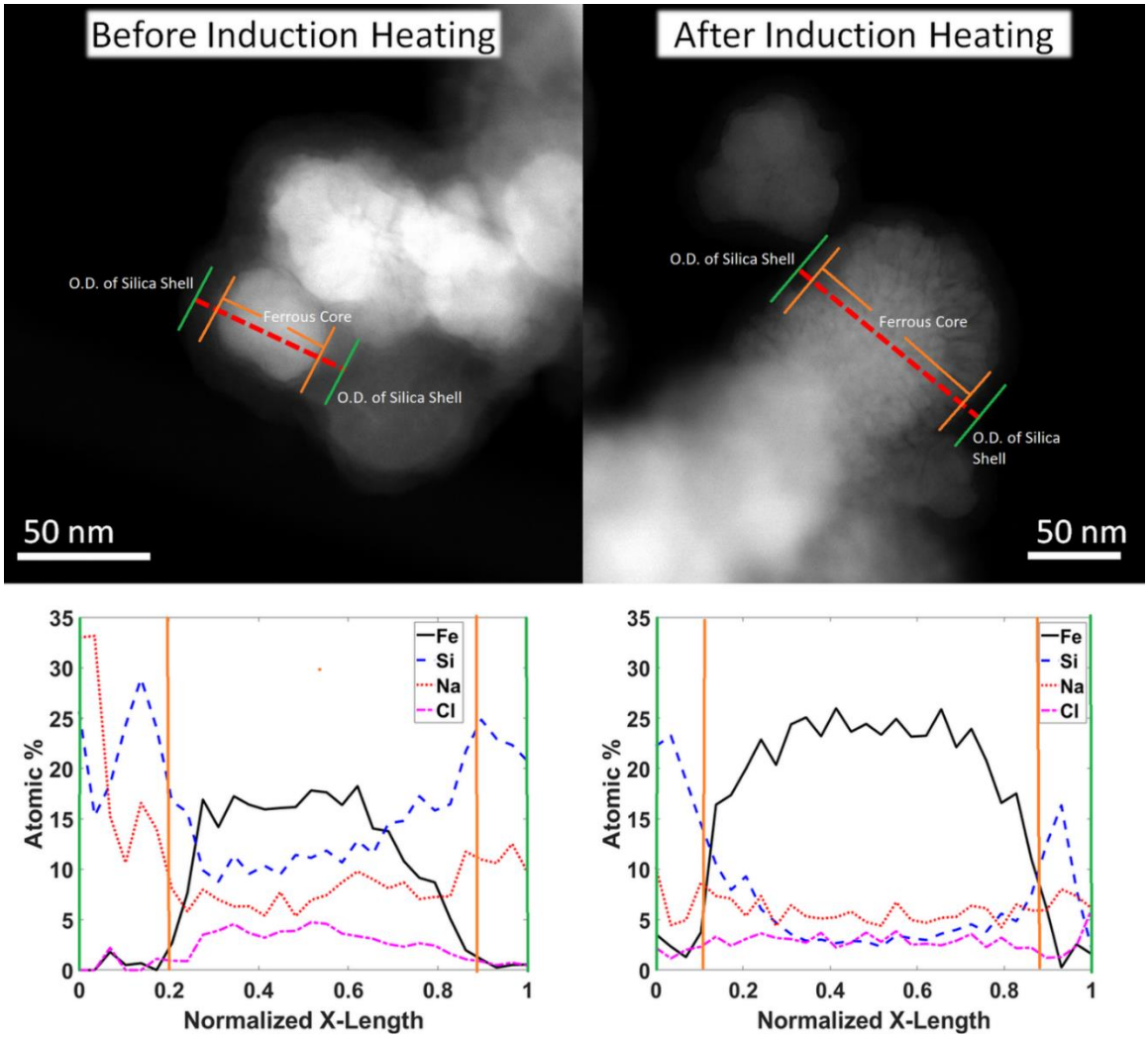


Figure 4: (left column) Nanoparticle EDX line scan before induction heating. A high concentration of sodium ions (orange dotted line) is observed towards the surface of the nanoparticle with decreasing concentration towards the iron core; (right column) After 30 minutes of induction heating, the concentration of sodium appears much more linear with respect to the "Before Induction Heating" measurement, which implies the occurrence of diffusion phenomena within the nanoparticle.

1.4. Rationale

This dissertation attempts to demonstrate an alternative methodology for estimating local nanoparticle temperature during induction heating using core/shell nanoparticles comprised of iron and silica. By correlating the temperature-dependent diffusion of sodium ions into the silica shell

with simulation values, the temperature of the area immediately around the ferrous cores can be estimated. An understanding of the nanoscale temperature can provide information regarding nanoscale convection between the particle and surrounding fluid. Quantifying convection between the particle and fluid relates to how efficient heat is transferred across the particle/fluid boundary, which proves useful to several applications. In order to accomplish this, several aspects are addressed:

- A numerical model is presented in order to simulate nanoscale diffusion within core-shell geometries. This simulation provides the backbone for estimating temperature by correlating simulated and experimental concentration gradients. (Chapter 2)
- Since the temperature-dependent properties of sodium diffusing into sol-gel silica are unknown, they are be determined empirically. X-Ray Photoelectron Spectroscopy Depth Profiling is utilized to measure the concentration profile of sodium in silica at a variety of process times and temperatures to determine the dependence of diffusivity on temperature. (Chapter 3)
- Using the diffusivity data and a working numerical model, nanoparticle temperature is estimated using Energy Dispersive X-ray (EDX) data from a High-Resolution Transmission Electron Microscope (HRTEM). After heating the particles via induction in a saline solution, the distribution of sodium within the silica shell is analyzed, compared to the numerical model, and used to estimate the core temperature. (Chapter 4)

Through the collective analysis of these various research approaches, parallels can be drawn between thermal and mass diffusivity in core-shell nanoparticles, which ultimately provides the context for the introduction and facilitation of estimating local nanoparticle temperature during induction heating.

2. NUMERICAL MODELLING OF CORE-SHELL GEOMETRY

While experimental methodologies greatly enhance the understanding of how a system performs in the real world, the most encompassing studies need to illustrate similitude between both empirical data and numerical models. For this geometry (spherical core-shell), analytical solutions are often quite complex and only account for a few ideal cases. Because of this difficulty associated with analytical solutions in non-ideal geometries, researchers widely utilize numerical, specifically finite element, methods to calculate a variety of spatially and/or temporally variant systems. Since this work largely depends on a detailed understanding of ion movement within spherical nanoparticles to extrapolate local temperature, utilizing numerical methods is necessary to equate the theoretical understanding of diffusion with the above experimental observations. Within this chapter, the development of a numerical diffusion model to analyze the temperature-dependent mass transfer of sodium ions within core-shell nanoparticles is discussed, which will serve as the first step towards nanoparticle temperature estimation.

2.1. Solving Partial Differential Equations with Core-Shell Geometry

As with any model, the first step is to create a diagram of the system in question: a spherical core-shell nanoparticle (Figure 5). The diagram illustrates not only where the system is spatially located, but also the locations of the respective boundary conditions. In this case, the system is comprised of a spherical iron core surrounded by a silica shell; the inside radius of the shell is denoted as R_i while the outside radius is R_o. For simplicity, a 1-D radial assumption is used for the simulations, or in this case, a quarter-circle arc with an inside and outside radius.

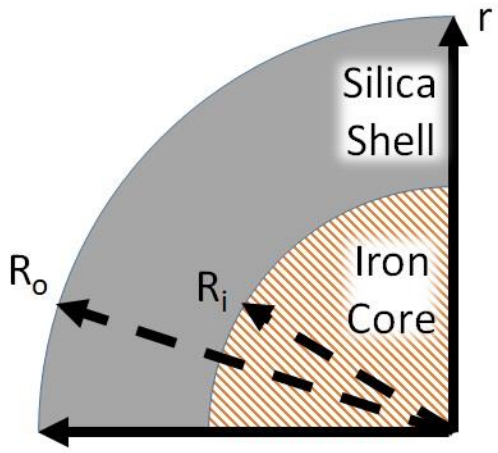


Figure 5: Illustrative model of the core-shell nanoparticle structure with an iron core and silica outer shell. This serves as the primary illustration for the 1-D radial model.

When diffusion is concerned, the standard model used for the transient state is Fick's second law, which relates how concentration (c) at a point in space, in this case, radius (r), changes with respect to time (t). In its general form, Fick's second law is shown in Equation 1.

$$\frac{\partial c}{\partial t} = -D\nabla^2 c \qquad (1)$$

The Laplacian term ($\nabla^2 c$) provides a general representation for a variety of geometries; among the more common are rectilinear, cylindrical, and spherical geometries. Luckily, the system of interest (spherical) can be represented easily, as seen in Equation 2, where θ and φ represent the concentration gradient in the angular dimensions.

$$\nabla^2 c = \left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c}{\partial r}\right)\right]_r + \left[\frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta\frac{\partial c}{\partial \theta}\right)\right]_\theta + \left[\frac{1}{r^2 \sin^2\theta}\frac{\partial^2 c}{\partial \varphi^2}\right]_\varphi \qquad (2)$$

While the above equation may have multiple components, some assumptions must be taken into consideration. It is assumed that the ionic concentration on the exterior surface of the shell (R_o) is constant, which implies that the ions should diffuse uniformly in the radial dimension (r). Ideally, this also means that the concentration should not vary in the angular dimensions (θ and φ), such that the angular components of Equation 2 would be zero throughout the simulation. Real nanoparticles often detract from the assumption of a perfect sphere; however, the nanoparticle geometry in the preliminary data tended towards spherical. This assumption greatly simplifies the simulation by only considering the radial term. Removing the unnecessary terms, Fick's second law becomes Equation 3.

$$\frac{\partial c}{\partial t} = -D\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c}{\partial r}\right) \qquad \textbf{(3)}$$

Still, the PDE can be simplified even further. If a simple substitution is made, *u = cr (concentration x radius)*, the equation mimics that of a rectilinear geometry [64], shown in Equations 4-7. By making the simple substitution, the potentially difficult spherical PDE becomes a rather simple 1-D rectilinear PDE (Equation 7).

$$\frac{1}{r}\frac{\partial u}{\partial t} = -D\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{r\frac{\partial u}{\partial r}-u}{r^2}\right) \qquad \textbf{(4)}$$

$$\frac{\partial u}{\partial t} = -D\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}-u\right) \qquad \textbf{(5)}$$

$$\frac{\partial u}{\partial t} = -D\frac{1}{r}\left(r\frac{\partial^2 u}{\partial r^2}+\frac{\partial u}{\partial r}-\frac{\partial u}{\partial r}\right) \qquad \textbf{(6)}$$

$$\frac{\partial u}{\partial t} = -D\frac{\partial^2 u}{\partial r^2} \qquad \textbf{(7)}$$

2.2. Numerical Formulation of Interior Nodes

With a general analytic model established, the next step was to discretize the system model to develop a numerical algorithm. With both spatial and temporal variance, the numerical solver can be visualized in Figure 6, where u_i^j, refers to the solution to Equation 7 at radius index *i* ($0 \leq i \leq R$) and time index *j* ($0 \leq j \leq T$). In this case, *R* refers to the length of the radius vector *r* and *T* refers to the length of the time vector *t*. The vector *r* is comprised of radius values separated by Δr within the range $R_i \leq r \leq R_o$ (within the silica shell); similarly, the vector *t* consists of time values separated by Δt within the range $0 \leq t \leq \max(t)$.

For this type of PDE (parabolic), which primarily models heat transfer or diffusion phenomena, the Crank-Nicolson (CN) algorithm is widely implemented, due to its unconditional stability [65]. In this case, Equation 7 can be discretized to the general CN form, as shown in Equation 8.

$$\frac{u_i^{j+1}-u_i^j}{\Delta t} = \frac{D}{2}\left[\frac{u_{i-1}^{j+1}-2u_i^{j+1}+u_{i+1}^{j+1}}{\Delta r^2} + \frac{u_{i-1}^{j}-2u_i^{j}+u_{i+1}^{j}}{\Delta r^2}\right] \quad \textbf{(8)}$$

The CN algorithm represents an implicit numerical algorithm. i.e., the values for the current time step (u^{j+1}) must be solved through analyzing a system of equations using the previous time step (u^j). Because of this, a common and efficient method for solving the CN algorithm is through matrix methods, where all the interior radius nodes for the current time step are solved simultaneously. After rearranging the terms in Equation 8 and compiling all the constants into a single term ($C = \frac{D\Delta t}{\Delta r^2}$), the setup for the matrix solver can be seen in Equations 9 and 10.

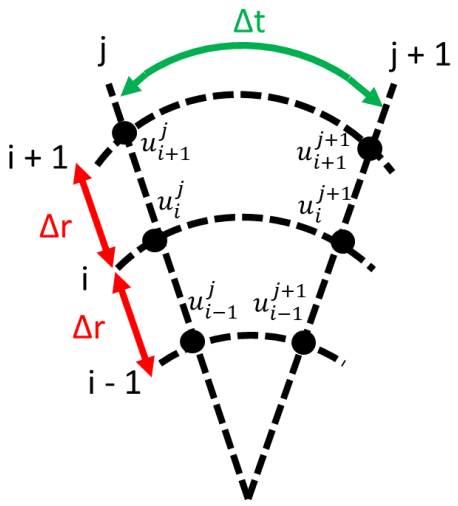


Figure 6: Visualization of the system discretization in both time and space dimensions. In this case, the radius is divided into sections of Δr and the time is divided into sections of Δt.

$$\frac{-C}{2}u_{i-1}^{j+1} + (1+C)u_{i}^{j+1} - \frac{C}{2}u_{i+1}^{j+1} = \frac{C}{2}u_{i-1}^{j} + (1-C)u_{i}^{j} + \frac{C}{2}u_{i+1}^{j} \qquad (9)$$

$$-u_{i-1}^{j+1} + 2\left(\frac{1}{C}+1\right)u_{i}^{j+1} - u_{i+1}^{j+1} = u_{i-1}^{j} + 2\left(\frac{1}{C}-1\right)u_{i}^{j} + u_{i+1}^{j} \qquad (10)$$

In the case of Equation 10, if this same format is applied at each radius step, the following matrix equation can be used to solve the systems of equations $A \times u = d$ (Equation 11), where $B = 2\left(\frac{1}{C}+1\right)$, and $d_i = u_{i-1}^{j} + 2\left(\frac{1}{C}-1\right)u_{i}^{j} + u_{i+1}^{j}$, or the right half of Equation 10.

$$\begin{bmatrix} B & -1 & 0 & \cdots & 0 & 0 \\ -1 & B & -1 & \ddots & \vdots & \vdots \\ 0 & -1 & B & -1 & 0 & \vdots \\ \vdots & 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & 0 & \cdots & 0 & -1 & B \end{bmatrix}_{A} * \begin{bmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ \vdots \\ \vdots \\ u_{R-1}^{j+1} \end{bmatrix}_{u} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ \vdots \\ d_{R-1} \end{bmatrix}_{d} \qquad (11)$$

For every time step after the initial condition (0 < t), the inverse of the *A* (R-2 x R-2) matrix is calculated and multiplied with the *d* (R-2 x 1) matrix for the current step (j+1). The resulting (R-2 x 1) solution matrix contains the "u" values for the interior radius nodes ($1 \leq i \leq R-1$) at the j+1 time step.

2.3. Boundary Conditions

Among the most important part of any numerical simulation is the selection of the proper boundary conditions. For the current model, three boundary conditions are necessary: 1 initial time condition and 2 spatial conditions.

2.3.1. Initial Condition

The easiest condition to implement into the algorithm is the initial condition. Presumably, the concentration of sodium within the silica layer is zero before diffusion occurs; however, this assumption will need to be verified with experimental data. In terms of the CN model, a zero-concentration initial condition can be represented by $c_i^0 = 0 \; or \; u_i^0 = 0$.

2.3.2. Dirichlet B.C.

Next, the spatial boundary conditions were determined. As stated earlier in this section, an easy starting assumption for the exterior of the silica shell is that the concentration of sodium atoms remains constant throughout induction heating, which would mean, on a normalized scale, the concentration would simply be 1 at r = R_o for the entirety of the simulation, or a Dirichlet boundary condition: $c_R^j = 1 \; or \; u_R^j = R_o$. This condition is trivial to implement into the numerical algorithm, as shown in Equation 12.

$$-\boldsymbol{u_{R-2}^{j+1}} + \mathbf{2}\left(\frac{\mathbf{1}}{\boldsymbol{C}} + \mathbf{1}\right)\boldsymbol{u_{R-1}^{j+1}} = \boldsymbol{u_{R-2}^{j}} + \mathbf{2}\left(\frac{\mathbf{1}}{\boldsymbol{C}} - \mathbf{1}\right)\boldsymbol{u_{R-1}^{j}} + \boldsymbol{u_R^j} + \boldsymbol{u_R^{j+1}} \qquad \textbf{(12)}$$

With all the values at u_R set to a constant value (R_o), the term u_R^{j+1} can simply be added to the right side of the equation, which contains the known values from the previous step. By doing this the last term in the *d* matrix changes: $d_{R-1} = u_{R-2}^{j} + 2\left(\frac{1}{C} - 1\right)u_{R-1}^{j} + 2 * R_o$.

2.3.3. Neumann B.C.

The next spatial condition must pertain to the other side of the silica shell (R_i), the barrier between the silica shell and the ferrous core. The boundary condition at this location was chosen to be a Neumann boundary condition or insulated surface. i.e., no material diffuses through the silica/iron interface. An insulated boundary condition was chosen namely for simplicity. Adding an interface between two unlike materials would have greatly increased the complexity of the model. Also, due to the pure interest in transient diffusion, the investigated diffusion length will be designed to not exceed the thickness of the shell. Mathematically, this boundary condition is represented $\frac{dc_0^{j+1}}{dr} = 0$, where there is no concentration gradient across the inside boundary. In terms of the numerical model, this slope needs to be determined through a second-order forward finite difference equation originating at the first concentration value (c_0) (Equation 13).

$$\frac{dc_0^{j+1}}{dr} \approx \frac{-3c_0^{j+1} + 4c_1^{j+1} - c_2^{j+1}}{2\Delta r} = 0 \qquad (13)$$

If we solve this equation in terms of the concentration value at the boundary (c_0^{j+1}), then the boundary condition is defined by the equations below, both in terms of concentration (Equation 14) and the *u* term (Equation 15).

$$c_0^{j+1} = \frac{4}{3}c_1^{j+1} - \frac{1}{3}c_2^{j+1} \qquad (14)$$

$$\boldsymbol{u_0^{j+1} = \frac{4r_0}{3r_1} u_1^{j+1} - \frac{r_0}{3r_2} u_2^{j+1}} \qquad \textbf{(15)}$$

Unlike the Dirichlet condition, the Neumann condition is a bit more difficult to implement into the algorithm. The difficulty lies in the CN equation at R_i (u_0) shown in Equation 16.

$$\boldsymbol{-u_0^{j+1} + 2\left(\frac{1}{C} + 1\right) u_1^{j+1} - u_2^{j+1} = u_0^j + 2\left(\frac{1}{C} - 1\right) u_1^j + u_2^j} \qquad \textbf{(16)}$$

In order to calculate this equation through the CN matrix method, the term u_0^{j+1} must be known. This seems counterintuitive since u_0^{j+1} depends on the two neighboring terms in the current time step, which have yet to be calculated. However, the equation simplifies by substituting in Equation 15 to 16 (Equation 17).

$$\boldsymbol{\left[2\left(\frac{1}{C} + 1\right) - \frac{4r_0}{3r_1}\right] u_1^{j+1} + \left(\frac{r_0}{3r_2} - 1\right) u_2^{j+1} = u_0^j + 2\left(\frac{1}{C} - 1\right) u_1^j + u_2^j} \qquad \textbf{(17)}$$

With respect to matrix *A*, this modification changes the top left corner respectively: $\mathbf{A(1,1) = 2\left(\frac{1}{C} + 1\right) - \frac{4r_0}{3r_1}}$ and $\mathbf{A(1,2) = \frac{r_0}{3r_2} - 1.}$ The modified *A* matrix allows for all the interior nodes to be calculated, while still keeping the Neumann condition into consideration. After calculating the interior nodes for the j+1 step, the value of u_0^{j+1} can be calculated using Equation 15.

2.4. From Math to MATLAB

Several coding languages could effectively model this system with similar degrees of accuracy; however, due to familiarity with the language, MATLAB was chosen as the primary

IDE to implement the core-shell diffusion model. While MATLAB excels at matrix-centric programming, the primary difficulty was that the indices for MATLAB start at "1" rather than "0". From this point forward, u_i^j represents the simplified solution value at radius index "i" and time index "j", where i, j ≥ 1. Other than this change, the implementation of this numerical system into MATLAB code was straightforward.

2.4.1. CN Algorithm: MATLAB Pseudocode

The pseudocode for the CN Implicit solver can be seen below, and the full script file can be found in the Appendix:

```
>> D = 1          % Define diffusivity in nm^2/s
>> t = 0:delt:totalTime          %Define time discretization
>> r = ri:delr:ro          %Define radius discretization
>> c = zeros(length(r), length(t))          %Create solution matrix
>> u = zeros(length(r), length(t))          %Create simplified "u" matrix (u=cr)
>> u(: ,1) = 0          %Initial condition of zero concentration at t = 1
>> u(length(r), :) = ro                    %Dirichlet B.C. at shell boundary
>> C = D*delt/(delr)^2          %Calculate the non-dimensional constant
>> A=tridiagonal matrix          %Create the tridiagonal matrix "A"
>>A(1,1) = 2*(1/C+1)-(4*r(1))/(3*r(2)); A(1,2) = r(1)/(3*r(3))-1; %Account for Neumann B.C
>> d = zeros(length(R)-2,1)          %Create matrix "d" containing previous step solutions
>> for j =2:length(t)          %Iterate over all time steps
        >> for i=length(r)-2:-1:1          %Iterate over interior nodes and build "d" vector
                >> if i == length(R)-2          %Accounting for Dirichlet B.C.
                        >> d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+2*u(i+2,j-1;
```

```
>> else          %All other interior points
        >> d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+u(i+2,j-1);
>> end
>> end
>> u(2:R-1,j) = A\d;  %Multiply inverse of A matrix by newly formed d vector
>> u(1,j) = (4*r(1))/(3*r(2))*u(2,j)-r(1)/(3*r(3))*u(3,j); %Account for Neumann B.C.
>> end
>> c(:,:) = u./repmat(r',1,T); %Divide simplified u matrix by radius vector and store as conc.
```

While the pseudocode goes into more depth, the base algorithm is simple:

1. Determine initial condition and matrix “A”.
2. Create “d” vector (Equation 11) by using values at the previous time step (j).
3. Multiply the inverse of tridiagonal “A” matrix by new “d” vector to determine solution at current time step (j+1).
4. Repeat 2 and 3 for each time step.
5. Divide each column of the final solution matrix by radius vector to return to concentration

2.4.2. Plotting Functions

The algorithm comprised the bulk of the mathematical calculations, yet additional functions would be needed to properly plot and visualize the results. MATLAB’s built-in “plot” function provided a simple 1-D representation of the data; however, for a 2-D view of the diffusion within the shell, the popular user function “3D Polar Plot” [66] was utilized to create still images as well as progression videos. An example of both the “plot” and “3D Polar Plot” graphs can be seen in Figure 7.

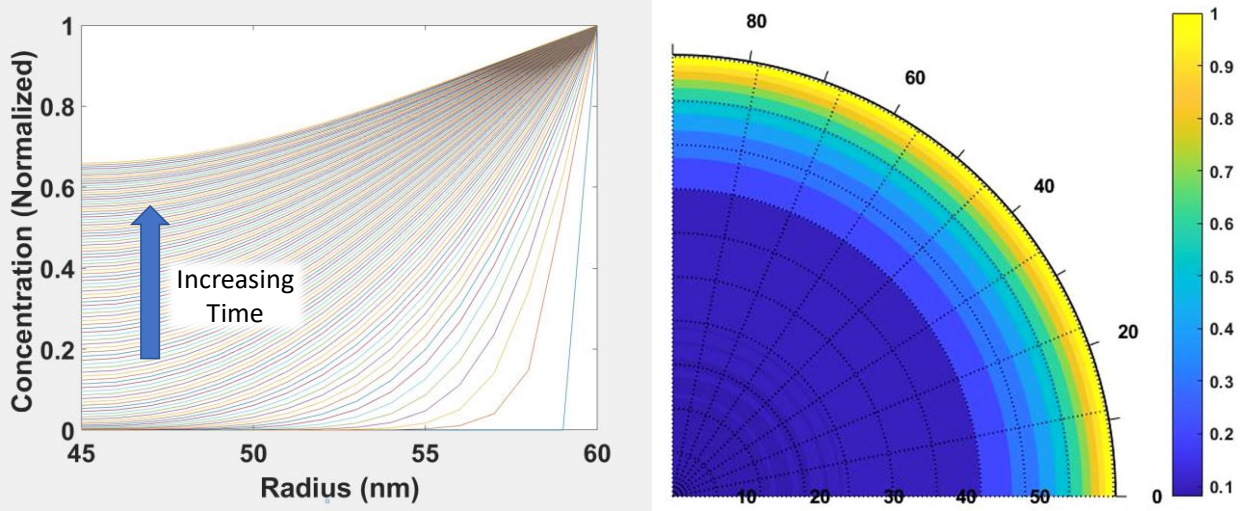


Figure 7: Example plots using two separate plotting functions: plot and 3D Polar Plot. In this case, the plots illustrate the concentration gradient across a silica shell, with an inner radius of 45 nm and an outer radius of 60 nm. (left) The standard "plot" function shows a simple 1-D gradient at progressing time steps (each color represents a new time step). (right) The "3D Polar Plot" function shows the same data, but only at a single time step. However, the data is more visually appealing and illustrates the core-shell geometry better.

2.5. Error Considerations

With the CN already proven to be unconditionally stable, this work will not delve further into its stability; however, due to the nature of numerical analysis, the presence of error is guaranteed. A main source of error in numerical calculations originates through discretization of a continuous function; in this case, the function of concentration with respect to both time and radius. Greater accuracy is achieved through finer discretization (smaller Δr and Δt); however, at the expense of greater computational time. In terms of this simulation, the dimensionless C parameter represents the primary term that combines both time and space discretization. To determine how this term impacts error propagation within the simulation, the value of C was increased gradually (Figure 8). After each increase, the results of the simulation were observed qualitatively.

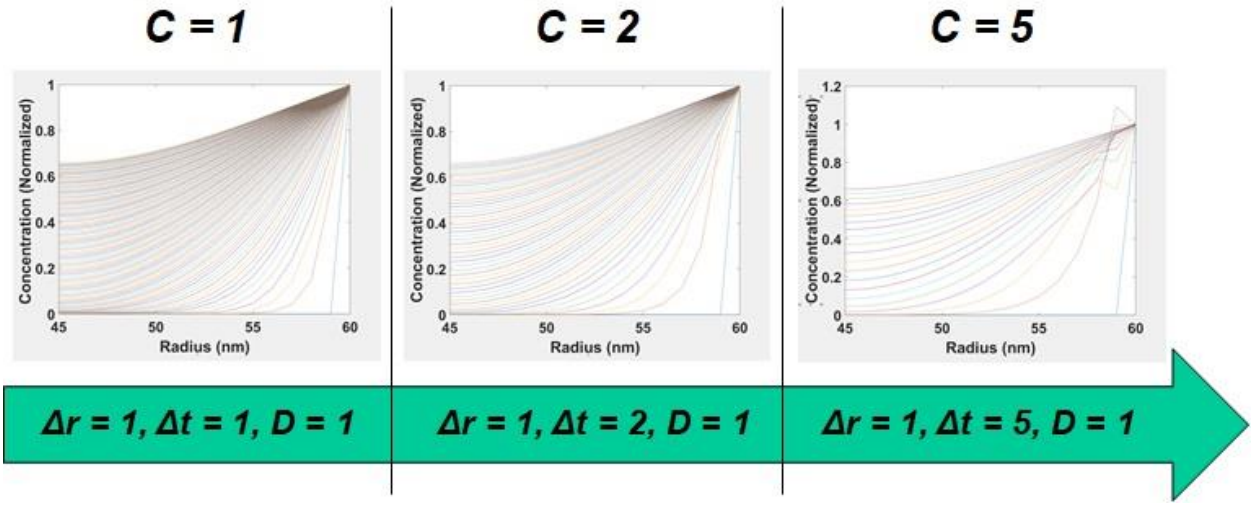


Figure 8: Modifying the dimensionless *C* parameter can greatly impact the presence of error within the simulation. As *Δt* is increased, perturbations begin to occur within the earliest time steps.

The values of *D* and *Δr* remained constant, but *Δt* was increased from 1 to 5 seconds. Initially, with *C = 1*, the simulation appeared to work fine; the boundary conditions were met, and there was no evidence of significant error propagation. However, as the value of *C* increased over 2, perturbations within the early time steps became apparent. With *Δt = 5*, there were some instances of the simulation exceeding the normalized scale (right panel of Figure 8). After these initial perturbations, the algorithm corrects itself to a solution similar to the previous simulations.

The goal of developing a numerical simulation is to reduce instances of the solution becoming unbounded or "blowing up". While the simulation did not blow up in these examples, the error still must be reduced as much as possible. An additional line of code prevents this same perturbation from occurring by adjusting *Δt* to keep $C = 1.5 < 2$: *>> delt = (1.5*delr^2)/D;*

2.6. Correlating with STEM-EDX Measurements

One of primary goals of this work is to be able to correlate between numerical simulation and experimental data. For the case of this study, diffusion within the nanoparticles is measured with an HRTEM in scanning-mode (STEM); essentially, the HRTEM scans the nanoparticle and

displays the distribution of elements within the core-shell structure with respect to atomic percentage. However, rather than show the distribution at the center of the nanoparticle, similar to the simulation results in Figure 7, the HRTEM actually measures a projection of the 3D elemental composition onto a 2D plane, similar to the data shown in Figure 9.

2.6.1. Implementing EDX Projection

Ideally, there should be a stark contrast between the core and shell on the EDX scan; however, the scan results in a diffuse boundary between the core and shell. With the EDX scan being a projection, this creates difficulty when trying to compare experimental and simulated results. In order to simulate this projection, the concentration was averaged with respect to the "z" direction, i.e., the direction of the electron beam. To accomplish this, the quarter-circle model used before was discretized on a rectilinear plane, as shown in Figure 10.

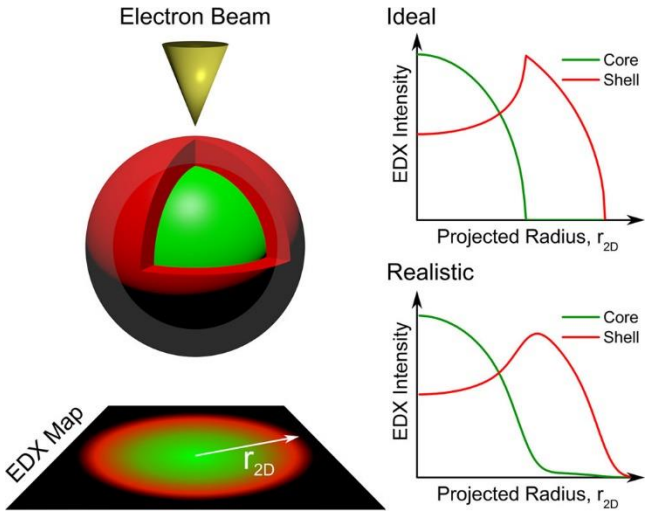


Figure 9: Schematic of STEM-EDX Map of core-shell nanoparticle. In an ideal case, there should be a sharp elemental transition between the shell and the core; however, this is not likely due to a finite probe size and a diffuse core/shell interface (Reprinted with permission from [67]. Copyright 2018 American Chemical Society)

As Figure 10 indicates, the projected concentration at each radius increment is determined by averaging the concentration values directly "above" with respect to the *z*-axis. Numerically, this can be represented through Equation 18, where c_{r_q} is the projected sodium concentration at radius index *q* and *Z* is the total number of indices on the *z*-axis within the particle dimensions.

$$\boldsymbol{c_{r_q} = \frac{\sum_{s=1}^{Z} c_{r_q}^{s}}{Z} = \frac{c_{r_q}^{1} + c_{r_q}^{2} + \cdots + c_{r_q}^{Z}}{Z}} \qquad \textbf{(18)}$$

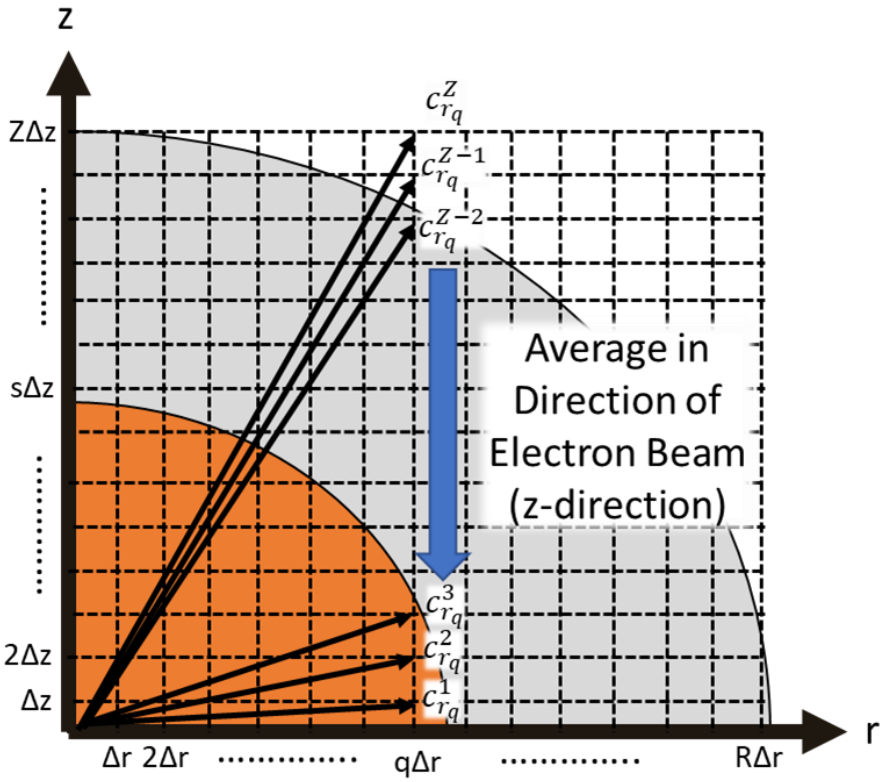


Figure 10: Diagram of the method used to emulate the projection created by EDX measurements. At each radius, the concentration values are averaged in the z-direction at each radius value. By doing so, the projected concentration value at each radius value can be estimated.

For the simulation at the center cross-section, the concentration of sodium at the core (r = 0) should be zero (left panel of Figure 11); however, the projection differs from the center-cross section in that there appears to be a noticeable concentration of sodium within the nanoparticle core, due to the electron beam interacting with the portion of the shell directly above the core (right panel of Figure 11).

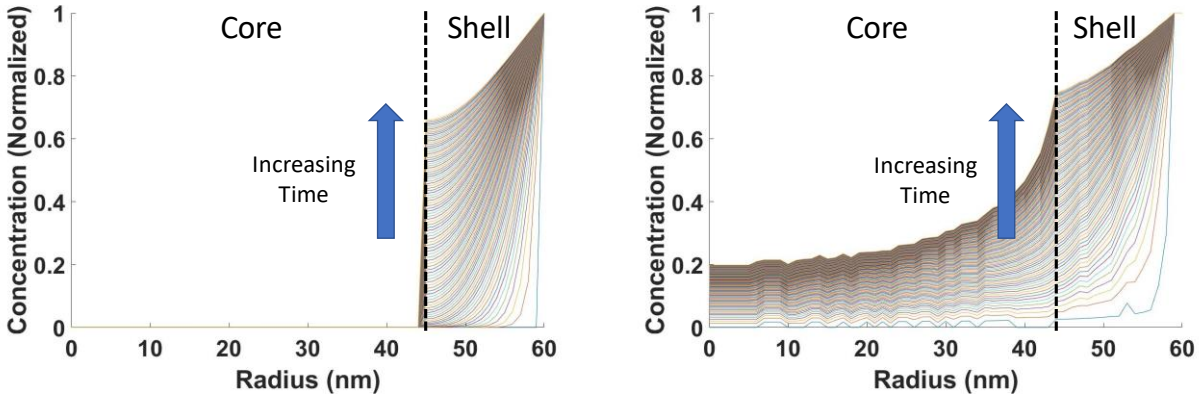


Figure 11: (left) Simulation results at the particle center over the entire radius. The sodium diffuses into the shell and stops when it reaches the core (45 nm); (right) The projected concentration gradient considers the average concentration with respect to the vertical *z*-axis, giving the impression of sodium ions present in the particle core. Simulating the projection provides greater similarity to the EDX measurements.

2.6.2. EDX Projection: MATLAB Pseudocode

The below pseudocode indicates how the projected concentration gradient is determined from the existing CN algorithm. Completed code can be found in the Appendix.

```
>> delz = delr; z = 0:delz:ro; Z = length(z);          %Set z-discretization size
>> c_z = zeros(rplotl,Z,T);   %Create matrix to store z values
>> c_avg = zeros(rplotl,T);   %Create matrix to store projected (averaged) concentrations
>> rplot = [0:delr:r(1),r(2:R)]; rplotl = length(rplot);       %Entire radius range (core and shell)
>> cplot = zeros(rplotl,thetaplotl,T);%Create plotting matrix with core concentration values
```

```
>> for p = 1:T
   cplot(rplotl-R+1:rplotl,:,p) = repmat(c(:,p),1,thetaplotl); %Fill plotting matrix with shell conc
values
>> end
>> for p = 1:T
   >> for q = 1:rplotl
      >> count = 0;
      >> add = 0;
      >> for s = 1:Z
         >> r_p = sqrt((q*delr)^2+(s*delz)^2);        %Calculate radius for selected z value
         >> if r_p <= ro        %Filter radii that exceed the particle size
            >> r_pcorr = interp1(rplot,rplot,r_p,'nearest'); %Interpolate to nearest radii within r
            >> index = find(rplot==r_pcorr);          %Find index for nearest radii
            >> c_z(q,s,p)=cplot(index,1,p);     %Using index, find associated concentration value
            >> add = add+c_z(q,s,p); count = count+1;       %Sum the z values and count once
         >> else
            >>c_z(q,s,p)=0;     %If z exceeds particle dimensions, don't count it and set conc to zero
         >> end
      >> end
      >> if count == 0
         >> c_avg(q,p) = c(R,p);       %Prevent division by zero if count is low
      >> else
         >> c_avg(q,p) = add/count;           %Average all the z concentrations at each radius
```

>> end

>> end

>> end;

In a sense, the code performs simple operations to calculate the projected concentration profile:

1. Particle is partitioned into square sections where the side length is equivalent to Δr.
2. The length between each node ($c_{r_q}^{z}$) and the particle is center is calculated and mapped to the closest value within the entire particle radius range (core and shell). If the length exceeds the particle size, the measurement is not counted.
3. Using the indexed radius value, the concentration at a node is determined from the simulation results.
4. All the values at a particular radius are averaged in the z-direction. The averaged value is then stored as a projected concentration.

2.7. Summary

A key element in facilitating nanoparticle temperature estimation offered in this dissertation is being able to effectively model the diffusion of sodium ions into a silica shell structure and compare the results of the model with experimental data. With the transient diffusion PDE possessing a parabolic structure, the Crank-Nicolson algorithm proved to be a stable and reliable algorithm to model the diffusion of ions within the silica shell, and MATLAB provided a simple, yet effective IDE. Since the diffusion within the nanoparticle will be measured with EDX functionality in an HRTEM, a significant part of the model accounted for the projected elemental map produced by the EDX measurement. This allowed for a better comparison between theoretical and empirical datasets.

While simulation development represents an important step in the understanding of nanoparticle diffusion behavior, the model is useless without empirical diffusion data (i.e., activation energy of diffusion and diffusivity at infinite temperature). Simulation results vary wildly depending on the diffusivity value “D” the user inputs. Diffusivity for a particular material system largely depends on the local temperature, which is the main idea for this work. If the diffusion of sodium into sol-gel silica is well characterized, then local temperature could be predicted based solely on the concentration gradient within the nanoparticle structure. The next section addresses the determination of these diffusion constants necessary to accurately predict diffusivity so that this model can be used to extract temperature.

3. DIFFUSION CONSTANT DETERMINATION

This work centers around estimating temperature by observing diffusion phenomena within nanoparticles. Diffusion possesses temperature dependency through an Arrhenius relation, similar to chemical reaction rates. Even with the simplicity of the equation, the activation energy (E) and diffusivity at infinite temperature (D_0) represent material constants unique to a particular system, which need to be determined empirically or through simulation. In the case of the SCNPs, the system of interest is sodium ions diffusing into a sol-gel silica layer. An early study exists in the literature that analyzes a similar system of sodium diffusing into silica glass and relates diffusivity with temperature experimentally [68]; however, the silica glass in this study, while similar in chemical composition, most likely possesses structural differences from sol-gel silica layer present on the surface of the SCNPs. Due to this assumption, the first aspect of this work aims to experimentally determine the diffusion constants associated with sodium ions diffusing into sol-gel silica to provide a better understanding of ion mobility into the shell of the SCNPs.

Determining the diffusion constants empirically is the most straightforward option for measuring temperature-dependent diffusion behavior. Rather than use theoretical analysis or computational material science, the diffusivity at various temperatures is determined by measuring the diffusion length of a material (sodium) into the bulk (silica). Ideally, the temperature is known and maintained at a set value throughout the experiment; however, in the case of this study, with diffusion into a nanoparticle shell, the temperature of interest (local nanoparticle temperature) is unknown. Without knowledge of the exact temperature of the system throughout the experiment, no meaningful conclusions can be drawn about the diffusivity. Because of this, a separate, micro-scale system must be studied, which both accurately mimics the material properties of the

nanoparticle silica shell as well as enables precise temperature control. For simplicity, this system would also need to possess a rectilinear geometry to accurately measure the diffusion depth.

3.1. Sol-Gel Silica Thin Film Creation and Characterization

In the case of this study, rather than coat a set of nanoparticles, a sol-gel silica coating will be formed on a polished silicon substrate. This will create a flat, rectilinear coating, which will be easier for empirical analysis, as well as enable environmental temperature control by heating the sample in an oven. The methodology used in this study utilizes sol-gel silica to create a uniform coating on the silicon substrate, similar to that on the SCNPs. Basically, sol-gel synthesis consists of first creating a colloidal suspension of particles (sol) and then encouraging the formation of an integrated particle network (gel), which comprises the coating. Sol-gel coatings are utilized ubiquitously in industry and academia and are quite easy to create in a variety of geometries.

3.1.1. Creating the Film

The chemicals utilized in this section were ACS grade and all acquired from commercial sources. This sol-gel coating procedure (Figure 12) was adapted from Reference [69] to both utilize a well-referenced source and to ensure simplicity. To create the colloidal silica suspension, a solution of tetraethyl orthosilicate (TEOS), methanol, water, and hydrochloric acid (HCl) at a respective molar ratio of 1:3:8:5E-5 needed to be prepared. To keep the batch small, the volume was scaled down to approximately 20 mL. First, 5.9 mL of deionized water, 4.98 mL of methanol, and 0.13 µL of HCl were added to a flask. While actively stirring the mixture with a stir bar at room temperature, 9.1 mL of TEOS was added dropwise to the solution; after introducing the TEOS, the mixture was covered with plastic wrap, heated to 60°C, and stirred vigorously for 90 minutes. Afterward, the addition of approximately 9.3 µL of HCl increased the concentration to 7.34 mM. Upon allowing the solution to cool, the mixture underwent additional stirring at room

temperature for 15 minutes. The solution was then aged at 50°C for 15 minutes. After aging, 60 mL of methanol diluted the solution by a factor of 3, which resulted in the completed sol-gel mixture.

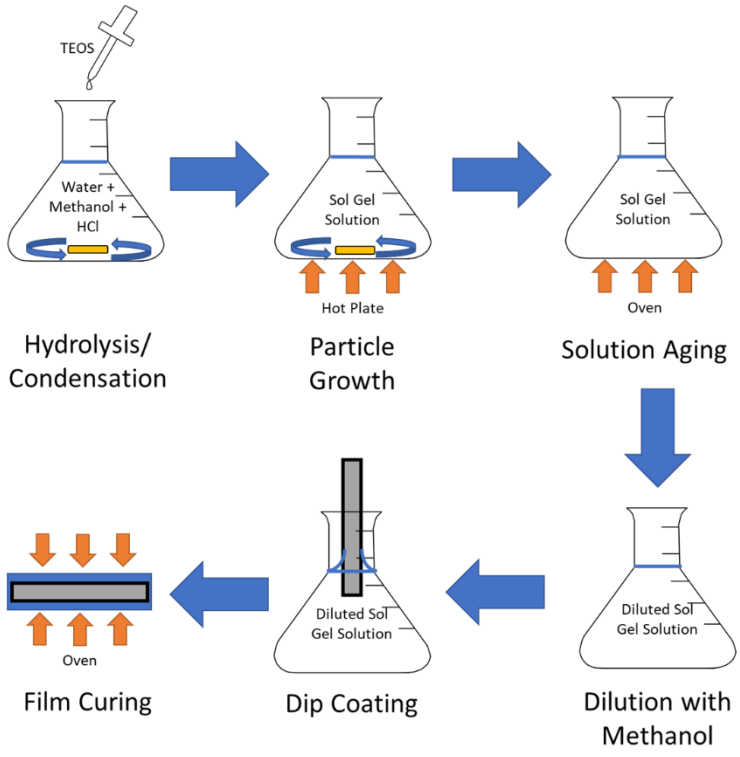


Figure 12: Sol-gel coating process used to create the silicon wafer samples for the diffusivity study. First, TEOS was introduced to the aqueous suspension to start the hydrolysis reaction. After heating and vigorous stirring, the final gel solution is diluted with methanol; the substrates are then dipped at a controlled speed into the solution to form the sol-gel silica coating.

The primary reaction occurs between TEOS (an alkoxysilane) and water, and it is comprised of two stages: hydrolysis and condensation. Hydrolysis first begins to replace the alkoxide groups with hydroxyl groups, and then condensation occurs, where the partially hydrolyzed molecules react to produce the network of Si-O-Si bonds [70, 71]. Due to the immiscibility of water and TEOS, methanol acted as a homogenizing solvent, while the addition

of HCl helped to keep the pH of the mixture low to encourage condensation. Early references described hydrolysis and condensation with three separate chemical reactions, where R represents the alkyl group, C_xH_{2x+1} [70, 72]:

$$\equiv Si - OR + H_2O = \equiv Si - OH + ROH \quad (1)$$

$$\equiv Si - OR + HO - Si \equiv = \equiv Si - O - Si \equiv + ROH \quad (2)$$

$$\equiv Si - OH + HO - Si \equiv = \equiv Si - O - Si \equiv + H_2O \quad (3)$$

The silicon substrates used in this study were created from a 100 mm diameter, single-side polished silicon wafer in the <100> orientation. Silicon substrates were used due to their low surface roughness, low cost, and possessing chemically favorable bonding conditions for silica. A low-speed saw diced the original wafer into several 10 x 10 mm pieces. After a thorough cleaning with acetone and water, each piece of the silicon wafer was then attached to a linear actuator (PI M-235.2DD), which lowered them into the sol-gel solution at a rate of 1.3 mm/s, kept still in the solution for 30 minutes, and then removed the piece at the same rate of speed. The speed of the substrate removal represents an important parameter, which both governs the thickness of the coating as well as the uniformity [71]. After removing the sample, the solution began to dry, and the coating became visible via the discoloration of the polished side of the silicon piece. Each coated piece was then cured in an oven at 90°C for 1 hour. Last, the cured silica-coated samples were then further cut with the low-speed saw to 2 x 2 mm squares.

3.1.2. EDX Characterization

An FEI Nova Nanolab 200 Scanning Electron Microscope (SEM) was one of the primary instruments used to characterize the sol-gel silica layers. In addition to the visual characterization,

EDX spectroscopy was utilized to provide an elemental breakdown of the films. Figure 13 shows a top-view micrograph in an area along the edge of the film where cracking was observed. To clarify, the sol-gel coating was purposefully cracked in order to generate the image; the majority of the film exhibited a uniform structure with minimal cracking. In the bottom panel, the EDX map of the same micrograph is also shown, which illustrates the predominant elements present within the view. Figure 13 provides a stark qualitative contrast between the sol-gel silica layer and the silicon substrate. Due to the oxidized structure of silica, oxygen atoms primarily dominate the signal from the sol-gel coating (yellow), as opposed to the silicon wafer substrate (purple); however, the oxygen-rich regions also contain silicon atoms as well, which provides further evidence for the presence of silica.

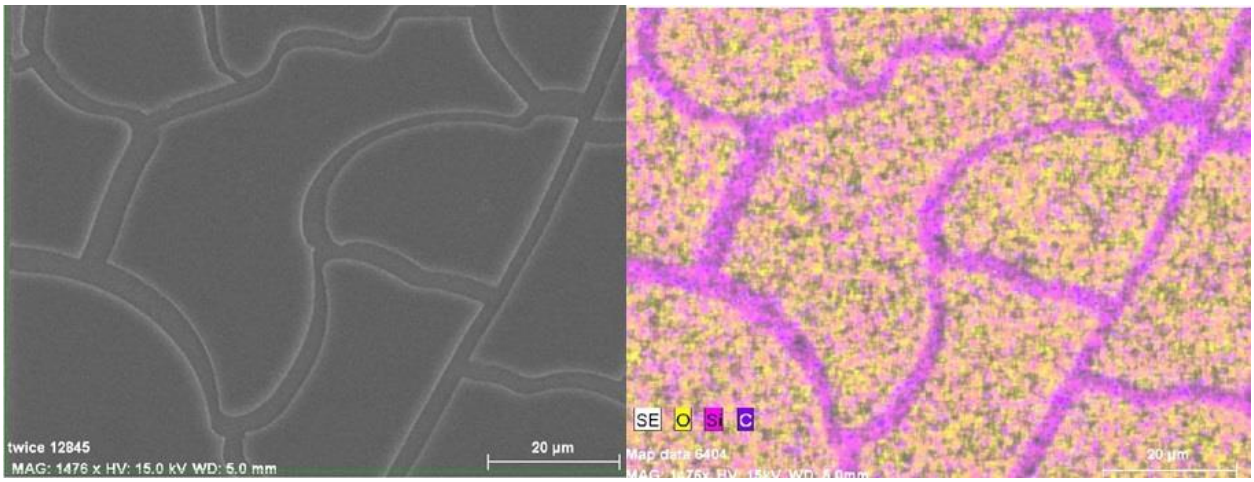


Figure 13: EDX scan of an SEM micrograph taken on the surface of a silica-coated sample. This particular area, with several micro-cracks, showcased the oxidized structure of the film. (top) Original micrograph; (bottom) EDX map of the region, where the yellow regions indicate oxidized silicon and the predominately purple regions illustrate the silicon substrate peeking through the cracks.

In addition to the top view images, the cross-section of the samples was also examined to characterize the film thickness. In this case, two of the samples were glued together using an epoxy

resin with the coated sides facing each other. The sandwiched sample was then cut in half and polished using polishing pads ranging from 0.5-90 µm in roughness. SEM and EDX analysis were then performed on the cross-section. Due to the small scale and lack of visible contrast, only the EDX image is shown in Figure 14.

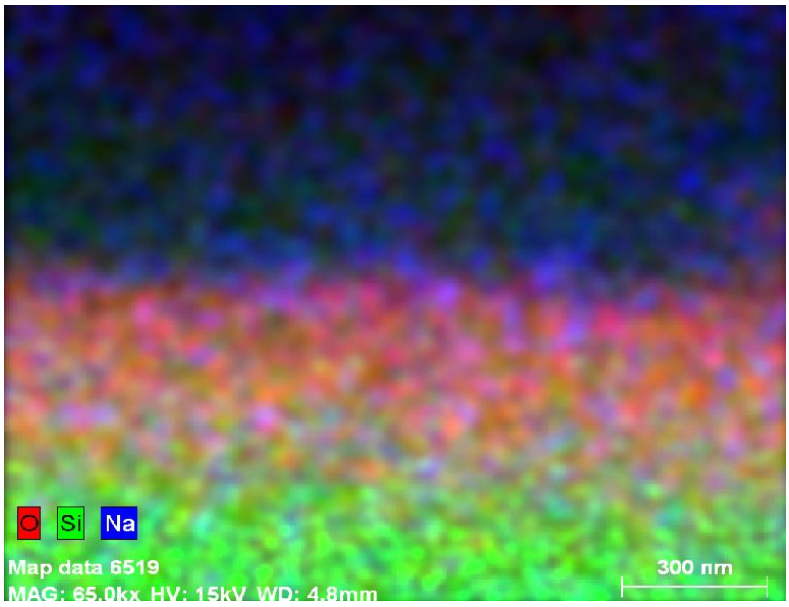


Figure 14: EDX profile of a cross-section of the silica layer. The image distinctly shows the predominately silicon substrate (bottom green) and the oxidized silica layer (middle red).

From Figure 14, the oxidized film (red) is clearly seen directly on top of the silicon substrate (green). Based on the scale and the measurements performed, the film is estimated to be 400 nm in thickness, but the cross-section is only a small portion of the film, which may not be a good representative sample of the thickness variance across the entire 2 x 2 mm square. However, these measurements do provide a good thickness estimation for the diffusion calculations.

3.1.3. Characterization with XPS Depth Profiling

While the EDX measurements distinctly show an oxidized thin film structure on top of the silicon substrate, these qualitative images do not distinctly prove whether the structure is

comprised of silica or some other oxidized material. In order to do provide quantitative characterization, X-Ray Photoelectron Spectroscopy (XPS) Depth Profiling was used to determine the binding energy of the silicon atoms present in the thin film. If the film is comprised of silica, there should be a distinct difference in binding energy between the oxidized film and the elemental silicon substrate. A PHI 5000 VersaProbe Scanning ESCA Microprobe was used to gather the depth profiling measurements. An initial scan was gathered on the surface of the coated 2 x 2 mm sample. Afterward, a 1 kV Argon sputtering beam etched into the sample for 30 seconds at a rate of approximately 8-15 nm/min, followed by an additional XPS scan, as shown in Figure 15.

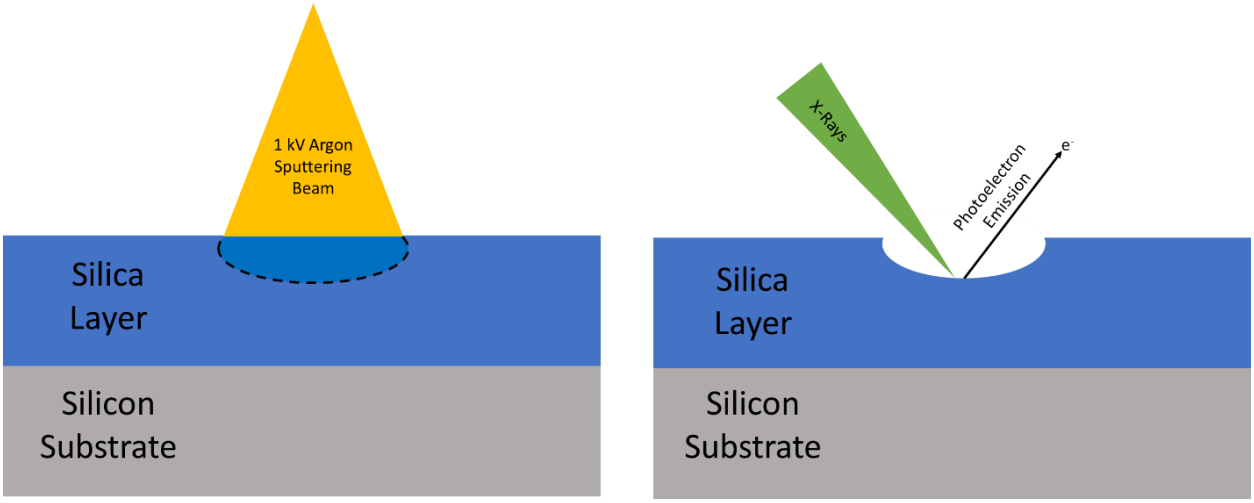


Figure 15: Illustration of XPS Argon Sputtering. (left) A 1 kV sputtering beam gradually removes material from the silica layer. (right) Following material removal, an XPS spectrum is gathered in the crater at a known depth.

The alternating cycle of sputtering and scanning was repeated 60 times. Figure 16 showcases a portion of the XPS spectra pertaining to the Si2p orbital; the scans after each sputtering cycle are overlaid to illustrate how the peak position changes with respect to sputtering time.

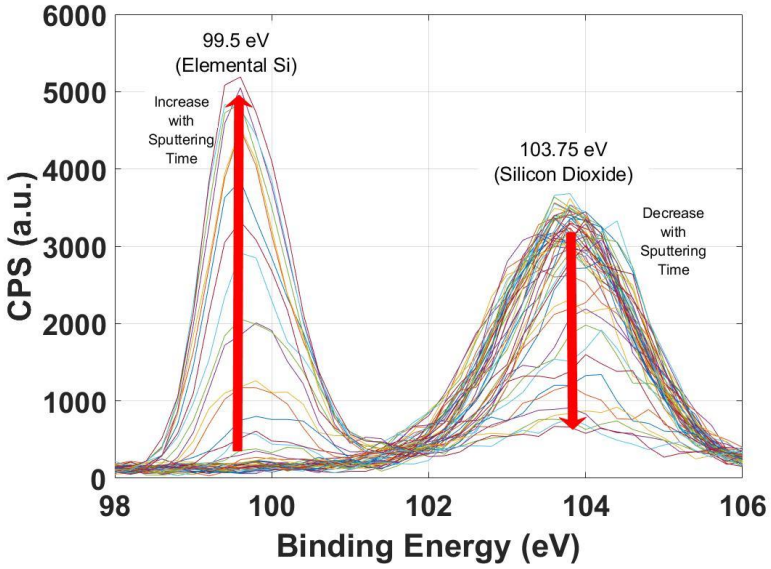


Figure 16: XPS depth profile of the sol-gel silica film focused on the Si2p peak. The initial scans indicated a strong peak at 103.75 eV, which corresponds to oxidized silicon; however, once the sputtering beam etched through the layer, the peak shifted to 99.5 eV, which corresponds to elemental silicon

For the first several scans, the predominant peak occurred at 103.75 eV; the high binding energy of this orbital correlates to an oxidized structure, which is characteristic of silica. As the beam gradually etches away the thin film, the intensity of the oxide peak diminishes, and a new peak appears at a lower binding energy (99.5 eV). The transition occurs when the sputtering beam has etched completely through the silica layer and is starting to etch into the silicon substrate, where no oxidized Si atoms are present. Similar to the EDX, the peak transition shows high contrast between the silicon substrate and the silica thin film. Because of the presence of the peak at 103.75 eV for the Si2p orbital, it can be definitively determined that the material present within the layer is in fact silica, rather than some other form of oxidation.

3.2. Measuring Diffusion Empirically

The purpose of the silica samples created in this chapter is to mimic the sol-gel silica shell that surrounds the nanoparticles. Unlike the silica present in glassware, which is formed through

high-temperature processes, silica films created through sol-gel procedures tend to be quite porous [73]. The presence of structural differences between sol-gel and traditional silica begs the question of how they impact diffusion behavior. As covered in previous sections, the commonly used model to describe the temperature dependence of diffusivity (Arrhenius model) uses a simple "activation energy" term (E_o) to describe the energy required to initiate a diffusion event. The general hypothesis is that this activation energy should be much lower in a porous sol-gel silica film than a solid silica film, such as with the study by Frischat et al. referenced earlier [68]. Micro/nano-size pores provide lower-energy routes for sodium atoms diffusing into the silica shell rather than vacancy or interstitial diffusion. The rest of this chapter will focus on the empirical determination of the diffusion of sodium ions into a sol-gel silica film through the use of XPS depth profiling.

3.2.1. Sample Creation

The 2x2 mm square samples created in the previous section were utilized as the primary means to measure diffusion. Characterization of the films indicated that a ~400 nm layer of silica was present on the surface of the silicon wafers after dip coating. However, the most challenging aspect of this study was to maintain a high concentration of sodium atoms at the surface of the silica layer, while also maintaining good surface contact. Dispersing the sodium atoms into an epoxy resin proved to be an effective method, which satisfied both conditions. To effectively introduce sodium into the epoxy, 2 g of non-iodized sodium chloride was dissolved in 20 mL of a 50/50 volume mixture of methanol and water. Afterward, 100 µL of the solution was added to a 1 mL of epoxy at a 1:1 resin to hardener ratio. The methanol served as a thinning agent for the epoxy, allowing it to be easily spread on the small sample surface, the water allowed for greater solubility of sodium chloride into the mixture, and the epoxy ensured good surface contact without compromising the integrity of the silica layer. The diagram in Figure 17 illustrates the process of

creating the samples for diffusion testing. After the epoxy dried onto the surface, each sample underwent heat treatment in a high-temperature oven at a set temperature for a given amount of time. After completing the heat treatment, each sample cooled to room temperature and soaked in an acetone bath for 1 hour to remove the epoxy. While the acetone works effectively at removing the epoxy from the surface, visual observation indicates that it does not impact the silica layer.

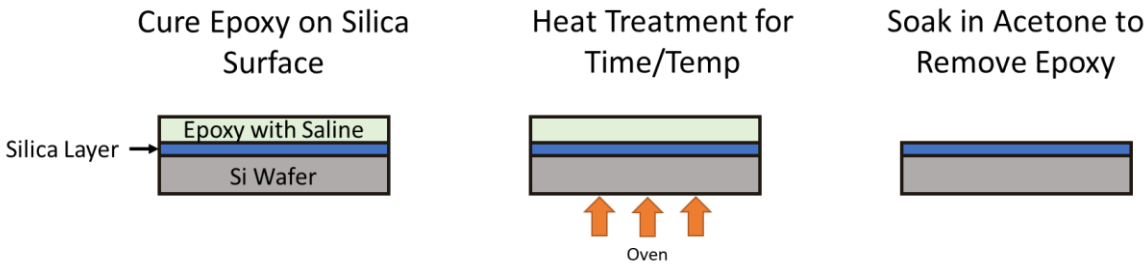


Figure 17: Process to create the samples for the diffusion study. The 2x2 silica samples are first coated with saline epoxy, allowed to dry, and then placed in an oven at a pre-determined temperature. After heat treatment, the wafer is soaked in acetone to remove the epoxy, but it does not dissolve the silica layer.

3.2.2. Heat Treatment

After coating each sample with the saline epoxy and allowing them to dry, the samples were then placed in an oven at a set temperature for a pre-determined amount of time. Heating in the oven promoted the diffusion of sodium from the epoxy layer into the silica layer. The diffusion creates a concentration gradient within the film, with the highest concentration being between the silica and epoxy layer and the lowest concentration being at the surface of the silicon wafer. The selection of the oven time and temperature required some forethought because multiple variables can influence the gathered results. The goal is to achieve a large concentration gradient between the top and bottom of the film that can be measurable with the XPS. If the sample temperature is too high or the diffusion time is too long, then the sodium atoms will reach a linear steady-state

concentration within the film, and the gradient with respect to the length will be difficult to discern. Also, according to the manufacturer, the stability of the epoxy significantly reduces when exceeding 200°C. Conversely, if the temperature or time is too low, then the sodium atoms will not diffuse far enough into the silica layer to measure with the XPS. An array of temperatures and times were chosen by calculating the estimated diffusion length ($D_l = \sqrt{Dt}$) using the results of the study by Frischat *et al.* [68], where "D" is temperature-dependent diffusivity and "t" is time. While the diffusion lengths most likely will not be exact, the estimations from an existing model provide a good starting point. Ideally, the diffusion length for the samples should not exceed the thickness of the silica film (~400 nm). Table 1 illustrates the testing matrix used for the diffusion constant determination as well as the estimated diffusion lengths.

Table 1: Testing matrix for the diffusivity study including estimated diffusion lengths. A sample will be prepared for each combination, heated for the given temperature and time, and tested with XPS.

	150°C/423 K	**175°C/448 K**	**200°C/473 K**	**250°C/523 K**
30 min	30.5 nm	78.1 nm	180.9 nm	762.8 nm
60 min	43.2 nm	110.4 nm	255.8 nm	1078.8 nm
90 min	-	135.2 nm	313.3 nm	1322.2 nm
120 min	61 nm	-	-	-

As can be seen, the selected testing matrix includes estimated diffusion lengths from 30 nm to over a micron; however, these are values estimated from sodium diffusing into solid silica rather than sol-gel. It is likely that the diffusion lengths will be much longer in the sol-gel samples; however, this is merely a hypothesis. To prepare for the eventuality that the diffusion is inhibited in the sol-gel, the 250°C case was explored.

3.2.3. Measuring Concentration Gradients

Similar to the characterization discussed in Section 3.1.3., XPS depth profiling (Figure 15) was the primary technique used in characterizing the diffusion of sodium into the silica layer after heat treatment. Each sample underwent an initial XPS scan to establish a surface concentration of sodium. Afterward, the alternating cycle of sputtering/measurement occurred 60 times, or until the beam had completely etched through the silica layer, which was determined by waiting for the peak shift observed in Figure 16. During each measurement cycle, the peak occurring in the region associated with the Na1s orbital (1066-1076 eV) was analyzed in terms of its area (Figure 18). As the sputtering beam etched through the silica, the peak intensity would be monitored and normalized with respect to the initial surface concentration. Since all the samples were made from the same dip-coated wafer, with a characterized silica thickness of approximately 400 nm, the sputtering speed was estimated to be 10 nm/min, with slight variations between samples.

After recording the intensity after each sputtering cycle, the resulting plots appeared similar to the left panel of Figure 19, which illustrates the 150°C/60 min heat treatment sample. The highest concentration is observed at the surface of the silica layer (0 nm), which on a normalized scale is set to "1". Further measurements illustrate an exponential decline in concentration as the sputtering beam etches further into the layer. When the concentration achieves a zero slope near 200 nm, this is assumed to be the baseline, or where the concentration is zero. When this occurred in the samples, the curve underwent correction to ensure the model could achieve a good fit, which made the concentration zero within this asymptotic region. The presence of the baseline noise most likely originates from the detection of photoelectrons outside the sample area; the peak, while rather small, still registers within the detectable area, resulting in an intensity measurement. Each sample shown in Table 1 underwent XPS depth profiling and a similar curve was generated.

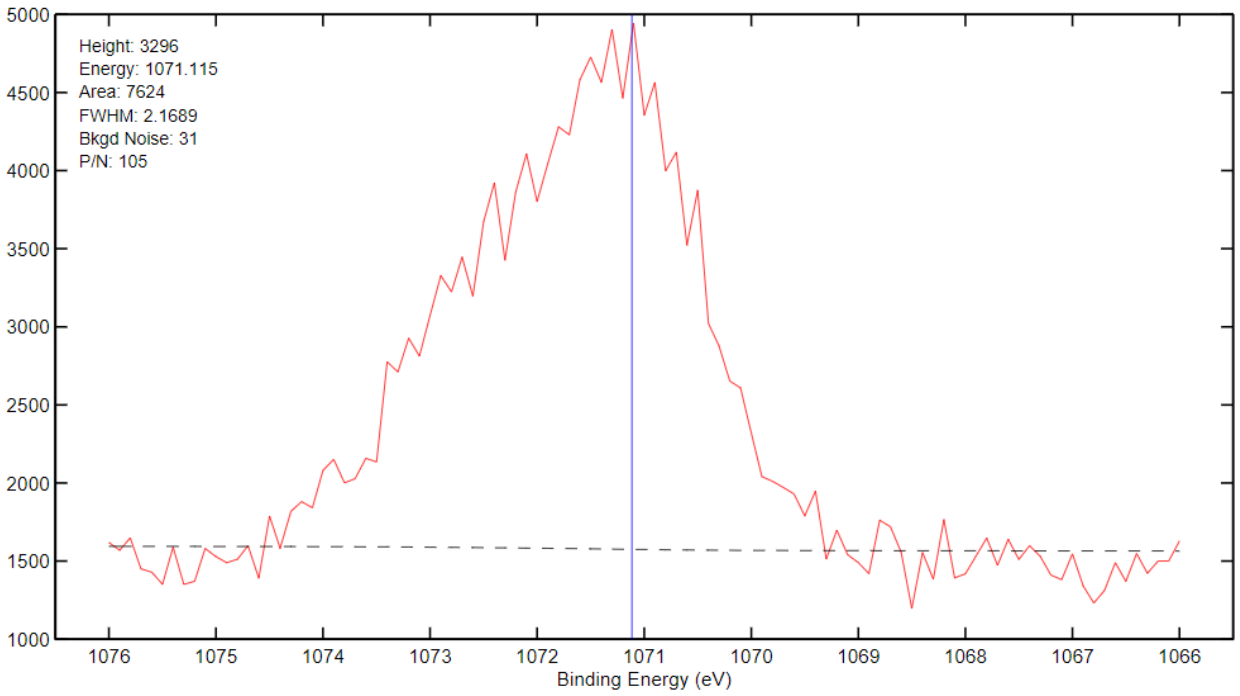


Figure 18: Example of sodium peak analysis during XPS depth profiling. After each sputtering time, the Na1s peak was analyzed and its area recorded.

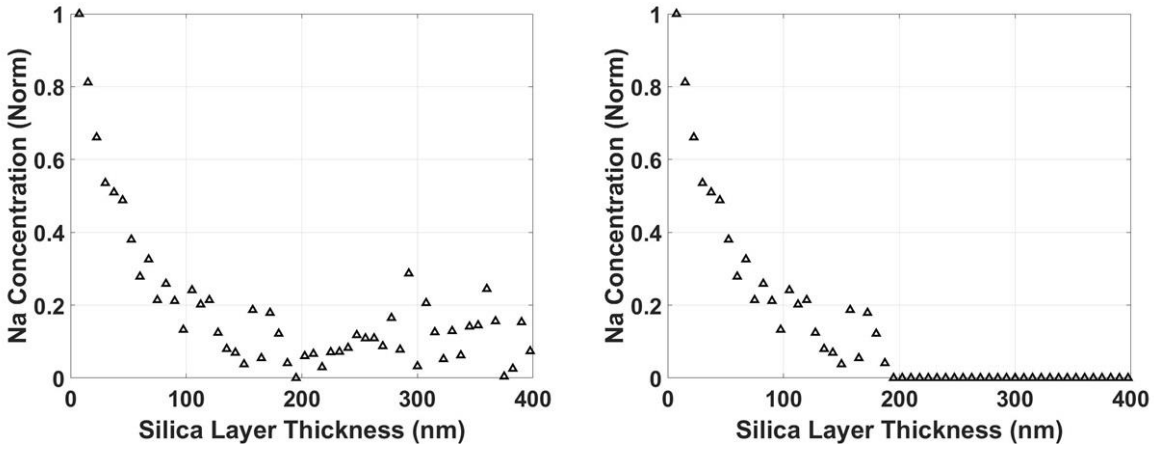


Figure 19: Intensity vs thickness plots for the 150°C/60 min heat treatment. For these measurements, the surface of the silica layer corresponds to 0 nm, while the silicon wafer corresponds to 400 nm. (left) Uncorrected raw data from the XPS. The highest concentration is observed near the surface with an exponential decay. However, the data approaches zero concentration and appears noisy as the peak intensity decreases. (right) Corrected data for use in the model, where data points in the asymptotic region are represented as zero. The main focus of this study is the initial region of concentration decay (for this example between 0 and 200 nm).

3.2.4. Qualitative Analysis of Edge Cases

As stated in the previous section, it was unknown how well diffusion would occur at the edge cases, i.e., highest temperatures and times. If the epoxy degraded at higher temperatures, then this most likely would result in poor sodium dispersal within the silica layer. Also, if the samples are heated for too long, then a steady-state condition might be reached with no discernable gradient present. The concern proved valid when observing the high temperature/time cases. As a case study, the qualitative differences between the 30-minute tests at 150°C, 200°C, and 250°C were taken into consideration (Figure 20).

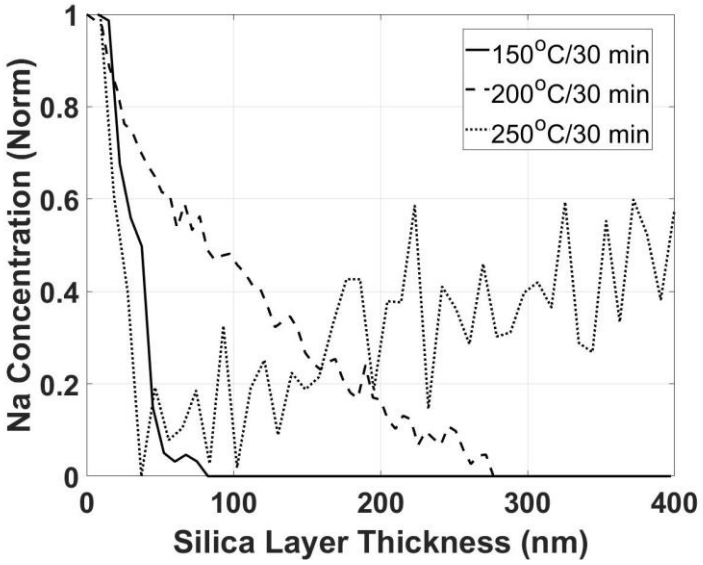


Figure 20: Comparison between 30-minute tests at 150°C, 200°C and 250°C. The curve for the 150°C and 200°C test results in a nice decreasing concentration curve; however, the 250°C test observed a sharp decline at the beginning of the layer and a gradual (and noisy) increase throughout the film.

The 150°C and 200°C tests resulted in a gradual and predictable exponential decay in concentration, consistent with transient diffusion. An increase in temperature to 200°C encourages further diffusion, which results in a higher sodium concentration deeper into the layer; however,

the trend stops once the temperature exceeds 250°C. Not only is the data much less intense, resulting in increased noise, but the concentration towards the surface, where it should be the highest, takes a steep decline before gradually increasing. After consideration, the culprit seemed to be the breakdown of the sodium delivery system: the epoxy. The original experimental design harnessed the properties of the epoxy to not only maintain a constant-concentration boundary condition but also allow for high surface contact. When the tests exceeded the temperature stability region of the epoxy, the delivery system collapsed. Thermal breakdown of the epoxy compound resulted in polymer degradation, which most likely significantly reduced the surface concentration of sodium atoms. Because this was a gradual process over the course of the entire test, some sodium atoms did diffuse into the silica layer but not a similar magnitude as the lower temperature testing.

While high temperature also alters the epoxy compound, the breakdown also possesses a time dependency. Even samples tested at longer testing times observed similar concentration curves, such as with the tests at 175°C (Figure 21). At 175°C/30 minutes, the concentration curve appeared normal with a very rapid decay and asymptotic behavior around 100 nm into the sample, and the 60-minute sample possessed a curve that indicated a little more diffusion into the sample. However, for the 90-minute sample, the curve possessed a structure similar to the 250°C tests, where a strong drop on the surface is followed by a gradual increase in concentration deeper into the silica layer. With the concentrations being higher into the silica layer, it is possible to conclude that the samples are reaching a linear steady-state condition before the epoxy eventually breaks down; however, the polymeric degradation observed in these tests puts into question the quality of the results for the edge cases. While the easy argument would be to omit the trials from the dataset,

some conclusions could still be derived during the quantitative analysis to determine diffusion constants.

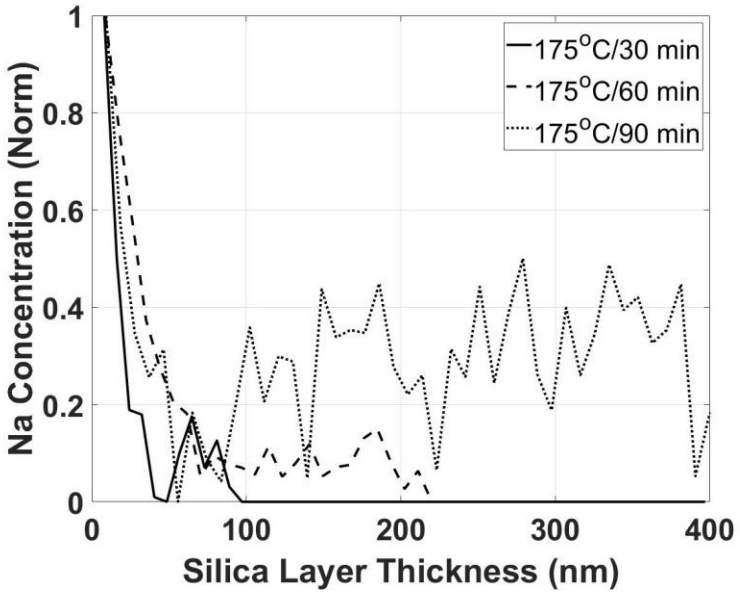


Figure 21: Comparison between the 175°C tests. Initially, the tests look normal at 30 and 60 minutes; however, at the 90 min test, the breakdown of the concentration curve is apparent, similar to the high-temperature testing.

3.3. Estimating Diffusion Constants

With the concentration curves gathered for each of the test cases using XPS analysis, the next step was to extract the relationship between diffusivity and temperature. Theoretically, the diffusivity for each temperature case (no matter the diffusion time) should be unique, e.g., the diffusivity values for 175°C at 30, 60, and 90 minutes should all be the same. After finding the average diffusivity at each temperature, the values were fit to the Arrhenius model to extract diffusion parameters.

3.3.1. Concentration Curve Fitting Algorithm

By using the numerical model in Chapter 2 to extract these values through an iterative process, the activation energy (E) as well as diffusivity at infinite temperature (D_o) can be estimated, which is essential for extrapolating nanoparticle temperature with this methodology. Before developing the code, a simple block diagram was used to illustrate the algorithm (Figure 22).

Figure 22: Block diagram for diffusivity determination algorithm. After importing a concentration curve, the theoretical concentration curve is solved for each diffusivity and compared against the experimental data. The diffusivity resulting in the highest R^2 is selected.

Essentially, a range of diffusivity values is guessed by the user. The simulation then solves the theoretical CN model for a rectilinear geometry and compares the output concentration curve to the experimental curve through a simple R^2 analysis. After testing all diffusivity values within the selected range, the value with the highest R^2 is chosen. Given the output, the user can then focus the diffusivity range to a finer scale. The below pseudocode outlines the algorithm implemented in MATLAB, where the complete source code can be found in the Appendix.

```
>> import('Experimental data');      %Import in single concentration curve
>> thickness = Imported array containing layer thickness;
>> intensity = Imported array containing concentration curve;
```

```
>> D = 1:1:50;          %Cycle over multiple diffusivities
>> for k = 1:length(D)
    >> Solve CN algorithm for Rectilinear Geometry with 400 nm thickness;
    >> for i = 1:length(thickness)          %Map experimental concentration gradient to model
        >> rnew(l) = interpl(r,r,thickness(l),'nearest');       %Find nearest model value
        >> index = find(r==rnew(l));          %Find index for that value
        >> unew(l,k) = u(index,T);    %Create of matrix of mapped values
    >> end
    >> data_mean = mean(intensity);
    >> sum_res = 0;
    >> sum_sq = 0;
    >> j = 1:length(thickness)
        >> sum_res = sum_res + (intensity(m) - unew(m,k))^2;       %Sum of residuals
        >> sum_sq = sum_sq + (intensity(m) - data_mean)^2;       %Sum of squares
    >> end
    >> R_squared(k) = 1- sum_res/sum_sq;       %Calculate R Squared
>> maxRSquared = find(R_squared==max(R_squared));   %Find max R Squared Value
>> D_opt = D(maxRSquared)          %Store diffusivity value with highest R Squared Value
>> R_2 = max(R_squared)            %Store R Squared value of optimal Diffusivity Value.
```

3.3.2. Quantitative Analysis of Edge Cases

The model used for this analysis assumed a constant concentration of sodium ions at the surface throughout the simulation. For most of the samples, loading epoxy with saline provided a simple, yet effective method to ensure this boundary condition during heat treatment; however, at

higher temperatures/times, it proved to be ineffective due to epoxy degradation, as shown in Figure 23. The left panel of Figure 23 illustrates experimental data which conforms well to the numerical model, resulting in an excellent fit; however, the right panel shows an edge case, where the model fit is quite poor. While typically R^2 values tend to be positive (on a scale from 0 to 1), a negative value indicates a particularly poor fit, where the model fits worse than a horizontal line drawn at the mean concentration.

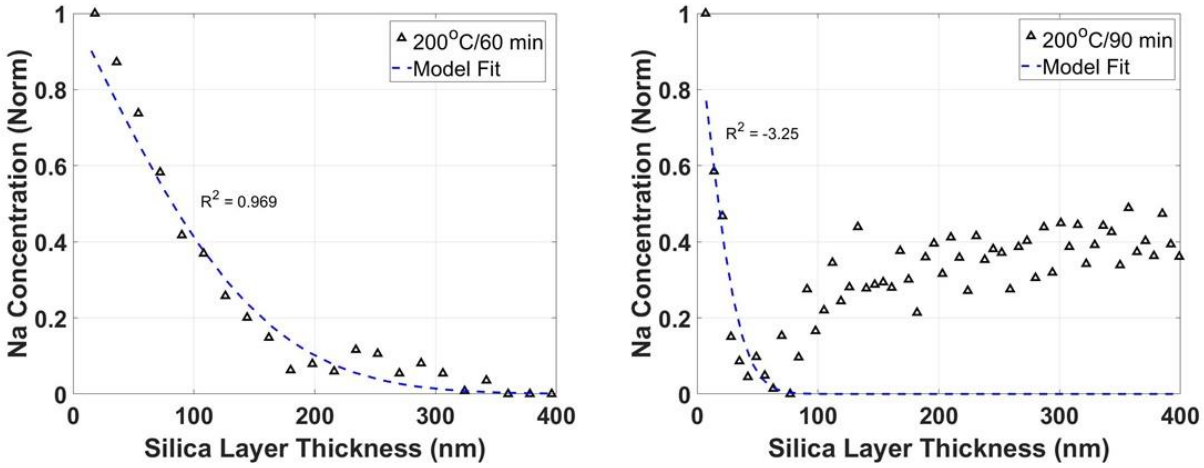


Figure 23: (left) Example of a good model fit to the 200°C/60 min trial; (right) Example of a bad model fit at one of the edge cases, 200°C/90 min.

It can be implied that the edge case fits are not valuable to the study since they deviate from the boundary conditions set by the model. However, rather than simply remove them from the data set, each concentration curve with a poor fit to the model was further analyzed by excluding the data points just below the surface, which from here on will be called the "trough region" (Figure 24). The hypothesis was that even though these cases exhibited transient diffusion behavior which deviated from the model, the data could still provide information regarding steady-state diffusion (when concentration gradient approaches linearity). Before excluding the points,

the algorithm fit the data shown in Figure 24 poorly; the R^2 value indicated a severely negative value. It did well in fitting the sharp decline right at the surface; however, it failed to capture the concentration gradient deeper into the silica layer. If the trough region was excluded, the algorithm focused on fitting the data points further into the film, which results in a significantly better fit (R^2=0.164). After performing this same exclusion for all the edge cases, the quality of their fits improved and the estimated diffusivity values increased, as shown in Table 2.

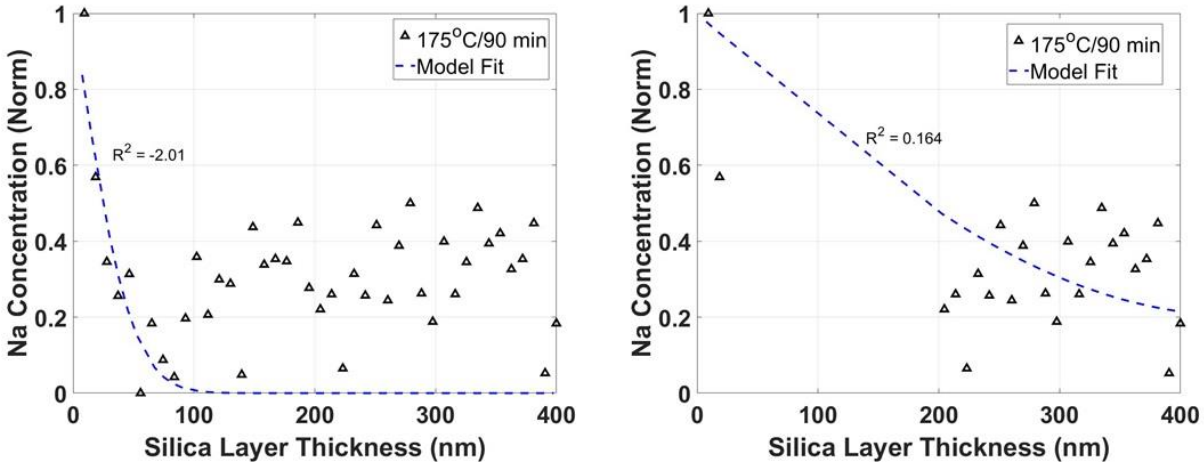


Figure 24: (left) The fit for the 175°C/90 min trial before data exclusion; (right) the same trial but after excluding data points collected near the surface of the silica layer.

Table 2: Analysis of the edge cases before and after excluding data points within the trough region. Every case not only noticed an improvement in the overall fit, but also an increase in the estimated diffusivity.

	Before Exclusion		**After Exclusion**	
Edge Case	**Diffusivity (nm^2/s)**	**R^2**	**Diffusivity (nm^2/s)**	**R^2**
175°C/90 min	0.13	-2.01	7.00	0.164
200°C/90 min	0.07	-3.25	7.34	-1.39
250°C/30 min	0.30	-2.09	28.2	0.04
250°C/60 min	8.43	-1.15	15.04	-0.19
250°C/90 min	11.91	-1.50	17.00	-0.01

In every case, excluding the data within the trough region resulted in an improved fit. Granted, the fits were still poor overall, but exclusion did help to improve the model. Such a result begs the question as to the cause of the trough region shown in nearly every edge case. A possible explanation could be a reversal in the diffusion direction caused by the breakdown of the epoxy. The experiment was designed to encourage diffusion into the silica layer, towards the silicon wafer, and it appears that nearly every edge case observed this to some degree. However, the surface of the silica layer could have become more sodium-deprived over time as the epoxy decomposed in the thermal environment, resulting in a reversal in the direction of mass transfer. Such an occurrence would theoretically cause a region of lower concentration near the surface. Regardless, the diffusivity values of the edge cases will not be used in estimating diffusion constants, due to their poor fits; however, they will be compared with respect to the final diffusivity vs temperature curve.

3.3.3. Calculating the Diffusivity Curve

Following the analysis of the edge cases, each concentration curve within the testing matrix was analyzed in terms of both calculated diffusivity and R^2 value, as shown in Table 3. As shown in the previous section, the edge case presented a more difficult challenge to fit than the lower temperature/shorter time cases. Table 3 highlights these values in yellow to indicate that the fitting process produced a fit of poor quality, while the green values indicated curves that with good overall fits. In some cases, the R^2 values went into the negative, indicating an extremely poor fit. For each of the green values within the same temperature set (e.g., 200°C at 30 and 60 min), an average and standard deviation were calculated, which represented the average diffusivity value for that particular temperature. The values in yellow were not used to calculate the average and standard deviation.

Table 3: Fitted values using the diffusivity determination algorithm. The green shaded values indicate a good fit, while yellow shaded values are those which fit poorly and were not included in the average/standard deviation calculations. *For 250°C, the values were averaged for investigative purposes only and were not included in the diffusivity analysis. $^{+}$ Denotes fits where data were excluded.

	Optimal Diffusivity Values (nm^2/s)			
	150°C	**175°C**	**200°C**	**250°C**
30 min	0.49	0.165	4.9	28.2^{+}
60 min	0.66	0.3	2.08	15.04^{+}
90 min	-	7.0^{+}	7.34^{+}	17^{+}
120 min	1.21	-	-	-
Average	0.786	0.23	3.49	20.1*
Stand. Dev.	0.37	0.095	2.0	7.1*
	Optimal R^2 Values			
	150°C	**175°C**	**200°C**	**250°C**
30 min	0.93	0.921	0.981	0.04^{+}
60 min	0.934	0.927	0.969	-0.19^{+}
90 min	-	0.164^{+}	-1.39^{+}	-0.01^{+}
120 min	0.95	-	-	-

Following the calculation of the diffusivity at each temperature/time combination, the values were then fit to the Arrhenius diffusivity equation (Equation 19) using MATLAB's Curve Fitting app. In this case, "R" represents the ideal gas constant (1.987E-3 kcal/K-mol), "E" corresponds to the activation energy for a diffusion event (kcal/mol), and "D_o" is the diffusivity at infinite temperature.

$$\boldsymbol{D(T) = D_o e^{\left(\frac{-E}{RT}\right)}} \qquad \textbf{(19)}$$

The average diffusivities at 150°C, 175°C, and 200°C were the data points used during the fitting process, with the diffusivity value at 250°C being excluded due to poor fits. The estimated diffusivity curve based on the data is shown in Figure 25, along with the estimated diffusion parameters and R^2 value. Based on the fitted equation, the "E" and "D_o" terms were determined to

be 20.84 kcal/mol and 1.406E10 nm^2/s, respectively with $R^2 = 0.86$. In terms of eV, the activation energy correlates with approximately 0.90 eV/atom.

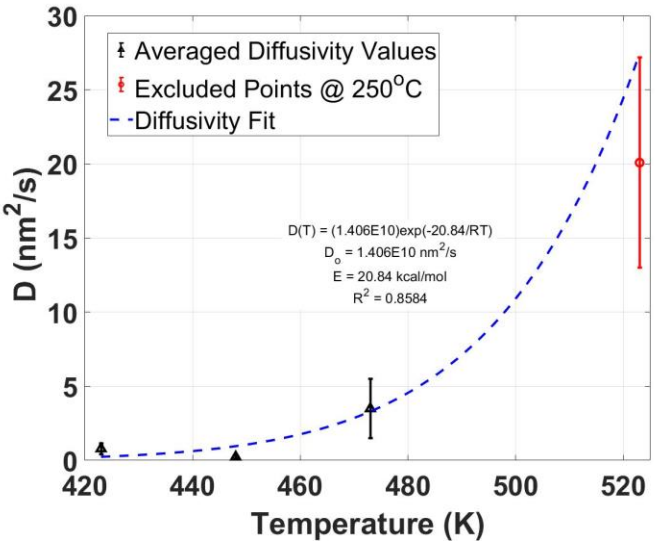


Figure 25: Fitting the diffusion data with an Arrhenius-type curve. The primary purpose is to illustrate the dependence of the diffusivity on temperature for this system. Data at 250°C was excluded from the fit, but the values were placed on the graph to observe any correlation with the fit.

As can be seen in the above figure, the averaged diffusivity values from the excluded 250°C trials were also plotted on the graph alongside the diffusivity curve. According to the fitted curve, the diffusivity of sodium into silica at 250°C should be approximately 27.46 nm^2/s, and the estimated value from the excluded data approximated the value to be 20.1 nm^2/s, a 27% relative difference. This comparison reinforced the efficacy of data exclusion in the edge cases, which provided a better approximation of the diffusivity at 250°C than the unmodified data (6.88 nm^2/s, 75% relative difference). By excluding the "trough region" of the concentration curves for these edge cases, the transition between transient and steady-state diffusion was better characterized and more representative of the material system.

At the beginning of the diffusivity analysis, the study by Frischat *et al.* provided the starting point for the model [68]. As stated earlier, the researchers studied a chemically similar system but varied structurally from the sol-gel silica indicated in this study. Within the range of 170°C to 250°C, similar to the range tested in this study, they suggested the diffusivity curve could be described by the following equation: $D(T) = 2.13\ \frac{nm^2}{s} e^{\left(\frac{-28.3\ kcal/mol}{RT}\right)}$. The activation energy (28.3 kcal/mol) corresponds to the energy requirement for a diffusion event to occur, which is probably either interstitial diffusion (smaller Na^+ ion diffuses in the space between silicon/oxygen atoms) or vacancy diffusion (Na^+ diffuses between vacancies in the atomic structure). In the case of sol-gel silica, another factor must be considered: porosity. Similar to diffusion along a grain boundary, sol-gel silica's intrinsic porosity provides the sodium atoms lower energy "shortcuts" to diffuse into the silica structure. Given the presence of porosity, it can be inferred that the diffusion of sodium into sol-gel silica would have a lower activation energy, i.e., the sodium atoms would diffuse easier. The fit from Figure 25 verifies this assumption by indicating the activation energy to be 20.84 kcal/mol, which is a 26% relative decrease from solid silica glass.

3.4. Summary

The numerical model introduced in the last chapter provided a good foundation for modeling sodium diffusion into sol-gel silica; however, it could not provide accurate results without knowledge of how diffusivity changes with respect to temperature. By recreating the sol-gel silica shell on a rectilinear geometry with accurate temperature control, the diffusion of sodium ions into a sol-gel silica thin film provided the basis for estimating the temperature dependence of diffusivity for this material system. XPS depth profiling accurately measured sodium concentration gradients at a variety of temperatures and times, where an estimated diffusivity was determined for each condition. By fitting the diffusivity values to the Arrhenius diffusion model,

the diffusion constants (D_o and E) were empirically determined to be, 1.406E10 nm^2/s and 20.84 kcal/mol, respectively. Knowledge of these parameters will aid in harnessing the power of diffusion to estimate nanoscale temperature during induction heating.

4. NANOPARTICLE TEMPERATURE ESTIMATION

The central thesis of this work focuses on diffusion occurring within a single core-shell nanoparticle for the purpose of estimating nanoscale temperature. Prior chapters have explained the steps required to achieve this goal. Developing a numerical model comprised the first step, which allowed for the simulation of diffusion in spherical coordinates; however, the model was incomplete without empirical diffusivity data. The next chapter concentrated on determining this empirical data by estimating the diffusivity vs temperature curve for Na diffusing into sol-gel silica. Herein, this chapter seeks to utilize all the tools and data developed thus far to provide an estimation of nanoparticle temperature using diffusion phenomena within a single core-shell nanoparticle.

4.1. Nanoparticle Fabrication and Characterization

Throughout this study, the nanoparticles in question consisted of a crystalline iron core and an amorphous sol-gel silica shell. In order to perform the diffusion analysis, it is necessary to ensure the existence of a conformal silica shell surrounding the nanoparticle core. The silica shell needs to be completely coating the core, while also maintaining a constant thickness to ensure uniform diffusion into the nanoparticle. The crystalline phase of the ferrous core will also be determined to provide a complete characterization of the particle.

4.1.1. SCNP Synthesis

The synthesis methodology used to create the silica capped nanoparticles, or SCNPs, was derived from a combination of two separate works: one for the fabrication of the iron cores [74] and the other for the silica shells [75]. All synthesis was performed in the Huitink Lab, while the SCNP characterization took place in the Arkansas Nano-Bio Materials Characterization Facility. The chemicals used were ACS grade and acquired from commercial sources. An illustration of the

synthesis process can be seen in Figure 26, which shows the creation of both the iron core and silica shell.

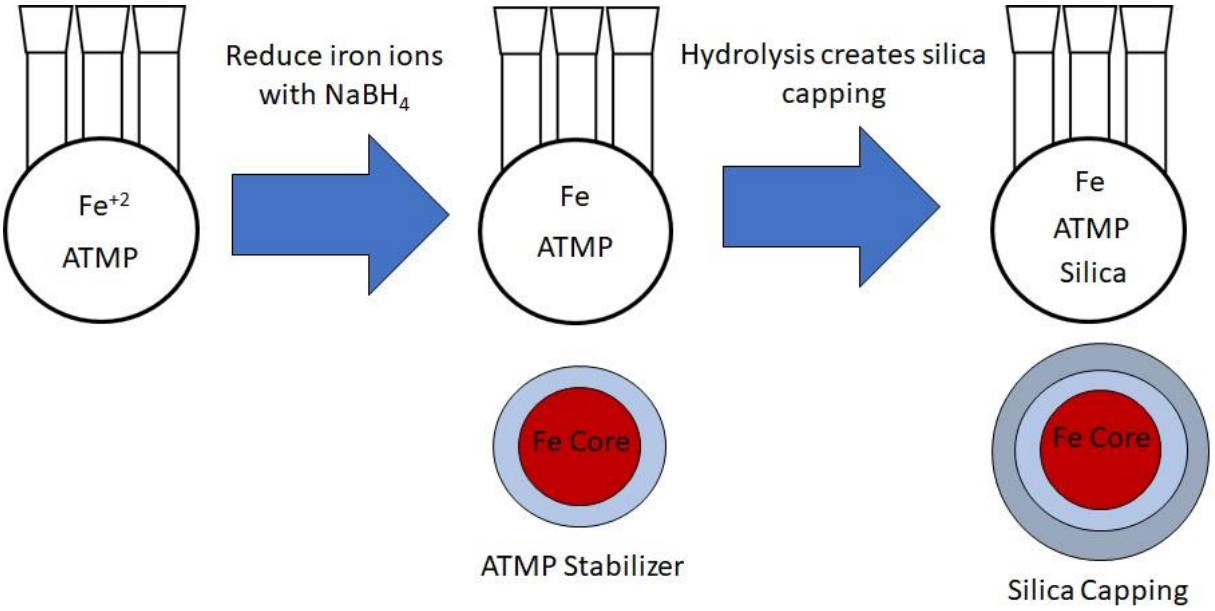


Figure 26: Diagram of the synthesis method used to make the SCNPs. First, the iron salt is added with solvent and ATMP. Under vacuum conditions, the iron ions are reduced with sodium borohydride to form iron core/ATMP stabilized nanoparticles. Silica precursors are then bubbled in the solution with argon to induce a hydrolysis reaction, which creates the silica shells.

Fabrication of the iron cores consisted of a methodology created by Greenlee *et al.* [74], which utilizes aminotris(methylenephosphonic acid), or ATMP to stabilize the nanoparticles. First, 23 µL of ATMP was added to a 3-neck flask containing 90 mL of deionized water and 40 mL of methanol. To ensure the removal of dissolved oxygen, the mixture was bubbled with argon gas and mixed using an orbital shaker table at 100 rpm for 10 minutes. In a separate flask, 17 mg of $FeSO_4 \cdot 7H_2O$ was added to 10 mL of deionized water and stirred to ensure it completely dissolves. While still maintaining the argon bubbling, the iron solution was immediately added to the 3-neck flask. The argon bubbling prevented unwanted oxidation of the iron within the aqueous suspension.

If the color of the solution turned a faint yellow, this indicated the solution had oxidized, and the synthesis must start over. The iron solution continued to bubble with argon for 30 additional minutes while stirring at 100 rpm. A few minutes before stopping the bubbling, 120 mg of $NaBH_4$ was dissolved in 10 mL of deionized water in a separate flask and then promptly added dropwise to the 3-neck flask. Assuming the lack of oxidation, the solution in the 3-neck flask should begin to turn black due to the $NaBH_4$ reducing the iron ions in the solution and creating iron cores. After turning off the argon gas and adding the reducing agent, the 3-neck flask was put under vacuum for 30 minutes to remove the subsequent hydrogen gas from the solution, while the solution continued to stir. Next, the iron cores underwent silica capping similar to the method by Yang *et al.* [75]. Under argon bubbling, 28 µL of (3-Aminopropyl) trimethoxysilane (APS) and 338 µL of tetraethyloxysilane (TEOS) were added incrementally every 30 minutes for the next 3 hours. The final products were centrifuged, washed several times, and suspended in deionized water.

4.1.2. Particle Characterization: STEM/EDX

After synthesizing the particles, a diluted sample of the final product solution was placed on a lacey carbon electron microscopy grid for imaging with a High-Resolution Transmission Electron Microscope (HRTEM). The HRTEM used was an FEI Titan 80-300, which enabled adequate resolution to observe the nanoparticle structure as well as any crystalline structures. To start, the nanoparticles were analyzed in terms of their relative size, as shown in Figure 27. As shown, the nanoparticles appear to have a wide range of sizes between approximately 70-115 nm. The less-dense silica layer contrasted against the iron core well enough to discern a noticeable core-shell structure with an average shell thickness of 13 nm; this visual characterization bolsters the conclusion that the synthesis was a success.

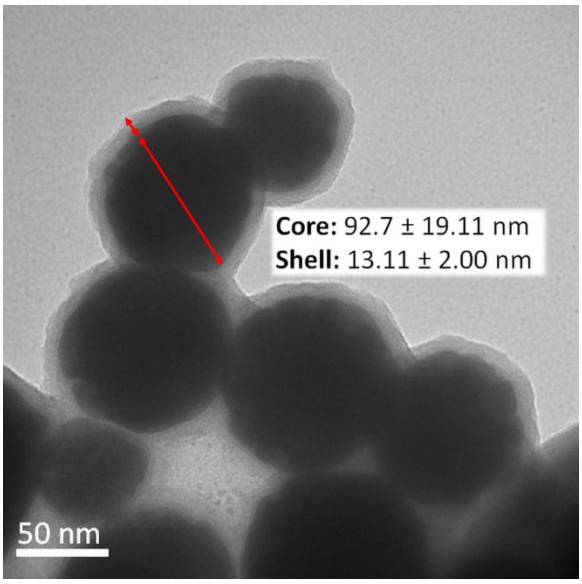


Figure 27: A grouping of SCNPs of various sizes at low magnification. The picture shows a wide variation in core sizes; however, even at low magnification, the conformal silica shell is easily distinguishable from the denser iron core.

Once a single particle was isolated, the HRTEM transitioned to Scanning TEM (STEM) mode, where the electron beam functioned more like a scanning electron microscope (SEM) rather than a TEM. In this mode, energy dispersive analysis (EDX) could be leveraged to not only discern physical characteristics, but also an elemental spectrum of the sample. Figure 28 illustrates a STEM scan of a grouping of particles and the results of a single line scan across the diameter of one particle. STEM mode portrays the sample with respect to the proton density (Z). In the left panel of Figure 28, the dense core appears much brighter than the surrounding shell, indicating the presence of heavier atoms. Subsequently, the shell appears dimmer, or in some cases invisible when compared against the bright cores, indicating a shell comprised of lighter atoms. When scanning across the diameter of a single particle, the atomic percentage of iron increases when the electron beam reaches the core, indicating a core rich in iron, and along the edges, the atomic percentage of silicon increases, which provides evidence of a silica shell. While EDX does well in

showing the elements that are present, it does not show the phase of the materials within the SCNPs. Specifically, the iron core could be any number of iron or iron oxide phases, which indicates that crystal analysis is necessary.

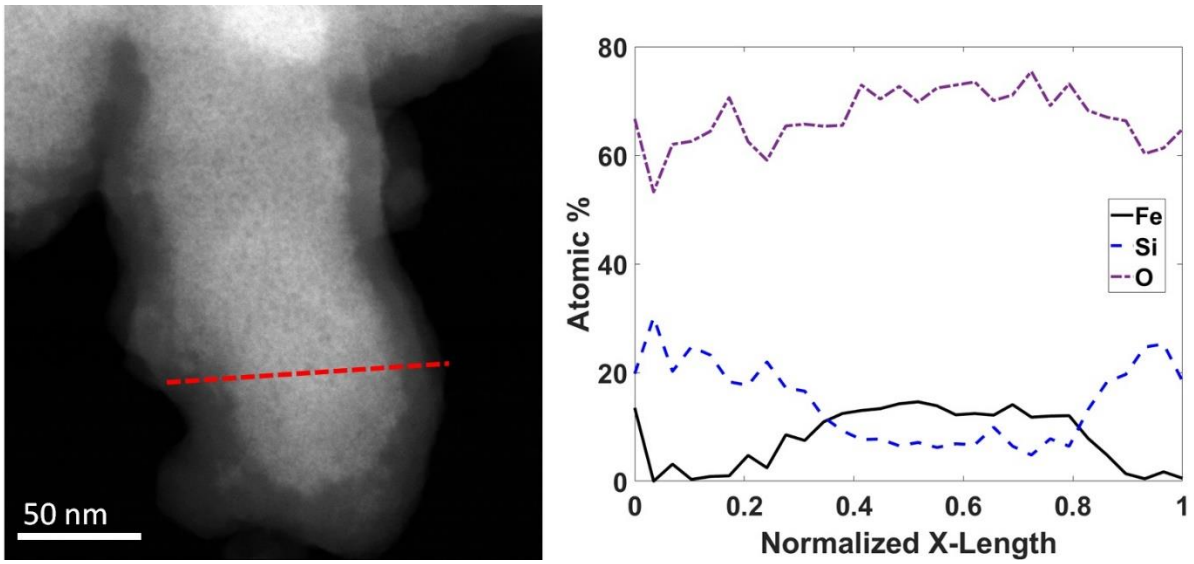


Figure 28: (left) STEM image of several SCNPS as well as the scanning direction; (right) An EDX plot of iron, silicon, and oxygen along the indicated scanning direction. As shown, the core of the sample appears to be rich in iron, indicating a ferrous core, while silicon has a higher intensity along the edges of the scan, indicating a silicon-rich shell.

4.1.3. Particle Characterization: Electron Diffraction

While EDX could not show crystalline phase, electron diffraction patterns can provide a glimpse into the crystal structure of a single particle. Essentially, similar to X-Ray diffraction, the diffraction of the electrons off the sample can be analyzed and used to predict interatomic distances and planar orientations. In this case, the HRTEM was focused at high magnification at the center of the core, and an image of the sample in Fourier space was generated, as shown in Figure 29.

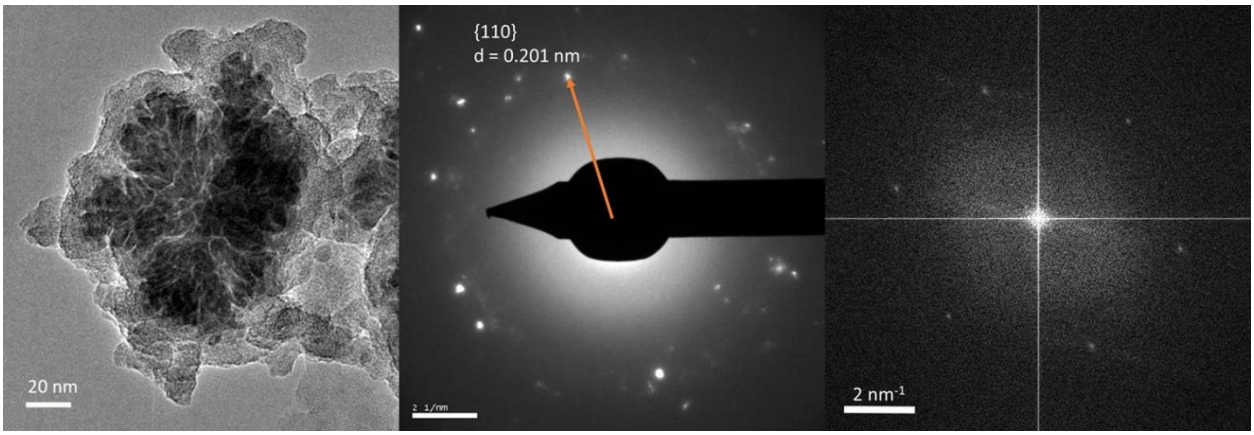


Figure 29: (left) A HRTEM image of the particle used for the electron diffraction analysis; (middle) A diffraction pattern at higher magnification from multiple particles indicating a similar hexagonal pattern; (right) The diffraction pattern of the particle in the left panel indicating a monocrystalline structure.

The presence of single bright points on the electron diffraction pattern indicates the presence of crystalline material within the nanoparticle(s). Further observation of the particle indicated that the repeating units came from the core rather than the shell, indicating that the shell was amorphous. The diffraction pattern seen in the middle image of Figure 29 was taken at a lower magnification over an area comprised of several individual particles; the pattern corroborates this by showing several discrete points, which form a ring around the central beam, indicating a multi-crystalline structure. As the magnification focuses on a single particle (right image of Figure 29) the diffraction pattern simplifies to a single pattern of six discrete points in a hexagonal shape, indicating a single crystalline structure. The distance between the center and each of the points correlates to an interplanar spacing within the crystalline structure. Using the scale provided with the Fourier image, several radial measurements were averaged and converted back to real space; the interplanar distance (d) corresponding to these points was approximately 0.201 nm. If this interplanar distance is converted to an x-ray diffraction peak, assuming an X-Ray wavelength of 0.154 nm, this measurement would correspond to a 2θ value of approximately 45.0°. When

comparing this interplanar distance to known phases of iron, it corresponds almost exactly with the principle peak at the (110) plane of the body-centered cubic (BCC) ferrite (Fe-α) phase, or elemental iron. Thus, this diffraction pattern provides definitive proof that the core present in the SCNPs is majority elemental iron and not iron oxide.

The hexagonal shape of the diffraction pattern originates from the orientation of the BCC crystal with the electron beam. There exist several permutations of the densely packed {110} family of planes within a single BCC unit cell; however, if the crystal is viewed along the {111} vector (isometric view of the unit cell), 3 planes can be seen within the lattice: (-101), (1-10), and (0-11), as shown in Figure 30. In a more general sense, the crystal can be viewed from any direction within the <111> family of vectors and still observe 3 different permutations of the {110} planar family, but this is merely for demonstration. The discrete points on the electron diffraction pattern are orthogonal to their respective planes, which are color-coded within the figure. Measuring these planes with the electron beam at this orientation produces the hexagonal electron diffraction pattern shown in the above image.

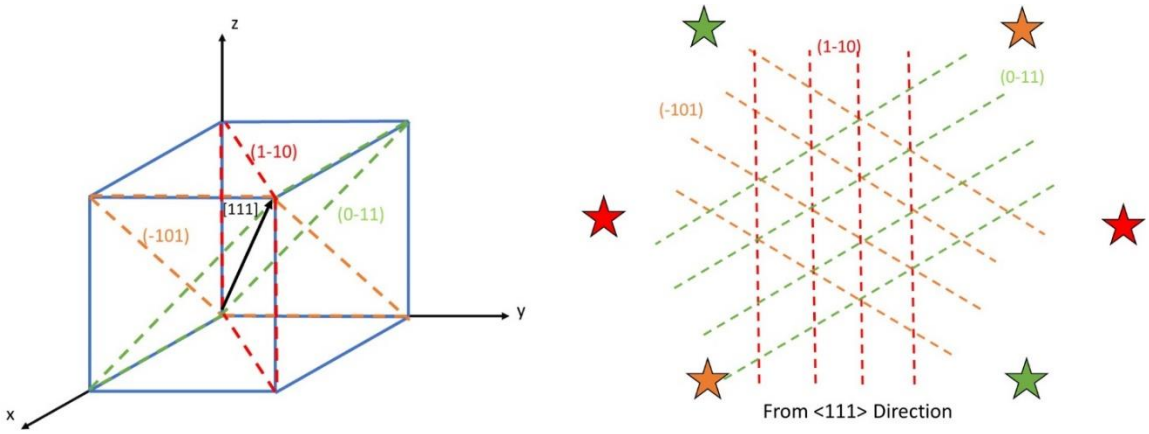


Figure 30: (left) A unit cell illustrating the 3 planes from the {110} family of planes visible along the [111] vector; (right) Diagram of these planes superimposed over each other. The stars represent the discrete points on the electron diffraction pattern, which are each orthogonal to the respective planes, thus creating the hexagonal pattern.

4.2. Thermal Output via Induction Heating

After synthesizing the nanoparticles and characterizing their structure, the final suspension of SCNPs underwent induction heating to characterize the macro-scale heat transfer occurring within the suspension. Essentially, by heating the suspension with an induction heater, the thermal output of the SCNPs can be characterized and calculated on a per unit mass basis. The total power output per unit mass is known commonly as the specific loss power (SLP) or the specific absorption rate (SAR). From this value, an estimate internal heat general per particle can be determined, using an averaged particle volume.

4.2.1. Induction Heating Setup

The thermal testing utilized an Ambrell LI 8310 10 kW induction heater with variable frequency between 150-400 kHz. Induction heating coil geometry consisted of a 7-turn coil with an inside and outside diameter of 25 mm and 44 mm, respectively. Two 1 μF capacitors in series coupled with the induction coil to produce a resonant frequency of 216 kHz. The power supply provided a current of 500 A to the coil, which, according to the datasheet provided by the manufacturer resulted in a magnetic field strength of approximately 100 kA/m (1256 Oe). The arrangement of the induction heating coil and the sample is shown in Figure 31.

In the setup, alternating current from the power supply is run through the coil to produce the intense alternating magnetic field necessary to heat nanoparticle suspensions. As a result, the coil itself undergoes significant resistive losses due to the high current being passed through the copper tubing. To combat this, filtered water from a nearby chiller is circulated through the center of the coil, throughout the transformer, and within the power supply. Regardless, the temperature of the coil is still heightened during the heating process, which in turn slightly heats the sample via radiation. To limit the amount of this background heat, circulating air lines act as an insulating

barrier between the coil and the sample. The alternating magnetic field also prohibits the usage of metallic or conductive materials within the confines of the coil; because of this, a polymeric sample holder is used, which will not interact with the magnetic field. In addition, a fiber optic thermocouple acts as the primary temperature measurement device rather than a metallic probe, so that the results are not skewed.

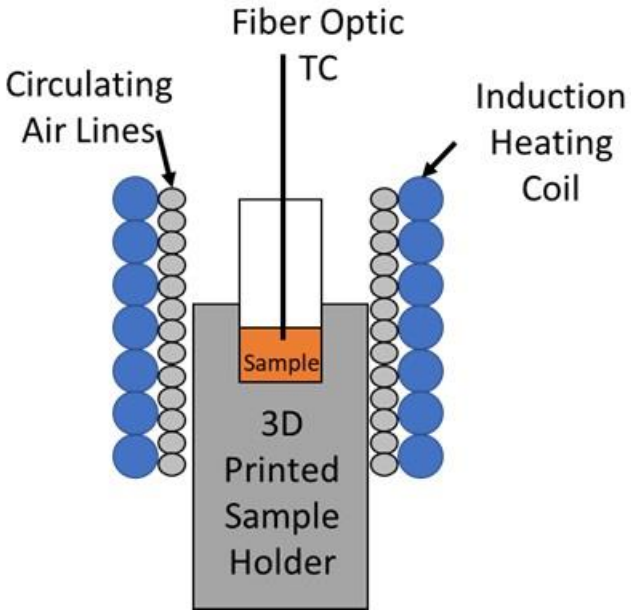


Figure 31: Cross-section of the induction heating setup. The heating coil provides the magnetic field; the sample is shielded from in radiated heat transfer via the sample holder and circulating air lines.

4.2.2. Sample Concentration

Minimal effort is required to prepare the samples for measurement; however, in order to calculate the heat produced (SAR) by the sample per unit mass, a precise measurement of sample concentration is necessary. A rough estimation could be attempted based on the synthesis methodology, but this method would not be able to account for sample loss during washing and centrifugation. As a more precise methodology, inductively coupled plasma mass spectrometry

(ICP-MS) provided the means to accurately determine the quantity of iron atoms within the solution. To start, a 2% nitric acid solution digested the particles present within a small sample of the suspension to create a solution comprised of iron atoms. The ICP-MS instrument then determined the concentration of iron atoms on a ppb basis, which can be equated back to mg/mL. Based on the ICP-MS analysis, the concentration of iron within the solution was approximately 0.32 mg/mL; however, this concentration was just the concentration of iron cores, not the silica shells. While the SCNPs contain both silica and iron, the iron comprises the majority of the particle size and weight, in addition to being the only aspect of the particle that interacts with the magnetic field. Thus, the value provided by the ICP-MS analysis was accepted as being a valid estimate of the total particle concentration for SAR analysis.

With the goal of the work being to differentiate between the temperature of the liquid and the temperature of the nanoparticle, testing a dilute colloidal suspension was necessary. Ideally, the nanoparticle suspension should be dilute enough to where the temperature gradients between neighboring nanoparticles do not intersect, i.e., single nanoparticles floating ins suspension, where the thermal energy release from one nanoparticle does not impact the particle next to it. A simple equation calculates the interparticle separation using the Wigner-Seitz radius, $d = 2\left(\frac{3}{4\pi n}\right)^{1/3}$, where "n" is the particle density, and the equation is doubled since the original equation calculates only half of the average inter-particle separation. Assuming uniform iron nanoparticles at a concentration of 0.32 mg/mL, the estimated inter-particle separation was calculated to be 2.7 μm. Since several references in the introduction indicated the temperature gradient to extend only a few nanometers from the nanoparticle surface, the concentration was considered low enough to be used for these experiments.

4.2.3. SAR Measurement

Using the described setup, the particle suspension underwent induction heating, while monitoring the temperature simultaneously. Each measurement consisted of heating the suspension for 1 minute while recording the sample temperature every 2 seconds. This method was repeated 3 times to ensure statistical significance, and the first 30 seconds were fit linearly to determine the initial slope of the temperature vs time plot (Figure 32).

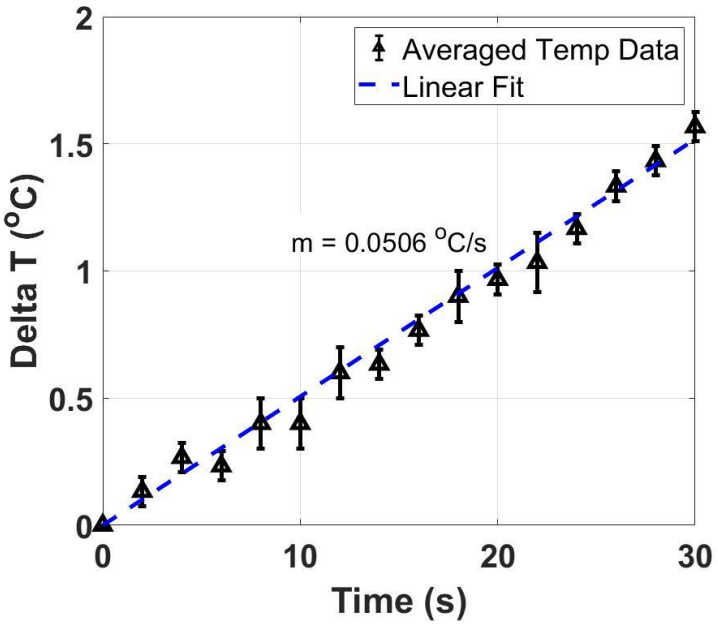


Figure 32: Heating curve of SCNP suspension. After averaging three separate measurements, the data was fit linearly to determine the temperature change with respect to time for SAR calculation.

As implied by the measurement, the SAR was calculated via the initial slope method used commonly in practice and represented in Equation 20, where "m" is the weight fraction of SCNPs (related to concentration), "c_p" is the specific heat of the suspended liquid (4.182 J/g°C) for water in this case), and dT/dt correlates to the initial slope measured with the thermocouple (0.0506 °C/s) [76].

$$SAR\ \left(\frac{W}{g}\right) = \frac{1}{m} c_p \frac{dT}{dt} \qquad (20)$$

After inputting the initial slop corresponding to the SCNPS, the resulting SAR can be seen below in Equation 21.

$$SAR_{SCNPs}\ \left(\frac{W}{g}\right) = \left(\frac{1}{0.00032}\right)\left(4.182\,\frac{J}{g°C}\right)\left(0.0506\,\frac{℃}{s}\right) = 661.3\,\frac{W}{g} \qquad (21)$$

Due to its simplicity, the initial slope method operates under a few key assumptions. The first being that the sample is perfectly insulated from exterior heat sources or sinks, and the second is that the suspension behaves like a lumped mass with a uniform temperature. Essentially, the ideal measurement environment would be a uniformly dispersed nanoparticle system that is perfectly insulated from the surroundings. Within the short time frame, losses to the room surroundings are unlikely, and with the presence of the circulating air barrier, external heating from the coil is minimized to drastically affect the result. Regarding nanoparticle dispersal, the nanoparticle sample underwent sonication for 15 minutes followed by 1 minute of vortex mixing immediately before each measurement. Because of the experimental setup, the initial slope method provides a valid estimation of the nanoparticle SAR, which was used to directly compare to the nanoparticle temperature measurement.

4.3. Nanoparticle Temperature Estimation via Diffusion

With the macroscale rate of heat transfer for the SCNP suspension determined, the next step involves leveraging the numerical model and knowledge regarding Na diffusion into sol-gel silica to estimate the temperature of the nanoparticles during the heating trial. By comparing the

concentration gradient across the nanoparticle before and after heating, the numerical model can be used to estimate the temperature required to cause the gradient to change.

4.3.1. STEM/EDX Measurements

For determining nanoscale temperature, two different nanoparticle samples were created from the synthesized batch: control and heated sample (Table 4). The control sample served as the means to illustrate the initial distribution of Na within the SCNPs shell without any heating. Knowledge of this initial distribution was used as an initial condition for the model for temperature estimation. Creating the control consisted of dispersing the nanoparticles in a 0.15 M NaCl aqueous solution, sonicating the sample, and then thoroughly mixing the sample. Immediately after mixing, a small sample of the suspension was used to make a TEM sample. The second sample underwent the same preparation as the control sample; however, the suspension was immediately heated with the induction heater at 100 kA/m for 10 seconds. Afterward, a TEM sample was created from the heated sample.

Table 4: Outlining the two samples analyzed with EDX

	Control	**Heated**
SCNP Conc. (mg/mL)	0.32	0.32
Saline Conc. (M)	0.15	0.15
Mag Field Strength (kA/m)	0	100
Time Heated (s)	0	10
r_i (shell inner radius, nm)	40	44
r_o (shell outer radius, nm)	62	70

For measuring the Na concentration gradient within the SCNPs, the STEM/EDX methodology discussed earlier in the chapter was utilized. Each TEM sample was analyzed with the HRTEM to find a single core-shell particle. After finding one, the mode switched to STEM,

which measured the elemental cross-section of the selected nanoparticle from each sample (Figure 33).

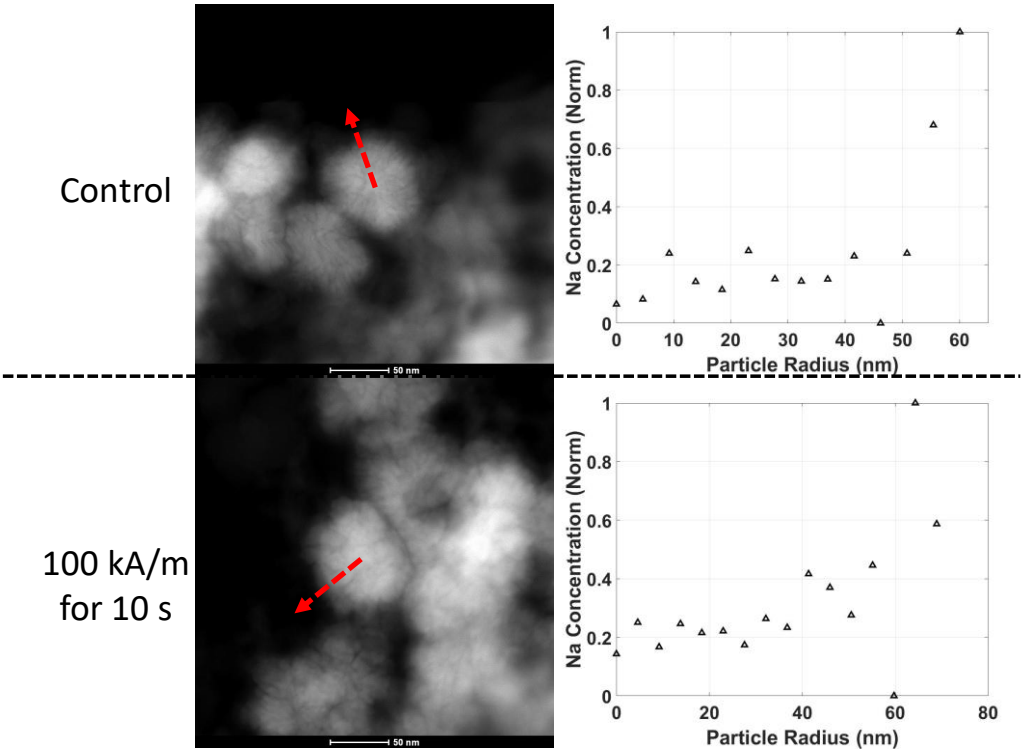


Figure 33: Raw STEM/EDX data from the control (top) and heated (bottom) sample. The data measures the atomic percentage of Na across the entire particle radius.

While the size range of the SCNPs varies by ~20 nm either way, the chosen particles were intended to be similar in size, so that they could be more easily compared. By visual observation of the data sets, the control sample Na concentration curve declines steeply the further the beam measures into the core, while the heated sample concentration curve appears to be a more gradual decline, with a heightened concentration near the center, indicating that some mass transfer may have occurred within the particles. The control measurement provides evidence that the Na atoms adsorb onto the silica shell easily, and that the initial concentration gradient is quite high. The heated sample possessed some possible outliers around 60 nm, which may be due to areas with a

non-uniform distribution of Na atoms on the particle surface. Outliers will be discussed in subsequent sections.

4.3.2. Temperature Estimation: MATLAB Pseudocode

The algorithm to estimate temperature possesses similar qualities to the previously discussed algorithm to estimate diffusivity (Figure 22). Basically, the algorithm selects a temperature range and solves the numerical model at each temperature until R^2 between the experimental data and model is maximized (Figure 34).

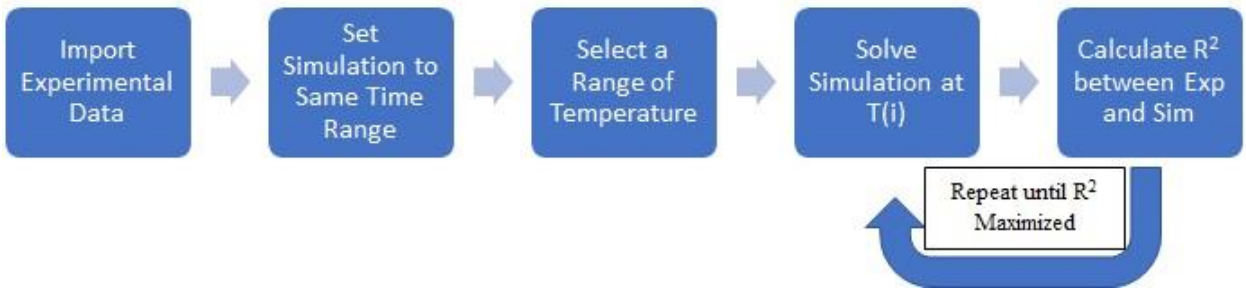


Figure 34: Algorithm to estimate nanoparticle temperature using both theoretical and experimental data. The theoretical model is solved at each temperature within the range until the best fit of the experimental data is determined.

The below abbreviated MATLAB pseudocode outlines the important aspects of the algorithm to estimate nanoparticle temperature; the complete body of code can be found in the Appendix.

```
>> import('Experimental data');      %Import in single concentration curve
>> radius = Imported array containing particle radii;
>> intensity = Imported array containing concentration curve;
>> temp = 400:1:500;          %Define array of temperatures (K)
>> R_squared = zeros(length(temp),1); %Create matrix to store R_squared values
>> rnew = zeros(length(radius),1); %Nearest diffusion sim radius values
```

```
>> cnew = zeros(length(radius),length(temp)); %Nearest diffusion sim conc values
>> for k = 1:length(temp)      %Cycle over temperatures (K)
        >> D = 1.406E10*exp(-20.84/(1.987E-3*temp(k))); %Calculate diffusivity from exp data
        >> totalTime = 10;      %Total simulation time, equal to 10 second heating time
        >> u = zeros(R,T);      %Create simplified "u" matrix
        >> u(:,1) = input initial condition;    %Define initial condition (control test)
        >> Solve CN algorithm for spherical geometry with same radius as measured particle;
        >> Simulate effects of EDX projection and store in matrix called "c_avg"
        >> for i = 1:length(radius)    %Map experimental concentration gradient to model
                >> rnew(l) = interpl(r,r,radius(l),'nearest');          %Find nearest model value
                >> index = find(r==rnew(l));          %Find index for that value
                >> cnew(l,k) = c_avg(index,T);        %Create of matrix of mapped values
        >> end
        >> data_mean = mean(intensity);
        >> sum_res = 0;
        >> sum_sq = 0;
        >> j = 1:length(radius)
                >> sum_res = sum_res + (intensity(m) - cnew(m,k))^2;       %Sum of residuals
                >> sum_sq = sum_sq + (intensity(m) - data_mean)^2;         %Sum of squares
        >> end
        >> R_squared(k) = 1- sum_res/sum_sq;       %Calculate R Squared
>> maxRSquared = find(R_squared==max(R_squared));   %Find max R Squared Value
>> temp_opt = temp(maxRSquared) %Store temperature value with highest R Squared Value
```

>> R_2 = max(R_squared) *%Store R Squared value of optimal temperature value.*

4.3.3. Initial Condition Approximation

As shown in the pseudocode, the initial condition must be defined within the model in order to get an accurate result. In this case, the initial condition represents the distribution of Na within the silica shell before induction heating (the Control sample). In order for the model to work properly, the experimental data from the Control (or representative fit) would need to be input into the first column of the "u" solution matrix, similar to how the spatial boundary conditions are defined. This proved difficult, due to how the model is solved. Specifically, after setting the initial condition at the particle center, the model outputs a solution, and that solution undergoes correction to account for EDX projection (Chapter 2). The "initial condition" data from the Control sample measured by the EDX is already projected data, which must be converted back to non-projected data to be utilized by the model. In order to accomplish this, the un-projected initial condition function needs to be extracted from the EDX data gathered from the Control sample.

The base function for the initial condition was decided to model a power function, where the "scale" and "shift" terms were varied (Equation 22). Within the silica shell, the concentration gradient will be represented by the top function, but within the core, the initial concentration is assumed zero.

$$\begin{cases} c = scale^{(r - r_{max})} + shift & r \geq r_i \\ c = 0 & r < r_i \end{cases} \qquad \textbf{(22)}$$

An algorithm was developed to determine which shift and scale parameters resulted in an initial condition (after projection) that fit the Control data best (see Appendix). After cycling

through several scale and shift combinations, the function $c = 1.26^{(r-r_{max})} + 0.034$ for the concentration within the silica layer resulted in the best fit with an $R^2 = 0.87$ (Figure 35).

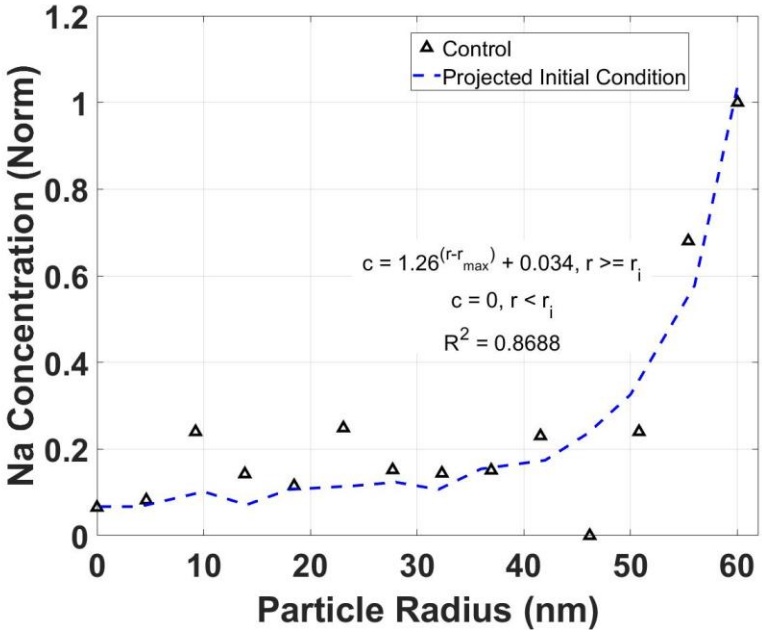


Figure 35: Proposed initial condition after projection fit to the control data.

The initial condition function determined by the algorithm served as the starting point for the model; essentially what the concentration gradient is at room temperature. One of the key assumptions in using this initial condition is that all future changes to the concentration curve are assumed to be a direct result of the input of thermal energy (induction heating). In the case of the temperature estimation algorithm, inputting the initial condition function can be seen below. As the line of code indicates, the function must be multiplied by the array of radii in order to be input into the simplified "u" matrix.

*>> u(:,1) = (1.26.^(r-max(r))+0.034).*r;*

4.3.4. Estimating Nanoparticle Temperature

Using the algorithm discussed earlier in the section to calculate nanoparticle temperature, the experimental data for the Heated sample was uploaded to MATLAB and analyzed. After setting the simulation time to 10 seconds and ensuring the selected radii values matched those measured with the HRTEM, the algorithm worked to find the temperature which best fit the Na concentration curve present in the Heated sample. As shown in the top panel of Figure 36, the algorithm estimated the local nanoparticle temperature to be approximately 459 K or 186°C during the 10-second heating trial; however, the fit seemed to be skewed namely by two outlier points, highlighted in red in Figure 36. Given the small size range of the EDX measurements, these outliers could originate from an unseen, non-uniform aggregate of Na ions on the surface of the silica layer, resulting in a localized spike, or it could simply be a measurement artefact. For investigative purposes, the same sample was analyzed without the two outlier points and re-normalized with respect to the Na concentration at the edge of the shell (~68 nm), as shown in the bottom panel of Figure 36. Removing the outlier data points resulted in a higher temperature (215°C) and a better fit ($R^2 = 0.78$).

The temperature estimation derived from the diffusion model suggests that the nanoscale temperature exceeds that of the surrounding fluid by well over 150°C, at a field intensity of 100 kA/m over a time of 10 seconds. Given the nature of the measurements, this estimation likely comes with a certain amount of error, originating from inhomogeneities within the sample on the nanoscale, as well as STEM/EDX measurement error. However, the fits indicate a noticeable difference between the Na concentration curves of the Control and Heated samples. With such a high-temperature difference between a single particle and the surrounding fluid, it begs the

question as to whether traditional macro-scale thermal modeling can properly characterize the heat transfer occurring within the particle suspension.

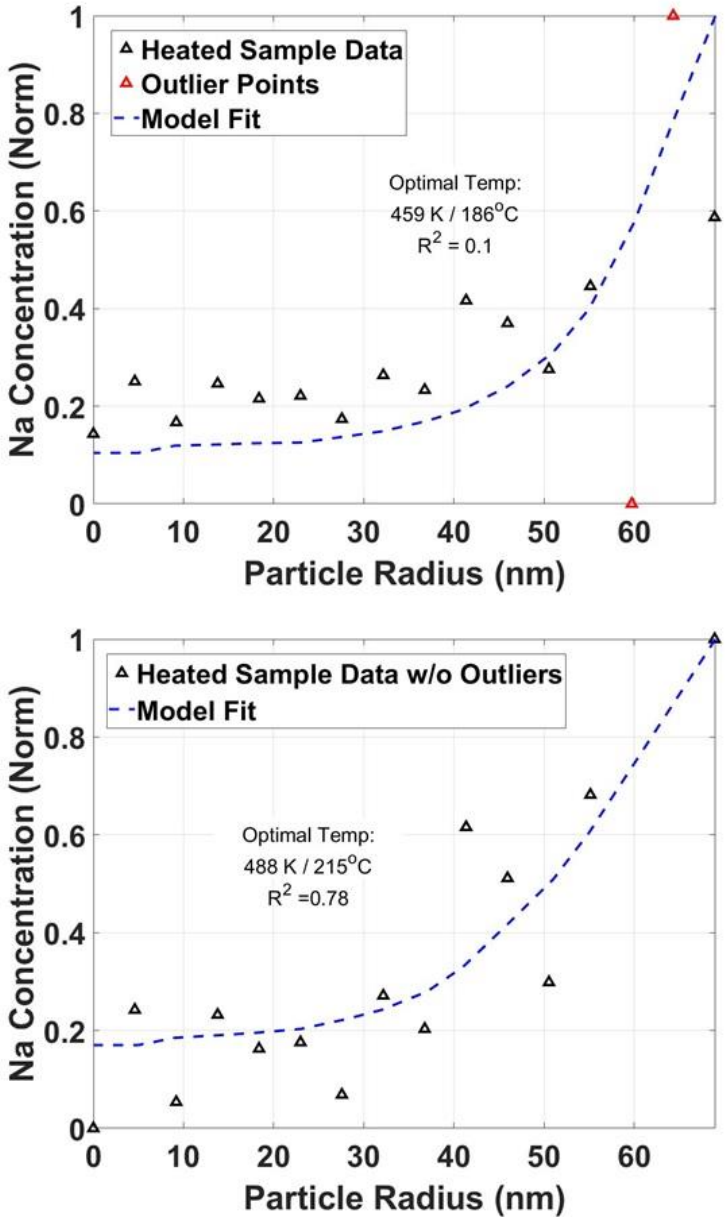


Figure 36: Sodium concentration curves of the Heated sample (10 seconds at 100 kA/m intensity) with fit from the algorithm. (top) The temperature which resulted in the best fit to the raw data was 459 K or 186°C (R^2 = 0.1). The poor fit likely originated from the presence of the two denoted points that measured extremely high/low concentration values; (bottom) The same sample fit without the outlier points, resulting in an improved fit (R^2 = 0.78) and a heightened temperature estimation (218°C).

4.4. Heat Transfer Modelling

When deriving the diffusion model in Chapter 2, the primary assumption was that the concentration of sodium within an individual particle varied spatially as well as temporally. EDX data corroborates this assumption. The highest concentrations of sodium tended to be towards the surface of the particle, which created a concentration gradient within the core-shell structure, and when comparing the Control and Heated sample, this gradient also varied with respect to heating time. Conversely, the particle temperature assumes a model which only possesses temporal variance (no spatial terms), known colloquially as the lumped-capacitance temperature model. Essentially, in order to calculate diffusivity, the user must model the particle as a uniform temperature with no thermal gradients within the particle.

Typically, a simple calculation of the Biot number would verify the validity of the lumped-capacitance assumption, where if the Biot number is less than 0.1, the lumped-capacitance model proves valid. Equation 23 estimates the Biot number for a single SCNP, where "h" is the heat transfer coefficient, "r" is the particle radius, and "k" is the thermal conductivity (assumed to be solid ferrite (iron). The heat transfer coefficient is assumed to be 1 $W/m^2°C$ for simplicity.

$$\boldsymbol{Bi_{sphere} = \frac{hr}{3k} = \frac{\left(1\frac{W}{m^2C}\right)(60E-9\ m)}{(3)\left(80\frac{W}{mC}\right)} = 2.5E-10 \ll 0.1} \qquad \textbf{(23)}$$

The Biot number calculation indicates that the lumped-capacitance assumption is valid for a model of a single nanoparticle. Due to the incredibly small size range, the radius term of the Biot number equation completely washes out the effects of the heat transfer coefficient and material thermal conductivity. Basically, even with an extremely efficient convective boundary at the particle surface and an extremely un-conductive particle material, the particle is so small that any

thermal gradient within the particle would be too small to disseminate. Yet, the heat transfer coefficient is the one aspect of Equation 23 which remains unknown. Heat transfer coefficients are traditionally estimated empirically or determined through advanced CFD simulations; however, in this case, the physical properties of the nanoparticles, particle temperature, fluid temperature, and internal heat generation are all known parameters, which allows for the heat transfer coefficient to be estimated using the lumped-capacitance model.

As illustrated in Figure 37, the entire particle was modeled as a homogenous temperature (T_p) with internal heat generation (E_{gen}, from the induction heating coil) and a convective boundary condition (suspended in water at temperature T_s). With regards to material properties, the entire particle was assumed to be comprised of iron, and the radius was set at 60 nm.

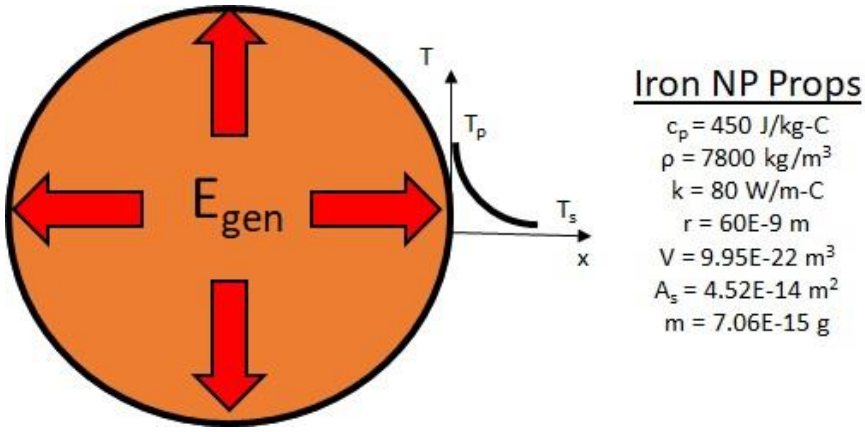


Figure 37: Illustration of the lumped-capacitance model for a single SCNP. The core and shell exist at a constant temperature which varies depending on the internal heat generation (E_{gen}), material properties, and convective heat removal.

Early in this chapter, the SAR for the solution was determined calorimetrically via the initial slope method. Measuring the change in water temperature over time aided in quantifying

the total thermal power produced by the nanoparticles, which, by knowing the particle size, was used to extrapolate the internal heat generation, as shown in Equation 24.

$$E_{gen} = \frac{SAR*m}{V} = \frac{\left(661.3\frac{W}{g}\right)*(7.06E-15\ g)}{(9.95E-22\ m^3)} = 4.69E9\ \frac{W}{m^3} \qquad (24)$$

The lumped capacitance model starts with the conservation of energy (Equations 25-27), which was modified to include aspects of the nanoparticle system, where $\theta = (T_p - T_s)$.

$$Q_{in} = Q_{out} + Q_{stored} \qquad (25)$$

$$E_{gen}V = hA_s\theta + mc_p\frac{d\theta}{dt} \qquad (26)$$

$$\frac{d\theta}{dt} + \frac{hA_s}{mc_p}\theta = \frac{E_{gen}V}{mc_p} \qquad (27)$$

The conservation of energy quickly evolves into a simple linear ordinary differential (Equation 27), which can be solved easily (Equation 28), assuming a zero initial condition ($\theta(0)=0$), i.e., the particle and fluid are the same temperature at the start.

$$\theta(t) = \frac{E_{gen}V}{hA_s}\left(1 - e^{-\left(\frac{hA_s}{mc_p}t\right)}\right) \qquad (28)$$

After inputting material parameters, the function becomes Equation 29, which varies with respect to heat transfer coefficient and time.

$$\theta(t) = \frac{103.242}{h}\left(1 - e^{-\left(\frac{ht}{0.0703}\right)}\right) \qquad (29)$$

In the SAR tests, the suspensions always started near room temperature (22°C), and given that the diffusion model estimated the particle temperature to be approximately 186°C, θ(10s) = 164°C, resulting in the following heat transfer coefficient (Equation 30).

$$\mathbf{164℃} = \frac{\mathbf{103.242}}{\boldsymbol{h}}\left(\mathbf{1} - \boldsymbol{e}^{-\left(\frac{\boldsymbol{h}(\mathbf{10}\boldsymbol{s})}{\mathbf{0.0703}}\right)}\right) \therefore \boldsymbol{h} = \mathbf{0.63}\ \frac{\boldsymbol{W}}{\boldsymbol{m}^{\mathbf{2}}℃} \qquad \mathbf{(30)}$$

After incorporating the calculated heat transfer coefficient into the lumped-capacitance model, a temperature plot with respect to time was generated to show the theoretical temperature response of the particle (Figure 38). As the temperature plot indicates, the particle approaches steady state quickly (< 500 ms), due to its overall small thermal mass.

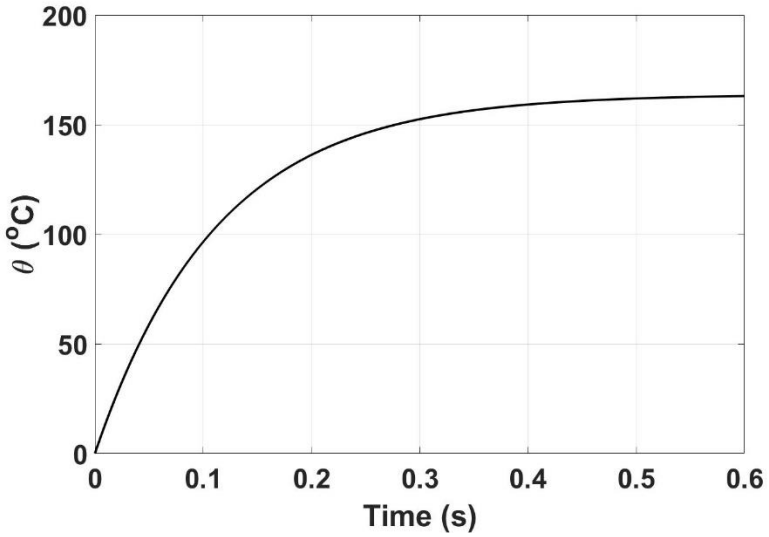


Figure 38: Estimated temperature over time of a single SCNP using the lumped capacitance model

4.5. Convection on the Nanoscale: Brief Discussion

The existence of such a large temperature gradient between the particle core and the surrounding fluid suggests largely inefficient convective heat removal. Using the lumped-capacitance assumption, a 164°C temperature difference indicates that the heat transfer coefficient between the particle and the surrounding fluid was approximately 0.63 W/m^2°C during induction heating, which is certainly not a very high value considering forced liquid convection solutions can attain values on the order of kW/m^2°C. Originally, the heat transfer coefficient derives from Newton's law of cooling, which functions as a "catch-all" term for convection effectiveness. Several different aspects influence the heat transfer coefficient of a given system, such as geometry and fluid velocity, and much of the properties depend on the continuum assumption (fluid is treated as continuous rather than comprised of discrete molecules). Typically, the criteria for the continuum assumption stems from the Knudsen number, which relates the size of the object in question to the mean free path within the fluid/gas ($Kn = \frac{\lambda}{L}$), where "λ" represents the mean free path and "L" is some characteristic length of the system. As long as Kn remains below 0.01, the continuum assumption remains valid. Given that liquids consist of loosely bound molecules, the distance between molecules will serve as the mean free path for liquid water (~0.3 nm) [77]. After choosing the diameter of an SCNP (120 nm) as "L", Kn = 0.0025, which implies that the continuum assumption still holds true, even for such a small system. Verifying the continuum assumption further bolsters the usage of the lumped-capacitance model with a convective boundary condition; however, it does not provide much evidence regarding the inefficiency of the convection at the particle surface.

When considering the surface of a nanoparticle in an ionic solution, such as the human body, several works indicate the existence of an electric double layer (EDL) in the immediate area

around a nanoparticle. Basically, the surface of a single nanoparticle becomes charged while in suspension, resulting in electrostatic attraction of oppositely charged ions within the vicinity; these ions congregate at the nanoparticle surface, creating a compact layer of ions. After the first layer formation, a diffuse second layer is ultimately formed, which possesses a much looser attraction to the nanoparticle in question [78, 79]. The Gouy-Chapman-Stern model of the EDL is among the more widely cited models in this field of study, which works to define the size of each of these layers [80]. Consider applying this model of the EDL to the suspension of SCNPs (Figure 39); given that EDX data shows a high concentration of Na^+ ions near the surface, let's infer that the surface charge of the silica shell is largely negative. Assuming a negative charge and the presence of dissolved Na^+ ions in the solution, the compact first layer should be comprised of hydrated Na^+ ions followed by a diffuse second layer of both hydrated Na^+ and Cl^- ions.

By visual observation of the layered structure, one could surmise that the EDL acts as a nanoscale insulating layer. The thermal conductivity of water compared to that of iron or silica is relatively low, and the static nature of the hydrated compact layer could dampen the benefit of natural convection in the direct vicinity of the silica shell. Basically, it is suggested that the EDL prevents the bulk fluid from coming in direct contact with the actual nanoparticle surface, so the thermal energy must first pass through the first layer via conduction before convection can take over. A few references in the literature corroborate this assumption when analyzing the effect of the EDL on microchannel cooling applications [81, 82]; the references highlight that the EDL constrains the flow and results in a decrease in the Nusselt number, which describes the effectiveness of convection. Also, there exists the concern of nanoscale vapor bubbles surrounding the nanoparticles during heating, which could further dampen convection by creating a layer of resistive vapor around the particle. A complete convection analysis at the SCNP/liquid boundary

exceeds the scope of this dissertation; however, further investigation pertaining to nanoscale convection would result in a more holistic understanding of nanoparticle induction heating.

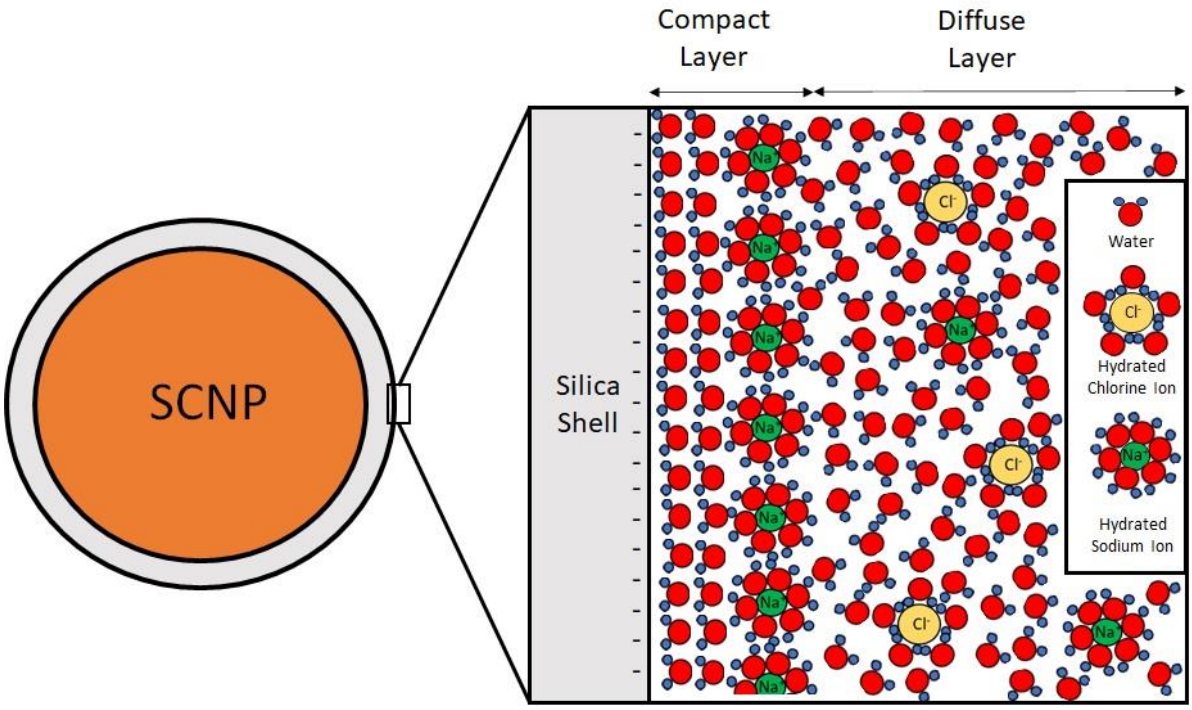


Figure 39: Illustration of the electric double layer at the liquid boundary of the SCNPs, according to the Gouy-Chapman-Stern model. Close to the silica shell exists the compact layer consisting of immobile Na ions which have adsorbed onto the surface of the silica shell, followed by a diffuse outer layer. (Image derived from [78])

5. CONCLUSION

5.1. Summary of Results

The work started by creating a robust model of a spherical core-shell nanoparticle, which served as the basis of the diffusion study. Utilizing the Crank-Nicolson algorithm and MATLAB IDE, a numerical model of radial diffusion in a single particle was implemented. The model, while unconditionally stable, proved to be sensitive to both time and space discretization. Using a dimensionless parameter, an established range for spatial and temporal discretization limited perturbations in the data. A plotting function was also developed to visualize diffusion in both 1 and 2D. In order to correlate the numerical simulation results with the experimental diffusion data, an additional function was developed to project the diffusion of a 3D particle onto a 2D plane. (Chapter 2)

The next aspect of the study established the diffusivity vs temperature relation for Na diffusing into sol-gel silica. Polished silicon wafers underwent sol-gel silica coating in order to mimic the coating on the SCNPs. After characterizing the film, a temperature-controlled oven heated various coated samples with a Na-rich layer located at the surface to encourage diffusion into the layer. Using XPS depth profiling, the concentration gradient with respect to layer depth was extracted in order to determine the diffusivity values at various temperatures. An Arrhenius relation fit the diffusivity data with respect to temperature in order to be used in the numerical simulation; the fit resulted in an estimated activation energy and diffusivity at infinite temperature of 20.84 kcal/mol (0.90 eV/atom) and 1.406E10 nm^2/s, respectively. (Chapter 3)

The next chapter served as the culmination of this study, which combined both theory and experiment covered in the previous two chapters to estimate nanoparticle temperature. Using a room-temperature synthesis method, iron/silica core/shell nanoparticles were fabricated for use in

this study. EDX and diffraction characterization techniques verified the existence of an iron-dense ferrite core with a silica shell, which matched the developed theoretical model introduced in Chapter 2. After analyzing their internal heat generation calorimetrically, the change in sodium concentration within the silica shell was used to extract an estimated nanoscale temperature using the diffusivity and temperature relations determined in Chapter 3. At a magnetic field strength of 100 kA/m over a period of 10 s, the change in concentration gradient correlated with a nanoscale temperature of approximately 186°C. Afterward, a lumped-capacitance transient temperature model estimated the heat transfer coefficient at the convective boundary condition to be approximately 0.63 $W/m^2°C$, which is much lower than typical values for water. (Chapter 4)

5.2. Literature Comparison and Future Outlook

Nanoscale heat transfer during nanoparticle heating (induction, plasmonic, etc.) is a new area of study, which assumes that the local nanoparticle temperature differs from the observed temperature on the macro scale. The work presented herein introduces a novel method of analyzing local nanoparticle temperature during induction heating from the perspective of nanoscale interactions with the surrounding fluid suspension. Based on the literature review in Chapter 1, this method describes a single-particle method of thermometry. Single-particle methods offer the greatest accuracy in temperature determination due to their proximity to the particle; however, when comparing the results gathered in this work to those published in the literature, the temperature gradient observed in the SCNPs appeared to be much higher than the other studied particle systems, as shown in Table 5.

Table 5: Comparison of the results of this study to various single-particle induction heating thermometry results from the literature

Particle Core	Method of Thermometry	Core Size (nm)	Field Strength/Frequency	Measured Temperature Gradient	Ref
α-Fe	Diffusion Gradient	100	100 kA/m @ 216 kHz	164°C	-
γ-Fe_2O_3	Optical	23	18.3 kA/m @ 97 kHz	3°C	[50]
$MnFe_2O_4$	Optical	6	0.66 kA/m @ 40 MHz	5°C	[57]
Fe_xO_y	Optical	15	13 kA/m @ 335 kHz	45°C	[63]

The above table provides 3 separate examples from the literature with both large and small temperature gradients. Primarily, the differences between this work and the previous efforts are the nanoparticle structure and magnetic field properties. Elemental iron (ferrite) comprises the cores of the SCNPs, which tends to have a much higher saturation magnetization than iron oxide. Also, the SCNPs are the largest in the de at 100 nm. As stated in the introduction, particle thermal output tends to increase with both saturation magnetization and particle size. That combined with a high magnetic field intensity (100 kA/m), compared to the next highest of 18.3 kA/m, explains why this diffusion study measured such a high thermal gradient; yet the selection of these parameters had a purpose. Using nanoparticles with heightened magnetic properties and large inter-particle distances (low concentrations) helped to induce as large of a thermal gradient as possible between the particle core and surrounding fluid, and an exorbitantly large magnetic field strength further enhanced the transient thermal response. For actual applications, like magnetic nanoparticle hyperthermia, the magnetic field strength will most likely be much lower; however, for the purpose of this study, a high magnetic field strength proved to be beneficial to showcase the potentially large thermal gradients occurring on the nanoscale.

The aspiration for this work is to encourage future studies involving nanoscale thermal transport and to search for creative applications for this novel heat transfer phenomenon. Several

aspects of the particles, surrounding fluid, and AMF all impact thermal output, yet the literature is scarce. Understanding temperature gradients between nanoparticles and surrounding medium would provide enhanced insight into the mechanisms which drive conduction and convection on the nanoscale. Such knowledge would allow for the improvement of heat transfer models to include nanoscale effects and allow for nanomaterials to be engineered to maximize heat transfer to the fluid medium, which would prove invaluable to several biomedical applications.

6. REFERENCES

[1] Brown, G. H., Hoyler, C. N., and Bierwirth, R. A., 1947, Theory and Application of Radio-Frequency Heating, Van Nostrand, New York City, NY, USA.

[2] Haimbaugh, R. E., 2001, Practical Induction Heating, ASM International, Materials Park, OH.

[3] Martin, D. L., and Wiley, F. E., 1945, "INDUCTION HARDENING OF PLAIN CARBON STEELS - A STUDY OF THE EFFECT OF TEMPERATURE, COMPOSITION, AND PRIOR STRUCTURE ON THE HARDNESS AND STRUCTURE AFTER HARDENING," Transactions of the American Society for Metals, 34, pp. 351-406.

[4] Acero, J., Burdio, J. M., Barragan, L. A., Navarro, D., Alonso, R., Garcia, J. R., Monterde, F., Hernandez, P., Llorente, S., and Garde, I., 2010, "DOMESTIC INDUCTION APPLIANCES," Ieee Industry Applications Magazine, 16(2), pp. 39-47.

[5] Hedayatnasab, Z., Abnisa, F., and Daud, W., 2017, "Review on magnetic nanoparticles for magnetic nanofluid hyperthermia application," Materials & Design, 123, pp. 174-196.

[6] Ali, A., Zafar, H., Zia, M., Haq, I. U., Phull, A. R., Ali, J. S., and Hussain, A., 2016, "Synthesis, characterization, applications, and challenges of iron oxide nanoparticles," Nanotechnology Science and Applications, 9, pp. 49-67.

[7] Khot, V. M., Salunkhe, A. B., Thorat, N. D., Ningthoujam, R. S., and Pawar, S. H., 2013, "Induction heating studies of dextran coated MgFe2O4 nanoparticles for magnetic hyperthermia," Dalton Transactions, 42(4), pp. 1249-1258.

[8] Dennis, C. L., and Ivkov, R., 2013, "Physics of heat generation using magnetic nanoparticles for hyperthermia," International Journal of Hyperthermia, 29(8), pp. 715-729.

[9] Kolhatkar, A. G., Jamison, A. C., Litvinov, D., Willson, R. C., and Lee, T. R., 2013, "Tuning the Magnetic Properties of Nanoparticles," International Journal of Molecular Sciences, 14(8), pp. 15977-16009.

[10] Bedanta, S., Barman, A., Kleemann, W., Petracic, O., and Seki, T., 2013, "Magnetic Nanoparticles: A Subject for Both Fundamental Research and Applications," Journal of Nanomaterials, 2013, p. 22.

[11] Dennis, C. L., Jackson, A. J., Borchers, J. A., Gruettner, C., and Ivkov, R., 2018, "Correlation between physical structure and magnetic anisotropy of a magnetic nanoparticle colloid," Nanotechnology, 29(21), p. 9.

[12] Noh, S. H., Na, W., Jang, J. T., Lee, J. H., Lee, E. J., Moon, S. H., Lim, Y., Shin, J. S., and Cheon, J., 2012, "Nanoscale Magnetism Control via Surface and Exchange Anisotropy for Optimized Ferrimagnetic Hysteresis," Nano Letters, 12(7), pp. 3716-3721.

[13] Jeong, U., Teng, X. W., Wang, Y., Yang, H., and Xia, Y. N., 2007, "Superparamagnetic colloids: Controlled synthesis and niche applications," Advanced Materials, 19(1), pp. 33-60.

[14] Brown, W. F., 1963, "THERMAL FLUCTUATIONS OF A SINGLE-DOMAIN PARTICLE," Physical Review, 130(5), pp. 1677-+.

[15] Tompkins, J., and Huitink, D., 2020, "Induction heating response of iron oxide nanoparticles in varyingly viscous mediums with prediction of brownian heating contribution," Nanoscale and Microscale Thermophysical Engineering, 24(3-4), pp. 123-137.

[16] Tackett, R. J., Thakur, J., Mosher, N., Perkins-Harbin, E., Kumon, R. E., Wang, L. H., Rablau, C., and Vaishnava, P. P., 2015, "A method for measuring the Neel relaxation time in a frozen ferrofluid," Journal of Applied Physics, 118(6), p. 5.

[17] Neel, L., 1949, "Theorie du trainage magnetique des ferromagnetiques en grains fins avec applications aux terres cuites," Annals of Geophysiques, 5, pp. 99-136.

[18] Abenojar, E. C., Wickramasinghe, S., Bas-Concepcion, J., and Samia, A. C. S., 2016, "Structural effects on the magnetic hyperthermia properties of iron oxide nanoparticles," Progress in Natural Science-Materials International, 26(5), pp. 440-448.

[19] Carlton, H., Xu, S., Hong, M. N., Begishev, I., and Huitink, D., 2018, "TX-100 capped iron oxide nanoparticle transformation and implications for induction heating and hyperthermia treatment," Journal of Nanoparticle Research, 20(9), p. 13.

[20] Carlton, H., Krycka, K., Bleuel, M., and Huitink, D., 2020, "In Situ Dimensional Characterization of Magnetic Nanoparticle Clusters during Induction Heating," Particle & Particle Systems Characterization, 37(1), p. 10.

[21] Carlton, H., 2019, "The Role of Inter-Particle Behavior in Iron Oxide Nanoparticle Induction Heating," University of Arkansas, ProQuest.

[22] Nemati, Z., Alonso, J., Rodrigo, I., Das, R., Garaio, E., Garcia, J. A., Orue, I., Phan, M. H., and Srikanth, H., 2018, "Improving the Heating Efficiency of Iron Oxide Nanoparticles by Tuning Their Shape and Size," Journal of Physical Chemistry C, 122(4), pp. 2367-2381.

[23] Mohapatra, J., Zeng, F. H., Elkins, K., Xing, M. Y., Ghimire, M., Yoon, S., Mishra, S. R., and Liu, J. P., 2018, "Size-dependent magnetic and inductive heating properties of Fe3O4 nanoparticles: scaling laws across the superparamagnetic size," Physical Chemistry Chemical Physics, 20(18), pp. 12879-12887.

[24] Tong, S., Quinto, C. A., Zhang, L. L., Mohindra, P., and Bao, G., 2017, "Size-Dependent Heating of Magnetic Iron Oxide Nanoparticles," Acs Nano, 11(7), pp. 6808-6816.

[25] Filippousi, M., Angelakeris, M., Katsikini, M., Paloura, E., Efthimiopoulos, I., Wang, Y. J., Zambouis, D., and Van Tendeloo, G., 2014, "Surfactant Effects on the Structural and Magnetic Properties of Iron Oxide Nanoparticles," Journal of Physical Chemistry C, 118(29), pp. 16209-16217.

[26] Liu, X. L., Fan, H. M., Yi, J. B., Yang, Y., Choo, E. S. G., Xue, J. M., Di Fan, D., and Ding, J., 2012, "Optimization of surface coating on Fe3O4 nanoparticles for high performance magnetic hyperthermia agents," Journal of Materials Chemistry, 22(17), pp. 8235-8244.

[27] Woodard, L. E., Dennis, C. L., Borchers, J. A., Attaluri, A., Velarde, E., Dawidczyk, C., Searson, P. C., Pomper, M. G., and Ivkov, R., 2018, "Nanoparticle architecture preserves magnetic properties during coating to enable robust multi-modal functionality," Scientific Reports, 8, p. 13.

[28] Smolkova, I. S., Kazantseva, N. E., Babayan, V., Smolka, P., Parmar, H., Vilcakova, J., Schneeweiss, O., and Pizurova, N., 2015, "Alternating magnetic field energy absorption in the dispersion of iron oxide nanoparticles in a viscous medium," Journal of Magnetism and Magnetic Materials, 374, pp. 508-515.

[29] Suto, M., Hirota, Y., Mamiya, H., Fujita, A., Kasuya, R., Tohji, K., and Jeyadevan, B., 2009, "Heat dissipation mechanism of magnetite nanoparticles in magnetic fluid hyperthermia," Journal of Magnetism and Magnetic Materials, 321(10), pp. 1493-1496.

[30] Lee, J. H., Jang, J. T., Choi, J. S., Moon, S. H., Noh, S. H., Kim, J. W., Kim, J. G., Kim, I. S., Park, K. I., and Cheon, J., 2011, "Exchange-coupled magnetic nanoparticles for efficient heat induction," Nature Nanotechnology, 6(7), pp. 418-422.

[31] Jang, J. T., Nah, H., Lee, J. H., Moon, S. H., Kim, M. G., and Cheon, J., 2009, "Critical Enhancements of MRI Contrast and Hyperthermic Effects by Dopant-Controlled Magnetic Nanoparticles," Angewandte Chemie-International Edition, 48(7), pp. 1234-1238.

[32] Pilati, V., Gomide, G., Gomes, R. C., Goya, G. F., and Depeyrot, J., 2021, "Colloidal Stability and Concentration Effects on Nanoparticle Heat Delivery for Magnetic Fluid Hyperthermia," Langmuir, 37(3), pp. 1129-1140.

[33] Pearce, J., Giustini, A., Stigliano, R., and Hoopes, J. P., 2014, "Magnetic Heating of Nanoparticles: The Importance of Particle Clustering to Achieve Therapeutic Temperatures," Journal of Nanotechnology in Engineering and Medicine, 4.

[34] Branquinho, L. C., Carriao, M. S., Costa, A. S., Zufelato, N., Sousa, M. H., Miotto, R., Ivkov, R., and Bakuzis, A. F., 2013, "Effect of magnetic dipolar interactions on nanoparticle heating efficiency: Implications for cancer hyperthermia," Scientific Reports, 3, p. 10.

[35] Fu, R., Yan, Y. Y., Roberts, C., Liu, Z. Y., and Chen, Y. Y., 2018, "The role of dipole interactions in hyperthermia heating colloidal clusters of densely-packed superparamagnetic nanoparticles," Scientific Reports, 8, p. 10.

[36] Rajan, A., and Sahu, N. K., 2020, "Review on magnetic nanoparticle-mediated hyperthermia for cancer therapy," Journal of Nanoparticle Research, 22(11), p. 25.

[37] Raouf, I., Khalid, S., Khan, A., Lee, J., Kim, H. S., and Kim, M. H., 2020, "A review on numerical modeling for magnetic nanoparticle hyperthermia: Progress and challenges," Journal of Thermal Biology, 91, p. 12.

[38] Oei, A. L., Korangath, P., Mulka, K., Helenius, M., Coulter, J. B., Stewart, J., Velarde, E., Crezee, J., Simons, B., Stalpers, L. J. A., Kok, H. P., Gabrielson, K., Franken, N. A. P., and Ivkov, R., 2019, "Enhancing the abscopal effect of radiation and immune checkpoint inhibitor therapies with magnetic nanoparticle hyperthermia in a model of metastatic breast cancer," International Journal of Hyperthermia, 36, pp. 47-63.

[39] Korangath, P., Barnett, J. D., Sharma, A., Henderson, E. T., Stewart, J., Yu, S. H., Kandala, S. K., Yang, C. T., Caserto, J. S., Hedayati, M., Armstrong, T. D., Jaffee, E., Gruettner, C., Zhou, X. C., Fu, W., Hu, C., Sukumar, S., Simons, B. W., and Ivkov, R., 2020, "Nanoparticle interactions with immune cells dominate tumor retention and induce T cell-mediated tumor suppression in models of breast cancer," Science Advances, 6(13).

[40] Chiu-Lam, A., and Rinaldi, C., 2016, "Nanoscale Thermal Phenomena in the Vicinity of Magnetic Nanoparticles in Alternating Magnetic Fields," Advanced Functional Materials, 26(22), pp. 3933-3941.

[41] Rabin, Y., 2002, "Is intracellular hyperthermia superior to extracellular hyperthermia in the thermal sense?," International Journal of Hyperthermia, 18(3), pp. 194-202.

[42] Keblinski, P., Cahill, D. G., Bodapati, A., Sullivan, C. R., and Taton, T. A., 2006, "Limits of localized heating by electromagnetically excited nanoparticles," Journal of Applied Physics, 100(5).

[43] Merabia, S., Shenogin, S., Joly, L., Keblinski, P., and Barrat, J. L., 2009, "Heat transfer from nanoparticles: A corresponding state analysis," Proceedings of the National Academy of Sciences of the United States of America, 106(36), pp. 15113-15118.

[44] Rajabpour, A., Seif, R., Arabha, S., Heyhat, M. M., Merabia, S., and Hassanali, A., 2019, "Thermal transport at a nanoparticle-water interface: A molecular dynamics and continuum modeling study," Journal of Chemical Physics, 150(11).

[45] Lombard, J., Biben, T., and Merabia, S., 2015, "Nanobubbles around plasmonic nanoparticles: Thermodynamic analysis," Physical Review E, 91(4), p. 12.

[46] Lapotko, D., 2009, "Optical excitation and detection of vapor bubbles around plasmonic nanoparticles," Optics Express, 17(4), pp. 2538-2556.

[47] Lukianova-Hleb, E., Hu, Y., Latterini, L., Tarpani, L., Lee, S., Drezek, R. A., Hafner, J. H., and Lapotko, D. O., 2010, "Plasmonic Nanobubbles as Transient Vapor Nanobubbles Generated around Plasmonic Nanoparticles," Acs Nano, 4(4), pp. 2109-2123.

[48] Lukianova-Hleb, E. Y., and Lapotko, D. O., 2012, "Experimental techniques for imaging and measuring transient vapor nanobubbles," Applied Physics Letters, 101(26), p. 5.

[49] Munoz-Menendez, C., Serantes, D., Chubykalo-Fesenko, O., Ruta, S., Hovorka, O., Nieves, P., Livesey, K. L., Baldomir, D., and Chantrell, R., 2020, "Disentangling local heat contributions in interacting magnetic nanoparticles," Physical Review B, 102(21).

[50] Pinol, R., Brites, C. D. S., Bustamante, R., Martinez, A., Silva, N. J. O., Murillo, J. L., Cases, R., Carrey, J., Estepa, C., Sosa, C., Palacio, F., Carlos, L. D., and Millan, A., 2015, "Joining Time-Resolved Thermometry and Magnetic-Induced Heating in a Single Nanoparticle Unveils Intriguing Thermal Properties," Acs Nano, 9(3), pp. 3134-3142.

[51] Gupta, A., Kane, R. S., and Borca-Tasciuc, D. A., 2010, "Local temperature measurement in the vicinity of electromagnetically heated magnetite and gold nanoparticles," Journal of Applied Physics, 108(6), p. 7.

[52] Maestro, L. M., Haro-Gonzalez, P., Coello, J. G., and Jaque, D., 2012, "Absorption efficiency of gold nanorods determined by quantum dot fluorescence thermometry," Applied Physics Letters, 100(20).

[53] Kucsko, G., Maurer, P. C., Yao, N. Y., Kubo, M., Noh, H. J., Lo, P. K., Park, H., and Lukin, M. D., 2013, "Nanometre-scale thermometry in a living cell," Nature, 500(7460), pp. 54-U71.

[54] Carlson, M. T., Khan, A., and Richardson, H. H., 2011, "Local Temperature Determination of Optically Excited Nanoparticles and Nanodots," Nano Letters, 11(3), pp. 1061-1069.

[55] Alaulamie, A. A., Baral, S., Johnson, S. C., and Richardson, H. H., 2017, "Targeted Nanoparticle Thermometry: A Method to Measure Local Temperature at the Nanoscale Point Where Water Vapor Nucleation Occurs," Small, 13(1).

[56] Dong, J. Y., and Zink, J. I., 2014, "Taking the Temperature of the Interiors of Magnetically Heated Nanoparticles," Acs Nano, 8(5), pp. 5199-5207.

[57] Huang, H., Delikanli, S., Zeng, H., Ferkey, D. M., and Pralle, A., 2010, "Remote control of ion channels and neurons through magnetic-field heating of nanoparticles," Nature Nanotechnology, 5(8), pp. 602-606.

[58] Lin, F. C., and Zink, J. I., 2020, "Probing the Local Nanoscale Heating Mechanism of a Magnetic Core in Mesoporous Silica Drug-Delivery Nanoparticles Using Fluorescence Depolarization," Journal of the American Chemical Society, 142(11), pp. 5212-5220.

[59] Freddi, S., Sironi, L., D'Antuono, R., Morone, D., Dona, A., Cabrini, E., D'Alfonso, L., Collini, M., Pallavicini, P., Baldi, G., Maggioni, D., and Chirico, G., 2013, "A Molecular Thermometer for Nanoparticles for Optical Hyperthermia," Nano Letters, 13(5), pp. 2004-2010.

[60] Ebrahimi, S., Akhlaghi, Y., Kompany-Zareh, M., and Rinnan, A., 2014, "Nucleic Acid Based Fluorescent Nanothermometers," Acs Nano, 8(10), pp. 10372-10382.

[61] Carattino, A., Caldarola, M., and Orrit, M., 2018, "Gold Nanoparticles as Absolute Nanothermometers," Nano Letters, 18(2), pp. 874-880.

[62] Holub, M., Adobes-Vidal, M., Frutiger, A., Gschwend, P. M., Pratsinis, S. E., and Momotenko, D., 2020, "Single-Nanoparticle Thermometry with a Nanopipette," Acs Nano, 14(6), pp. 7358-7369.

[63] Riedinger, A., Guardia, P., Curcio, A., Garcia, M. A., Cingolani, R., Manna, L., and Pellegrino, T., 2013, "Subnanometer Local Temperature Probing and Remotely Controlled Drug Release Based on Azo-Functionalized Iron Oxide Nanoparticles," Nano Letters, 13(6), pp. 2399-2406.

[64] Crank, J., 1975, The Mathematics of Diffusion, Oxford University Press, Ely House, London.

[65] Crank, J., and Nicolson, P., 1947, "A PRACTICAL METHOD FOR NUMERICAL EVALUATION OF SOLUTIONS OF PARTIAL DIFFERENTIAL EQUATIONS OF THE HEAT-CONDUCTION TYPE," Proceedings of the Cambridge Philosophical Society, 43(1), pp. 50-67.

[66] Garrard, K., 2021, "3D Polar Plot,"MATLAB Central File Exchange.

[67] Held, J. T., Hunter, K. I., Dahod, N., Greenberg, B., Hickey, D. R., Tisdale, W. A., Kortshagen, U., and Mkhoyan, K. A., 2018, "Obtaining Structural Parameters from STEM-EDX Maps of Core/Shell Nanocrystals for Optoelectronics," Acs Applied Nano Materials, 1(2), pp. 989-996.

[68] Frischat, G. H., 1968, "SODIUM DIFFUSION IN SIO2 GLASS," Journal of the American Ceramic Society, 51(9), pp. 528-+.

[69] Lu, Y. F., Ganguli, R., Drewien, C. A., Anderson, M. T., Brinker, C. J., Gong, W. L., Guo, Y. X., Soyez, H., Dunn, B., Huang, M. H., and Zink, J. I., 1997, "Continuous formation of supported cubic and hexagonal mesoporous films by sol gel dip-coating," Nature, 389(6649), pp. 364-368.

[70] Artaki, I., Bradley, M., Zerda, T. W., and Jonas, J., 1985, "NMR AND RAMAN-STUDY OF THE HYDROLYSIS REACTION IN SOL-GEL PROCESSES," Journal of Physical Chemistry, 89(20), pp. 4399-4404.

[71] Sheridan, F., 1995, "Characterisation and Optimisation of Sol-Gel-Derived Thin Films for Use in Optical Sensing," Master of Science, Dublin City University, Dublin, Ireland.

[72] Chen, S. L., Dong, P., Yang, G. H., and Yang, J. J., 1996, "Kinetics of formation of monodisperse colloidal silica particles through the hydrolysis and condensation of tetraethylorthosilicate," Industrial & Engineering Chemistry Research, 35(12), pp. 4487-4493.

[73] McDonagh, C., Sheridan, F., Butler, T., and MacCraith, B. D., 1996, "Characterisation of sol-gel-derived silica films," Journal of Non-Crystalline Solids, 194(1-2), pp. 72-77.

[74] Greenlee, L. F., and Rentz, N. S., 2014, "ATMP-stabilized iron nanoparticles: chelator-controlled nanoparticle synthesis," Journal of Nanoparticle Research, 16(11), p. 16.

[75] Yang, T. I., Brown, R. N. C., Ckempel, L., and Kofinas, P., 2011, "Controlled synthesis of core-shell iron-silica nanoparticles and their magneto-dielectric properties in polymer composites," Nanotechnology, 22(10), p. 8.

[76] Andreu, I., and Natividad, E., 2013, "Accuracy of available methods for quantifying the heat power generation of nanoparticles for magnetic hyperthermia," International Journal of Hyperthermia, 29(8), pp. 739-751.

[77] Sinha, N., and Antony, B., 2021, "Mean Free Paths and Cross Sections for Electron Scattering from Liquid Water," Journal of Physical Chemistry B, 125(21), pp. 5479-5488.

[78] Goel, A., 2018, "Structure of the Electrical Double Layer at the Silica Nanoparticle-Electrolyte Water Interface," Doctor of Sciences, ETH Zurich.

[79] Brown, M. A., Abbas, Z., Kleibert, A., Green, R. G., Goel, A., May, S., and Squires, T. M., 2016, "Determination of Surface Potential and Electrical Double-Layer Structure at the Aqueous Electrolyte-Nanoparticle Interface," Physical Review X, 6(1).

[80] Stern, O., 1924, "The theory of the electrolytic double-layer," Z. Elektrochem, 30(508), pp. 1014-1020.

[81] Tardu, S., 2004, "The electric double layer effect on the microchannel flow stability and heat transfer," Superlattices and Microstructures, 35(3-6), pp. 513-529.

[82] Zhao, Q. K., Xu, H., and Tao, L. B., 2021, "Two-layer nanofluid flow and heat transfer in a horizontal microchannel with electric double layer effects and magnetic field," International Journal of Numerical Methods for Heat & Fluid Flow, 31(7), pp. 2347-2372.

APPENDIX

Source Code: CN Algorithm, Plotting, and Animation for Core-Shell Particles

```
%% Fick's second law of diffusion numerically for core-shell nanoparticle
%% Diffusivity %%

%Define particle temperature (K)
 T = 500;

%Determine diffusivity (nm^2/s) based on experiments
D = 1.406E10*exp(-20.84/(1.987E-3*T));

%% Discretize the problem %%

%Radius (nm)
ri = 45; %core radius (nm)
ro = 60; %core+shell radius (nm)
delr = 1; %radius discretization
r = ri:delr:ro; %matrix of radius steps
R = length(r); %total number of radius steps

%Time (s)
totalTime = 100; %total simulation time
delt = (1.5*delr^2)/D; %time discretization
t = 0:delt:totalTime; %matrix of time steps
T = length(t); %total number of time steps

%% Create Solution Matrices %%

%Create solution matrix (at%)
c = zeros(R,T);

%Create simplified "u" matrix (u=c*r)
u = zeros(R,T);

%% Initial Condition %%

%Define initial condition
u(:,1) = 0;

%% BC at outer shell (r = ro) %%

%Diriclet B.C.
u(R,:) = ro;

%% Solve Crank Nicolson Implicitly (A x u = d) %%

%Define nondimensional constant "C"
C = D*delt/(delr)^2;

%Create tridiagonal matrix "A"
A = diag((2*(1/C+1))*ones(R-2,1)) + diag(-1*ones(R-3,1),1) + ...
```

```
    diag(-1*ones(R-3,1),-1);

%Account for insulated boundary condition at r=ri
A(1,1) = 2*(1/C+1)-(4*r(1))/(3*r(2));
A(1,2) = r(1)/(3*r(3))-1;

%Create solution matrix "d"
d = zeros(R-2,1);

for j = 2:T %Perform solution iteration for given number of time steps
    for i = R-2:-1:1 %Build d vector for current time step
        if i == R-2
            %Dirichlet BC at outer shell
            d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+2*u(i+2,j-1);
        else
            d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+u(i+2,j-1);
        end
    end
    u(2:R-1,j) = A\d; %Solve for intermediate points
    u(1,j) = (4*r(1))/(3*r(2))*u(2,j)-r(1)/(3*r(3))*u(3,j); %Neumann B.C.
end
c(:,:) = u./repmat(r',1,T);

%% Plot your results at each time step %%
h1 = figure();
for n = 1:T
    if n == T
        plot(r,c(:,n));
        hold off
    elseif n == 1
        plot(r,c(:,n));
        hold on
    else
        plot(r,c(:,n));
    end
end

%% Create video of diffusion at center cross section %%
%Include entire radius of particle, including the core
rplot = [0:delr:r(1),r(2:R)];
rplotl = length(rplot);

%Assume rotational symmetry, plot with respect to theta for quarter sphere
thetaplot = 0:(pi/2)/(length(rplot)-1):pi/2;
thetaplotl = length(thetaplot);

%Create plotting matrix: a giant matrix which includes all
%(theta,r,c) values for each time step
cplot = zeros(rplotl,thetaplotl,T);

%Make the figure and make it invisible
h2 = figure();
h2.Visible = 'off';

%Initialize video
```

```
centerCross = VideoWriter('Diffusion at Particle Center');
centerCross.FrameRate = 10;
open(centerCross);

h3 = figure();
hold on;
for p = 1:T
    %Store diffusion values for single time iteration
    cplot(rplotl-R+1:rplotl,:,p) = repmat(c(:,p),1,thetaplotl);

    %Create 1D plot
    if p == T
       plot(rplot,cplot(:,1,p));
       hold off
    else
       plot(rplot,cplot(:,1,p));
    end

    %Create 2D plot for that time step
    polarplot3d(cplot(:,:,p),'PlotType','contour',...
        'AngularRange',thetaplot,'RadialRange',rplot, 'RadLabels', 5);
    drawnow

    %Save the frame
    frame_cross = getframe(gcf);
    writeVideo(centerCross,frame_cross);
end

%Close the video file
close(centerCross);

%% Create video showing diffusion of EDX projection %%

%Create storage matries for concentration with respect to r and z at each
%time step
delz = delr;
z = 0:delz:ro;
Z = length(z);
c_z = zeros(rplotl,Z,T);
c_avg = zeros(rplotl,T);

%Fill c_z with values and average the values at each radii
for p = 1:T
    for q = 1:rplotl
        count = 0;
        add = 0;
        for s = 1:Z
            %Hypoteneuse length
            r_p = sqrt((q*delr)^2+(s*delz)^2);

            %Filter radii that exceed the particle size
            if r_p <= ro

                %Interpolate to nearest radii within r
                r_pcorr = interp1(rplot,rplot,r_p,'nearest');
```

```
                %Find index for nearest radii
                index = find(rplot==r_pcorr);

                %Using index, find associated concentration value
                c_z(q,s,p)=cplot(index,1,p);

                %Sum the values together and count once
                add = add+c_z(q,s,p);
                count = count+1;
            else
                c_z(q,s,p)=0;
            end
        end
        %Average all the concentrations at each radius
        if count == 0 %Prevent division by zero
            c_avg(q,p) = c(R,p);
        else
            c_avg(q,p) = add/count;
        end
    end
end

%Make the figure and make it invisible
h4 = figure();
h4.Visible = 'off';

%Initialize video
projection = VideoWriter('Diffusion of EDX Projection');
projection.FrameRate = 10;
open(projection);

%Create plotting matrix: a giant matrix which includes all
%(theta,r,c) values for each time step
c_avgplot = zeros(rplotl,thetaplotl,T);

for p = 1:T
    %Store diffusion values for single time iteration
    c_avgplot(:,:,p) = repmat(c_avg(:,p),1,thetaplotl);

    %Create plot for that time step
    polarplot3d(c_avgplot(:,:,p),'PlotType','contour',...
        'AngularRange',thetaplot,'RadialRange',rplot, 'RadLabels', 5);
    drawnow

    %Save the frame
    frame_projection = getframe(gcf);
    writeVideo(projection,frame_projection);
end

%Close the video file
close(projection);

%% Plot your results at each time step %%
h5 = figure();
hold on;
```

```
for n = 1:T
    if n == T
       plot(rplot,c_avg(:,n));
       hold off
    else
       plot(rplot,c_avg(:,n));
    end
end
```

Source Code: Diffusivity Estimator for Rectilinear Geometry

```
%% Fick's second law of diffusion numerically for rectilinear geometry

%% Select exp data

thickness = t20090_corr;
intensity = i20090_corr;
L = length(thickness);
%% Cycle over multiple diffusivities %%

D = 1:1:50; %nm^2/s
D_l = length(D);
R_squared = zeros(D_l,1);
rnew = zeros(L,1); %nearest diffusion sim radius values
unew = zeros(L,D_l); %nearest diffusion sim conc values
for k = 1:D_l
    %% Discretize the problem %%

    %Radius (nm)
    ri = 0; %inside length (nm)
    ro = 450; %outside length (nm)
    delr = 7.5; %radius discretization
    r = ri:delr:ro; %matrix of radius steps
    R = length(r); %total number of radius steps

    %Time (s)
    totalTime = 5400; %total simulation time
    delt = (1.5*delr^2)/D(k); %time discretization
    t = 0:delt:totalTime; %matrix of time steps
    T = length(t); %total number of time steps
    %% Create Solution Matrices %%

    %Create simplified "u" matrix (u=c*r)
    u = zeros(R,T);
    %% Initial Condition %%

    %Zero initial condition
    u(:,1) = 0;
    %% BC at outer shell (L = 0) %%

    %Diriclet B.C.
    u(1,:) = 1;
    %% Solve Crank Nicolson Implicitly (A x u = d) %%
```

```
    %Define nondimensional constant "C"
    C = D(k)*delt/(delr)^2;

    %Create tridiagonal matrix "A"
    A = diag((2*(1/C+1))*ones(R-2,1)) + diag(-1*ones(R-3,1),1) + ...
        diag(-1*ones(R-3,1),-1);

    %Account for insulated boundary condition at r=ro
    A(R-2,R-2) = 2*(1/C+1)-(4/3);
    A(R-2,R-3) = -(2/3);

    %Create solution matrix "d"
    d = zeros(R-2,1);

    for j = 2:T %Perform solution iteration for given number of time steps
        for i = 1:1:R-2 %Build d vector for current time step
            if i == 1
                %Dirichlet BC at outer shell
                d(i) = 2*u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+u(i+2,j-1);
            else
                d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+u(i+2,j-1);
            end
        end
        u(2:R-1,j) = A\d; %Solve for intermediate points
        u(R,j) = (4/3)*u(R-1,j)-(1/3)*u(R-2,j);
    end

    %% Calculate R^2 %%

    %Create comparison plot between exp and theoretical data
    for l=1:L
          %Interpolate to nearest common radii
          rnew(l) = interp1(r,r,thickness(l),'nearest');
          index = find(r==rnew(l));

          %Find concentration value for the common radii
          unew(l,k) = u(index, T);
    end

    %Calculate sum of squares of residuals and sum of squares
    data_mean = mean(intensity);
    sum_res = 0;
    sum_sq = 0;
    for m=1:L
          sum_res = sum_res + (intensity(m) - unew(m,k))^2;
          sum_sq = sum_sq + (intensity(m) - data_mean)^2;
    end
    %Calculate R^2
    R_squared(k) = 1- sum_res/sum_sq;
end

%% Find optimum diffusivity %%

%locate D that gives best R^2 value
maxRSquared = find(R_squared==max(R_squared));
```

```
D_opt = D(maxRSquared)
R_2 = max(R_squared)
%% Plot exp and optimal sim solution

plot(thickness,intensity);
hold on;
plot(rnew,unew(:,maxRSquared));
hold off;
```

Source Code: Nanoparticle Temperature Estimator for Spherical Geometry

```
%% Algorithm to estimate NP temperature from diffusion data
%% Select exp data

radius = r_500;
intensity = normi_500;
L = length(radius);
%% Cycle over multiple temperatures %%

temp = 400:1:500; %Kelvin
temp_l = length(temp);
R_squared = zeros(temp_l,1);
rnew = zeros(L,1); %nearest diffusion sim radius values
cnew = zeros(L,temp_l); %nearest diffusion sim conc values

for k = 1:temp_l
    %% Discretize the problem %%
    %Radius with just shell(nm)
    ri = 44; %core radius (nm)
    ro = 70; %core+shell radius (nm)
    delr = 2; %radius discretization
    r = ri:delr:ro; %matrix of radius steps
    R = length(r); %total number of radius steps

    %Include entire radius of particle, including the core
    rplot = [0:delr:r(1),r(2:R)];
    rplotl = length(rplot);

    %Assume rotational symmetry, plot with respect to theta for quarter
sphere
    thetaplot = 0:(pi/2)/(length(rplot)-1):pi/2;
    thetaplotl = length(thetaplot);

    %Diffusivity
    D = 1.406e+10*exp(-20.84/(1.987E-3*temp(k)));

    %Time (s)
    totalTime = 10; %total simulation time
    delt = 0.1; %time discretization
    t = 0:delt:totalTime; %matrix of time steps
    T = length(t); %total number of time steps

    %% Create Solution Matrices %%
```

```
%Create solution matrix (at%)
c = zeros(R,T);

%Create simplified "u" matrix (u=c*r)
u = zeros(R,T);

%% Initial Condition %%

%Define initial condition
u(:,1) = (1.26.^(r-max(r))+0.034).*r; %Determined experimentally

%% BC at outer shell (r = ro) %%

%Diriclet B.C.
u(R,:) = ro;

%% Solve Crank Nicolson Implicitly (A x u = d) %%

%Define nondimensional constant "C"
C = D*delt/(delr)^2;

%Create tridiagonal matrix "A"
A = diag((2*(1/C+1))*ones(R-2,1)) + diag(-1*ones(R-3,1),1) + ...
    diag(-1*ones(R-3,1),-1);

%Account for insulated boundary condition at r=ri
A(1,1) = 2*(1/C+1)-(4*r(1))/(3*r(2));
A(1,2) = r(1)/(3*r(3))-1;

%Create solution matrix "d"
d = zeros(R-2,1);

for j = 2:T %Perform solution iteration for given number of time steps
    for i = R-2:-1:1 %Build d vector for current time step
        if i == R-2
            %Dirichlet BC at outer shell
            d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+2*u(i+2,j-1);
        else
            d(i) = u(i,j-1)+2*(1/C-1)*u(i+1,j-1)+u(i+2,j-1);
        end
    end
    u(2:R-1,j) = A\d; %Solve for intermediate points
    u(1,j) = (4*r(1))/(3*r(2))*u(2,j)-r(1)/(3*r(3))*u(3,j); %Neumann B.C.
end
c(:,:) = u./repmat(r',1,T);

%% Calculate EDX Projection %%

%Create plotting matrix: a giant matrix which includes all
%(theta,r,c) values for each time step
cplot = zeros(rplotl,thetaplotl,T);

%Create storage matries for concentration with respect to r and z at each
%time step
```

```
delz = delr;
z = 0:delz:ro;
Z = length(z);
c_z = zeros(rplotl,Z,T);
c_avg = zeros(rplotl,T);

%Fill c_z with values and average the values at each radii
for p = 1:T
    %Store diffusion values for single time iteration
    cplot(rplotl-R+1:rplotl,:,p) = repmat(c(:,p),1,thetaplotl);
    for q = 1:rplotl
        count = 0;
        add = 0;
        for s = 1:Z
            %Hypoteneuse length
            r_p = sqrt((q*delr)^2+(s*delz)^2);

            %Filter radii that exceed the particle size
            if r_p <= ro

                %Interpolate to nearest radii within r
                r_pcorr = interp1(rplot,rplot,r_p,'nearest');

                %Find index for nearest radii
                index = find(rplot==r_pcorr);

                %Using index, find associated concentration value
                c_z(q,s,p)=cplot(index,1,p);

                %Sum the values together and count once
                add = add+c_z(q,s,p);
                count = count+1;
            else
                c_z(q,s,p)=0;
            end
        end
        %Average all the concentrations at each radius
        if count == 0 %Prevent division by zero
            c_avg(q,p) = c(R,p);
        else
            c_avg(q,p) = add/count;
        end
    end
end

%% Calculate R^2 for given temperature %%

%Create comparison plot between exp and theoretical data
for l=1:L
      %Interpolate to nearest common radii
      rnew(l) = interp1(rplot,rplot,radius(l),'nearest');
      index = find(rplot==rnew(l));

      %Find concentration value for the common radii
      cnew(l,k) = c_avg(index, T);
```

```
    end

    %Calculate sum of squares of residuals and sum of squares
    data_mean = mean(intensity);
    sum_res = 0;
    sum_sq = 0;
    for m=1:L
          sum_res = sum_res + (intensity(m) - cnew(m,k))^2;
          sum_sq = sum_sq + (intensity(m) - data_mean)^2;
    end
    %Calculate R^2
    R_squared(k) = 1- sum_res/sum_sq;

end
%% Find optimum temp %%

%locate D that gives best R^2 value
maxRSquared = find(R_squared==max(R_squared));
temp_opt = temp(maxRSquared)
R_2 = max(R_squared)
%% Plot exp and optimal sim solution

plot(radius,intensity);
hold on;
plot(rnew,cnew(:,maxRSquared));
hold off;
```

Source Code: Initial Condition Estimator for Core-Shell Nanoparticles

```
%% Algorithm to extract initial condition for model
%% Select exp data

radius = r_control;
intensity = normi_control;
L = length(radius);
data_mean = mean(intensity);
%% Define parameters %%

%Range of initial condition parameters
scale = 1.0:0.01:1.3;
scale_l = length(scale);
shift = 0:0.01:0.1;
shift_l = length(shift);
R_squared = zeros(scale_l,shift_l); %matrix to store R^2 values
rnew = zeros(L,1); %nearest diffusion sim radius values
cnew = zeros(L,scale_l,shift_l); %nearest diffusion sim conc values

%Discretization
ri = 40; %core radius (nm)
ro = 62; %core+shell radius (nm)
delr = 2; %radius discretization
r = ri:delr:ro; %matrix of radius steps
R = length(r); %total number of radius steps
```

```
rplot = [0:delr:r(1),r(2:R)]; %radius of whole particle
rplotl = length(rplot);
delz = delr; %Discretizing with respect to electron beam direction
z = 0:delz:ro;
Z = length(z);

%% Create Solution Matrices %%

%Create solution matrix (at%)
cplot = zeros(rplotl,1);
c_z = zeros(rplotl,Z);
c_avg = zeros(rplotl,1);
%% Cycle Over all Initial Conditions %%

for a=1:scale_l
    for b=1:shift_l
        %% Calculate initial condition %%
        for e = 1:rplotl
            if e >= rplotl-R+1 %Initial distribution in shell
                cplot(e) = scale(a)^(rplot(e)-max(rplot))+shift(b);
            else %Zero concentration in core
                cplot(e) = 0;
            end
        end

        %% Calculate EDX Projection %%

        %Fill c_z with values and average the values at each radii
        for q = 1:rplotl
            count = 0;
            add = 0;
            for s = 1:Z
                %Hypoteneuse length
                r_p = sqrt((q*delr)^2+(s*delz)^2);

                %Filter radii that exceed the particle size
                if r_p <= ro

                    %Interpolate to nearest radii within r
                    r_pcorr = interp1(rplot,rplot,r_p,'nearest');

                    %Find index for nearest radii
                    index = find(rplot==r_pcorr);

                    %Using index, find associated concentration value
                    c_z(q,s)=cplot(index);

                    %Sum the values together and count once
                    add = add+c_z(q,s);
                    count = count+1;
                else
                    c_z(q,s)=0;
                end
            end
            %Average all the concentrations at each radius
```

```
            if count == 0 %Prevent division by zero
                c_avg(q) = cplot(rplotl);
            else
                c_avg(q) = add/count;
            end
        end
        %% Calculate R^2 for given temperature %%

        %Create comparison plot between exp and theoretical data
        for l=1:L
              %Interpolate to nearest common radii
              rnew(l) = interp1(rplot,rplot,radius(l),'nearest');
              index = find(rplot==rnew(l));

              %Find concentration value for the common radii
              cnew(l,a,b) = c_avg(index);
        end

        %Calculate sum of squares of residuals and sum of squares
        sum_res = 0;
        sum_sq = 0;
        for m=1:L
              sum_res = sum_res + (intensity(m) - cnew(m,a,b))^2;
              sum_sq = sum_sq + (intensity(m) - data_mean)^2;
        end
        %Calculate R^2
        R_squared(a,b) = 1- sum_res/sum_sq;
    end
end
%% Find optimum temp %%

%locate D that gives best R^2 value
[maxrow,maxcol,value]= find(R_squared==max(R_squared(:)));
scale_opt = scale(maxrow)
shift_opt = shift(maxcol)
R_2 = R_squared(maxrow,maxcol)
%% Plot exp and optimal sim solution

plot(radius,intensity);
hold on;
plot(rnew,cnew(:,maxrow,maxcol));
hold off;
```