Title
The Effect of Particle Properties on Hot Particle Spot Fire Ignition

Author
Zak, Casey David

Publication Date
2015

Peer reviewed|Thesis/dissertation

The Effect of Particle Properties on Hot Particle Spot Fire Ignition

by

Casey David Zak

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

in

Engineering – Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor A. Carlos Fernandez-Pello, Chair
Professor J.Y. Chen
Professor Scott Stephens

Summer 2015

The Effect of Particle Properties on Hot Particle Spot Fire Ignition

Abstract

The Effect of Particle Properties
on Hot Particle Spot Fire Ignition

by

Casey David Zak

Doctor of Philosophy in Engineering – Mechanical Engineering

University of California, Berkeley

Professor A. Carlos Fernandez-Pello, Chair

The ignition of natural combustible material by hot metal particles is an important fire ignition pathway by which wildland and wildland-urban-interface spot fires are started. There are numerous cases reported of wild fires started by clashing power-lines or from sparks generated by machines or engines. Similarly there are many cases reported of fires caused by grinding, welding and cutting sparks. Up to this point, research on hot particle spot fire ignition has largely focused on particle generation and transport. A small number of studies have examined what occurs after a hot particle contacts a natural fuel bed, but until recently the process remained poorly understood.

This work describes an investigation of the effect of particle size, temperature and thermal properties on the ability of hot particles to cause flaming ignition of cellulosic fuel beds. Both experimental and theoretical approaches are used, with a focus on understanding the physics underlying the ignition process. For the experimental study, spheres of stainless steel, aluminum, brass and copper are heated in a tube furnace and dropped onto a powdered cellulose fuel bed; the occurrence of flaming ignition or lack thereof is visually observed and recorded. This procedure is repeated a large number of times for each metal type, varying particle diameter from 2 to 11 mm and particle temperature between 575 and 1100°C. The results of these experiments are statistically analyzed to find approximate ignition boundaries and identify boundary trends with respect to the particle parameters of interest. Schlieren images recorded during the ignition experiments are also used to more accurately describe the ignition process. Based on these images, a simple theoretical model of hot particle spot fire ignition is developed and used to explore the experimental trends further. The model under-predicts the minimum ignition temperatures required for small spheres, but agrees qualitatively with the experimental data. Model simulations identify the important physics controlling ignition for different sized particles and clarify many of the experimental trends.

The results show a hyperbolic relationship between particle size and temperature, with the larger particles requiring lower temperatures to ignite the cellulose than the smaller particles. For very small spheres, the temperature required for ignition is very sensitive to

particle size, while for very large spheres, ignition temperature shows only a weak dependence on that variable. Flaming ignition of powdered cellulose by particles $\leq$ 11 mm in size requires particle temperatures of at least 600°C. Ignition has not been observed for 2 mm particles at temperatures up to 1100°C, but the statistical analysis indicates that ignition by particles 2 mm and smaller may be possible at temperatures above 950°. No clear trend is observed with particle metal type, but copper particles require slightly higher ignition temperatures and seem more sensitive to experimental variation, likely due to their relatively high thermal conductivity. High-speed Schlieren images taken during the ignition experiments show that once particles land, they volatilize the powdered cellulose and the fuel vapor diffuses out into the surrounding air. Ignition occurs in the mixing layer between the vapor and the air, either during the initial expansion of the pyrolyzate away from the particle, or after a stable plume of volatiles has formed.

Modeling results indicate that in the large-particle, high-conductivity limit, the particle's surface temperature remains close to its impact temperature over the timescales of ignition. As a result, particle thermal properties are unimportant and ignition occurs when heat generation in the mixing layer overcomes losses to the surrounding air. When the large-particle limit does not apply, the particle cools upon impact with the fuel bed. In addition to the losses to the surrounding air, the reaction zone experiences losses to the cooling particle and must generate a larger amount of heat for ignition to occur. Because cooling is so important, the initial bulk energy is more useful than impact temperature for predicting ignition by smaller particles. Along those lines, the additional heat of melting available to molten particles helps to resist particle cooling; as such, molten aluminum particles $3.5 - 7$ mm in diameter can ignite at lower temperatures than solid particles of the same size with similar thermal properties. Decreasing volumetric heat capacity does increase minimum ignition temperature somewhat, but this effect is reduced for larger particles. Emissivity does not appear to have a significant effect on ignition propensity, suggesting that, over the timescales of ignition, radiation heat transfer is small relative to other modes of particle heat loss.

To my family, for all that you've taught me,
and especially my grandfathers, Rayburn David Zak and Alvin Dale Koenig,
for igniting my curiosity about the world around me.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbols

a	Upwind scheme coefficient
α	Thermal diffusivity
b	Ratio of particle to solid (or gas) volumetric specific
$\boldsymbol{\beta}$	Logistic regression fitting constants
c	Speed of light in a medium OR specific heat
c_0	Speed of light in a vacuum
d	Diameter
δr	Thickness of differential volume
$\Delta\tau$	Time step
$\dot{m}$	Mass flow rate
$\dot{q}$	Heat transfer rate
d	Generic vector of constants
ϵ	Emissivity
ε	Dimensionless radial coordinate
$\varepsilon_{x,y}$	Angular deflections due to optical inhomogeneity
f_1	Focal length
γ	Dimensionless specific heat
h	Enthalpy OR external heat transfer coefficient
$\tilde{h}_g$	Dimensionless gas-phase enthalpy

$h_{\mathbf{f}}^{\circ}$	Heat of formation
k	Kinetic coefficient
k_{GD}	Gladstone-Dale coefficient
l	Likelihood OR characteristic length of a body
λ	Thermal conductivity
$\tilde{\lambda}$	Dimensionless thermal conductivity
n	Index of refraction OR number of trials conducted
ν	Stoichiometric coefficient (by mass)
ω	Reaction or consumption rate
p	Ignition probability
$\hat{p}$	Observed ignition probability
$\dot{m}''$	Mass flux
ϕ	Generic transported scalar
ψ	Dimensionless radiation term
r	Radial coordinate
ρ	Density
σ	Stefan-Boltzmann constant
t	Time coordinate
τ	Dimensionless time coordinate
θ	Dimensionless temperature
u	Velocity
$\tilde{u}$	Dimensionless velocity
x, y, z	Cartesian coordinates
A	Surface area
A,B,C,D	TDMA coefficient vectors

$\mathbf{A}$	Ratio of solid fuel to gas-phase thermal diffusivity
Ar	Arrhenius number
Bi	Biot number
D	Gas-phase mass diffusivity OR generic diffusion coefficient
Da	Damkohler number
$\Delta\varepsilon$	Thickness of finite volume
ΔH	Heat of reaction or melting
$\overline{\Delta H}_{vol}$	Dimensionless heat of volatilization
ΔH°_{vol}	Standard heat of volatilization reaction
ΔT	Melting temperature range
$\dot{H}_{comb}$	Heat release rate due to combustion
$\dot{Y}_{i,comb}$	Consumption rate for species i
E	Bulk energy
E_a	Activation energy
$\mathbf{F}$	Generic square coefficient matrix
Fo	Fourier number
L	Line-of-sight distance OR log-likelihood
N	Number of grid points
P	Pressure OR proportion of submersion
P_{χ^2}	Pearson χ^2 test statistic
Py	Pyrolysis number
Q	Heat of combustion
$\tilde{Q}$	Dimensionless heat of combustion
R	Ratio of solid fuel and gas phase densities
R_u	Universal gas constant

S	Generic volumetric source
T	Temperature
T_a	Activation temperature
V	Volume
W	Molecular mass
Y	Random variable OR species mass fraction
$\tilde{Y}$	Scaled mass fraction
Z	Pre-exponential factor

Subscripts

0	First (innermost) finite volume
$conv$	Convective
$diff$	Diffusive
eff	Effective
f	Fuel volatiles
g	Gas phase
i	Initial state of particle OR generic gas-phase species
∞	Ambient
j	Associated with finite volume j
$j-$	Inner boundary of finite volume j
$j-1$	Associated with the volume inside of finite volume j
$j+$	Outer boundary of finite volume j
$j+1$	Associated with the volume outside of finite volume j
liq	Liquidus
m	Melting

n	Inerts or product species
neg	Negative
o	Oxidizer
p	Particle or sphere
$pore$	Associated with solid fuel pores
pos	Positive
rad	Radiation
ref	Reference
s	Solid cellulosic fuel
sen	Sensible
sol	Solidus OR solid-phase
st	Stainless steel
$surf$	Surface
vol	Volatilization
A, B	Denotes separate bodies
B	Boundary

Superscripts

$\circ$	Value at previous time step
$*$	Guess value from previous iteration

Acknowledgments

First, I would like to thank my family for the love and support that they have given me, particularly my parents and my brother. They have been, and continue to be, role models of how to work hard and take care of the people that are important to you. They have always pushed me to do my best, and supported me in both my successes and failures. These words and a thousand others could not express the gratitude that I feel. Thank you.

To Jessica, for loving me, making me laugh, and for listening to me ramble about lighting stuff on fire even when you had no idea what I was talking about. For taking care of me when I was not taking care of myself and for putting up with my shortcomings and quirks. I look forward to when I can support you as you have supported me.

Of course, none of this amazing experience would have been possible if Carlos Fernandez-Pello had not accepted me into his research group. He has been an amazing mentor, and I hope that our 'discussions' will not end simply because I am leaving the lab. He is supportive, fun, direct with both his praise and his criticism, and allows his students to make their own choices (and mistakes) even if that means they get a bit distracted along the way. I don't think one could ask for a better advisor.

During my time here at Berkeley, I have had a number of other academic mentors to whom I am hugely grateful. Chris Lautenberger has probably spent more time on the phone with me than any other person in recent memory, and was instrumental in helping me to make sure my numerical model was performing the way it should. I am also grateful for the opportunities that he and David Rich gave me to work on real projects with real clients at Reax Engineering. Besides being one of my best friends, Colin Smith instilled in me a devotion to scientific pursuits and helped me appreciate the amazing opportunities I've been given. Professor Robert Dibble has been an amazing friend and always made time for me when I had a question to ask. Professor J.Y. Chen has been an excellent instructor and was kind enough to sit on both my qualifying exam and dissertation committees. Professor Mario Sanchez-Sanz at La Universidad de Carlos III has been a great friend and mentor and also contributed to my initial efforts to develop a theortical model of hot-particle ignition. I would also like to thank Professor Scott Stephens for his involvement in my dissertation approval process. Despite his busy schedule and having only met me in person a few times, he made the time to contribute his expertise, and I am truly in his debt. Finally, I would like to thank Professor Janet Ellzey. As my undergraduate research advisor at the University of Texas, she mentored me and gave me incredible opportunities that set me on the path to where I am today.

In addition to great mentors, I have also been surrounded by great friends and colleagues. In particular, I am indebted to my friend and partner-in-science James Urban. As I am fond of saying, if it were not for James, I would not be finishing my graduate education. His

inquisitiveness and desire to learn when he first began working on the hot-particle project pushed me to refocus and really get back into the work after a period of low productivity. I also owe a great deal to my friend Daniel Murphy and his constant psyche for science. When working on a project, I am often reminded of something Dan told me once, paraphrased as "instead of finding reasons you can't do what you want to do, find a reason to do *something*". It was at Dan's suggestion that we first tried Schlieren imaging of the ignition process, and I have had more enlightening theoretical discussions with Dan than I can count. Of course, I was lucky enough to have a massive group of friends in Hesse Hall, and I'd really like to thank Shmuel Link, John Stevens (an adopted Hesse Hall member), Andres Osorio, Ben Wolk, Daniel Pineda, Tiernan Casey, Miguel Aznar, and Tim Sennott for their support and camaraderie over the past 5 years.

They say it takes a village to raise a child, but it takes all of the staff in Hesse Hall and Etcheverry Hall to complete a PhD. In particular I have to thank the real boss of Hesse Hall, Mary Anne Peters. Her support and help in all things administrative has truly been a blessing; the fact that she's just fun to be around is just icing on the cake. In Etcheverry, Yawo Akpawu has helped me patch up numerous missed deadlines and advised me on how best to handle all of the paperwork that comes along with being at Berkeley. Also many thanks to the Hesse Hall technical staff, including Micheal Neufer, Alex Jordan, Pete Graham, Scott McCormick and Tom Clark.

Finally, I would like to earnestly acknowledge the support of the National Science Foundation. They have made my graduate education possible, through funding the hot-particle ignition project and endowing me with a graduate fellowship. I am still not sure whether I deserved any of it, but I am thankful nonetheless.

Chapter 1

INTRODUCTION

1.1 Overview

Our understanding of, and ability to suppress, destructive fires has greatly increased since the 1950s when fire science research began in earnest [1]. In the United States, this is evidenced by the continual decline in the number of structure fires that occur each year [2, 3]. Even so, the occurrence and severity of wildfires and wildland-urban interface (WUI) fires has drastically increased, due to both increased human-wildland interaction and global climate change [4, 5]. According to US Fire Administration statistics, 96,000 wildfires burned over 4 million hectares in 2006, an area larger than the state of Maryland, the District of Columbia, and the surrounding suburbs combined [6]. In 2007, "outdoor and other" (primarily wildland) fires resulted in 45 deaths, 650 injuries, and $2.6 billion in property damage [7]. The latter figure includes $1.8 billion in losses from the October 2007 California firestorm. The largest single fire in that firestorm was the Witch Creek fire, ignited on August 21, 2007. It would eventually become the second largest fire in California history, burning almost 81,000 hectares and destroying over 1,100 homes. The California Department of Forestry and Fire Protection (CalFire) Investigation Report for the Witch Creek Fire alleges it was ignited by hot particles generated by power lines. Specifically, the report states "hot particles were created when the lines faulted and landed in the light grassy fuels in the Specific Origin Area. These fuels ignited, and the fire spread with the wind in a Westerly direction" [8]. Another recent large WUI fire is the Bastrop County Complex wildfire in Texas, which was the largest-loss fire in the United States in 2011 and burned almost 13,000 hectares [9]. Similar to the Witch Fire, the Bastrop fire allegedly started when power lines shorted out during high winds and sparks ignited dried grass and vegetation [10]. In addition to these examples, particles and sparks produced by welding, grinding, and various forms of hot work have also been involved in several other notable incidents [11].

In fact, it is estimated that power lines, equipment, and railroads cause approximately 28,000 natural fuel fires annually in the United States [12, 13]. There are many other

examples in New Zealand, Australia, and Southern Europe of wildfires that were initiated by metal particles. It has been reported that in New Zealand, 275 fires were ignited by sparks or flying brands between 2005 and 2010 [14]. In Australia, some of the devastating wild fires in Sydney in 1994 and of the Black Saturday event of February 2009 were also generated by sparks [15, 16]. Many other similar fires remain unreferenced because they have been involved in litigation or the origin of the fire could not be determined.

Clearly, the spot fire ignition of cellulosic materials (such as duff, litter and dry grass) by hot metal particles is an important fire ignition pathway by which wildfires and WUI fires are started. Structural fires may also be started when these particles ignite cellulosic insulation or by-products such as sawdust. The established literature discusses many potential particle sources in addition to those mentioned previously, including bullet fragmentation, friction between metal components, overheated brakes, and vehicle exhaust systems [11, 17, 18]. Small particles generated by these sources can be carried large distances by wind and even combustible material some distance away is at risk of ignition. Depending on the thermal characteristics of a particle and the fuel bed on which it lands, the particle may initiate a smoldering or flaming reaction that develops into a spot fire.

If we can predict whether ignition will occur in a given scenario based on particle and fuel bed properties and ambient conditions, we can use this information to prevent destructive accidental fires. High risk fuels can be identified and given higher priority during fuel treatments. Work site regulations and clearance distances around highways, railroads, and power line runs can be determined in an intelligent way. The predictive wildland fire models that are currently used by land managers and response personnel can be improved to account for this ignition possibility.

The work presented here is part of a larger effort directed towards this goal. It describes our efforts to understand the effect of particle properties (specifically particle size, temperature and thermal properties) on the flaming ignition of cellulosic fuels. The problem is studied both experimentally and analytically. In the remainder of the chapter, background information is given that will help the reader understand the contribution of the current work, particularly in the context of the scientific literature.

1.2 Background

The problem of hot particle spot fire ignition can be separated into three independent processes: (1) the generation of particles, (2) their flight due to ejection and wind drag, and (3) the ignition of the fuel bed after the landing of the particle. The subsections below briefly describe the work done thus far in each of these areas, as well as the work conducted on other types of hot particle ignition. It should be noted that while the discussion below focuses on hot particles, there are a large number of phenomena capable of initiating a spot fire. In particular, a number of studies have looked at the transport and ignition capabilities

of glowing embers and firebrands [19–22]. In addition to being a possible primary ignition source, spotting by embers is also an important mechanism for front propagation in established wildfires. Ignition by electric arcs and hot surfaces has also been studied, and the reader is directed to the references for more information on these topics [23, 24].

1.2.1 Hot particle production

The generation of particles, even from a single process, is difficult to characterize. Even so, a number of experimental studies have been conducted on particle generation from different sources. In response to bush fires in the 1970s, the State Electric Commission of Victoria (SECV) in Australia commissioned several studies on the ignition potential of particles produced by aluminum conductor clashing [25]. Unfortunately, the results of these studies are found in unpublished internal reports that present contradictory evidence. The most rigorous of these investigations found ejection velocities between 1 and 5 m/s, but another less rigorous effort claimed much higher values [26]. According to a literature review by Coldham, a later SECV-commissioned study by Joynt used high speed photography to measure particle temperature and ejection velocity of particles produced by copper conductor clashing [27, 28]. Coldham states that they found a maximum ejection velocity of 20 m/s and that particles up to 1.3 mm in diameter could be produced. Maximum temperatures were observed as high as 2240°C, but due to particle cooling during flight, temperatures were between 1100 and 1500°C upon impact. However, Coldham also states that this study followed from earlier SECV studies, so it is not clear whether this data is based off of separate copper conductor experiments or restated from previous, perhaps inaccurate, work on aluminum conductors. More recently, Ramljak et al. studied the statistical distribution of particle sizes produced by low-voltage aluminum power line clashing [29]. They investigated a number of different power system protection scenarios and found that in both live network and laboratory tests the mean particle size was on the order of 1 mm; a Dagum distribution was found to best characterize the size distribution. The introduction to their study also discusses a number of previous investigations conducted on particle production due to clashing.

In another recent study, Mikkelsen performed an investigation of the particles generated by various forms of hot work, including welding and abrasive and thermal cutting [30]. The focus of this work was to determine the spatial distribution of particles created by the various processes and to estimate safe working clearances. The authors found that welding processes have a higher risk of igniting material close to the work piece, but that many cutting and grinding processes are capable of ejecting particles a longer distance. Although the size of individual particles is not characterized, images from the study demonstrate that large globules of hot material can be produced from several processes, as shown in Figure 1.1. It was also noted that particle oxidization greatly affects the impact temperature, and that different materials have different oxidation characteristics (i.e. higher or lower impact temperatures). Mikkelsen's work includes a comprehensive literature review of past studies of ignition by industrial hot work processes.

Figure 1.1: Thermal Paper Set Alight by Oxyacetylene Slag [30].

Bullet fragmentation is another generation process capable of producing larger particles. Finney et al. studied this scenario experimentally by firing a variety of rifle bullets at an angled steel plate [17]. Upon impact with the plate, the bullets fragmented and deflected downwards onto a fuel bed of dry peat moss or excelsior. They measured size distributions and found that fragments weighing up to 5.7 g were produced by solid copper bullets and a deflector angle of 30°. They also estimated the temperature rise due to impact based on conversion of kinetic energy into thermal energy. As an example, the temperature rise for a copper bullet impacting at 750 m/s was estimated as 660°C. The other results of their study will be discussed in subsequent sections.

One of the more unique studies of particle generation is that of Arulmoli et al. [31]. They studied the production of sparks from titanium alloy and stainless steel golf clubs abraded against rock surfaces during a swing. The authors showed that the titanium alloy clubs were capable of producing particles (up to 0.5 mm in size) that oxidized and glowed for up to 1 s. The stainless steel clubs were not found to produce any sparks.

1.2.2 Hot particle transport

The particle transport step of the spot fire formation process has been studied the most rigorously, albeit almost exclusively with computational models. These models have all generally been concerned with particles produced by power line interaction, but the models, which consider the kinematics, cooling and oxidation of ejected particles, are applicable to other particle sources, and in some cases even the transport of firebrands [32].

The first analysis of how aluminum particle temperature changes during flight was conducted as part of the previously mentioned SECV study [25]. This was a very limited model that considered the particle as stationary with constant properties and imposed a single heat transfer coefficient based on constant estimates of the Reynolds number and drag coefficient. The mass and volume of the particle were also assumed constant. Finally, the heat generation due to surface oxidation was estimated using empirical evidence, but resulted in a particle temperature at landing that was much higher than the temperature at ejection. The end results of the analysis were physically unrealistic, and the authors pointed out a number of shortcomings in the model. This effort highlighted to need for careful consideration of particle transport and heat generation.

Rallis and Mangaya applied a similar model in an attempt to determine if there was a high probability that a particular veld-grass fire had been ignited by particles produced by conductor arcing [33]. They also neglected the effect of surface oxidation or combustion. Particle flight times and distance traveled were considered with decoupled equations in the horizontal and vertical directions, resulting in greater distances traveled compared to coupled equations.

Mills and Hang developed a more complex model of the trajectory and burning of hot aluminum particles produced by conductor arcing [34]. This work showed that smaller particles (less than 0.5 mm in size) would burn up before impact, but that larger particles ($\sim$1.5 mm and greater) could travel up to 200 m and remain burning upon impact. However, they assumed that during flight the particle had an effective diameter twice its original diameter due to oxide layer bubbling. They also modeled particle burning as the combustion of a fuel droplet. These assumptions yielded reasonable particle flight times but over-predicted particle lifetimes relative to later studies.

Psarros et al. formulated an analytical model for the transport and cooling of molten (not flaming) aluminum particles produced by low voltage conductor clashing [35]. They assumed that the clashing particles were ejected at the melting point of aluminum ($\sim$660 °C), and that they were ellipsoidal in shape. They considered only the losses due to convection but allowed the heat transfer coefficient to vary as a function of temperature and velocity. Ejection velocity was assumed to be zero and it is unclear how the drag coefficient was treated. The authors specifically examined the case of 0.5 mm particles falling from a 10 m

height in a uniform 20 m/s cross flow, and concluded that in this case such particles would not represent a serious ignition risk.

Tse and Fernandez-Pello developed a comprehensive model and used it to investigate the transport, combustion and ignition risk of aluminum and copper particles and firebrands produced by conductor interaction during high winds [32]. They solved the coupled ballistic equations in three dimensions subject to a wind velocity profile corresponding to grassy terrain. They also treated particle burning using experimental data for aluminum combustion in air. Their model produced reasonable results without the need for an effective particle diameter. They noted aluminum particles could both combust and travel further than their heavier, non-combusting copper counterparts. Even so, the higher volumetric heat capacity and unchanging mass of the copper particles still allowed them to impact with a significant amount of thermal energy.

1.2.3 Other forms of hot particle ignition

As a small localized source of thermal energy, a hot particle has the potential to ignite a number of different substances. Hot particle ignition of gases in particular has been a topic of great interest, particularly with respect to mines. In these mines, various processes can produce sparks capable of igniting the combustible gases that have built up in the enclosure. Although a number of authors have studied this phenomena, one early study that the author of this study found particularly enlightening was performed by Silver [36]. His investigation was prompted by the possibility of unintentional ignition of coal-gas within mines. The study involved injecting platinum and quartz particles into coal-gas, pentane and hydrogen mixtures to determine the minimum ignition temperature as a function of particle size. Silver proposed, and then demonstrated, that the temperature required for ignition would vary inversely with particle size. Additionally, his results showed little difference between the ignition behavior of platinum and quartz spheres, implying that particle composition may not play a major role. He also developed a theoretical model to explain this behavior, and predicted a linear relationship between $\log_{10}\frac{T_p - T_o}{a}$ and $|T_p|^{-1}$. Here, a is the particle radius and the subscripts p and o denote the particle and ambient, respectively. As seen in Figure 1.2, the experimental data exhibits this behavior, although more so for pentane than coal gas or hydrogen.

More generally, hot particle ignition problems can also be thought of as a subset of the classic 'hot spot' problem. A hot spot is a volume of heated material surrounded by exothermically reactive material at a lower temperature. Under the right conditions, this hot spot can initiate thermal runaway, i.e. an explosion. Researchers have looked at a number of variations of this problem; the hot spot may be inert or reactive, it may be cooling or heating up, and it may have similar or dissimilar thermal properties compared to the surrounding material [37–39]. In many of the treatments, the classical thermal ignition approach of Frank-Kemenetski is employed to find a critical value of the hot spot parameter,

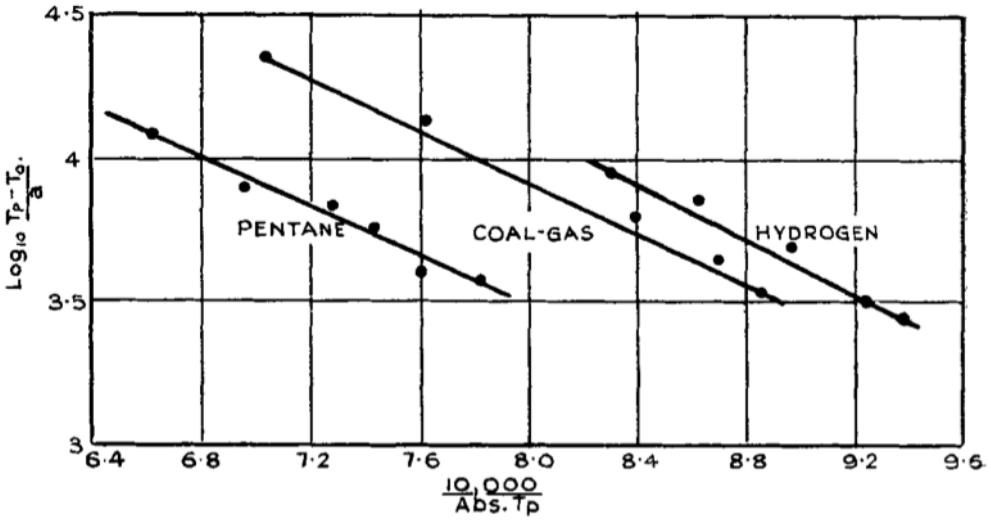

Figure 1.2: Experimental data with trend lines for platinum spheres ejected into three fuel-oxidizer mixtures [36].

denoted δ [40]. For example, Gol'dshleger et al. numerically solved the governing equations for an inert particle with a high initial temperature embedded in a reacting fuel. They assumed a global one-step reaction and found the critical value of δ (δ_{cr}) as a function of other parameters such as particle conductivity and chemical constants. As will be discussed later, their hot spot solution has been applied to the ignition of natural fuels by several investigators.

The modeling work of Gol'dshleger et al. was focused on the ignition of solid explosives and propellants due to the impact of ballistic fragments. Understanding this potential hazard is important for safely guarding and transporting weapons. A number of studies (both experimental and numerical) have been conducted on this specific process, known as hot fragment conduction ignition (HFCI) [41, 42]. Kuo et al. numerically modeled HFCI of a propellant in two dimensions, considering solid-phase decomposition and an exothermic gas phase reaction [41]. Later chapters will discuss the importance of considering both the solid and gas phase for properly modeling hot particle ignition of cellulosic fuels.

More recently, concern over fire safety in buildings has prompted research into the ignition of non-cellulosic building materials by hot particles. Specifically, Wang et al. studied the ignition of expandable polystyrene foam by hot particles [43]. Experimentally, they found flaming ignition probability as a function of particle size and temperature. Flaming ignition probability was defined as the number of flaming ignitions per number of tests at a given condition. Interestingly, for the size range tested (3 - 7 mm), the temperature required to achieve 50% ignition probability varied only from ~930 - 1030°C, with smaller particles

requiring higher temperatures to ignite. They also developed a numerical model that considered chemistry and heat transfer in the solid and gas phase. Comparisons of their numerical and experimental results indicated fair agreement.

1.2.4 Ignition of cellulosic fuels by hot particles

Despite a number of experimental and computational studies, what happens after a particle contacts a combustible fuel bed remains poorly understood. The SECV (mentioned previously) conducted experiments on the ignition potential of conductor-clashing particles in 1977 [25]. In those experiments, aluminum fragments were heated in an electric furnace and then dropped onto beds of dry barley grass or cotton wool. One of three possible observations was recorded: no ignition, smoldering ignition, or flaming ignition. Between tests, the size and temperature of the particles were varied between 1 and 6 mm and 600 and 1400°C, respectively. Based on the results, approximate smoldering and flaming ignition boundaries were established, as seen in Figure 1.3.

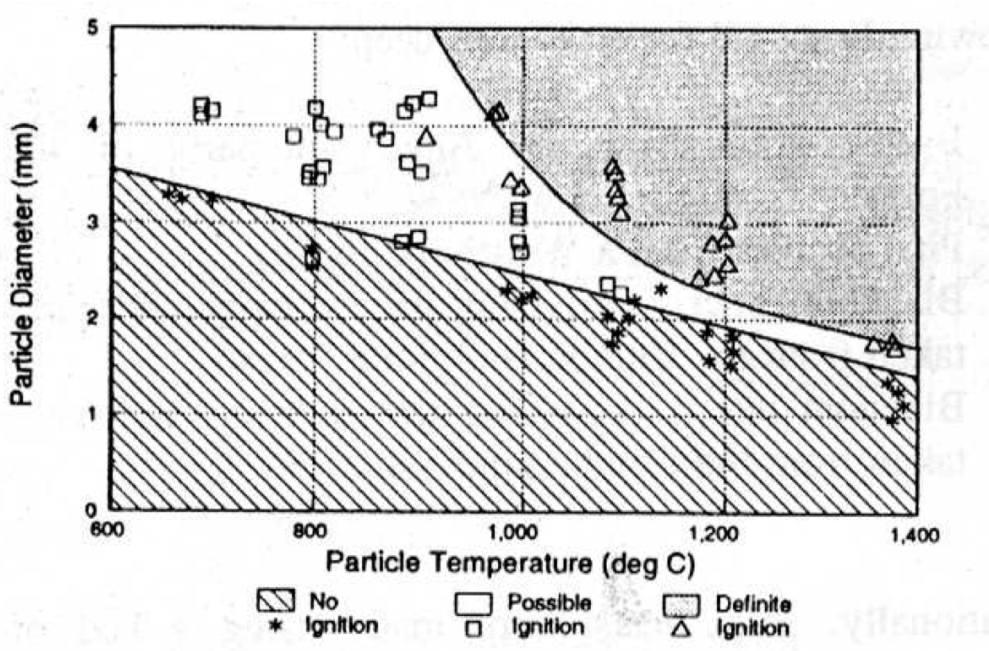

Figure 1.3: Ignition results and approximate ignition boundaries for aluminum particles igniting barley grass fuel beds [25].

The SECV study found that smaller particles required higher temperatures to ignite compared to larger particles and stated that "the results generally accord with the plausible notion that a particle must be both 'hot' and 'large' to cause ignition..." Note that in Figure 1.3, the three possible ignition regions appear to overlap. This implies that ignition

experiments are stochastic events, and Silver's results exhibit similar behavior [36]. There was also a marked difference in the behavior of the barley grass and cotton wool fuel beds. In one test, a < 2 mm particle ignited cotton wool at just 800°C, whereas the same sized particle in barley grass required a temperature near 1400°C. Also of note is the fact that the investigators found the flaming ignition boundary to have a slight hyperbolic curvature while the smoldering boundary appears linear.

Stokes later investigated the same ignition scenario, but generated particles through an actual electrical arc [44]. He arced steel, aluminum, copper, and brass conductors above a number of different fuel beds, as shown in Table 1.1. For each conductor and fuel bed combination, he recorded how many arcing events resulted in an ignition and recorded the size distribution of the particles produced. From these tests he deduced that steel and aluminum particles were more likely to ignite than the other particle metals due to their flaming oxidation, and that grasses were the most ignitable natural fuel. In subsequent tests with copper conductors, he attempted to control the size of particles that landed on the fuel bed by allowing the particles to fall through different meshes before impacting the fuel bed [45].

Rowntree and Stokes then applied this same experimental approach to aluminum particles and fuel beds composed of pine needles, barley grass and blackbutt and bluegum litter [46]. The particle temperature (at least at ejection) was apparently measured with video pyrometry, and mention is made that particles could reach temperatures in excess of 1400°C. In an attempt to align the work with previous energy based hazards standards, Rowntree and Stokes used their experimental data to determine minimum particle energies required for ignition: 0.2 J for the pine needles and barley grass and 1.2 J for the two types of litter. These energies were calculated assuming the particle impacted at its melting point ($\sim$660°C), and included the latent heat of melting. Thus, particle energy was assumed to be a function of size alone. However, when lines of constant energy are plotted along with ignition boundaries extrapolated from the SECV data, it appears that particle energy alone is not capable of predicting ignition (see Figure 1.4).

Tanaka also investigated the ignition of sawdust by steel splatter from welding and oxy-acetylene cutting [47]. In this study, steel spheres of varying diameters were heated to 950°C using a torch and dropped onto sawdust fuel beds of varying moisture contents. As shown in Figure 1.5, Tanaka observed ignition for particles as small as 3 mm and moisture contents as high as 40%. However, it is unclear how they defined their ignition criteria, so these ignition events may denote smoldering, flaming or some combination of both. As expected, larger particles were capable of igniting higher moisture-content fuels. They also observed variation in ignition outcome and conducted multiple tests at the same condition, such that ignition probability could be estimated and an ignition boundary could be drawn based on where this probability was sufficiently low. Based on their limited data, they assumed a linear shape for the ignition boundary.

Bed	IG/tests	Metering Sieve Mesh range, millimetres										
		≥ 2.8	2.79	2.36	2.00	1.65	1.41	1.17	0.99	0.84	0.70	0.59
		2.79	2.36	2.00	1.65	1.41	1.17	0.99	0.84	0.70	0.59	0.42
BC	48/100	4	9	59	176	274	410	705	620	635	1012	1723
PC	0/50	5	10	23	69	130	207	498	496	455	966	1588
PA	42/50	19	25	9	105	82	101	115	85	97	117	175
PS	47/50	20	19	16	85	80	83	195	378	168	254	1241
DC	28/50	18	25	55	79	113	177	238	236	216	421	815
DA	22/50	17	23	20								
DS	45/50	36	20	25	83	41	51	58	315	128	290	1211
HC	4/50	1	4	22	65	123	221	335	373	381	787	1414
HA	29/50	39	32	126	71	96	96	108	97	73	147	179
HS	45/50	20	17	33	50	37	52	60	232	107	208	848
CB	0/50	0	0	0	7	12	20	44	257	978	1787	882
WC	0/100	119	18	4	1	4	11	7	11	8	21	112
WB	0/50	0	0	5	14	65	87	181	258	475	1605	3031
WA	0/50	91	0	5	3	6	11	20	13	16	41	97
WS	0/50	136	5	6	7	23	23	72	154	263	533	2645

Legend: IG = Number of ignitions, MSM = Metering Mesh Size.

Code	First letter	Second letter	Code	First letter	Second letter
BC	Barley grass	Copper particles	HA	Hardwood	Aluminium particles
WS	Water bed	Steel particles	PB	Pine needles	Brass particles
DX	Decomposed hardwood	X = metal type	CX	Cotton fibre	X = metal type

Table 1.1: Ignition results and particle size distributions for arcing of various combinations of particle metal type and fuel bed composition [44].

Using an experimental approach similar to that used in the SECV study, Hadden et al. investigated the ignition of powdered cellulose by steel spheres [48]. The steel spheres were chosen as simplified representations of hot particles or firebrands and the powdered cellulose was chosen for its physical uniformity and availability of property data. The parameters of interest were the size and temperature of the particles, and the results are shown in Figure 1.6.

The data indicate that the smoldering ignition boundary is curved in shape, but the shape of the flaming boundary is less clear. Like the SECV study, there is some degree of overlap between no-ignition, smoldering and flaming conditions [25]. Hadden et al. also applied the hot spot solution of Gol'dshleger to the specific case of a steel particle and cellulose fuel bed and found that hot spot theory only gave general qualitative agreement with the experimental results (see Figure 1.7). Earlier work by Jones had also also applied the hot spot theories of Gol'dshleger and Zinn to the problem of hot particle spot fire ignition, but no comparison to experimental results was made [49–51]. The work by Hadden et al. also confirmed the observations of Babrauskas, showing that particle bulk energy was not

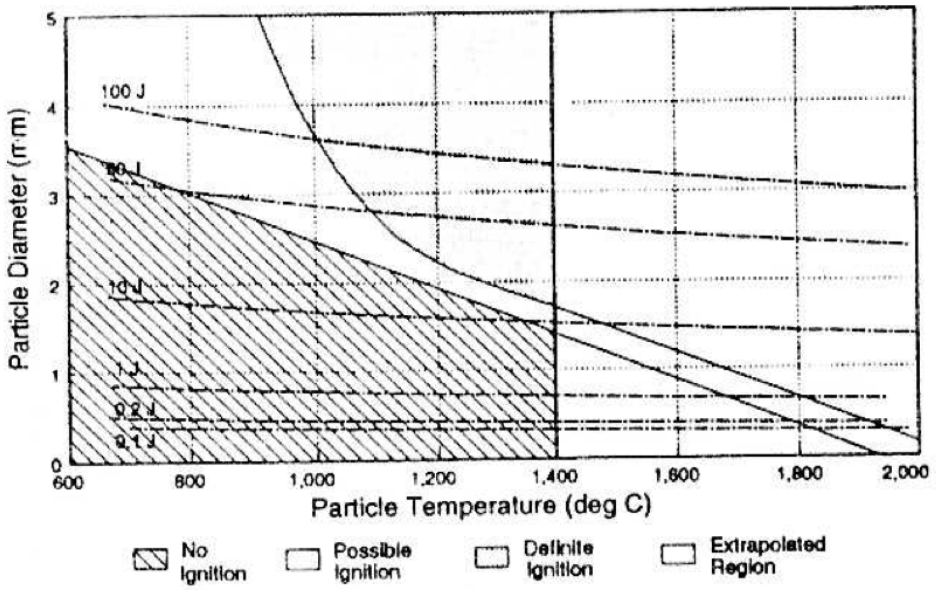

Figure 1.4: Lines of constant initial bulk particle energy (acc. to [46]) overlaid on the extrapolated ignition boundaries from [25].

sufficient to predict smoldering or flaming ignition [11].

The study by Finney et al, mentioned previously, also characterized the smoldering ignition capabilities of the bullet fragments upon impact with dried peat and excelsior fuel beds [17]. The peat fuel beds proved to be more ignition prone, likely due to the fact that their fine nature had improved surface contact with the irregular fragments. The study found that, compared to lead-copper bullets, solid copper bullets and steel components produced more ignitions, but mainly due to the fact that they produced larger fragments. Based on IR thermography and tests with temperature sensitive paints, the authors measured maximum impact temperatures between 550 and 800°C, but emphasized that these are conservative estimates. However, no information was given on the size or temperature of the particles that were capable of ignition.

Only a few computational studies have been conducted on spot fire ignition by hot particles or embers. Subbotin, and later Zvyagils'kaya and Subbotin, modeled the ignition of forest litter by a heated cylinder and a reacting ember, respectively [52, 53]. Their approach involved averaging over the axial dimension and solving the governing equations over the radial dimension. The model predicted ignition by reacting embers larger than 4.5 mm and moisture contents of less than 11%. Grishin et al. used a similar modeling approach to model experiments whereby forest litter was ignited by a heated nichrome wire coil [54]. In the model, electrical power was supplied to the coil for a set amount of time in an effort to determine the critical amount of energy required for ignition. The authors

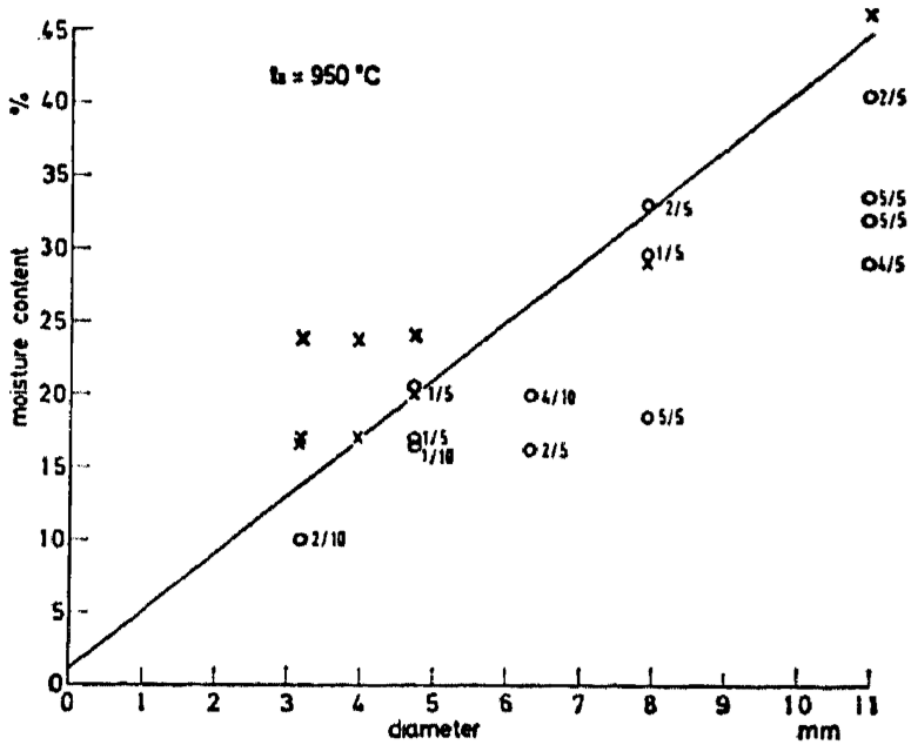

Figure 1.5: Observed ignition probabilities and approximate ignition boundary for steel spheres igniting sawdust [47].

then compared the model results with experimental ignition results for a heated nichrome coil as well as a number of real ignition sources. The comparison was used to estimate the minimum ignition energies for each of the real ignition sources. More recently, Lautenberger et al modeled the experiments of Hadden et al in two dimensions [55]. He treated the hot particle as a volumetric heat source and showed that coupling a condensed phase heat transfer and pyrolysis model with an available computational fluid dynamics solver was capable of representing both smoldering and flaming ignition.

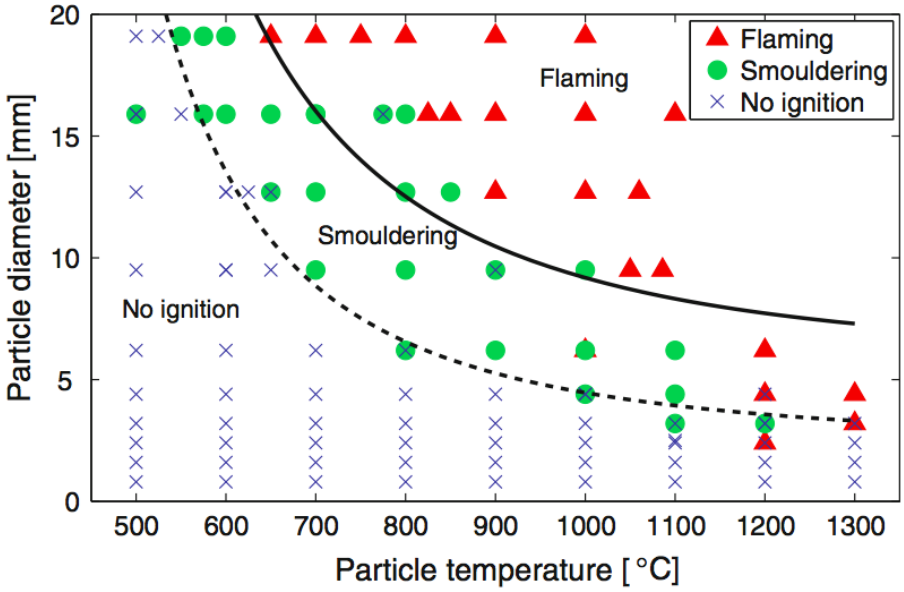

Figure 1.6: Ignition results and approximate ignition boundaries for steel spheres igniting powdered cellulose [48].

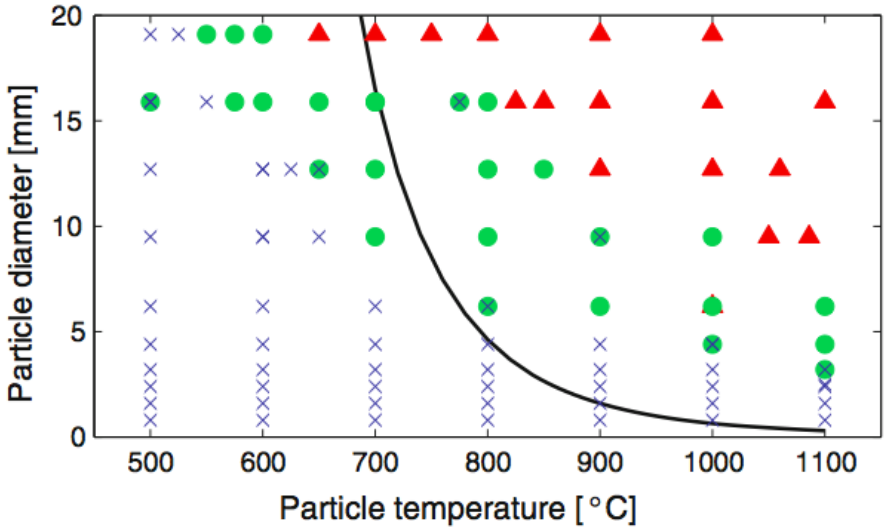

Figure 1.7: Ignition results for steel spheres and powdered cellulose along with the predicted ignition boundary according to hot spot theory [48].

1.3 Contribution of Current Work

Research on hot particle spot fire ignition up to this point has been mainly experimental and largely motivated by specific ignition incidents. For example, the first tests by the SECV focused on aluminum and copper particles produced by clashing because of a notable bush fire where conductor clashing may have been involved [28]. Researchers were less concerned about understanding the ignition process and more concerned about determining whether ignition *seemed possible* given the particular source. As a result, the focus of the work was often on how particles are produced and transported. What happens upon impact was often inferred based on the 'ignition temperature' of natural fuels, despite the fact that this value varies depending on the ignition scenario [27]. Theoretical treatments of the problem have largely applied hot spot theory, which has been shown to be inadequate due to its simplicity. While the ignition process has been studied computationally for projectiles and polymers, the few cases that applied such complexity to natural fuels have focused largely on embers as the ignition source.

As such, the present works aims to understand the process that occurs after a particle contacts a cellulosic fuel bed, and how particle properties affect the potential for flaming ignition. Such an understanding is important for improving our predictive capability as well as implementing preventative measures that are adequate without being overly burdensome [30]. The objective is accomplished by (1) conducting bench scale ignition experiments to establish ignition boundaries and determine the effect of particle size, temperature and metal type on these boundaries, (2) using high speed Schlieren videography of these experiments to gain insight into the ignition phenomenology, and (3) developing a simple theoretical model of the ignition process and using it to characterize the ignition physics and clarify the trends observed in experimental work.

Given that smoldering and flaming ignition are related but distinctly different, the scope of this work is restricted to flaming ignition (i.e. the establishment of a visible flame shortly after particle impact). The initiation of self-supported smoldering fronts and their associated transition to flaming are sufficiently complex as to deserve their own series of investigations. The particles discussed are also hot as opposed to flaming, which are representative of bullet fragments, some particles produced by friction, and flaming particles that cool off during flight or impact. This simpler scenario may in fact be the state in which the majority of particles land. Future efforts can build off of the work described here to include the complexities of flaming/glowing particles and a smoldering fuel bed. This dissertation is structured into an introductory and literature review chapter, a chapter focusing on experimental investigations, a chapter focusing on a theoretical model of hot particle spot fire ignition, and a conclusions and future work chapter.

Chapter 2 presents experiments focused on establishing boundaries between particle property combinations that result in flaming ignition and those that don't. The experiments involved dropping spheres of four metal types onto beds of powdered cellulose, and observ-

ing whether ignition occurred or not. The stochastic nature of ignition experiments that is so apparent in previous studies was addressed with a large number of tests and a rigorous experimental procedure. Logistic regression was used to establish the flaming ignition boundary for stainless steel spheres. Using this boundary, we discuss the effect of particle size and diameter. The ignition boundaries for aluminum, copper, and brass particles were established by a less rigorous method, but these boundaries are nonetheless compared to gain insight into the effect of particle thermal properties. The remainder of the chapter describes the use of high-speed Schlieren images of the stainless steel ignition tests to interpret the experimental ignition data and re-frame flaming ignition due to hot particles as a coupling of the solid and gas phase, not simply a solid-phase hot spot.

Chapter 3 discusses a simple theoretical model of hot particle ignition of cellulosic fuels. Based on the high-speed Schlieren observations, one can approximate the system as 2 one-dimensional spherical domains that are coupled by a thermally lumped hot particle. The resulting equations governing heat and mass transfer and chemistry in the solid and gas phases are derived. The chapter then nondimensionalizes these simplified governing equations in the style of previous ignition studies. After, the numerical implementation and solution of these dimensionless equations are discussed. Later, the chapter presents the model results, which show that ignition by large particles can be thought about in terms of a critical ignition Damkohler number. For small particles, the particle cools significantly upon contact with the fuel bed and losses from the gas phase reaction zone back to the particle dominate whether ignition occurs or not. Mirroring the experimental work, the remainder of the chapter uses the model to find flaming ignition limits and explore the effect of varying particle thermal properties on these limits. The results of this work clarify the experimental data. Model shortcomings are also discussed by comparing the experimental and numerical ignition boundaries and checking for internal consistency of the model assumptions.

1.4 References

[1] H.W. Emmons. "The growth of fire science". In: *Fire Safety J.* 3 (1981), pp. 95–106.

[2] United States Fire Administration. *U.S. Fire Statistics*. Feb. 2015. URL: `http://www.usfa.fema.gov/data/statistics`.

[3] M.J. Karter. *Fire Loss in the United States During 2013*. Tech. rep. Quincy, Massachusetts: National Fire Protection Association, U.S., 2014.

[4] A.L. Westerling et al. "Warming and Earlier Spring Increase Western U.S. Forest Wildfire Activity". In: *Science* 313 (2006), pp. 940–943.

[5] O. Pechony and D.T. Shindell. "Driving forces of global wildfires over the past millennium and the forthcoming century". In: *Proc. Nat. Aca. Sci. U.S.* 107 (2010), pp. 19167–19170.

[6] United States Fire Administration. Aug. 2009. URL: `http://www.usfa.dhs.gov/statistics/wildfire/index.shtm`.

[7] United States Fire Administration. Aug. 2009. URL: `http://www.usfa.dhs.gov/statistics/national/outside.shtm`.

[8] A. Maranghides and W. Mell. *NIST Technical Note 1635: A Case Study of a Community Affected by the Witch and Guejito Fires*. Tech. rep. Gaithersburg, Maryland: U.S. National Institute of Standards and Technology, 2009.

[9] S.G. Badger. *Large-Loss Fires in the United States - 2011*. Tech. rep. Quincy, Massachusetts: National Fire Protection Association, U.S., 2012.

[10] J. Jones, A. Saginor, and B. Smith. *2011 Texas Wildfires: Common Denominators of Home Destruction*. Tech. rep. College Station, Texas: Texas A&M Forest Service, 2012.

[11] V. Babrauskas. *Ignition Handbook: principles and applications to fire safety engineering, fire investigation, risk management and forensic science*. Issaquah, Washington: Fire Science Publishers, 2003.

[12] J.P. Prestemon et al. *Wildfire Ignitions: A Review of the Science and Recommendations for Empirical Modeling*. Tech. rep. United States Department of Agriculture: Forest Service, 2013.

[13] M. Ahrens. *BRUSH, GRASS, AND FOREST FIRES*. Tech. rep. Quincy, Massachusetts: National Fire Protection Association, U.S., 2013.

[14] New Zealand Fire Service. *Emergency Incident Statistics 2009-2010*. New Zealand Fire Service. Wellington, New Zealand, 2010.

[15] G. Ramsey and N. McArthur. *Building in the Urban Interface: Lessons from the January 1994 Sydney Brushfires*. Tech. rep. Hobart, Tasmania: National Institute of Standards and Technology, 1995.

[16] C. Egan and S. Holland. 2009.

[17] M.A. Finney et al. *A Study of Ignition by Rifle Bullets*. Tech. rep. Fort Collins, Colorado: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2013.

[18] U.S. National Fire Protection Association. *NFPA 51B, Standard for Fire Prevention During Welding, Cutting and Other Hot Work, 2014 ed., App. B*. National Fire Protection Association, U.S. Quincy, Massachusetts 02169, 2014.

[19] S.L. Manzello et al. "Experimental Investigation of Firebrands: Generation and Ignition of Fuel Beds". In: *Fire Safety J.* 43 (2008), pp. 226–233.

[20] R.A. Anthenian, S.D. Tse, and A.C. Fernandez-Pello. "On the trajectories of embers initially elevated or lofted by small scale ground fire plumes in high winds". In: *Fire Safety J.* 41 (2006), pp. 349–363.

[21] N. Sardoy et al. "Modeling Transport and Combustion of Firebrands from Burning Trees". In: *Comb. and Flame* 150 (2007), pp. 151–169.

[22] C.S. Tarifa, P.P. del Notario, and F.G. Moreno. "On the Flight Paths and Lifetimes of Burning Particles of Wood". In: *Proc. Comb. Inst.* 10 (1965), pp. 1021–1037.

[23] D. Coldham, A. Czerwinski, and T. Marxsen. *Probability of Bushfire Ignition from Electric Arc Faults*. Tech. rep. Victoria, Australia: Energy Safe Victoria, 2011.

[24] W.M. Pitts. *NIST Technical Note 1481: Ignition of Cellulosic Fuels by Heated and Radiative Surfaces*. Tech. rep. Gaithersburg, Maryland 20899: U.S. National Institute of Standards and Technology, 2007.

[25] G.E. Pleasance. *An Examination of Particles from Conductor Clashes as Possible Sources of Bushfire Ignition, App. 5*. Tech. rep. unpublished. Victoria, Australia: State Electricity Commission of Victoria, 1977.

[26] Sweeting Consulting Pty. Ltd. *Unkown*. Tech. rep. Victoria, Australia: State Electricity Commission of Victoria, 1977(est).

[27] D. Coldham. *Bushfire Ignition from Electric Faults: A Review of Technical Literature*. Tech. rep. Victoria, Australia: Energy Safe Victoria, 2011.

[28] R. Joynt. *The possibility of fires being caused by copper conductor clashing*. Tech. rep. Victoria, Australia: State Electricity Commission of Victoria, 1983.

[29] *Statistical Analysis of Particles of Conductor Clashing*. ENERGYCON 2014. 2014.

[30] K. Mikkelsen. "An Experimental Investigation of Ignition Propensity of Hot Work Processes in the Nuclear Industry". MA thesis. University of Waterloo, 2014.

[31] J. Arulmoli et al. "Spark production by abrasion of titanium alloys in golf club heads". In: *Fire Mater.* 39 (2015), pp. 119–126.

[32] S.D. Tse and A.C. Fernandez-Pello. "On the Flight Paths of Metal Particles and Embers Generated by Power Lines in High Winds, a Potential Source of Wildland Fires". In: *Fire Safety J.* 30 (1998), pp. 333–356.

[33] C.J. Rallis and B.M. Mangaya. "Ignition of Veld Grass by Hot Aluminium Particles Ejected from Clashing Overhead Transmission Lines". In: *Fire Tech.* 38 (2002), pp. 81–92.

[34] A.F. Mills and X. Hang. "Trajectories of Sparks from Arcing Aluminum Power Cables". In: *Fire Tech.* 20 (1984), pp. 5–14.

[35] E.G. Psarros et al. "A model for calculating the temperature of aluminium particles ejected from overhead low-voltage lines owing to a short-circuit". In: *Int. J. Wildland Fire* 18 (2009), pp. 722–726.

[36] R.S. Silver. "The Ignition of Gaseous Mixtures by Hot Particles". In: *Philos. Mag. and J. Sci.* 23 (1937), pp. 633–657.

[37] P.H. Thomas. "A Comparison of Some Hot Spot Theories". In: *Proc. Comb. Inst.* 10 (1965), pp. 369–372.

[38] U.I. Gol'dshleger, K.V. Pribytkova, and V.V.Barzykin. "Ignition of a Condensed Explosive by a Hot Object of Finite Dimensions". In: *Fiz. Gor. I Vzr.* 9 (1973), pp. 119–123.

[39] A. Liñan and M. Kindelan. "Ignition of a Reactive Solid by an Inert Hot Spot". In: *Comb. React. Sys.* Ed. by J.R. Bowen et al. Prog. Astro. Aero. Am. Inst. Aero. Astro., 1981, pp. 412–426.

[40] I. Glassman and R.A. Yetter. *Combustion*. Fourth. Burlington, Massachusetts: Elsevier, 2008.

[41] K.K. Kuo, W.H. Hsieh, and K.C. Hsieh. *MODELING OF HOT FRAGMENT CONDUCTIVE IGNITION OF SOLID PROPELLANTS WITH APPLICATION TO MELTING AND EVAPORATION OF SOLIDS, Report BRL-CR-607*. Tech. rep. Aberdeen, Maryland: Ballistic Research Laboratory, United States Laboratory Command, 1988.

[42] M.S. MILLER et al. *EFFECTIVE IGNITION KINETICS FOR LOVA PROPELLANT, Report BRL-MR-3724*. Tech. rep. Aberdeen, Maryland: Ballistic Research Laboratory, United States Laboratory Command, 1988.

[43] S. Wang, H. Chen, and N. Liu. "Ignition of expandable polystyrene foam by a hot particle: An experimental and numerical study". In: *J. Haz. Mat.* 283 (2015), pp. 536–543.

[44] A.D. Stokes. "Fire Ignition by Electrically Produced Incandescent Particles". In: 10 (1990), pp. 175–187.

[45] A.D. Stokes. "Fire Ignition by Copper Particles of Controlled Size". In: *J. Elec. Electronics Engr., Australia* 10 (1990).

[46] G.W.G. Rowntree and A.D. Stokes. “Fire Ignition by Aluminum particles of Controlled Size”. In: *J. of Elec. Electronics Engr., Australia* 14 (1994), pp. 117–123.

[47] T. Tanaka. “On the Flammability of Combustible Materials by Welding Splatter”. In: *Reports of the Natl. Res. Inst. of Police Sci.* 30 (1977), pp. 151–158.

[48] R.M. Hadden et al. “Ignition of Combustible Fuel Beds by Hot Particles: An Experimental and Theoretical Study”. In: *Fire Tech,* 47 (2011). http://dx.doi.org/10.1007/s10694-010-0181-x, pp. 341–355.

[49] J.C. Jones. “Predictive Calculations of the Effect of an Accidental Heat Source on a Bed of Forest Litter”. In: *J. Fire Sci.* 11 (1993), pp. 80–86.

[50] J.C. Jones. “Further Calculations Concerning the Accidental Supply of Heat to a Bed of Forest Material”. In: *J. Fire Sci.* 12 (1994), pp. 502–505.

[51] J.C. Jones. “Improved Calculations Concerning the Ignition of Forest Litter by Hot Particle Ingress”. In: *J. Fire Sci.* 13 (1995), pp. 350–356.

[52] A.N. Subbotin. “Study of the critical conditions of initiation of a surface fire”. In: *Phys. Mech. Eco., Abstracts of the Int. Sci. Conf.* Tomsk, Russia, 1994, pp. 149–150.

[53] A.I. Zvyagils‘kaya and A.N. Subbotin. “Influence of Moisture Content and Heat and Mass Exchange with the Surrounding Medium on the Critical Conditions of Initiation of Surface Fire”. In: *Comb. Exp. Shock Waves* 32 (1996), pp. 558–564.

[54] A.M. Grishin et al. “Ignition of a Layer of Combustible Forest Materials”. In: *Comb. Exp. Shock Waves* 34 (1998), pp. 613–620.

[55] C. Lautenberger and A.C. Fernandez-Pello. “Modeling Ignition of Combustible Fuel Beds by Embers and Heated Particles”. In: *Forest Fires 2008*. 2008.

Chapter 2

EXPERIMENTAL INVESTIGATIONS

2.1 Introduction

This chapter presents experiments aimed at determining the effect of particle properties on the flaming ignition limit of cellulosic fuel beds. Particle size, temperature and metal type were the specific properties of interest. While other researchers have investigated some of these aspects, the author sought to do so as rigorously as possible, such that the experimental results would yield clear trends and practically useful ignition boundaries. To that end, experimental variation was limited as much as possible. Different sized metal particles composed of Stainless Steel 302, Aluminum 1100, Brass 260, and Copper 110 were heated using a tube furnace and dropped onto a fuel bed seated inside the test section of a bench-scale wind tunnel. The author chose this heating method, which was also used in the SECV study, because it seemed to allow more consistent and accurate particle temperature measurement than the torch-heating methods used by Tanaka and Hadden et al. [1–3]. Similar to Hadden et. al, powdered α-cellulose was chosen as the fuel bed material for its chemical simplicity and physical uniformity. Other aspects of the experiments, such as fuel bed moisture content, fuel bed settled volume and air cross-flow were held constant.

Even so, many aspects of the problem were not easily controlled, such as local fuel bed properties, impact dynamics and inherent variability in the ignition process. The result was stochastic ignition data similar to the data of previous researchers, where there is some degree of overlap between conditions that will cause flaming ignition and those that will not [1–3]. Embracing the variable nature of the experiments, the author attempted to randomize aspects of the problem that were not easily controlled to eliminate systematic bias and conducted multiple experiments at each test condition, resulting in a large number of tests overall. In doing so, one finds that there is actually a significant region where flaming ignition is possible but not certain. Logistic regression and estimates of ignition probabil-

ity were used to find useful ignition boundaries. While conducting these tests the author recorded the ignition events using high-speed Schlieren imaging. In addition to highlighting the randomness of the ignition process, observing these events gave additional insight into the ignition process.

Section 2.2 describes the experiments in detail, including the experimental apparatus and the efforts made to control or randomize experimental variables not under study. The Schlieren imaging system is also described in detail. Section 2.3 discusses the statistical approach we used to analyze the experimental ignition data. Sections 2.4 presents the results of the experiments with stainless steel particles and uses them to demonstrate the hyperbolic relationship between particle size and minimum flaming ignition temperature. Section 2.5 gives the ignition results for the other three metal types. By comparing their flaming ignition boundaries, we show that particle melting reduces the minimum temperature required for ignition, and suggest that large particle thermal conductivity may reduce ignition likelihood. The final section discusses the high speed video taken of the ignition events, and uses these observations to develop an ignition phenomenology and possible explanation for the results of the previous two sections.

2.2 Experimental description

The metal particles studied were either spherical or approximately spherical in shape. The particles ranged in size from 2 to 11 mm in diameter. The temperatures tested ranged from 575 - 1100°C. The maximum particle temperature was limited by the maximum operating temperature of the tube furnace. For the range of particles tested no ignition was observed at 575°C, so further tests were not conducted below that temperature. Tests using brass and copper were limited to temperatures below their melting points (915°C and 1065°C, respectively) because when these metals were heated past their melting point the molten particle would wet to the ceramic crucible in which it was held. There was no commercially available coating found which would prevent this from occurring in an oxidizing environment. As such the maximum temperature used for brass was 900°C and 1050°C for copper. As discussed in the previous chapter, these different metal types are representative of particles produced by conductor clashing, bullet fragmentation, and hot work such as welding and torch cutting. While particles produced by electrical arcing are typically much smaller (often less than 1 mm) and much hotter (up to 3600°C), the results presented here still provided insight into the ignition process and important controlling parameters [4]. The larger diameters investigated may also sufficiently approximate the accumulation of small particles that can occur in real-world scenarios, making those results all the more relevant.

Fuel beds were composed of powdered α-cellulose ($(C_6H_{10}O_5)_n$). Cellulose was chosen for its chemical homogeneity and the availability of property data. Furthermore, cellulose is the largest component of woody biomass, making it a reasonable surrogate for more complex

fuels (such as sawdust or duff). According to data provided by the manufacturer (Sigma-Aldrich), the cellulose particles have a mean diameter of 0.36 mm and at least 80% are 0.42 mm or smaller. The fuel beds were laboratory-conditioned with a mean moisture content of $7 \pm 2\%$. The author measured the moisture content of the fuel for each daily series of experiments. This involved drying lab-conditioned samples in an oven at 110 ± 5°C for at least 4 hours. Each sample weighed at least 1.3 g and the mass was measured before and after drying.

Given that the volume of the fuel container was constant, the density of the fuel bed was determined by the cellulose mass. However, the volume occupied by a given mass of cellulose was found to vary greatly from fuel bed to fuel bed. Additionally, a large amount of settling was observed between testing days such that a fuel bed that completely filled the fuel container one day would be well below the level of the wind tunnel floor the next. Wanting to maintain a uniform surface along the bottom of the wind tunnel, the author elected to hold cellulose settled volume constant for all experiments. Here, the settled volume refers the minimum volume occupied by the fuel bed after vigorous vibration. Thus, the container was filled and vibrated until settling was no longer observed, and the mass was recorded to calculate the resulting density. Fuel bed bulk density varied normally about a mean of 338 ± 40 kg/m^3, slightly higher than the range stated by the manufacturer and much higher than the density used by Hadden et al. [3].

2.2.1 Apparatus

The experimental apparatus is shown in Figure 2.1. An isometric schematic with a cutout of the test section is given in Figure 2.2. The fuel bed was seated flush with the bottom of the wind tunnel. The wind tunnel was 550 mm in length with a 130 mm by 80 mm cross section. The fuel bed container was 150 mm long, 100 mm wide and 50 mm deep and its leading edge was 150 mm from the inlet of the tunnel. The bottom 18 mm of the fuel bed container were filled with sand to conserve limited supplies of cellulose and simulate the soil that would normally be below natural fuels. The densities discussed in the previous section were based on the volume remaining in the container. Viewing windows in the sides of the tunnel allowed optical access for Schlieren video recording. This experimental apparatus was identical to the one used in the study by Hadden et al., with the exception of the fuel container being partially filled with sand [3]. As discussed later in the chapter, this resulted in differences between the studies' results.

Laboratory air was flown through the wind tunnel with a centerline velocity of 0.5 m/s at the leading edge of the fuel bed. This particular velocity was chosen in an effort to simulate the cross-flowing wind that would most likely be encountered in real-world scenarios. However, this was the highest flow achievable that did not overly disturb the fuel bed surface. The Schlieren video footage indicates that it is on the order of buoyancy-induced flow speeds, implying a mixed flow regime. Because of the low flow velocities, the ignition conditions

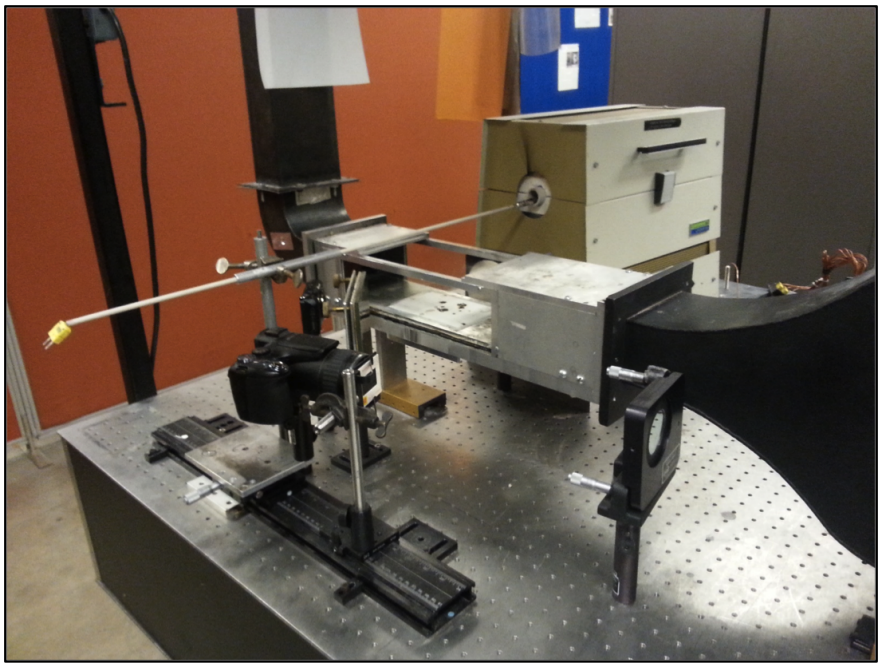

Figure 2.1: Photograph of experimental setup and elements of Schlieren system.

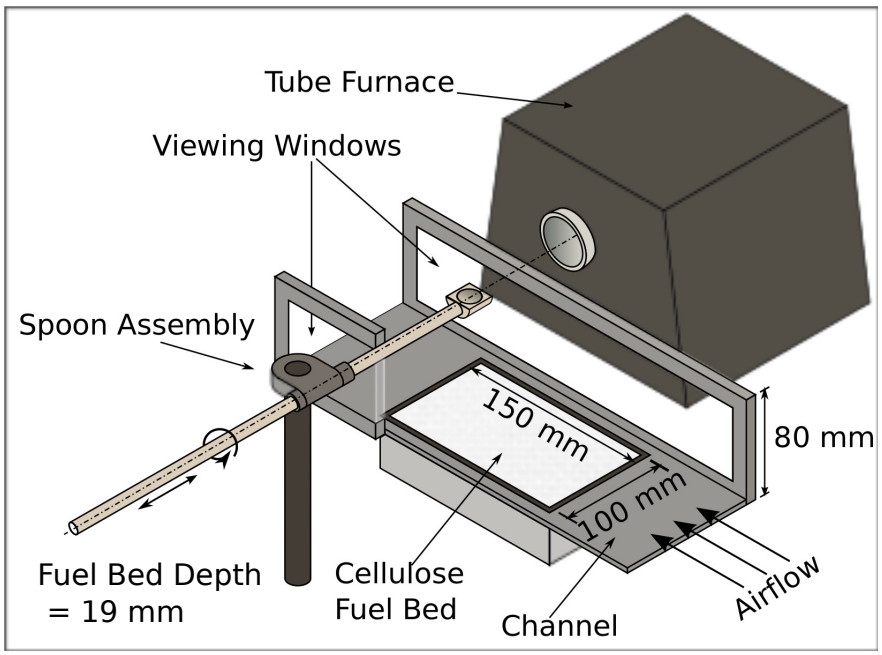

Figure 2.2: Experimental setup in an isometric view with cutout.

found in this work may not be representative of those that would be obtained at higher velocities. Flow uniformity was reduced when the tunnel top was opened to introduce the

spheres. To overcome this complexity and ensure a uniform cross-flow velocity between tests, tests were only conducted on the leading half of the fuel bed. Hot-wire anemometer probes indicated that the flow was sufficiently uniform in this region. The relative humidity and temperature of the flow were measured daily and found to be 13.1±1.8% and 26.3±0.45°C respectively.

A linear guide held a ceramic spoon-shaped crucible approximately 140 mm above the fuel bed. This guide was collinear and concentric with a high-temperature tube furnace such that the crucible could easily be inserted and removed from the furnace. A type K thermocouple was embedded in the crucible to provide a reliable measurement of the sphere temperature.

2.2.2 Experimental procedure

During each run, a sphere was placed in the ceramic crucible and the crucible was inserted into the tube furnace. The sphere remained in the furnace until the crucible thermocouple indicated that the crucible and sphere were at a steady state at the desired temperature. After thermal equilibration, the crucible was rapidly removed from the furnace and rotated, dropping the sphere onto a virgin area of the fuel bed. It should be noted that the particle temperature reported here is that of the particle in the furnace, not at landing. The particle temperature at landing is obviously lower and dependent on the particle size, temperature, thermal properties and emissivity. To conservatively estimate the temperature drop that occurs, we performed a large number of mock test runs and estimated the total time during which a particle was out of the furnace before contacting the bed as 0.4 s. Based on a lumped model, the temperature loss of the sphere during this time was no more than 50°C (for the case of a 2.38 mm sphere at 1100°C). This estimate neglects the insulating effect of the hot crucible holding the sphere, which would further reduce heat losses.

We conducted between five and twenty experiments for each particle metal type at a given diameter and temperature. For each specific parameter combination, drop locations were varied and no more than three tests were done on the same fuel bed. This reduced any potential systematic error due to variations in the fuel beds and penetration depth of the particles. Flaming ignition or lack thereof was visually observed for each test and recorded. We defined 'flaming ignition' as the appearance of a stable flame that persisted for more than 1 second after the sphere contacted the bed. As alluded to previously, incipient smoldering was not considered in this study and was recorded as a non-ignition event, and experiments focused on testing conditions near the flaming/non-flaming limit.

2.2.3 Schlieren imaging system

Schlieren imaging is a technique that allows one to see gradients in the index of refraction of a medium. When light interacts with matter it slows down, and the speed of light in the

medium is given as

$$c = \frac{c_0}{n} \tag{2.2.1}$$

where n is the medium's index of refraction and c_0 is the speed of light in a vacuum. For gases or gaseous mixtures, the relationship between density and the index of refraction is given as

$$n - 1 = k_{GD}\rho \tag{2.2.2}$$

where ρ is the density and k_{GD} is the Gladstone-Dale coefficient. The latter parameter depends largely on the composition of the gas; there is also a weak dependence on gas temperature and the wavelength of interacting light [5]. If one assumes that the behavior of the gas under consideration is ideal, then the density is further related to the pressure P and the temperature T by

$$P = \frac{\rho R_u T}{W} \tag{2.2.3}$$

where R_u is the universal gas constant and W is the molecular mass of the gas. Combining equations 2.2.2 and 2.2.3 and rearranging yields

$$n - 1 = \frac{k_{GD} P W}{RT}. \tag{2.2.4}$$

For open systems where pressure can be considered approximately constant, equation 2.2.4 shows that gradients in the index of refraction can result from temperature gradients, concentration gradients (through changing k_{GD} and W), or some combination of both. These gradients can occur at the boundary between hot and cold fluid streams or across the diffusion layer between different gases. Thus, Schlieren imaging can give information about density changes and the flow of energy and species that would normally be invisible to the naked eye. As an example, Figure 2.3 shows a Schlieren image of a candle flame. The boundary of the buoyant plume created by the flame is clearly visible, as are the eddies that indicate the plume is transitioning to the turbulent regime. Besides its capabilities, the appeal of this imaging method is that the optics are quite simple. As a note, the discussion that follows is largely adapted from [6]. When light encounters a gradient in the index of refraction, it is bent or *refracted* away from its original path. This same phenomena is responsible for the broken appearance of a pencil that has been placed in a glass of water. If we take the z-direction as the line of sight, then angular deflections in the x, y-plane some distance L

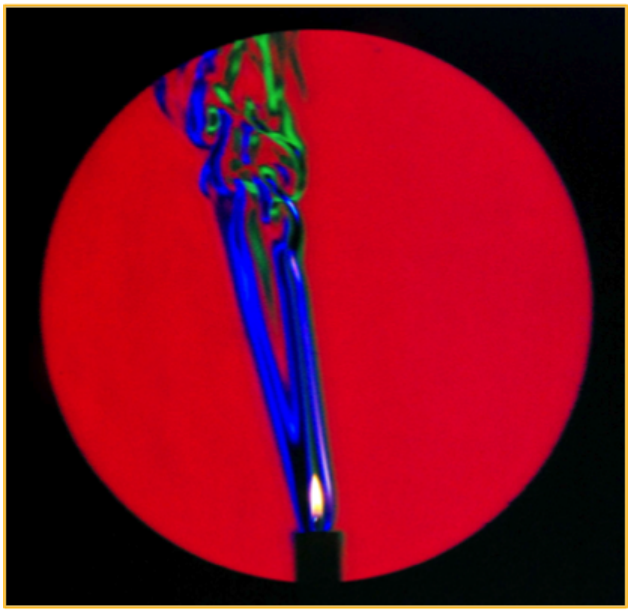

Figure 2.3: Schlieren image of candle flame [6].

from the optical inhomogeneity are given as

$$\varepsilon_x = \frac{L}{n}\frac{\partial n}{\partial x}, \qquad \varepsilon_y = \frac{L}{n}\frac{\partial n}{\partial y}. \tag{2.2.5}$$

These relations serve as the basis for Schlieren imaging, and their derivation can be found in Appendix A.6 of [6]. Figure 2.4 shows a simple lens-type schlieren system where this phenomena is used to visualize gradients of interest. On the left is an extended light source (as

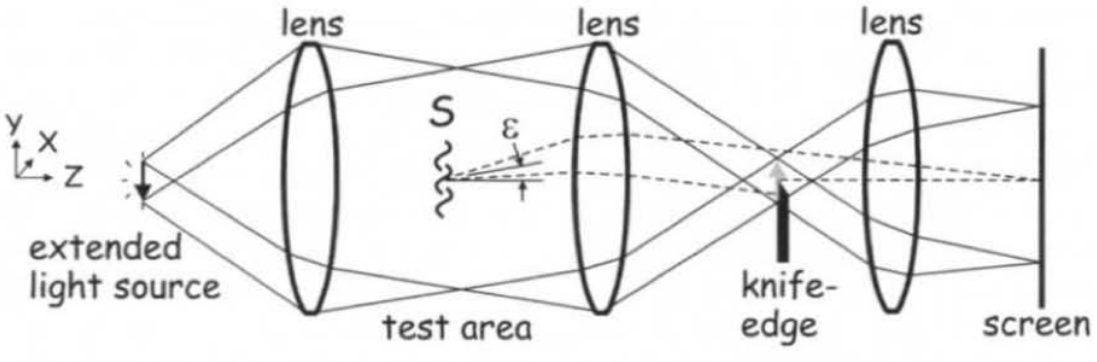

Figure 2.4: Simple lens-type schlieren system [6].

opposed to a point light source), which for demonstration purposes is shown as a downward facing arrow. In practice, the light source is often a bright light that is completely masked except for a narrow slit. The light from this source passes through three lenses. The first lens collimates the light from the source and the second focuses the light such that an inverted image of the source is produced at its focal point. The third lens creates a focused image of the *schlieren test area* (not the light source) on the viewing screen.

The extended light source can be thought of as an infinite array of small point light sources. The light from each of these sources passes through the entire test area, and the portion of this light that is caught by the final focusing lens produces a circle on the viewing screen. This is demonstrated by the edge rays emanating from the upper and lower points of the extended source in Figure 2.4. As such, what is seen on the viewing screen is a composite image made of the light from each of these point sources. If a knife edge is used to block the lower portion of the inverted source image, the light from some, but not all, of the point sources is blocked, and the circle on the screen darkens uniformly. This sets a background level of illumination against which brighter and darker regions can be seen.

Now, one can consider the light from a single point in the test area when a Schlieren object is present. The light that has passed through that point is now bent, and it becomes more or less occluded by the knife edge than it was previously (see Figure 2.4). However, the properties of lenses are such that this light bundle is then returned to its original relative position in the image on the screen. In this way, gradients in n brighten or darken the image at their true relative location. Use of a horizontal knife edge allows the detection of gradients in the y-direction as changes in gray scale intensity. A vertical knife edge or a pinhole would give gray scale images of gradients in the x- and r-directions, respectively. As evidenced by Figure 2.3, the knife edge can also be replaced by a colored bullseye, such that deflections cause the image to change colors.

Figure 2.5 shows a schematic of the Schlieren system used in this work. It made use of a single spherical mirror with a focal length of 1 m in place of the collimating and focusing lenses shown in Figure 2.4. This system was chosen based on the availability of optical components and the space constraints of the optical table where the wind tunnel was mounted. The light source was a single, white, high-powered light emitting diode (LED). Light from this source first encountered a beam splitter, as indicated by the edge rays shown in Figure 2.5. The beam splitter allowed a portion of the illumination to pass through and the beam then deflected off of a flat mirror and into the spherical focusing mirror. Between these two mirrors the light made its first pass through the wind tunnel and imaging test area. Due to reflection and focusing by the spherical mirror, the light retraced its path until it returned to the beam splitter. A portion of the beam passed through the splitter, but the remainder was deflected towards a colored bullseye (blue center, yellow and red rings). Given that the light beam passed through the test area twice (called a "double pass" configuration), density gradients within the tunnel had two opportunities to cause refraction. Therefore, a clear

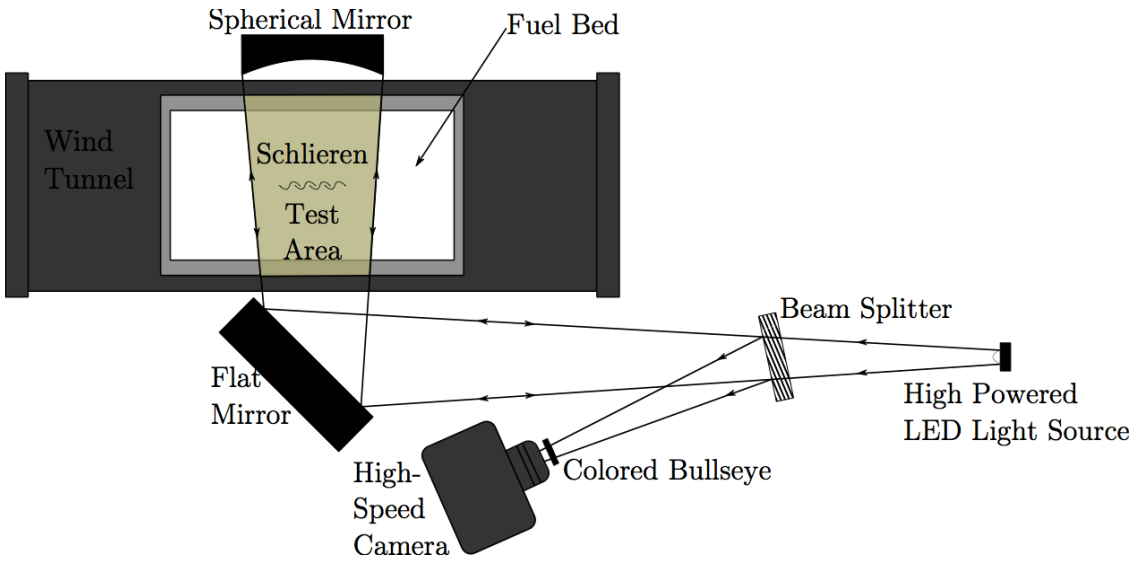

Figure 2.5: Top view schematic of schlieren system and wind tunnel.

image required that the incident and reflected optical paths were perfectly aligned and that the bullseye and light source were each a distance $f_1$away from the mirror along the path (f_1 being the focal length of the spherical mirror). After the bullseye, the light was focused onto the photo sensor panel of a high speed digital camera. Videos were recorded at 1200 frames per second (fps).

2.3 Statistical analysis of ignition data

As indicated in the previous chapter, experimental hot particle spot fire ignition results are stochastic, with the same parameter combinations sometimes resulting in both ignition and non-ignition events. The results presented in the remainder of this chapter are no different. As noted above, many difficult-to-control but important experimental factors were intentionally randomized to avoid systematic bias, and this random error must be accounted for if one desires to separate the underlying physical trends from the experimental noise. Additionally, a statistical approach helps identify conservative flaming ignition boundaries that are less specific to the experimental study. This section gives a description of the statistical methodology that was developed and used to find ignition boundaries. For results with stainless steel particles, this involved using logistic regression to find the approximate ignition probability distribution. However, for experiments with other metal types the author was unable to find sufficient probability distributions. Therefore, for the purposes of comparing the effect of metal type, a less rigorous method was used.

First, it is important to note that ignition data is *binary* or *categorical*. Each test results in only one of two possible outcomes: ignition (given a value of 1) or no-ignition (given a

value of 0). In this way, each ignition test is similar to a coin toss, where the outcome can only be heads or tails. One can't know before hand whether the outcome of a particular toss will be heads, but one can, based on multiple tests and/or reasoning, determine that the *probability* of getting heads is 50%. Similarly, while one can't know whether ignition will occur during a particular test, one can estimate the probability of ignition for that case through multiple tests, and use this probability to make meaningful decisions. In doing so one treats all of the tests conducted at a particular parameter combination as a binomial random variable with an unknown probability of ignition p that depends on the parameters of interest. Following the discussion of Hosmer et al., it is instructive to compare the analysis of categorical experiments to the analysis of more familiar experiments where the outcome is a continuous variable like time or species concentration [7]. In the latter case, the relationship between the independent and dependent variables is often found using linear least-squares (LLS) regression. The analogous regression technique for binary data is *logistic regression*, but the same general steps are taken in both types of regression.

For a least-squares regression, we assume that the relationship between the dependent variable Y and the independent variables $\mathbf{x}$ is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \tag{2.3.1}$$

where the variables $\mathbf{x}$ may or may not be interrelated and $\boldsymbol{\beta}$ are the constants to be fit by the regression. Implied in this relationship is that all of the variables involved are unbounded (i.e. can take any value between ∞ and $-\infty$). However, in the case of logistic regression the relationship between the probability of success (e.g. ignition) and the independent variables is more restricted because p can only vary between 0 and 1. The results discussed in the previous chapter indicate that there are conditions in which flaming ignition simply will not occur (i.e. $p = 0$) and conditions where it will always occur (i.e. $p = 1$). In between these regions of the parameter space, the ignition probability transitions smoothly between 0 and 1, resulting in an s-shaped probability curve. An example of this is as shown in Figure 2.6.

The canonical function used to represent this behavior is the *logistic function*,

$$p(x_i) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2 \dots}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2 \dots}}. \tag{2.3.2}$$

Rearranging yields an expression for the *logit* of p,

$$\text{logit}(p(x_i)) \equiv \ln[\frac{p(x_i)}{1 - p(x_i)}] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \dots \tag{2.3.3}$$

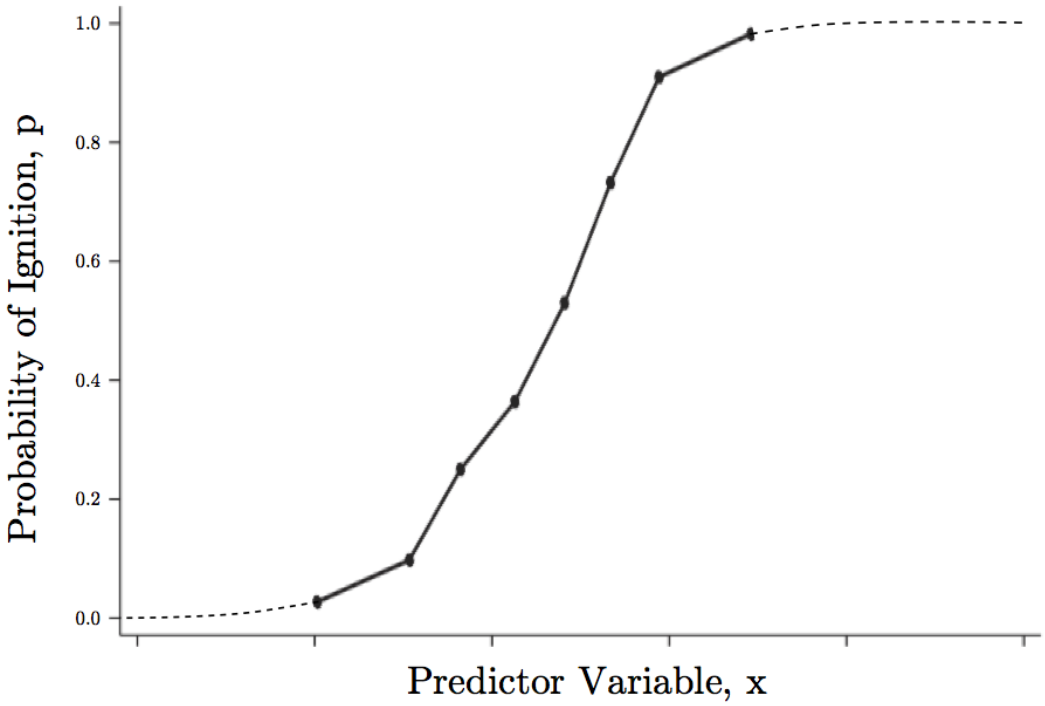

Figure 2.6: Example plot of ignition probability as a function of some predictor variable x, adapted from [7].

From equation 2.3.3 it is clear that logistic regression results in a specific case of the general linear model where the linearly-dependent variable is the unbounded logit of p as opposed to p itself. The next step in both least-squares and logistic regression is to use maximum likelihood estimation (MLE) to estimate the values of the fitting parameters $\boldsymbol{\beta}$.

In statistical inference, MLE is widely used to estimate distribution parameters (for example, the mean and standard deviation of a normal distribution) based on experimental sampling. Normally when considering random variables, the probability of a particular outcome is calculated for *a set of known parameters*. Going back to the binomial coin toss example, one can calculate the probability of getting 4 heads out of 5 tosses as

$$p(y = 4|n = 5, p = 0.5) = \frac{5!}{4!(5-4)!}0.5^4(1-0.5)^{(5-4)} = 0.156$$

On the other hand, likelihood is the probability that the underlying distribution has a particular parameter value given *a set of known outcomes*. So, if one gets 20 heads out

of 30 coin tosses, one can calculate the likelihood that the coin is fair (i.e. $p = 50\%$) as

$$l(p = 0.5|y = 20, n = 30) = \frac{30!}{20!(30-20)!}0.5^{20}(1-0.5)^{(30-20)} = 0.028$$

While the equations for probability and likelihood are equivalent, the information one has before making the calculation is different. Assuming the distribution governing a particular data set is known, MLE determines the parameter values that maximize the likelihood function of the data. Maximum likelihood estimators for statistical quantities have a number a desirable qualities compared to alternative estimators [7].

Given the similarity between the equation for logit(p) and the LLS relation, one might now expect that the regression procedure for the continuous and categorical variable would be identical. In LLS regression, the error between the experimental data and the true value of the dependent variable is assumed to be normally-distributed. In this case, minimizing the square of the residuals results in maximum likelihood estimations for $\boldsymbol{\beta}$. However, for categorical variables the probability of success is binomially, not normally, distributed. As such, least-squares regression applied to categorical data results in estimators that do not maximize likelihood of the experimental data, unless very large sample sizes are used [8]. Thus a more general approach must be taken to maximize the likelihood. This involves solving the system of equations

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_i} = 0, \ \frac{\partial^2 L(\boldsymbol{\beta})}{\partial \beta_i^2} < 0 \tag{2.3.4}$$

where $\boldsymbol{\beta}$ are the i unknown fitting constants and $L(\boldsymbol{\beta})$ is the log of the likelihood function. The log-likelihood function is used because it is easier to work with and its maximum coincides with the maximum of the likelihood function.

In this work, the tests conducted for each metal type were considered as a separate statistical data set. Within each of these data sets, each particle diameter-temperature pair was assumed to be represented by a binomial random variable. Thus, the $\boldsymbol{\beta}$ found for each metal type had to maximize the joint likelihood function, i.e. the product of all the component binomial likelihoods,

$$L(p|\boldsymbol{y}) = \prod_{i=0}^{N} \frac{n_i!}{y_i!(n_i - y_i)!} p_i^{y_i}(1-p_i)^{n_i - y_i}. \tag{2.3.5}$$

where each y_i is the number of ignitions out of n_i tests and p_i is the value of the unknown ignition probability p for that test condition. Again, p is related to the independent predictors $\mathbf{x}$ through the logistic function. Numerical methods are required to solve the system of equations shown in 2.3.4, but a number of commercial packages are currently available. In this particular case, the author used MATLAB software for the statistical analysis.

In both LLS and logistic regression, once the fitting parameters are determined, it is necessary to assess the fit of the assumed model. In least-squares regression, model fit is achieved with an R^2 value that is acceptably close to 1. If one choice for $\mathbf{x}$ results in a low R^2 value, different functional forms can be assumed (linear, polynomial, power law, etc.). In logistic regression there is no metric directly analogous to the R^2 value, and goodness-of-fit test statistics must be used. For binomial data, the Pearson χ^2 test and the deviance are the appropriate goodness-of-fit statistics, but the Pearson χ^2 test is more robust to sparseness. It can be defined as

$$P_{\chi^2} = \sum_{i=1}^{n} \frac{(p_i - \hat{p}_i)^2}{\hat{p}_i} \tag{2.3.6}$$

where p_i is the probability at each test condition according to the proposed distribution and $\hat{p}_i$ is the observed probability of ignition [9]. However, when a data set is sufficiently sparse, this test statistic can deviate strongly from the expected χ^2 distribution [8]. While sparseness is poorly defined, concerns over the statistically small sample size and the potential sparsity of the ignition data presented below led the author to use a bootstrap sum-of-squares test statistic with a p-value of 0.05 [9]. Bootstrapping is a way to find the approximate distribution of a random variable or test statistic when its functional form is unknown. The process leverages the power of a computer to perform artificial sampling, and uses the outcomes of these tests to build the unknown distribution.

After proposing particular forms for the various x_i and using MLE to estimate the corresponding β_i, the bootstrapping procedure was a follows:

- Numerically generate a random data set $y_i{}^*$ (of the same size as the real data set) from the proposed distribution.
- Using this generated data, perform MLE to get new values for $\boldsymbol{\beta}$ (denoted $\boldsymbol{\beta}^*$).
- Use the generated data set $y_i{}^*$ and the $\boldsymbol{\beta}^*$ to calculate a value for the Pearson sum-of-squares.
- Perform the first three steps a large number of times (e.g. 20,000).
- Build a frequency distribution for the sum-of-squares statistic based on the results of the numerical sampling.

- Compare the sum-of-squares value calculated for the true experimental data set to this frequency distribution. If the probability of producing that sum-of-squares value from the proposed distribution is less than 5%, the proposed form of $\mathbf{x}$ is rejected.

To find the approximate ignition probability distribution, different $\mathbf{x}$ were tried until the MLE algorithm yielded an acceptable goodness of fit according to the bootstrap procedure. It is interesting to note that the bootstrap statistic always agreed with the Pearson χ^2 test, indicating that the data was not overly sparse. Once the author found an approximate ignition probability distribution, the flaming ignition boundaries were defined as where the ignition probability was 5% (bounding the no-ignition region) and 95% (bounding the strongly flaming region).

The need to guess the initial form of the distribution makes it difficult to achieve a good fit with many data sets, especially those with a large amount of scatter or complicated underlying physics. In this work, the ignition results for aluminum spheres were sufficiently complex such that we never achieved an acceptable goodness of fit. Returning to the LLS comparison, even if least-squares regression results in unacceptably low R^2-values, one can estimate the true value of Y for each set of parameters by taking sample averages $\hat{Y}$. These averages can then be used to draw (by eye) a trend line describing the relationship between the dependent and independent variables. Similarly, if one desires to define an ignition boundary but an acceptable goodness-of-fit cannot be achieved with logistic regression, one can estimate the true ignition probability p at each set of conditions with the *observed* ignition probability $\hat{p}$, given by

$$\hat{p} = \frac{\text{number of tests resulting in flaming ignition}}{\text{number of total tests}}. \tag{2.3.7}$$

Parameter combinations that result in a $\hat{p}$ of 0% and 100% can be used to draw conservative ignition boundaries. Given that logistic regression failed to achieve a good fit for the results of all metal types, this approach was used to find ignition boundaries that could be compared.

The drawback of this approach for ignition data is that $\hat{p}$ is estimated treating the experiments at each test condition as an *independent* binomial random variable. Conducting tests across broad parameter ranges requires a relatively low number of tests to be performed at each parameter combination, resulting in large uncertainty in the probability estimation. This is demonstrated by the confidence intervals shown in Table 2.1 for a sample size of 5. Even so, later sections will show that the 'drawn-by-eye' and logistic regression boundaries for the stainless steel data agree fairly well, giving a greater degree of confidence in the boundaries found for the other three metal types.

Table 2.1: Clopper-Pearson confidence intervals for ignition probability at a 90% level of confidence for 5 tests.

y_i, number of ignitions	$\hat{p_l}$, lower confidence bound	$\hat{p_u}$, upper confidence bound
0	0	0.451
1	0.010	0.657
2	0.076	0.811
3	0.189	0.924
4	0.343	0.990
5	0.549	1.000

2.4 Effect of particle diameter and temperature

2.4.1 Results

For the purposes of understanding the effect of particle size and temperature, only the tests conducted with stainless steel particles will be discussed. As mentioned in the previous section, the use of logistic regression to find the approximate probability distribution was successful for this data set, making it ideal for identifying underlying trends. The results discussed here are also applicable to the tests conducted with other metal types. The ignition results are shown in Figure 2.7. Each circle shown denotes a combination of sphere diameter and temperature. The color of the circle indicates $\hat{p}$, the observed probability of ignition, as scaled by the associated color bar. Thus, a completely black circle corresponds to test conditions where flaming ignition was never observed and a white circle corresponds to a test condition where all of the tests ignited.

The approximate distribution of ignition probability was found to be

$$p = \text{logistic}(\beta_1 d_p + \beta_2 (T_p - 575)^{-3/2} + \beta_3 d_p T_p + \beta_4) \tag{2.4.1}$$

where p is the probability of ignition and d_p and T_p are the diameter (in mm) and temperature (in degrees Celsius) of the sphere, respectively. The values of $\beta_1 - \beta_4$ are shown in Table 2.2. As mentioned in section 2.3, the values of $\boldsymbol{\beta}$ were determined by maximum likelihood estimation. The functional form of the distribution was chosen by inspection such that the goodness of fit criteria was met. Along with the circles denoting observed ignition probability, figure 2.7 shows 5% and 95% probability curves according to the approximate distribution. These probability curves can be thought of as useful ignition limits, bounding regions of the parameter space where flaming ignition is very unlikely (No Ignition) and very likely (Flaming Ignition) to occur. The regression appears to fit the data well, and clarifies the probability scatter that results from random variation in the experiment.

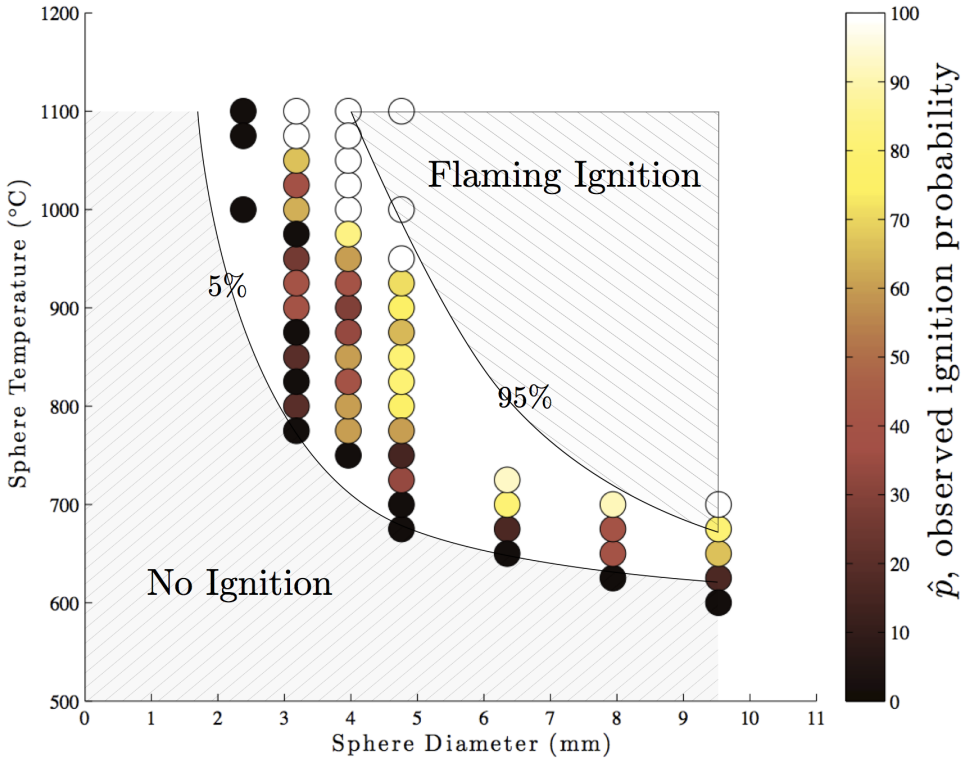

Figure 2.7: Observed ignition probability and approximate ignition limits for stainless steel spheres.

Table 2.2: Fitting constants for ignition probability regression

Constant	Value
β_1	-0.74
β_2	$-2.09 \cdot 10^3$
β_3	$-3.00 \cdot 10^{-3}$
β_4	-7.14

2.4.2 Discussion

Qualitatively, the results in Figure 2.7 show that spheres of decreasing size require higher temperatures to ignite, which agrees with previous studies [3, 10]. It is also clear that for very small spheres ($d_p < 3.5$ mm), increasing sphere temperature by a set increment (e.g. 100°C) has a mild effect, but increasing diameter by 1 mm has a marked one. Alternatively, for the largest spheres ($d_p > 8.5$ mm) temperature appears to have a much stronger effect than diameter. For medium sized spheres it appears that both sphere parameters are important

for determining ignition. The trend shown for small spheres and the overall hyperbolic shape of the no-ignition boundary agree qualitatively with the SECV study [1]. It is also similar to the boundary found by Hadden et al., but it exhibits a stronger curvature [3]. Recent experiments in the author's laboratory indicate that ignition temperature is sensitive to both fuel bed bulk density and the depth of the fuel bed. This, along with improvements in the determination of the particle temperature, explain the quantitative differences between these results and previous results from the author's laboratory [3, 10]. One also notes that the width of the transition region between the flaming and no ignition zones varies with diameter, suggesting that ignition is more sensitive to random effects (local fuel bed density, impact dynamics, etc) for medium and small sized spheres.

One possible explanation for the different sensitivities to sphere diameter for different sized particles is that there is some minimum bulk energy that a sphere must have in order to cause ignition. However, Hadden et al. showed that this is not the case [3]. Calculating sphere energy based on the current ignition results presented above confirms that ignition does not depend on energy alone. In this case bulk initial energy is calculated as

$$E_p = \frac{\pi}{6}(\rho c)_{p,st} d_p^3 (T_p - T_\infty)$$

where the product $(\rho c)_{p,st}$ is the volumetric heat capacity of steel and T_∞ is the baseline temperature of the system (here assumed to be room temperature). A 7 mm sphere with an initial temperature of 600°C contains 406 J, almost twice the energy in a 5 mm particle at 1100°C. And yet, the smaller particle will ignite and the larger particle will not. However, the next section will show that, while it is not solely capable of predicting ignition, bulk sphere energy is in fact related to ignition propensity.

2.5 Effect of particle metal type

2.5.1 Results

The experimental results for particle metal type are shown in Figure 2.8. Each of the sub-figures (a-d) corresponds to all of the tests conducted with a particular metal. Similar to Figure 2.7, the coloring denotes observed ignition probability and follows the center color bar. As discussed in the previous section, the boundaries shown were drawn based on the estimated location of 0% and 100% ignition probability cases. As shown in Figure 2.9, there is reasonable agreement between the stainless steel ignition boundaries found by logistic regression and those drawn by eye based on observed ignition probability. While the agreement is worse at smaller diameters, the overall trends are accurately represented and give credence to the boundaries drawn for the other metals. Overall the materials follow a similar trend. There is a hyperbolic relationship between particle size and temperature, with

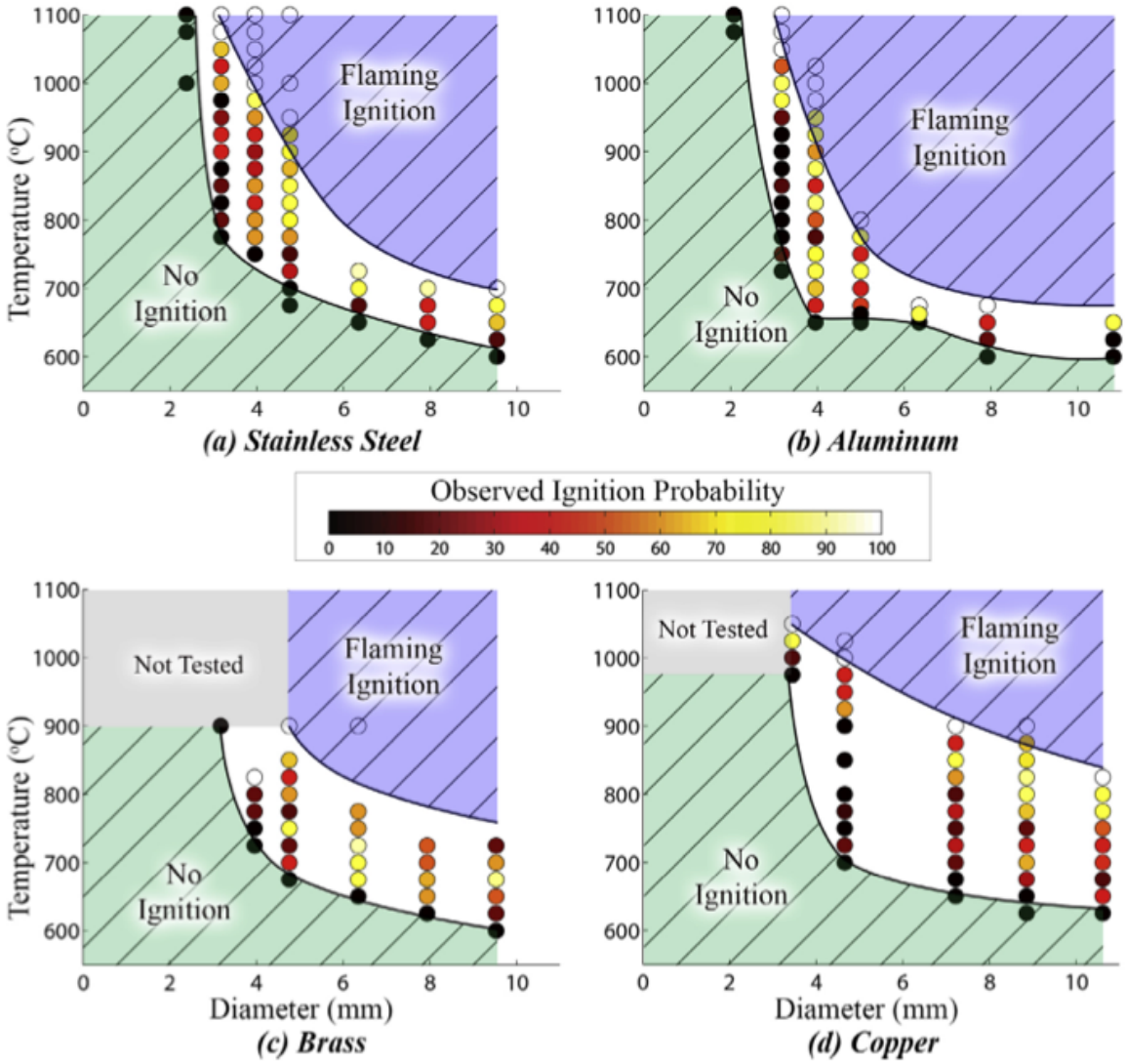

Figure 2.8: Observed ignition probabilities for all four metal types.

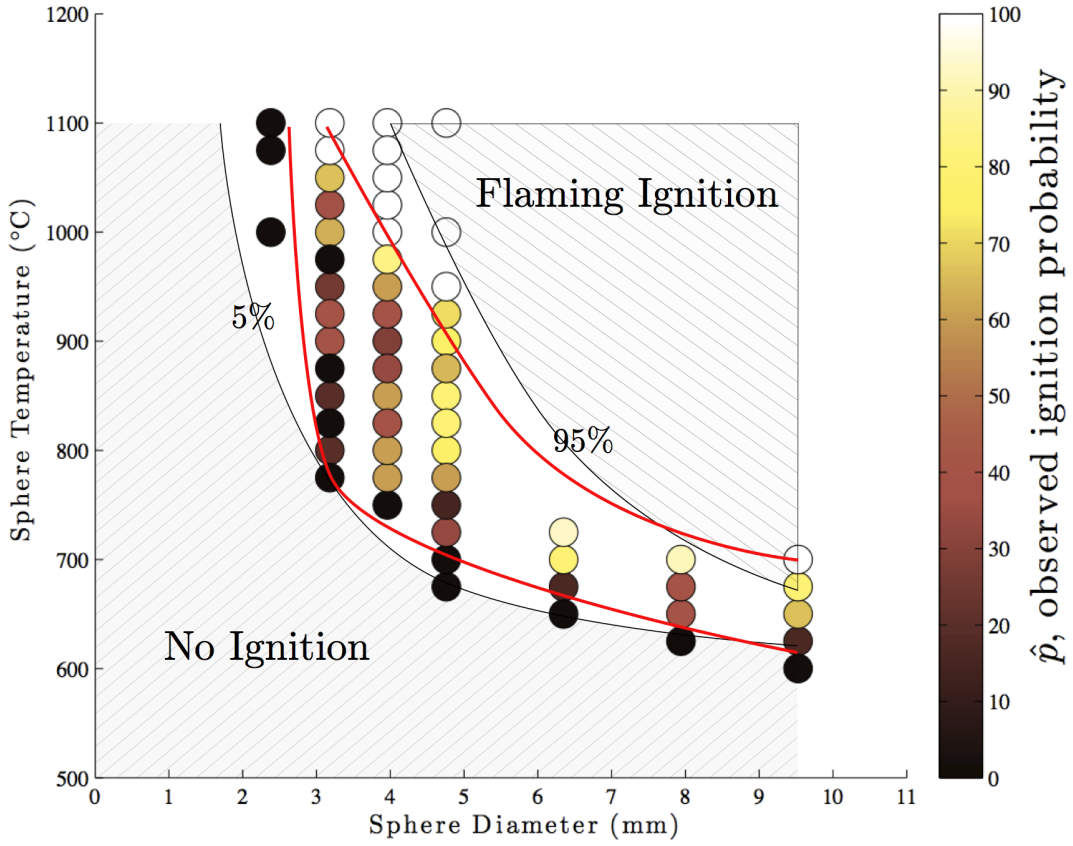

Figure 2.9: Results from Figure 2.7 with drawn-by-eye boundaries overlaid.

the larger particles requiring lower temperatures to ignite the cellulose bed. One marked difference between materials was the range of temperatures over which the observed chance of ignition was between 0% and 100%, for a given size particle. For large size particles, all of the materials except copper had a relatively small range of temperatures where the observed ignition probability was between 0 and 1 while for copper it was considerably larger. In the case of aluminum, additional tests were done with a finer temperature resolution because in some cases the range over which ignition occurred was so narrow.

2.5.2 Discussion

In addition to being representative of real-world hot particles, the four different alloys demonstrate the effect, or lack thereof, of several thermal properties. As seen in Table 2.3, the most notable difference between aluminum 1100 and the other metals is that its melting range is

within the temperature range studied. This allowed us to investigate the effect of particle melting and the additional energy imparted to the particle by phase change. Also of interest is the fact that, while the density and specific heat of the metals vary independently, the product of the density and specific heat $(\rho c)_p$ (more important when considering thermodynamics and heat transfer) only varies somewhat between the 4 alloys. Emissivity, which gives a sense of relative radiative heat losses, varies slightly more. However, the thermal conductivity varies by an order of magnitude across the 4 alloys. Given its prominent role in thermal inertia $(\lambda\rho c)_p$ and thermal diffusivity $\left(\lambda/\rho c\right)_p$, one would expect that its effect would dominate differences in the ignition boundaries not attributed to melting. Figure 2.10 shows

Table 2.3: Thermal properties of Stainless Steel 302, Aluminum 1100 (solid and liquid properties), Copper 110 and Brass 260.

Particle Material	Stainless Steel 302	Al 1100 (solid)	Al 1100 (molten)	Cu 110	Brass 260
T_{sol}, **Solidus temperature (°C)**	1400	643	643	1065	915
T_{liq}, **Liquidus temperature (°C)**	1420	657	657	1083	955
ΔH_m, **Heat of melting (kg/m^3)**	2122	1057	1057	1822	1433
ρ, **Density (kg/m^3)**	7860	2710	2375	8890	8530
c, **Specific heat capacity (J/kgK)**	500	900	1141	385	375
(ρc), **Volumetric heat capacity (MJ/m^{3}K)**	3.9	2.4	2.7	3.4	3.2
ϵ, **Emissivity (approx.)**	0.9	0.2	0.2	0.8	0.6
λ, **Thermal conductivity (W/mK)**	21.5	220	90.7	390	120

the no-ignition limits for each of the four metals. All the metals show similar minimum ignition temperatures for a given diameter, despite differences in the metals' thermal properties. What variations do exist do not show any clear trend with volumetric heat capacity, emissivity, or thermal conductivity. Copper does however exhibit some noticeable deviation in ignition behavior; specifically, copper particles require slightly larger temperatures to ignite than the other metals. At temperatures above the liquidus temperature of aluminum, 657°C, the thermal conductivity and thermal diffusivity of copper are at least a factor of three larger than those of the other materials. From this, one can infer that small changes in thermal conductivity have little effect, but that large changes result in faster heat losses, especially from small particles. The modeling results discussed in the next chapter indicate that small metal particles are thermally lumped and conductivity should be unimportant, but this may not accurately represent the ignition experiments. Thus, the deviation of the copper limit may be attributable to its high thermal conductivity, but the exact reason for it remains unclear. It may even be associated with the formation of cuprous oxide on the surface of the particle during particle heating, but additional work is required to determine if this is the case.

Furthermore, examining the aluminum ignition boundary, it can be seen that there is a sharp corner at a diameter of approximately 3.5 mm. Between 4 mm and 7 mm diameters

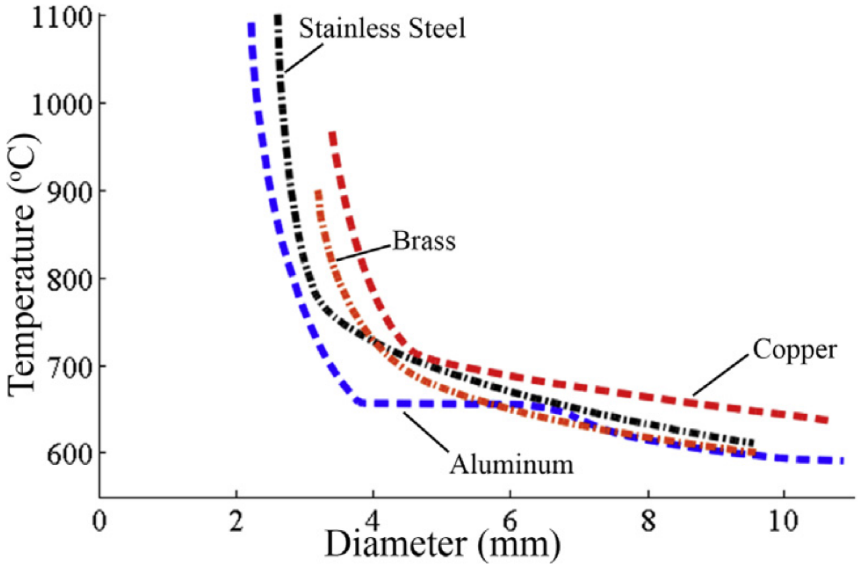

Figure 2.10: No-ignition limits of the four metals tested

the ignition limit follows the melting temperature range and 4 mm aluminum particles ignite at temperatures roughly 100°C lower than their stainless steel counterparts. For diameters greater than 7 mm the ignition limit drops below the melting temperature. Clearly, melting has a significant effect on the potential for the particles to ignite for diameters in the range of 3.5 - 7 mm. Possible explanations for this behavior are that molten particles change shape upon impact or that molten and solid aluminum have different thermal properties. The most likely explanation is that molten particles have considerably greater energy than particles that are not molten at the same temperature. Comparing the volumetric heat of melting to the volumetric heat capacity shows that the latent heat of melting adds energy equivalent to the energy gained from heating the aluminum particle an additional 433°C. Thus, a completely molten aluminum particle has considerably more energy than a barely molten particle despite the temperature difference of only 15°C. This indicates that, while bulk particle energy is not solely capable of determining ignition propensity, bulk energy does affect ignition behavior over some range of particle size. Plotting the bulk sphere energy along the no-ignition boundaries for the 4 alloys confirms that this is the case (see Figure 2.11). The energy values shown in Figure 2.11 were calculated assuming constant material properties for each material phase. The energy released from melting was treated as a linear variation over the melting temperature range. All energy values were calculated as energy above the ambient temperature.

For particle sizes greater than 7 mm, Figure 2.10 shows that there is less than 100°C difference in the temperature required for ignition between the different alloys or diameters.

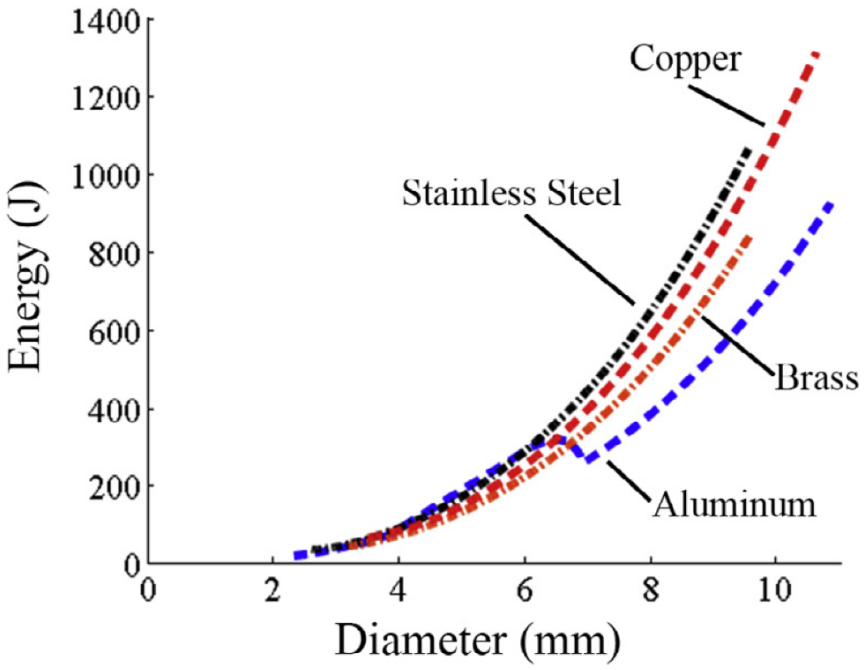

Figure 2.11: Bulk initial energy vs. particle diameter along the no ignition limit

However, Figure 2.11 shows that the aluminum particles, due to their relatively low volumetric heat capacity, have significantly less energy in that size range. This indicates that particle temperature is more important than energy or size for determining ignition by larger particles. For particle diameters less than 7 mm, the energy along the ignition boundary is similar for all 4 metals, and in the case of aluminum this is due to the additional energy from phase change. This indicates that particle energy and size are more important than temperature for determining ignition by smaller particles. Despite the additional energy from melting, as size decreases the aluminum ignition temperature boundary behaves more and more like the others, indicating that the contribution from the melting energy is not as important. For these sizes, increasingly higher temperatures (i.e. more sensible energy in the molten phase) are needed to ignite the fuel. Chapter 3 reveals why large-particle ignition is sensitive to temperature, while small-particle ignition requires a critical bulk energy. The effect of melting and why it varies with particle size is also discussed.

2.6 Ignition phenomenology based on observation

Following completion of the experiments, the high-speed Schlieren video footage was analyzed. The events described here will largely focus on ignition by stainless steel spheres, but these observations are descriptive of ignition by particles of other metal types. Pre-

ceding this discussion, it should be noted that the Schlieren technique is ideal for imaging two-dimensional phenomena within a test plane normal to the system's line of sight. In many practical systems, including the one used in this work, the viewing plane is replaced by a "viewing volume"; phenomena at different depths of field become superimposed into a single image, reducing the acquirable spatial information in any given frame. Despite this drawback, schlieren imaging yielded important observations. The small bright circle that appears near the lower middle of all frames is an artifact of the Schlieren imaging technique and is not related to the ignition process.

A general sequence of events is common to all recorded tests with solid spheres (refers to Figure 2.12). First, the sphere, surrounded and trailed by a Schlieren contour, is seen impacting the fuel bed (frames I-II). All spheres bounced to some degree upon impact with the fuel bed (frames III). Smaller spheres generally bounced higher, but the penetration depth, exact bounce height, and number of residual bounces all appear to be random. This random behavior is assumed to be captured by the statistical approach applied in this work. Shortly after impact, a Schlieren contour expands away from the sphere (frames III), indicating heat and/or pyrolyzate gas diffusing away from the sphere into the surrounding air. In general, heat and mass transfer appear to be diffusion-dominated, with little advective mixing observed. For sphere temperatures of 750°C or less, a dark fluid plume also em-

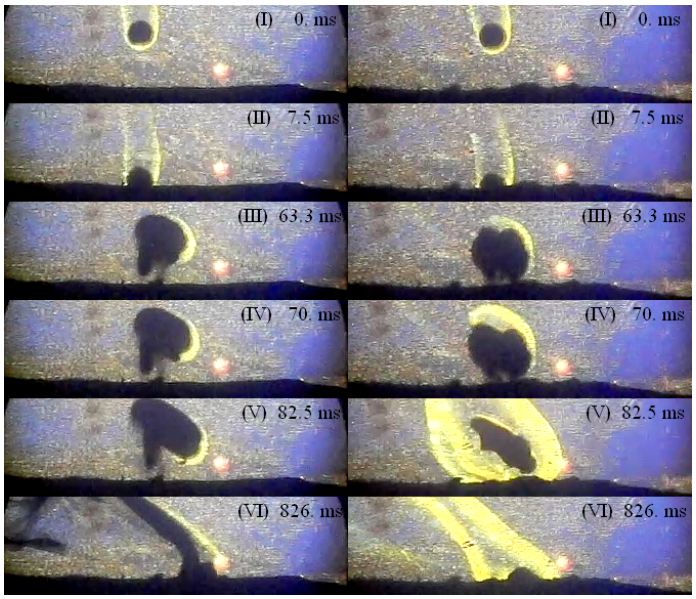

Figure 2.12: No-ignition (left) and ignition (right) events for stainless steel spheres with $d_s = 6.35$ mm and $T_s = 675°$ C.

anates from the cellulose that is near the sphere surface (see figure 2.12). A dark appearance in a Schlieren image means an object is opaque, suggesting liquid components (condensed pyrolysis products or fuel-bound water) and/or solid components (ash or lofted unreacted

cellulose). The current resolution of the videos (0.22 mm) means that the presence of small lofted solids cannot be resolved. The condensed pyrolysis products are mainly composed of anhydrosugars and are referred to as 'tar' [11].

Melting has a noticeable effect on the impact behavior of the molten aluminum spheres, although this effect is less pronounced for smaller spheres (see Figure 2.13). All of the molten aluminum spheres have an thin outer layer of solid aluminum oxide due to being heated in an oxidizing environment (i.e. the open air furnace). This layer allows the particles to keep a spherical shape, at least up to the point of impact, and the smaller spheres seem to retain their spherical shape even after impact. The solid aluminum spheres appear to exhibit the same phenomenology as the steel spheres (left side of figure 2.13). On the other hand, the molten spheres impact much more violently, and loft large cellulose particles, as seen in the rightmost portions of frames II-V in figure 2.13. Despite keeping their shape, even small molten spheres create a large dark plume, and when ignition occurs it appears to initiate at the edges of this plume; this suggests that lofted cellulose dust may be igniting. Interestingly, the time to ignition for the solid and molten spheres is very similar (frame V, figure 2.13). It is unclear how much this different impact behavior would increase ignition propensity; the model results presented in Chapter 3 suggest that the shape of the aluminum ignition boundary can be largely explained by the added heat of melting alone. The particle lofting, enhanced mixing and heat transfer that result from the flattening of the sphere may be balanced by an increase in the surface area to volume ratio of the shape. This would result in increased losses due to radiation and convection, such that there would be little net effect.

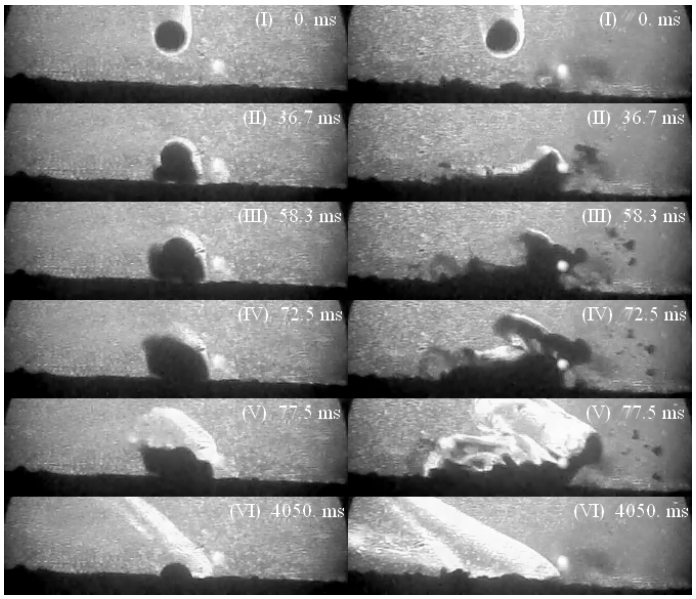

Figure 2.13: Ignition events for 7.92 mm aluminum spheres: molten at 675 °C (left) and solid at 625 °C (right).

When ignition does occur, it can be generally categorized as one of two different cases regardless of sphere diameter or temperature. In the first case, ignition happens shortly after an impact and is observed as a rapid expansion of the relatively thin schlieren contour surrounding the sphere (see the right side of figure 2.12 and figure 2.14). As seen in figure 2.14,

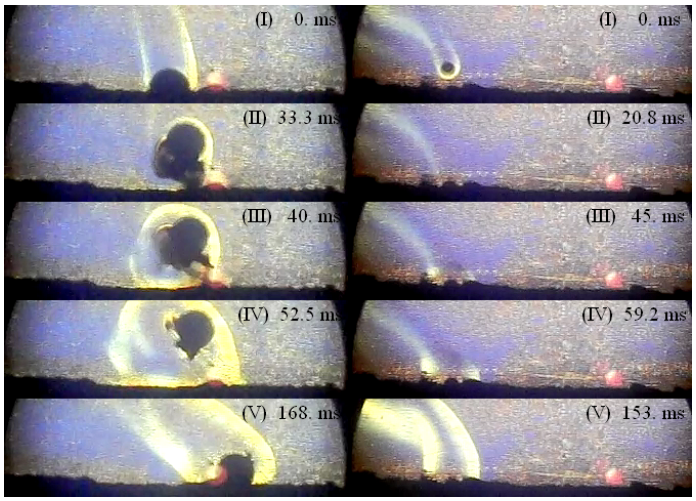

Figure 2.14: Examples of the first ignition scenario: a 9.53 mm sphere heated to 700°C (left) and a 3.18 mm sphere heated to 1100°C (right).

for the smallest spheres the schlieren intensity does not greatly increase and the expansion appears relatively weak (frames III-IV, right side), but for the largest spheres the schlieren intensifies and expands violently (frames III-IV, left side). This seems due to the fact that different sphere sizes results in different amounts of gasified fuel present at the time of ignition. In fact, this "flashing" is also observed in several of the no-ignition events with smaller spheres, sometimes when the sphere is mid-bounce and other times when it is in contact with the bed. In these situations the flash is unable to anchor to the fuel bed and form a stable flame, due to an insufficient amount of fuel. This suggests that the particle must be capable of pyrolyzing the fuel even after ignition occurs, at least until thermal feedback from the flame can take over the process.

In the second case, ignition occurs much later after the sphere has settled and the schlieren contour has formed a stable plume. Ignition is observed as a bright new schlieren contour that forms within the plume and rapidly expands to match the shape of the plume before continuing out into the surrounding gas. In this second scenario, ignition may initiate away from the ball, and sometimes along the leading edge of the schlieren plume, as shown in frames III-V in figure 2.15. This delayed ignition was also observed by Silver during his gas ignition experiments, but only for larger spheres [12]. Both of the observed ignition scenarios imply that a reacting layer of gaseous fuel and air is formed (a thin layer near the sphere in the first scenario and along the boundary of the plume in the second) and that ignition occurs within this layer and not at the surface of the fuel bed. The particular test shown

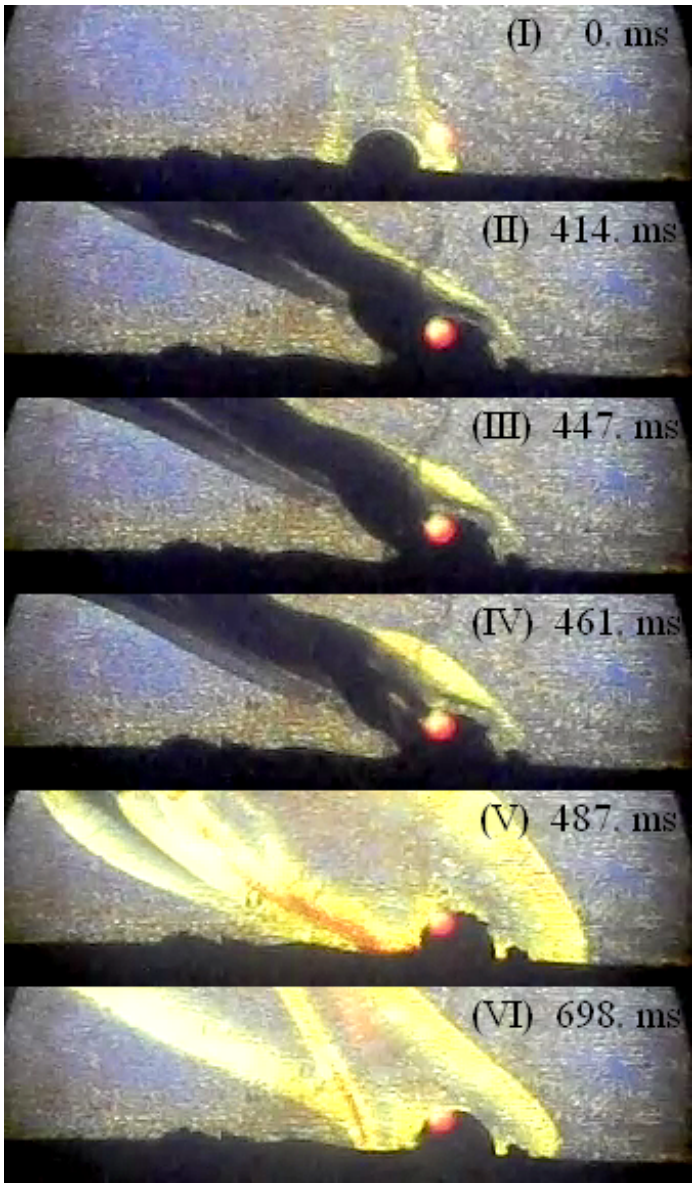

Figure 2.15: Example of second ignition scenario: a 9.53 mm sphere heated to 700°C.

in figure 2.16 clearly demonstrates both of these scenarios. After the initial flash occurs and fails to anchor (frames II-III), the sphere settles and a stable plume forms (frames IV-V). Ignition then occurs within this plume (frame VI) and the flame is able to anchor to the bed.

In both ignition scenarios, after the flame initiates it anchors to the fuel bed and grows out into the cross flow, becoming a diffusion flame (frame VI, right hand side of figure 2.12). In cases of no-ignition, a stable schlieren contour eventually extends out into the cross flow

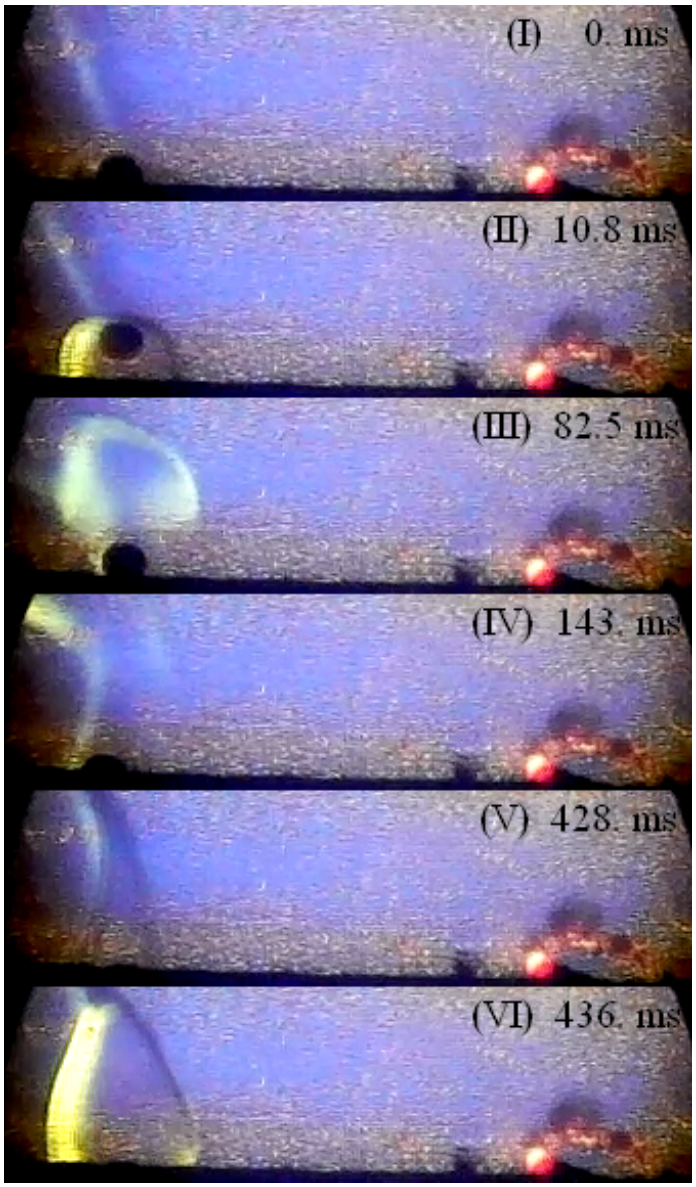

Figure 2.16: A 4.76 mm sphere heated to 775°C demonstrating a flash ignition followed by a plume ignition.

from the leading edge of the sphere (frame VI, left hand side of figure 2.12). Depending on the temperature of the sphere, this contour is accompanied by a dark plume as described above. Both structures fade once the sphere has cooled sufficiently.

Most ignition and no-ignition events for a given d_s and T_s appear very similar up to the point of ignition, as in figure 2.12. A particularly interesting case of this similarity is shown in figure 2.17. Both spheres are 3.18 mm and heated to 800°C, and both impact and bounce in

a very similar manner (frames I-III). After the spheres come to rest, the Schlieren contours appear to form a stable plume and then begin to fade in intensity (frames IV). At some later time, both videos show an increase in schlieren intensity (frames V) and one eventually forms a rapidly expanding inner contour characteristic of ignition (frame VI, left side). The other eventually dissipates (frame VI, right side), implying that in both cases an exothermic reaction began to take off but local variations produced different outcomes. This highlights the tenuous nature of the ignition process and the following chapter discusses how small deviations in particle temperature (like those that are within experimental uncertainty) can effect ignition by smaller spheres.

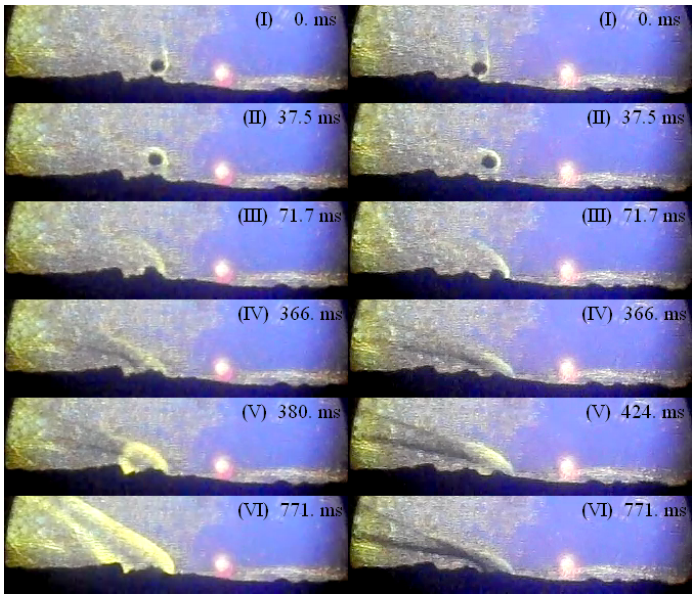

Figure 2.17: Two 3.18 mm spheres heated to 800°C; after initial similarity one ignites and the other does not.

In closing, ignition within the mixing layer occurs when heat generated by the incipient combustion reaction is greater than the losses to the surrounding air and the particle. The temperature near the surface of very large spheres remains near its impact value, and particle thermal properties are relatively unimportant. Therefore, larger spheres are capable of producing larger amounts of pyrolyzate, but a high heat generation rate requires a sufficiently initial temperature. Thus, ignition by large spheres is more sensitive to temperature. As the size of the sphere decreases, particle cooling over the timescales of ignition becomes more significant. Heat losses from the mixing layer to the particle become more important and increasingly high temperatures are needed for heat generation rates to overcome these losses. Changing thermal properties and phenomena such as melting effect the degree of particle heat loss, but to varying degrees. The next chapter elaborates and corroborates these findings with results from the theoretical model.

2.7 References

[1] G.E. Pleasance. *An Examination of Particles from Conductor Clashes as Possible Sources of Bushfire Ignition, App. 5*. Tech. rep. unpublished. Victoria, Australia: State Electricity Commission of Victoria, 1977.

[2] T. Tanaka. "On the Flammability of Combustible Materials by Welding Splatter". In: *Reports of the Natl. Res. Inst. of Police Sci.* 30 (1977), pp. 151–158.

[3] R.M. Hadden et al. "Ignition of Combustible Fuel Beds by Hot Particles: An Experimental and Theoretical Study". In: *Fire Tech,* 47 (2011). http://dx.doi.org/10.1007/s10694-010-0181-x, pp. 341–355.

[4] V. Babrauskas. *Ignition Handbook: principles and applications to fire safety engineering, fire investigation, risk management and forensic science*. Issaquah, Washington: Fire Science Publishers, 2003.

[5] A.W. Burner and W.K. Goad. *Gladstone-Dale Constant for CF_4*. Tech. rep. 80228. Hampton, Virginia: Langley Research Center, National Aeronatics and Space Administration, 1980.

[6] G.S. Settles. *Schlieren and Shadowgraph Techniques*. First. Berlin, Germany: Springer-Verlag, 2001.

[7] D.W. Hosmer, S. Lemeshow, and R.X. Sturdivant. *Applied Logistic Regression*. Third. John Wiley and Sons, Inc., 2013.

[8] R.B. Millar. *Maximum Likelihood Estimation and Inference*. West Sussex, UK PO19 8SQ: John Wiley and Sons, Ltd, 2011.

[9] M. von Davier. "Bootstrapping Goodness-of-Fit Statistics for Sparse Categorical Data, Results of a Monte Carlo Study". In: *Meth. Psych. Res. Online* 2 (1997).

[10] C.D. Zak, D.C. Murphy, and A.C. Fernandez-Pello. "Understanding Ignition of Natural Fuels by Heated Particles". In: *WIT Trans. on Built Env.* 134 (2013). Safety and Security Engineering V, Editors F. Garzia, C.A. Brebbia and M. Guarascio, WIT Press, pp. 607–614.

[11] J. Cho, J.M. Davis, and G.W. Huber. "The Intrinsic Kinetics and Heats of Reactions for Cellulose Pyrolysis and Char Formation". In: *ChemSusChem* 3 (2010). http://dx.doi.org/10.1002/cssc.201000119, pp. 1162–1165.

[12] R.S. Silver. "The Ignition of Gaseous Mixtures by Hot Particles". In: *Philos. Mag. and J. Sci.* 23 (1937), pp. 633–657.

Chapter 3

ANALYSIS OF IGNITION PHYSICS

3.1 Introduction

When collected, the ignition data and Schlieren images presented in the previous chapter showed clear trends, but these trends had multiple potential explanations. In situations like these, it is useful to develop a mathematical model of the physical system; one can then see how quantities of interest change with space and time and investigate conditions that are difficult to investigate experimentally. This chapter presents a theoretical model of hot particle spot fire ignition developed by the author, and the results of investigations performed with the model.

In general, the modeling approach was to simplify the problem down to its most important physics and use this simple model to better understand the ignition process and experimentally observed trends. Based on the spherical geometry of the particle, the system is modeled as a thermally lumped particle halfway embedded in two 1D hemispherical domains: a solid fuel and a gas phase. Treating this problem as an approximately 1D system requires one to neglect the non-radial motion of pyrolyzate around the particle. Instead, the model assumes that the pyrolyzate flows towards the surface of the particle in the solid fuel and originates from very near the particle surface in the gas phase. Buoyant flow is neglected since transport in the Schlieren images appears to be diffusion dominated. These and other approximations are discussed in greater detail in the following section.

Once the model was developed, it was implemented with a fully-implicit iterative finite volume scheme, largely following the approach of Patankar and later Lautenberger [1, 2]. The author then used it to analyze the ignition physics. The experimental results of Chapter 2 indicate that the physics controlling ignition may be different for particles of different sizes. To explore this idea further, the model was used to simulate ignition by particles of different sizes, and investigate how the ignition process changes for different size regimes. Then, the author ran a large number of simulations to find a flaming ignition boundary like

those found experimentally. In all of these simulations, the various fuel bed, particle and ambient properties were chosen to approximate the stainless steel and powdered cellulose experiments. This allows one the opportunity to compare model and experimental results, assess the validity of the model framework and get a better idea of what physics are important to the process. From there, one can identify aspects of the problem that require further investigation.

After the simulations of ignition by stainless steel particles were conducted, the model was used to explore the effect of various particle properties on the flaming ignition limit. Unencumbered by experimental limitations and noise, the results of these simulations confirm and clarify the experimental observations made in Chapter 2. In particular, several aspects that were simultaneously changed when changing particle metal type were investigated to determine their individual effects on the ignition process. Specifically, the author focused on the effects of particle volumetric heat capacity, emissivity, and the possibility of particle melting.

As mentioned, the model is described in more detail in section 3.2. Section 3.3 derives and nondimensionalizes the governing equations for heat and mass transfer in the solid fuel and gas domains. Section 3.4 describes the numerical implementation of the equations and their numerical solution. With the model completely characterized, the final two sections present and discuss the results of the investigations carried out with the model.

3.2 Model description

The system under study is the simplified case of a spherical particle igniting a solid but porous natural fuel. We model the system as two semi-infinite hemispherical domains (the solid fuel bed and the surrounding gas phase) with inner boundaries at the particle surface. Figure 3.1 illustrates the model system. The boundary between these domains is assumed to be adiabatic and closed, except where they are connected by the particle. For this study the particle is assumed to be embedded halfway in the fuel bed such that the fraction of particle surface area contacting each domain is 0.5. These simplifications imply that fluid flow/diffusion is only in the radial direction, which agrees qualitatively with early time in the Schlieren videos. For simplicity, it is assumed that once the vapor leaves the solid phase, it flows around the particle and enters the gas domain at the boundary of the particle and with the surface temperature of the particle. The solid fuel and the gas domain are initially at room temperature (298 K), as are the surroundings.

The solid is considered dry (i.e. moisture content of 0%) and solid phase bulk density is taken to be constant. Also, fuel bed porosity is not explicitly accounted for except that the bulk density is lower than the solid nonporous density of the fuel. Solid phase specific heat is assumed constant. Fuel thermal conductivity is comprised of a solid and radiative component; the solid component has a fixed value while the radiative component depends

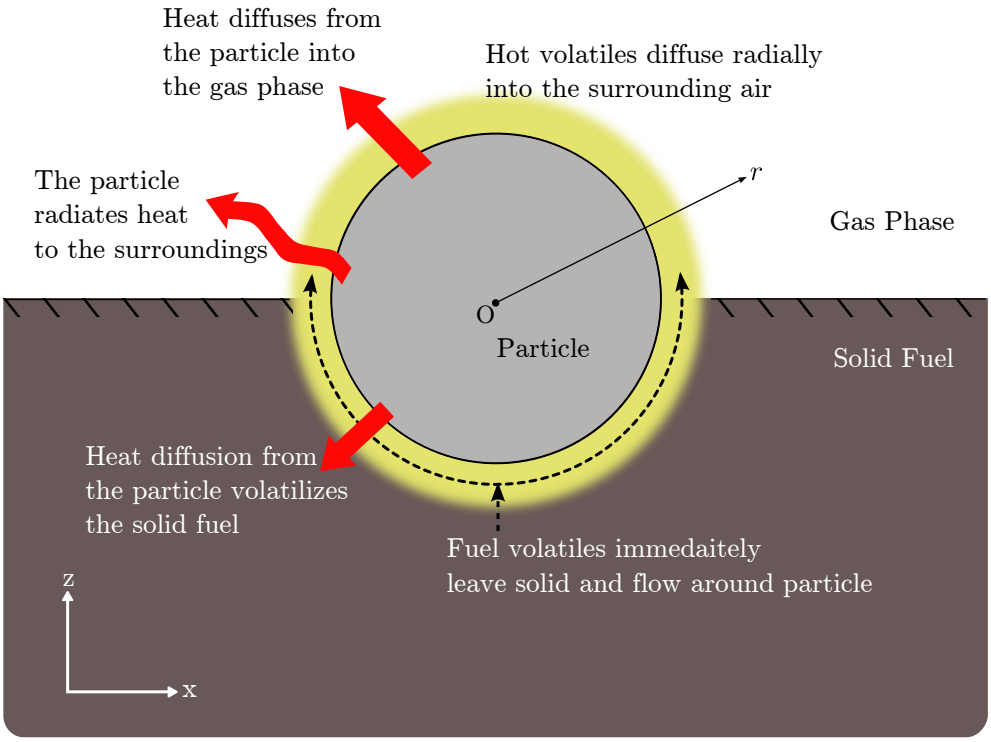

Figure 3.1: Model system.

on temperature [2]. The solid fuel decomposes endothermically, and even if it is a charring material it is assumed that the high temperatures and heat fluxes that occur near the particles of interest would leave negligible amounts of char. Oxidative pyrolysis reactions are also neglected.

As such, the solid fuel is treated as undergoing a single-step volatilization reaction

$$\text{solid fuel} \rightarrow \text{volatiles} \tag{3.2.1}$$

with no intermediate fraction. The vapor is assumed to instantly leave the solid domain. Because there is only one solid-phase species (the virgin fuel), conservation of mass is equivalent to conservation of species (i.e. there is one less equation to solve for the solid-phase). Fuel vaporization in a 1D spherical system requires the fuel bed to recede in the radial direction. In reality, the particle would sink into the fuel bed in the negative z direction (see Figure 3.1), but we assume for early times and small motions of the particle that the downward motion of the ball can be approximated by 'radial' motion. In the reference frame of the particle, the virgin solid fuel is moving towards the particle at a velocity such that the mass within the solid fuel domain is conserved. The motion of the particle relative to the gas

phase (downward or radially) is neglected.

Pressure in both the solid fuel and gaseous phases is taken to be approximately constant at 1 atm due to the open nature of the system. Considering times before actual ignition, density in the gas domain is assumed to be constant as that of air at STP. All other gas properties (including those of the fuel vapor) are assumed to be equivalent across species and to take the values of pure nitrogen where applicable. A unity Lewis number is assumed, and the ratio $\frac{\lambda_g}{c_g} = \rho_g D_g$is treated as constant given its insensitivity to pressure and weak dependence on temperature [3]. However, thermal conductivity and specific heat are individually allowed to vary with temperature. Flow within the gas phase is considered inviscid and irrotational and buoyancy effects are neglected. As such, the velocity field can be described by the potential flow due to a spherical source. In this case it is not necessary to solve the momentum conservation equation, as the velocity field can be determined from the continuity equation. Combustion in the gas phase is modeled with a second order global one step reaction

$$\text{volatiles} + \nu_o \cdot \text{oxygen} \rightarrow \text{products} \tag{3.2.2}$$

where $\nu_o$is the number of kilograms of oxygen consumed per kilogram of volatiles. The gas initially surrounding the particle and fuel bed is treated as air with composition of 23% oxygen and 77% nitrogen by weight. Thus, the gas domain is treated as containing only three species: fuel, oxygen, and inerts (nitrogen and combustion products).

The particle itself is treated as lumped body, due to the particle sizes of interest and the relatively high conductivity of metals. The validity of this assumption will be explored in later sections. The particle is capable of melting over a certain temperature range, but the specific heats of the solid and molten phases are assumed equal and constant with respect to temperature. Other particle properties including density and thermal conductivity are assumed constant. Surface oxidation or any other processes that might change the particle size are neglected. The particle loses heat through conduction into the two domains, through radiation to the surroundings, and due to the convective flux of fuel vapor flowing from the solid to the gas domain. The gray body assumption is made for the radiation behavior of the particle and the total emissivity is assumed constant. The particle acts as an interface condition between the solid fuel and gas phase domains through the lumped heat equation and as such the inner boundary temperature of the two domains is equal to the particle temperature.

3.3 Model formulation

In the following subsections, the conservation equations are derived by applying Reynolds Transport Theorem to a single one-dimensional control volume at location $r$and with thick-

ness δr.For the present simple 1D system, this reduces to setting the sum of the inflow, outflow, and generation terms equal to the storage term. Although the control volume is actually a hemispherical shell, it is drawn as an incomplete centerline slice of the shell (i.e. a circular arc) as shown in Figure 3.2. The normals (denoted by n) to the two surfaces at

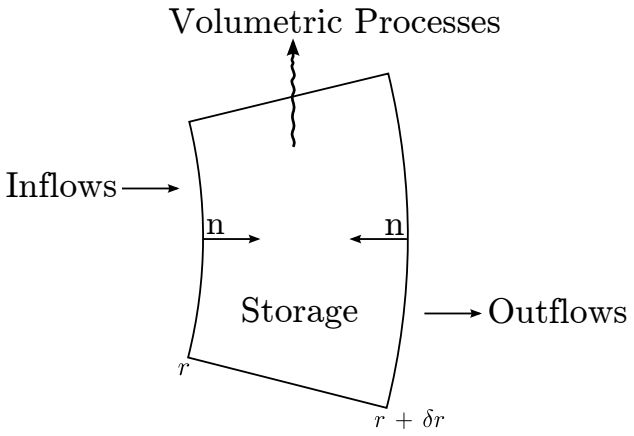

Figure 3.2: Representative differential volume.

r and $r + \delta r$ point inward by convention. Flows (of mass, species or energy) across these two surfaces are denoted by straight arrows, while oscillating arrows indicate volumetric processes due either to source terms (formation or destruction of some quantity per unit volume per unit time) or transfer of mass from the solid fuel to the gaseous phase.

The subsections deriving the conservation equations are bookended by three additional subsections: the first defines the many variables used in the other subsections, and the last two derive the particle interface conditions and nondimensionalize the governing equations to yield characteristic dimensionless groups.

3.3.1 Preliminaries and definitions

This subsection outlines the model variables and defines several of the thermal properties used throughout the remainder of the chapter. In the subsections that follow, ρ denotes density, u denotes velocity,uppercase T denotes temperature, and lowercase t denotes time.As mentioned previously, r is the radial space coordinate. The mass fraction of gaseous species $i$is represented as Y_i and defined as

$$Y_i = \frac{\rho_i}{\rho_g}$$

where ρ_g is the constant mixture density in the gas phase. Generally, subscript $s$refers to a solid phase variable, and subscript g refers to a gas phase variable. The other subscripts used to differentiate quantities are: p for particle, ∞ for ambient, f for fuel volatiles,o for oxidizerand n for inerts and product species.

Enthalpy (denoted h) is made up of a heat of formation (defined at some reference temperature) and a sensible enthalpy component, as in

$$h(T) = h_{\mathbf{f}}^{\circ} + h_{sen}(T) = h_{\mathbf{f}}^{\circ} + \int_{T_{ref}}^{T} c(T')\,\mathrm{d}T'. \tag{3.3.1}$$

For the purposes of this model, the reference temperature will be taken as room temperature, 298K. In subsection 3.3.3, it proves useful to use the general form for h_s given by equation 3.3.1. However, in the gas phase, the assumption of equal specific heats means that all gaseous species have identical sensible enthalpies. Thus, during the combustion reaction, the only enthalpy change is due to the difference between the heats of formation of the reactants and products; there is no change in sensible enthalpy. If one defines a heat of combustion, Q, as

$$Q = h_{\mathbf{f},f}^{\circ} + h_{\mathbf{f},o}^{\circ} - h_{\mathbf{f},n}^{\circ}\,,$$

then the gas mixture enthalpy h_g can be replaced by the mixture sensible enthalpy $h_{g,sen}$ in the conservation equations. For simplicity, in the remainder of the chapter the *sen* subscript will be dropped such that $h_g = h_{g,sen}$.

As described in the previous section, the solid fuel thermal conductivity is assumed to be composed of a constant 'solid' component and a temperature-dependent radiative component,

$$\lambda_s(T_s) = \lambda_{s,sol} + \lambda_{s,rad}(T_s).$$

The definition of $\lambda_{s,rad}$ is taken from Di Blasi,

$$\lambda_{s,rad}(T_s) = \frac{13.5\sigma d_{pore} {T_s}^3}{\epsilon_{pore}},$$

where σ is the Stefan-Boltzmann constant, d_{pore} is the estimated pore diameter in the solid fuel, T_s is the solid fuel temperature and ϵ_{pore} is the pore emissivity [4]. In the gas phase, both the thermal conductivity and the specific heat vary with temperature according to the

NASA polynomials for nitrogen. The functional forms of these relations are

$$\lambda_g = \exp\left(A \ln T_g + \frac{B}{T_g} + \frac{C}{{T_g}^2} + D\right),$$

$$c_g = R_u \left(a_1 + a_2 T_g + a_3 {T_g}^2 + a_4 {T_g}^3 + a_5 {T_g}^4\right),$$

where R_u is the universal gas constant and $a_{1...5}$ and A, B, C, and D are fitting constants whose values can be found in the literature [5].

Particle melting occurs over a melting range ΔT_m and with a heat of melting ΔH_m. The heat of melting is assumed to apply linearly over the melting range, such that it can be represented by a specific heat of melting,

$$c_{p,m} = \frac{\Delta H_m}{\Delta T_m}.$$

Based on the assumption that both the solid and molten particle phases have the same constant specific heat, an effective particle specific heat, $c_{p,eff}$, can be defined as

$$c_{p,eff}(T_s) = \begin{cases} c_{p,m} & \text{for } T_{sol} \leq T_s < T_{liq} \\ c_p & \text{all other } T_s \end{cases}$$

where T_{sol} and T_{liq} are the particle solidus and liquidus temperatures, respectively.

3.3.2 Solid fuel mass conservation

Consider a spherical differential volume in the solid phase domain, as shown in Figure 3.3. The term $\dot{m}_f$ represents the conversion of solid fuel into fuel volatiles that leave the volume and flow around the particle and into the gas phase. As discussed in the previous section, due to the non-charring nature of the solid at high temperatures and heat fluxes, the particle sinks into the fuel bed, continually encountering virgin fuel. If the differential volume is fixed in the reference frame of the particle, solid fuel 'flows' into and out of the volume so as to retain its original mass. This is represented by the quantities $\dot{m}_{s,r}$ and $\dot{m}_{s,r+\delta r}$. With the mass within the volume constant, the storage term $\frac{\partial \rho_s}{\partial t}$ is zero and summing the contributions

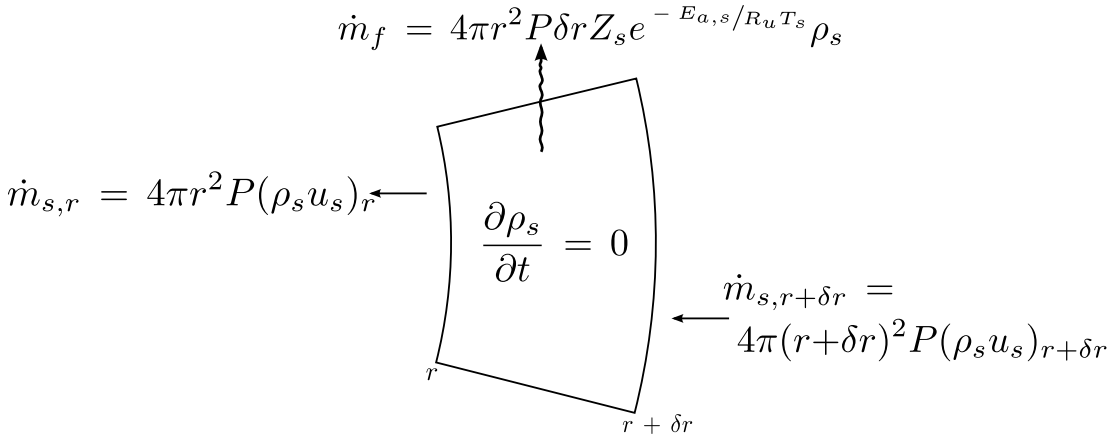

Figure 3.3: Solid fuel mass balance.

in Figure 3.3 yields

$$4\pi r^2 P(\rho_s u_s)_r - 4\pi(r+\delta r)^2 P(\rho_s u_s)_{r+\delta r} - 4\pi r^2 P \delta r Z_s e^{-E_{a,s}/R_u T_s} \rho_s = 0 \tag{3.3.2}$$

where P is the fraction of particle surface area contacting the solid fuel (equal to 0.5 but retained for generality), ρ_s is the virgin bulk density of the fuel, u_s is the local velocity relative to the particle, and T_s is the temperature of the solid within the volume. The volatilization reaction has an Arrhenius dependence on temperature, as indicated by the pre-exponential factor Z_s and the activation energy $E_{a,s}$. R_u is the universal gas constant. Dividing all terms by $4\pi r^2 P \delta r \rho_s$ and taking the limit where $\delta r \to 0$ gives the equation for the solid phase conservation of mass,

$$\frac{\partial u_s}{\partial r} + \frac{2u_s}{r} = -Z_s e^{-E_{a,s}/R_u T_s}. \tag{3.3.3}$$

This first order equation in space requires only a single boundary condition. In the reference frame of the particle, the fuel-particle interface is stationary, stated mathematically as

$$u_s(r_p) = 0 \tag{3.3.4}$$

where r_p is the radius of the particle. At each instant, the mass flowing into the solid domain across its open outer boundary ($r \to \infty$) is equivalent to the total mass leaving due to volatilization.

3.3.3 Solid fuel energy conservation

The energy balance shown in Figure 3.4 is similar to the mass balance of the preceding sub-section. However, in this case the flows into and out of the control volume are diffusive

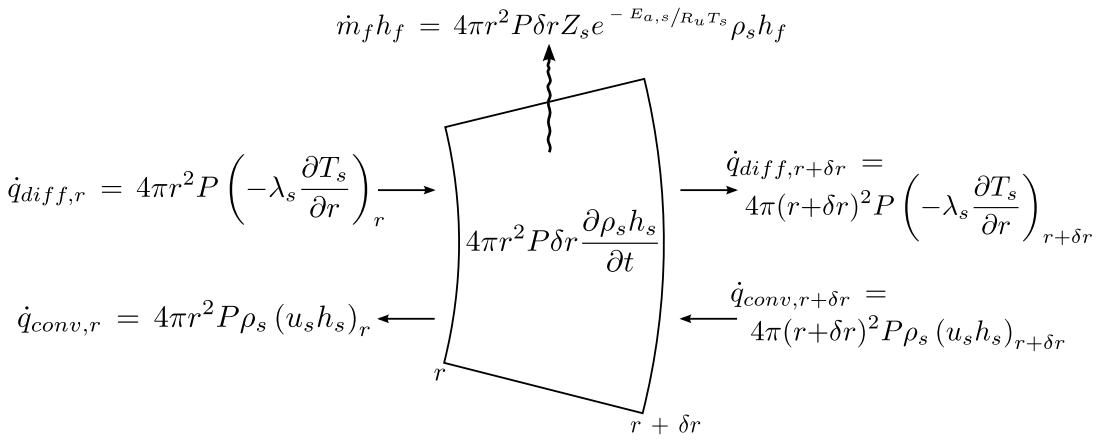

Figure 3.4: Solid fuel energy balance.

(denoted $\dot{q}_{diff}$) and convective (denoted $\dot{q}_{conv}$) in nature. The term $\dot{m}_f h_f$ represents the energy leaving the volume due to the outflow of volatiles. Note that kinetic and potential energy contributions have been neglected and that internal energy has been replaced by enthalpy based on the assumptions of constant pressure and density. The storage term can be simplified by expanding the derivative $\frac{\partial \rho_s h_s}{\partial t}$ and noting that $\frac{\partial \rho_s}{\partial t} = 0$, as in

$$\begin{aligned} 4\pi r^2 P \delta r \frac{\partial \rho_s h_s}{\partial t} &= 4\pi r^2 P \delta r \left[\rho_s \frac{\partial h_s}{\partial t} + h_s \frac{\partial \rho_s}{\partial t}\right] \\ &= 4\pi r^2 P \delta r \left[\rho_s \frac{\partial h_s}{\partial t} + 0\right] \\ &= 4\pi r^2 P \delta r \rho_s \frac{\partial h_s}{\partial t} \end{aligned}$$

Setting this storage term equal to the sum of the inflows and outflows, dividing by $4\pi r^2 P\delta r$ and rearranging yields

$$\rho_s \frac{\partial h_s}{\partial t} = \frac{\lambda_s}{\delta r}\left[\left(\frac{\partial T_s}{\partial r}\right)_{r+\delta r} - \left(\frac{\partial T_s}{\partial r}\right)_r\right] + \frac{2\lambda_s}{r}\left(\frac{\partial T_s}{\partial r}\right)_{r+\delta r} - \frac{\rho_s}{\delta r}\left[(u_s h_s)_{r+\delta r} - (u_s h_s)_r\right] - \frac{2\rho_s (u_s h_s)_{r+\delta r}}{r} - Z_s e^{-E_{a,s}/R_u T_s}\rho_s h_f \quad (3.3.5)$$

Taking the limit $\delta r \to 0$ and expanding derivatives, equation 3.3.5 becomes

$$\rho_s c_s \frac{\partial T_s}{\partial t} = \frac{\partial}{\partial r}\left(\lambda_s \frac{\partial T_s}{\partial r}\right) + \frac{2\lambda_s}{r}\frac{\partial T_s}{\partial r} - \rho_s h_s\left[\frac{\partial u_s}{\partial r} + \frac{2u_s}{r}\right] - \rho_s c_s u_s \frac{\partial T_s}{\partial r} - Z_s e^{-E_{a,s}/R_u T_s}\rho_s h_f \quad (3.3.6)$$

where c_s is the solid fuel specific heat and the relation $\partial h_s = c_s \partial T_s$ has been applied. Noting that the left hand side of equation 3.3.3 is contained in the third term on the right hand side of equation 3.3.6, the two equations can be combined and rearranged, as in

$$\rho_s c_s \frac{\partial T_s}{\partial t} = \frac{\partial}{\partial r}\left(\lambda_s \frac{\partial T_s}{\partial r}\right) + \frac{2\lambda_s}{r}\frac{\partial T_s}{\partial r} - \rho_s c_s u_s \frac{\partial T_s}{\partial r} - [h_f - h_s] Z_s e^{-E_{a,s}/R_u T_s}\rho_s. \quad (3.3.7)$$

The difference $h_f - h_s$ is the heat of the volatilization reaction given by equation 3.2.1. Although there is ambiguity in the literature concerning the term 'heat of reaction' for pyrolysis reactions, here the term includes the standard heat of reaction and the difference between the sensible enthalpies of the fuel and volatiles,

$$h_f - h_s = \Delta H_{vol} = \Delta H^\circ_{vol} + h_{f,sen} - h_{s,sen} \quad (3.3.8)$$

where the standard heat of reaction, ΔH°_{vol}, is the difference between $h^\circ_{\mathbf{f},f}$ and $h^\circ_{\mathbf{f},s}$, the respective heats of formation of the volatiles and solid fuel. In experimental studies aimed at determining solid phase heats of reaction, such as those using Differential Scanning Calorimetry (DSC) or Differential Thermogravimetric Analysis (DTA), ΔH_{vol} is the most commonly reported quantity [6]. Inserting ΔH_{vol} into equation 3.3.7 and dividing all terms by ρ_s yields

the final form of the conservation of energy equation,

$$c_s \frac{\partial T_s}{\partial t} = \frac{1}{\rho_s}\left[\frac{\partial}{\partial r}\left(\lambda_s \frac{\partial T_s}{\partial r}\right) + \frac{2\lambda_s}{r}\frac{\partial T_s}{\partial r}\right] - c_s u_s \frac{\partial T_s}{\partial r} - \Delta H_{vol} Z_s e^{-E_{a,s}/R_u T_s}. \tag{3.3.9}$$

Equation 3.3.9 is second-order in space and first-order in time, requiring two boundary conditions and one initial condition. The boundary condition at $r = r_p$ is provided by the particle interface conditions discussed in subsection 3.3.7. Because of the semi-infinite nature of the solid-phase domain, the additional boundary condition and the initial condition can be stated together as

$$T_s(t < 0, r) = T_s(t, r \to \infty) = T_\infty \tag{3.3.10}$$

where T_∞ is the ambient temperature (room temperature in this case).

3.3.4 Gas phase mass conservation

In the gas phase domain, no volumetric processes result in mass loss or gain, as shown in Figure 3.5. Given the assumption that gas density is constant, storage within the control

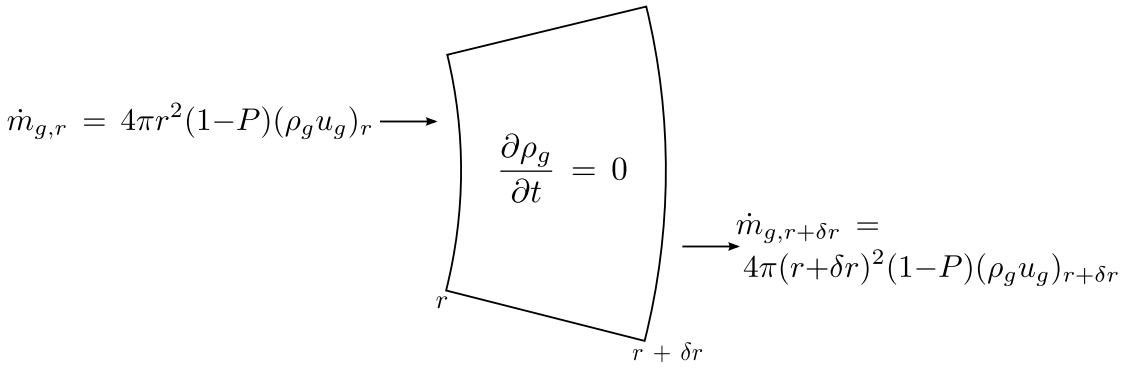

Figure 3.5: Gas phase mass balance.

volume is neglected and a mass balance yields

$$4\pi r^2(1-P)(\rho_g u_g)_r \; - \; 4\pi(r+\delta r)^2(1-P)(\rho_g u_g)_{r+\delta r} \; = \; 0. \tag{3.3.11}$$

Dividing all terms by $4\pi r^2(1-P)\delta r \rho_g$ and taking the limit where $\delta r \to 0$ gives the equation for the gas phase mass conservation,

$$\frac{\partial u_g}{\partial r} \; + \; \frac{2u_g}{r} \; = \; 0. \tag{3.3.12}$$

This first-order ordinary differential equation can be solved by separating and subsequently integrating the variables u_g and r,

$$\frac{\partial u_g}{\partial r} \; = \; -\frac{2u_g}{r}$$

$$\int -\frac{1}{2}\frac{\partial u_g}{u_g} \; = \; \int \frac{\mathrm{d}r}{r}$$

$$-\frac{1}{2}\ln u_g \; = \; \ln r + C$$

$$u_g(r) \; = \; \frac{C}{r^2} \tag{3.3.13}$$

where C is an unknown integration constant. The boundary condition required to find C and complete the solution results from equating the flow into the gas domain at the particle surface ($r = r_p$) to the total amount of volatiles produced in the solid phase,

$$4\pi {r_p}^2(1-P)\rho_g u_g(r_p) \; = \; 4\pi P \rho_s \int_{r_p}^{\infty} Z_s e^{\,-\,E_{a,s}/R_u T_s} \; r'^2 \, \mathrm{d}r'. \tag{3.3.14}$$

Equation 3.3.14 can be rearranged and simplified to

$$u_g(r_p) = \left(\frac{P}{1-P}\right)\left(\frac{\rho_s}{\rho_g}\right)\frac{\int_{r_p}^{\infty} Z_s e^{-E_{a,s}/R_u T_s}\ r'^2\,\mathrm{d}r'}{{r_p}^2}. \tag{3.3.15}$$

$u_g(r_p)$ can then be used to find C,

$$\begin{aligned} C &= u_g(r_p){r_p}^2 \\ &= \left(\frac{P}{1-P}\right)\left(\frac{\rho_s}{\rho_g}\right)\int_{r_p}^{\infty} Z_s e^{-E_{a,s}/R_u T_s}\ r'^2\,\mathrm{d}r'. \end{aligned}$$

With C determined, the equation for the velocity field in the gas phase is

$$u_g(r) = \left(\frac{P}{1-P}\right)R\ \frac{\int_{r_p}^{\infty} Z_s e^{-E_{a,s}/R_u T_s}\ r'^2\,\mathrm{d}r'}{r^2}, \tag{3.3.16}$$

where R is defined as ρ_s/ρ_g. As such, the mass flowing out of the gas domain across its open outer boundary ($r \to \infty$) is equivalent to the mass of volatiles entering at the particle surface.

3.3.5 Gas phase energy conservation

In the gas phase, unlike in the solid phase, there is no energy leaving the system due to mass loss. Instead, there is a volumetric source, $\dot{H}_{comb}$ due to the exothermic combustion reaction, as shown in Figure 3.6. Here, ω_f is the volumetric consumption rate of the fuel volatiles,

$$\omega_f = Z_g(\rho_g Y_f)(\rho_g Y_o)e^{-E_{a,g}/R_u T_g} \tag{3.3.17}$$

where Y_f and Y_o are the mass fractions of the fuel and oxidizer, respectively. As mentioned previously, Q is the heat of combustion. Again note that kinetic and potential energy contributions have been neglected and that the constant density assumption has been used to simplify the storage term. Summing the terms shown in Figure 3.6 and dividing by

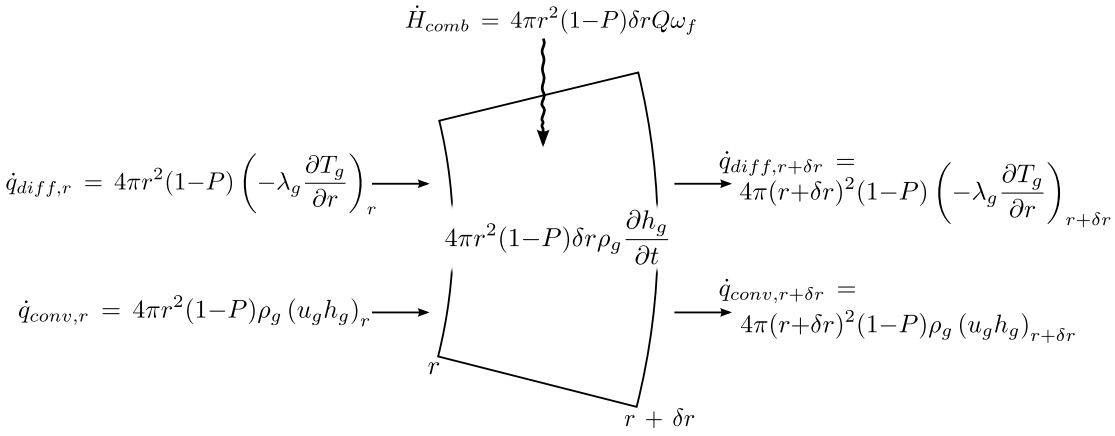

Figure 3.6: Gas phase energy balance.

$4\pi r^2(1-P)\delta r$ gives

$$\rho_g \frac{\partial h_g}{\partial t} = \frac{\lambda_g}{\delta r}\left[\left(\frac{\partial T_g}{\partial r}\right)_{r+\delta r} - \left(\frac{\partial T_g}{\partial r}\right)_r\right] + \frac{2\lambda_g}{r}\left(\frac{\partial T_g}{\partial r}\right)_{r+\delta r}$$

$$- \frac{\rho_g}{\delta r}\left[(u_g h_g)_{r+\delta r} - (u_g h_g)_r\right] - \frac{2\rho_g (u_g h_g)_{r+\delta r}}{r} + Q\omega_f \qquad (3.3.18)$$

Taking the limit $\delta r \to 0$, applying gas phase continuity, and dividing all terms by ρ_g yields

$$c_g \frac{\partial T_g}{\partial t} = \frac{1}{\rho_g}\left[\frac{\partial}{\partial r}\left(\lambda_g \frac{\partial T_g}{\partial r}\right) + \frac{2\lambda_g}{r}\frac{\partial T_g}{\partial r}\right] - c_g u_g \frac{\partial T_g}{\partial r} + Q Z_g \rho_g Y_f Y_o e^{-E_{a,g}/R_u T_g}. \qquad (3.3.19)$$

As in the solid phase, the boundary condition for equation 3.3.19 at $r = r_p$ comes from the particle interface conditions. The far field boundary condition and the initial condition can be stated together as

$$T_g(t < 0, r) = T_g(t, r \to \infty) = T_\infty \qquad (3.3.20)$$

3.3.6 Gas phase species conservation

The control volume balance for the gas phase species i is shown in Figure 3.7. In the

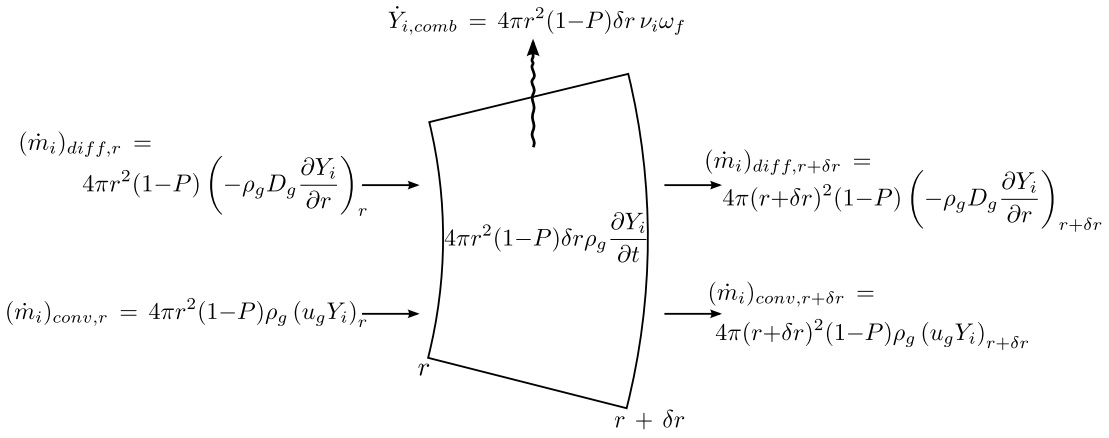

Figure 3.7: Gas phase species balance.

expression for species consumption rate, $\dot{Y}_{i,comb}$, ν_i is 1 for the fuel species, and ν_o for oxygen. ω_f is the fuel consumption rate defined in the previous subsection. Species conservation equations are only required for these two species because the mass fraction of inerts can always be found from the relation

$$Y_f + Y_o + Y_n = 1, \tag{3.3.21}$$

where the subscript n denotes inerts. The terms of the species balance are analogous to those of the gas phase energy balance, where $\dot{q}_{conv}$ and $\dot{q}_{diff}$ have been replaced by $\dot{m}_{i,conv}$ and $\dot{m}_{i,diff}$ and the transported scalar is now mass fraction instead of enthalpy or temperature. As such, the two equations for fuel mass fraction and oxygen mass fraction conservation can

be written by inspection from equation 3.3.19,

$$\frac{\partial Y_f}{\partial t} = \alpha_g \left[\frac{\partial^2 Y_f}{\partial r^2} + \frac{2}{r}\frac{\partial Y_f}{\partial r} \right] - u_g \frac{\partial Y_f}{\partial r} - Z_g \rho_g Y_f Y_o e^{-E_{a,g}/R_u T_g}, \tag{3.3.22}$$

$$\frac{\partial Y_o}{\partial t} = \alpha_g \left[\frac{\partial^2 Y_o}{\partial r^2} + \frac{2}{r}\frac{\partial Y_o}{\partial r} \right] - u_g \frac{\partial Y_o}{\partial r} - \nu_o Z_g \rho_g Y_f Y_o e^{-E_{a,g}/R_u T_g}. \tag{3.3.23}$$

In equations 3.3.22 and 3.3.23, the unity Lewis number assumption (i.e. $\lambda_g/c_g = \rho_g D_g$) has been been applied and $\alpha_g = \lambda_g/\rho_g c_g$ is the thermal diffusivity of the gas mixture. To find the boundary conditions for the two species at $r = r_p$, one first considers the equation for total mass flow,

$$\begin{aligned} \dot{m}_i &= 4\pi(1-P)r^2 \left[\dot{m}''_{i,conv} + \dot{m}''_{i,diff} \right] \\ &= 4\pi(1-P)r^2 \left[\rho_g u_g Y_i - \rho_g D_g \frac{\partial Y_i}{\partial r} \right] \\ &= 4\pi(1-P)r^2 \left[(\dot{m}''_f + \dot{m}''_o + \dot{m}''_n) Y_i - \left(\frac{\lambda_g}{c_g} \right) \frac{\partial Y_i}{\partial r} \right], \end{aligned} \tag{3.3.24}$$

where $\dot{m}''$ indicates a mass flux. All of the mass flowing into the gaseous domain at the particle surface is fuel, such that $\dot{m}''_o(r_p) = \dot{m}''_n(r_p) = 0$. Thus, equation 3.3.24 can be rearranged for the fuel and oxidizer species as

$$\left(\frac{\partial Y_f}{\partial r} \right)_{r_p} = \left[\left(\frac{\lambda_g}{c_g} \right)^{-1} \frac{\dot{m}_f(r_p)}{4\pi(1-P){r_p}^2} \right] \left(Y_f(r_p) - 1 \right) \tag{3.3.25}$$

$$\left(\frac{\partial Y_o}{\partial r} \right)_{r_p} = \left[\left(\frac{\lambda_g}{c_g} \right)^{-1} \frac{\dot{m}_f(r_p)}{4\pi(1-P){r_p}^2} \right] Y_o(r_p) \tag{3.3.26}$$

As discussed in subsection 3.3.4, the total mass of fuel volatiles flowing from the solid

phase into the gas phase is

$$\dot{m}_f(r_p) \;=\; 4\pi P \rho_s \int_{r_p}^{\infty} Z_s e^{\,-\,E_{a,s}/R_u T_s}\; r'^2 \,\mathrm{d}r'. \tag{3.3.27}$$

Plugging equation 3.3.27 into equations 3.3.25 and 3.3.26 and rearranging gives the final form for the inner boundary conditions for the fuel and oxygen species,

$$\left(\frac{\partial Y_f}{\partial r}\right)_{r_p} \;=\; \left[\left(\frac{P}{1-P}\right)\left(\frac{R}{\alpha_g}\right)\frac{\int_{r_p}^{\infty} Z_s e^{\,-\,E_{a,s}/R_u T_s}\; r'^2 \,\mathrm{d}r'}{{r_p}^2}\right]\big(Y_f(r_p) - 1\big) \tag{3.3.28}$$

$$\left(\frac{\partial Y_o}{\partial r}\right)_{r_p} \;=\; \left[\left(\frac{P}{1-P}\right)\left(\frac{R}{\alpha_g}\right)\frac{\int_{r_p}^{\infty} Z_s e^{\,-\,E_{a,s}/R_u T_s}\; r'^2 \,\mathrm{d}r'}{{r_p}^2}\right] Y_o(r_p) \tag{3.3.29}$$

The far field boundary condition and the initial condition for the two species can be written together as

$$Y_f(t<0, r) \;=\; Y_f(t, r\to\infty) \;=\; 0 \tag{3.3.30}$$

$$Y_o(t<0, r) \;=\; Y_o(t, r\to\infty) \;=\; Y_{o,\infty} \;=\; 0.23 \tag{3.3.31}$$

3.3.7 Particle interface conditions

The particle interface conditions required to solve the two conservation of energy equations (3.3.9 and 3.3.19) are derived by considering the lumped particle and a region of thickness δr surrounding it (see Figure 3.8). The upper boundary of this region is in perfect contact with the gas domain, and the lower boundary is in perfect contact with the porous solid fuel. Within this narrow region, there are only the fuel volatiles that are leaving the solid phase, flowing around the particle, and then entering the gas phase. If one draws a control volume

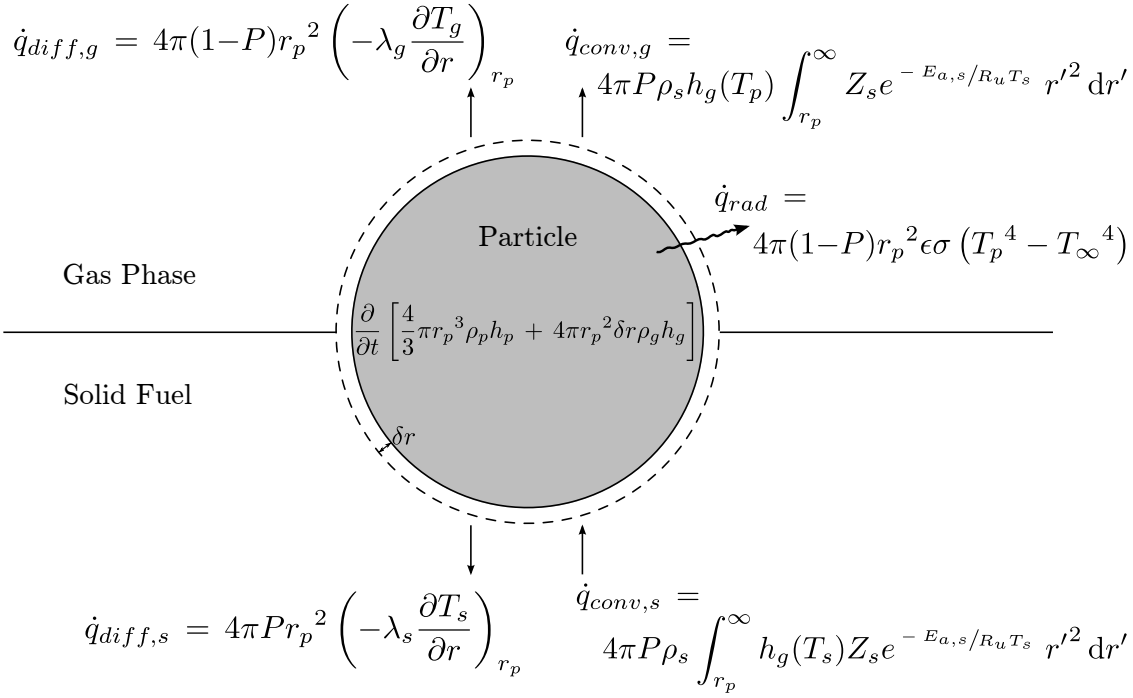

Figure 3.8: Energy balance for hot particle and narrow adjacent region.

that surrounds the particle and the narrow zone, an energy balance yields

$$\begin{aligned}\frac{\partial}{\partial t}\left[\frac{4}{3}\pi{r_p}^3\rho_p h_p + 4\pi{r_p}^2\delta r\rho_g h_g\right] = {} & 4\pi P{r_p}^2\left(\lambda_s\frac{\partial T_s}{\partial r}\right)_{r_p} + 4\pi(1-P){r_p}^2\left(\lambda_g\frac{\partial T_g}{\partial r}\right)_{r_p} \\ & + 4\pi P\rho_s\int_{r_p}^{\infty} h_g(T_s) Z_s e^{-E_{a,s}/R_u T_s}\, r'^2\,\mathrm{d}r' \\ & - 4\pi P\rho_s h_g(T_p)\int_{r_p}^{\infty} Z_s e^{-E_{a,s}/R_u T_s}\, r'^2\,\mathrm{d}r' \\ & - 4\pi(1-P){r_p}^2\epsilon\sigma\left({T_p}^4 - {T_\infty}^4\right), \end{aligned} \tag{3.3.32}$$

where ϵ is the particle emissivity and σ is the Stefan-Boltzmann constant. If one assumes that $\delta r \lll r_p$, the second term on the left hand side can be neglected and the particle can be assumed to be in thermal contact with the solid fuel and gas phase. Rearranging, simplifying

and dividing all terms by $4\pi {r_p}^2$ transforms equation 3.3.32 into

$$\frac{r_p \rho_p c_{p,eff}}{3} \frac{\partial T_p}{\partial t} = P \left[\left(\lambda_s \frac{\partial T_s}{\partial r} \right)_{r_p} - \frac{\rho_s}{{r_p}^2} \int_{r_p}^{\infty} (h_g(T_p) - h_g(T_s)) Z_s e^{-E_{a,s}/R_u T_s} \, {r'}^2 \, \mathrm{d}r' \right]$$

$$+ (1 - P) \left[\left(\lambda_g \frac{\partial T_g}{\partial r} \right)_{r_p} - \epsilon \sigma \left({T_p}^4 - {T_\infty}^4 \right) \right]. \tag{3.3.33}$$

In equation 3.3.33, $c_{p,eff}$ accounts for the possibility of particle melting, as described in subsection 3.3.1. This equation alone does not sufficiently characterize the interface between the solid fuel and gas domains; two conditions are required, one for each domain. The additional condition is provided by considering that the particle is lumped and the two domains are in good thermal contact with the particle at $r = r_p$, such that

$$T_s(t > 0, r_p) = T_g(t > 0, r_p) = T_p. \tag{3.3.34}$$

The first-order time derivative in equation 3.3.33 also requires an initial condition for the particle temperature,

$$T_p(t < 0) = T_i \,, \tag{3.3.35}$$

where T_i is the initial 'impact' temperature of the particle.

3.3.8 Nondimensionalization of conservation equations

The preceding subsections derive 6 governing equations (2 in the solid phase and 4 in the gas phase) and numerous associated conditions that must be solved to study ignition in this model system. These equations can be nondimensionalized to give meaningful insight into the ignition process and ease numerical implementation of the model. Since ignition occurs within a gas phase mixing layer, it is intuitive to start by scaling the gas-phase variables by their characteristic values and then carry these definitions to the solid fuel equations and the

various associated conitions. The resulting dimensionless forms of the major variables are

$$\varepsilon = \frac{r}{r_p}, \qquad \tau = \frac{\alpha_g t}{{r_p}^2}, \qquad \theta = \frac{T}{T_i}, \qquad \tilde{h}_g = \frac{h_g}{c_g T_i},$$

$$\tilde{u} = \frac{r_p u}{\alpha_g}, \qquad \tilde{Y}_f = \frac{Y_f}{Y_{f,B}}, \qquad \tilde{Y}_o = \frac{Y_o}{\nu_o Y_{f,B}},$$

where $Y_{f,B}$ is the fuel mass fraction at some boundary. Here $Y_{f,B}$ is taken to be 1 since the gas mixture at the particle surface is often solely fuel. The thermal diffusivity in the gas phase is denoted as $\alpha_g = \left({}^{\lambda}/_{\rho c}\right)_g$. The lack of subscripts on θ and $\tilde{u}$ indicates that these relations apply to all instances of the dimensional symbol (i.e. all velocities and all temperatures are nondimensionalized the same way). Note that the time variable t has been scaled by the characteristic diffusion time in the gas phase. Based on the model simplifications, diffusion is the dominant transport mechanism in both domains, but the choice to scale by $t_{diff,g}$ instead of $t_{diff,s}$ is due to the fact that $t_{diff,g} \ll t_{diff,s}$. When chemistry becomes important, the smaller steps taken (in units of τ) are not as small for $t_{diff,g}$ as they are for $t_{diff,s}$; as a result, there is less risk of round-off error during model implementation.

Plugging these variables into equation 3.3.3 for solid fuel mass conservation yields

$$\frac{\partial \tilde{u}_s}{\partial \varepsilon} + \frac{2\tilde{u}_s}{\varepsilon} = -\mathbf{A}\mathrm{Py} \cdot e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}, \tag{3.3.36}$$

with the boundary condition

$$\tilde{u}_s(\varepsilon_p) = 0. \tag{3.3.37}$$

Similarly, the equation for gas phase velocity (3.3.16) is

$$\tilde{u}_g(\varepsilon) = \left(\frac{P}{1-P}\right) R\mathbf{A}\mathrm{Py}\, \frac{\int_{\varepsilon_p}^{\infty} e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}\, \varepsilon'^2\, \mathrm{d}\varepsilon'}{\varepsilon^2}. \tag{3.3.38}$$

In the above equations, $\mathbf{A} = {}^{\alpha_s}/_{\alpha_g}$ is the ratio of solid fuel to gas thermal diffusivity, and

Py is the pyrolysis number, defined as

$$\mathrm{Py} \;=\; \left(\frac{{r_p}^2}{\alpha_s}\right) Z_s e^{\,-\,E_{a,s}/R_u T_i}. \tag{3.3.39}$$

The pyrolysis number gives a measure of the time required to volatilize the fuel relative to the diffusion time in the solid. Equations 3.3.36 and 3.3.38 also include a solid phase Arrhenius number, Ar_s. For chemical reactions with an Arrhenius dependence on temperature, one can define an Arrhenius number as

$$\mathrm{Ar} \;=\; \frac{T_a}{T_i} \;=\; \frac{E_a}{R_u T_i}. \tag{3.3.40}$$

where $T_a \;=\; {E_a}/{R_u}$ is the activation temperature of the reaction. Again, subscripts have been dropped to imply that the relation can apply to both solid and gas phase reactions. Ar gives a measure of the activation temperature relative to the characteristic temperature of the system prior to ignition.

Nondimensionalizing the solid phase energy equation (3.3.9) results in

$$\frac{\partial \theta_s}{\partial \tau} \;=\; \mathbf{A}\left[\frac{\partial}{\partial \varepsilon}\left(\tilde{\lambda}_s \frac{\partial \theta_s}{\partial \varepsilon}\right) \;+\; \frac{2\tilde{\lambda}_s}{\varepsilon}\frac{\partial \theta_s}{\partial \varepsilon}\right] \;-\; \tilde{u}_s \frac{\partial \theta_s}{\partial \varepsilon} \;-\; \overline{\Delta H}_{vol}\mathbf{A}\mathrm{Py}\cdot e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}, \tag{3.3.41}$$

with semi-infinite conditions

$$\theta_s(\tau < 0, \varepsilon) \;=\; \theta_s(\tau, \varepsilon \to \infty) \;=\; \theta_\infty \tag{3.3.42}$$

Here, $\tilde{\lambda}_s \;=\; {\lambda_s}/{\lambda_{s,\,sol}}$ is a dimensionless solid phase thermal conductivity and $\overline{\Delta H}_{vol} \;=\; {\Delta H_{vol}}/{c_s T_i}$ is a dimensionless heat of reaction.

The dimensionless forms of the equations for gas phase energy and species (and their associated boundary and semi-infinite conditions) follow similarly as

$$\gamma_g \frac{\partial \theta_g}{\partial \tau} \;=\; \left[\frac{\partial}{\partial \varepsilon}\left(\tilde{\lambda}_g \frac{\partial \theta_g}{\partial \varepsilon}\right) \;+\; \frac{2\tilde{\lambda}_g}{\varepsilon}\frac{\partial \theta_g}{\partial \varepsilon}\right] \;-\; \gamma_g \tilde{u}_g \frac{\partial \theta_g}{\partial \varepsilon} \;+\; \tilde{Q}\mathrm{Da}\,\tilde{Y}_f\,\tilde{Y}_o\, e^{\mathrm{Ar}_g\left(1-\frac{1}{\theta_g}\right)}, \tag{3.3.43}$$

$$\theta_g(\tau < 0, \varepsilon) \,=\, \theta_g(\tau, \varepsilon \to \infty) \,=\, \theta_\infty, \tag{3.3.44}$$

$$\frac{\partial \tilde{Y}_f}{\partial \tau} \,=\, \frac{\partial^2 \tilde{Y}_f}{\partial \varepsilon^2} \,+\, \frac{2}{\varepsilon}\frac{\partial \tilde{Y}_f}{\partial \varepsilon} \,-\, \tilde{u}_g \frac{\partial \tilde{Y}_f}{\partial \varepsilon} \,-\, \mathrm{Da}\, \tilde{Y}_f\, \tilde{Y}_o\, e^{\mathrm{Ar}_g\left(1-\frac{1}{\theta_g}\right)}, \tag{3.3.45}$$

$$\frac{\partial \tilde{Y}_o}{\partial \tau} \,=\, \frac{\partial^2 \tilde{Y}_o}{\partial \varepsilon^2} \,+\, \frac{2}{\varepsilon}\frac{\partial \tilde{Y}_o}{\partial \varepsilon} \,-\, \tilde{u}_g \frac{\partial \tilde{Y}_o}{\partial \varepsilon} \,-\, \mathrm{Da}\, \tilde{Y}_f\, \tilde{Y}_o\, e^{\mathrm{Ar}_g\left(1-\frac{1}{\theta_g}\right)}, \tag{3.3.46}$$

$$\left(\frac{\partial \tilde{Y}_f}{\partial \varepsilon}\right)_{\varepsilon_p} \,=\, \left[\left(\frac{P}{1-P}\right) R\mathbf{A}\mathrm{Py} \int_{\varepsilon_p}^{\infty} e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}\, \varepsilon'^2\, \mathrm{d}\varepsilon'\right] \big(Y_f(r_p) - 1\big), \tag{3.3.47}$$

$$\left(\frac{\partial \tilde{Y}_o}{\partial \varepsilon}\right)_{\varepsilon_p} \,=\, \left[\left(\frac{P}{1-P}\right) R\mathbf{A}\mathrm{Py} \int_{\varepsilon_p}^{\infty} e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}\, \varepsilon'^2\, \mathrm{d}\varepsilon'\right] Y_o(r_p), \tag{3.3.48}$$

$$\tilde{Y}_f(\tau < 0, \varepsilon) \,=\, \tilde{Y}_f(\tau, \varepsilon \to \infty) \,=\, 0, \tag{3.3.49}$$

$$\tilde{Y}_o(\tau < 0, \varepsilon) \,=\, \tilde{Y}_o(\tau, \varepsilon \to \infty) \,=\, \tilde{Y}_{o,\infty}. \tag{3.3.50}$$

In the above gas phase equations, c_g and λ_g have been scaled by their values at room

temperature, as in

$$\tilde{\lambda}_g \;=\; {}^{\lambda_g}\!/_{\lambda_{g,\infty}},$$

$$\gamma_g \;=\; {}^{c_g}\!/_{c_{g,\infty}}.$$

$\tilde{Q} \;=\; {}^{Q}\!/_{c_{g,\infty}T_i}$ is a dimensionless heat of combustion and Da is the Damkohler number, defined as

$$\mathrm{Da} \;=\; \left(\frac{{r_p}^2}{\alpha_g}\right) Z_g \nu_o \rho_g e^{-E_{a,g}/R_u T_i}. \tag{3.3.51}$$

Analogous to the pyrolysis number, the Damkohler number is a measure of the chemistry timescale in the gas phase relative to the diffusion timescale.

The particle interface conditions are nondimensionalized as

$$\gamma_p \frac{\partial \theta_p}{\partial \tau} \;=\; P\mathbf{A}\left[b_s\left(\tilde{\lambda}_s \frac{\partial \theta_s}{\partial \varepsilon}\right)_{\varepsilon_p} \;-\; b_g R\mathrm{Py}\int_{\varepsilon_p}^{\infty} (\tilde{h}_g(\theta_p) - \tilde{h}_g(\theta_s)) e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_s}\right)}\; {\varepsilon'}^2 \,\mathrm{d}\varepsilon'\right]$$

$$+\; (1-P)\left[b_g\left(\tilde{\lambda}_g \frac{\partial \theta_g}{\partial \varepsilon}\right)_{\varepsilon_p} \;-\; \psi\left({\theta_p}^4 - {\theta_\infty}^4\right)\right], \tag{3.3.52}$$

$$\theta_s(\tau > 0, \varepsilon_p) \;=\; \theta_g(\tau > 0, \varepsilon_p) \;=\; \theta_p. \tag{3.3.53}$$

$$\theta_p(\tau < 0) \;=\; 1. \tag{3.3.54}$$

In these particle equations, $\gamma_p \;=\; {}^{c_{p,eff}}\!/_{c_p}$ is the effective particle specific heat scaled by the true particle specific heat. The related variables b_s and b_g are ratios of the particle volumetric specific heat to those of the solid and gas, respectively, as in

$$b \;=\; \frac{3(\rho c)}{(\rho_p c_p)}.$$

The variable ψ is a dimensionless term associated with radiation losses from the particle and is defined as

$$\psi = \frac{3r_p\epsilon\sigma T_i^3}{(\rho_p c_p)\alpha_g}.$$

3.4 Solution Methodology

The 5 conservation equations for solid phase mass, solid phase energy, gas phase energy, and the two gas phase reactants are discretized using a finite volume formulation. This includes second order spatial differencing, an upwind scheme to handle convection terms and implicit time stepping. The N ordinary difference equations (where N is the number of grid points in the domain) that result from each conservation equation are solved at each time step using an efficient Tri-diagonal matrix (TDM) algorithm. Nonlinearities are handled with multiple iterations at each time step until sufficient convergence. Equation 3.3.38 for the gas phase velocity is also solved during each iteration, but it does not require discretization. The model is implemented using the Python programming language and a copy of the program can be found in Appendix A.

The first step in the discretization process is to divide up the solid and gas phase domains into N_s and N_g finite volumes, respectively. The numerical solution will be calculated for grid points located halfway between the boundaries of these volumes. The grid spacing in the solid and gas phase may be different, but within each domain the grid spacing is a uniform $\Delta\varepsilon_s$ or $\Delta\varepsilon_g$. An example of three contiguous volumes is shown in Figure 3.9. Quantities associated with volume centers are denoted by subscripts $j-1$, j, and $j+1$, and quantities associated with the left and right boundaries of volume j are denoted by subscripts $j-$ and $j+$ respectively.

Finite volume discretization of the conservation equations is identical to the derivation process, but with the differential volume of thickness $\delta\varepsilon$ replaced by the finite volume of thickness $\Delta\varepsilon$. Spatial and temporal derivatives are also replaced by their finite difference approximations. Inspection of the dimensionless conservation equations for solid phase energy and gas phase energy and species shows that they all have essentially the same form (that of a convective-diffusive system with a nonlinear source). As such, the next subsection outlines their discretization using the generic transported scalar ϕ. Equation 3.3.36 for solid phase mass conservation has a slightly simpler form, and its discretization is covered in the following subsection. The remaining subsections discretize the initial, boundary and interface conditions and discuss more practical aspects of model implementation.

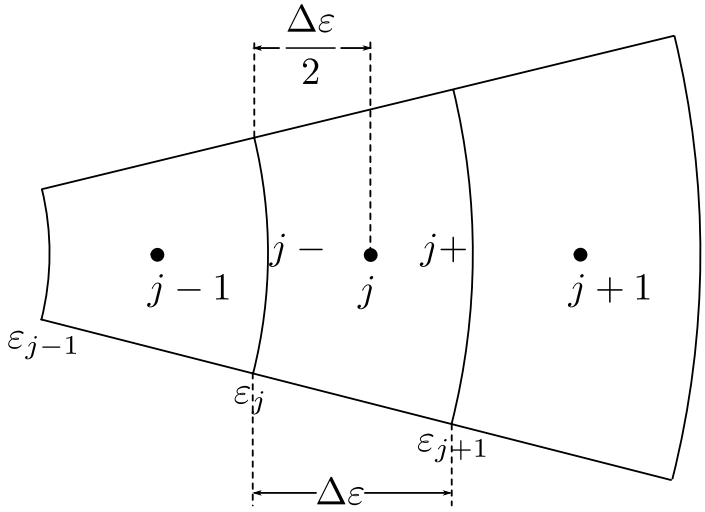

Figure 3.9: Three contiguous finite volumes and associated labels.

3.4.1 Discretization of convection-diffusion conservation equations

The representative conservation equation of interest has the form

$$\frac{\partial\phi}{\partial\tau} = \left[\frac{\partial}{\partial\varepsilon}\left(D\frac{\partial\phi}{\partial\varepsilon}\right) + \frac{2D}{\varepsilon}\frac{\partial\phi}{\partial\varepsilon}\right] - \tilde{u}\frac{\partial\phi}{\partial\varepsilon} + S, \tag{3.4.1}$$

where D is a representative diffusion coefficient, and S is a volumetric source. The derivation of the discrete counterpart to equation 3.4.1 begins with a ϕ-balance around a finite control volume as shown in Figure 3.10. Note that the factor 4π has been dropped from all terms for convenience and the superscript on the term ϕ_j° signifies the value of ϕ_j at the previous time step. Values and gradients of ϕ at the volume boundaries are approximated assuming a linear variation between volume centers. However, for the purposes of estimating the source term, ϕ_j is assumed to be constant over the entire j-th volume. Summing the terms in Figure

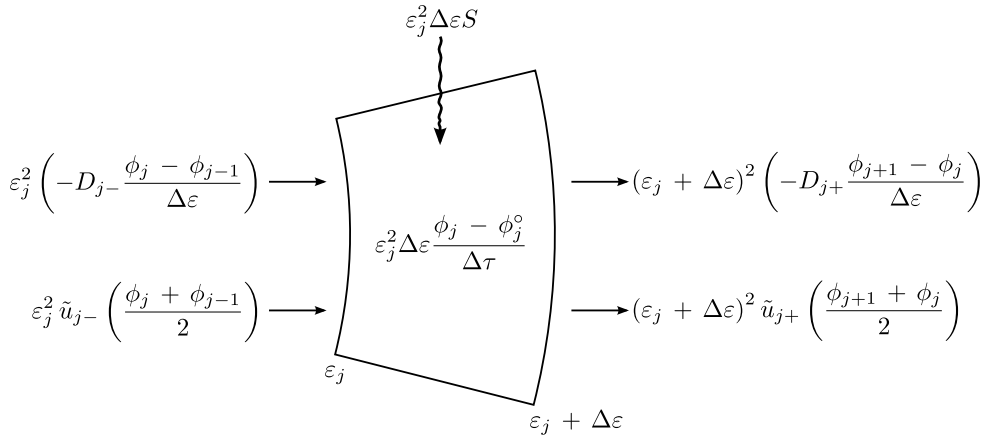

Figure 3.10: Discrete balance for scalar ϕ on finite volume j.

3.10 and dividing by ${\varepsilon_j}^2 \Delta\varepsilon$ yields

$$\frac{\phi_j - {\phi_j}^\circ}{\Delta\tau} = \frac{1}{{\Delta\varepsilon}^2} \left[D_{j+} \left(\phi_{j+1} - \phi_j\right) - D_{j-} \left(\phi_j - \phi_{j-1}\right)\right] + \frac{2D_{j+} \left(\phi_{j+1} - \phi_j\right)}{\varepsilon_j \Delta\varepsilon}$$

$$- \frac{1}{2\Delta\varepsilon} \left[\tilde{u}_{j+} \left(\phi_{j+1} + \phi_j\right) - \tilde{u}_{j-} \left(\phi_j + \phi_{j-1}\right)\right]$$

$$- \frac{\tilde{u}_{j+} \left(\phi_{j+1} + \phi_j\right)}{\varepsilon_j} + S, \tag{3.4.2}$$

where $\Delta\tau$ is the time step. The discrete form of the continuity equation,

$$\frac{\tilde{u}_{j+} - \tilde{u}_{j-}}{\Delta\varepsilon} + \frac{2\tilde{u}_{j+}}{\varepsilon_j} = 0. \tag{3.4.3}$$

can be used to simplify equation 3.4.2 further to

$$\frac{\phi_j - \phi_j^{\circ}}{\Delta\tau} = \frac{1}{\Delta\varepsilon^2}\left[D_{j+}\left(\phi_{j+1} - \phi_j\right) - D_{j-}\left(\phi_j - \phi_{j-1}\right)\right] + \frac{2D_{j+}\left(\phi_{j+1} - \phi_j\right)}{\varepsilon_j \Delta\varepsilon}$$

$$- \frac{1}{2\Delta\varepsilon}\left[\tilde{u}_{j-}\left(\phi_{j+1} + \phi_j\right) - \tilde{u}_{j-}\left(\phi_j + \phi_{j-1}\right)\right] + S. \tag{3.4.4}$$

While equation 3.4.3 does not actually apply to the solid phase, discretization of the solid phase energy equation will yield the same form as equation 3.4.4.

In the gas phase, the value for $\tilde{u}_{j-}$ comes directly from equation 3.3.38, while in the solid phase, $\tilde{u}_{j-}$ is taken as equal to $\tilde{u}_{j-1}$ as explained in the following subsection. Also, to ensure that the flux leaving one cell is equal to the flux entering the adjacent cell, the diffusion coefficients D_{j-} and D_{j+} are calculated as harmonic means,

$$D_{j-} = \frac{2D_{j-1}D_j}{D_{j-1} + D_j}$$

$$D_{j+} = \frac{2D_{j+1}D_j}{D_{j+1} + D_j}.$$

In the above formulation, D, $\tilde{u}$ and S are all functions of ϕ. However, solution of equation 3.4.4 is greatly simplified if these variables are linearly dependent or constant with respect to ϕ. As such, they are evaluated at ϕ^*, the guess of ϕ from the previous iteration. One must also take special care with the source S; if the source term is negative and not properly handled, the transported scalar may become unrealistically negative [1]. One way to avoid this is to ensure that the negative source has a linear dependence on ϕ. To apply this generally, the source S can be broken into a positive and negative component, $S = S_{pos} + S_{neg}$, and the negative component can be multiplied by the ratio ϕ/ϕ^*, as in

$$S = S_{pos} + \frac{S_{neg}}{\phi^*}\phi.$$

In the convergent limit, $\phi/\phi^* = 1$ and the relation for S is mathematically unchanged. For the purposes of this model, S_{pos} or S_{neg} are zero, depending on the conservation equation.

One can now gather all of the terms in equation 3.4.4 that are multiplied by the values ϕ_{j-1} and ϕ_{j+1},

$$\left[\frac{D_{j-}}{\Delta\varepsilon^2} + \frac{u_{j-}}{2\Delta\varepsilon}\right]\phi_{j-1} = a_{j-1}\phi_{j-1}$$

$$\left[\frac{D_{j+}}{\Delta\varepsilon^2} + \frac{2D_{j+}}{\varepsilon_j\Delta\varepsilon} - \frac{u_{j-}}{2\Delta\varepsilon}\right]\phi_{j+1} = a_{j+1}\phi_{j+1}, \tag{3.4.5}$$

such that it can be rearranged as

$$-\Delta\tau a_{j-1}\phi_{j-1} + \left[1 + \Delta\tau\left(a_{j+1} + a_{j-1} + {}^{S_n}/_{\phi_j{}^*}\right)\right]\phi_j - \Delta\tau a_{j+1}\phi_{j+1} = \phi_j{}^{\circ} + \Delta\tau S_p. \tag{3.4.6}$$

Correctly solving this finite volume equation requires that the coefficients of ϕ_{j-1} and ϕ_{j+1} are always ≥ 0 and that the coefficient of ϕ_j is always positive [1]. However, equations 3.4.5 make it clear that if $\tilde{u}_{j-}$ is sufficiently large, $a_{j+1} < 0$ and the coefficient on ϕ_{j+1} is positive. Although a number of schemes exist to deal with numerical modeling of convection, the simple upwind scheme is used in this model. If the value of $\tilde{u}_{j-}$ in a particular cell is sufficiently high (in the $+r$ *or* $-r$ directions) then that cell is assumed to experience pure convective transport. Mathematically, this is accomplished by modifying the definitions of a_{j+1} and a_{j-1},

$$a_{j-1} = \text{maximum}\left[\tilde{u}_{j-},\ \frac{D_{j-}}{\Delta\varepsilon^2} + \frac{u_{j-}}{2\Delta\varepsilon},\ 0\right]$$

$$a_{j+1} = \text{maximum}\left[-\tilde{u}_{j-},\ \frac{D_{j+}}{\Delta\varepsilon^2} + \frac{2D_{j+}}{\varepsilon_j\Delta\varepsilon} - \frac{u_{j-}}{2\Delta\varepsilon},\ 0\right]. \tag{3.4.7}$$

With the upwind scheme in place, equation 3.4.6 can be recast in the form most conducive to solution by the TDM algorithm discussed in a later subsection,

$$\mathrm{A}_j\phi_{j-1} + \mathrm{B}_j\phi_j - \mathrm{C}_j\phi_{j+1} = \mathrm{D}_j, \tag{3.4.8}$$

where

$$\mathrm{A}_j \;=\; -\Delta\tau a_{j-1}$$

$$\mathrm{B}_j \;=\; 1 \;+\; \Delta\tau\left(a_{j+1} \;+\; a_{j-1} \;+\; {}^{S_n}\!/_{\phi_j{}^*}\right)$$

$$\mathrm{C}_j \;=\; -\,\Delta\tau a_{j+1}$$

$$\mathrm{D}_j \;=\; \phi_j{}^{\circ} \;+\; \Delta\tau S_p\,.$$

Note that the Roman D_j denotes the TDM coefficient while the italicized D_j represents the general diffusion coefficient at the j-th volume center.

3.4.2 Discretization of solid fuel mass conservation

Figure 3.11 shows the mass flows associated with the j-th solid phase finite volume. Again,

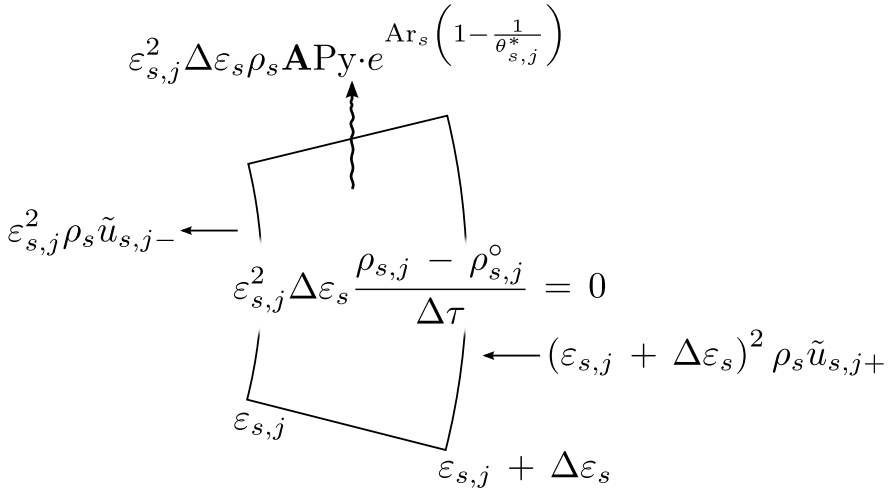

Figure 3.11: Discrete mass balance for solid phase finite volume j.

the factor 4π is dropped from all terms for convenience. Summing the inflows and outflows

and dividing by $\varepsilon_{s,j}^2 \Delta\varepsilon_s \rho_s$ yields

$$\frac{\tilde{u}_{s,j+} - \tilde{u}_{s,j-}}{\Delta\varepsilon_s} + \frac{2\tilde{u}_{s,j+}}{\varepsilon_{s,j}} = -\mathbf{A}\mathrm{Py} \cdot e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta^*_{s,j}}\right)}. \tag{3.4.9}$$

At this point it is useful to note that in both the solid and gas domains, the 0-th volume is different than all the other finite volumes; it is hemispherical in shape and represents the particle. Therefore, the solid phase boundary at 0+ is closed and $\tilde{u}_{s,0+} = 0$, in keeping with the solid phase mass boundary condition. To simplify implementation of this boundary condition, one can set $\tilde{u}_{j-}$ equal to $\tilde{u}_{j-1}$, such that $\tilde{u}_{s,0} = 0$ and $\tilde{u}_{s,1} = \tilde{u}_{s,1+}$. With this clarification, equation 3.4.9 can be rearranged for the TDM algorithm,

$$\mathrm{A}_{\tilde{u}_s,j}\tilde{u}_{s,j-1} + \mathrm{B}_{\tilde{u}_s,j}\tilde{u}_{s,j} - \mathrm{C}_{\tilde{u}_s,j}\tilde{u}_{s,j+1} = \mathrm{D}_{\tilde{u}_s,j}, \tag{3.4.10}$$

where

$$\mathrm{A}_{\tilde{u}_s,j} = -\frac{1}{\Delta\varepsilon_s} \qquad \mathrm{A}_{\tilde{u}_s,0} = 0$$

$$\mathrm{B}_{\tilde{u}_s,j} = \frac{1}{\Delta\varepsilon_s} + \frac{2}{\varepsilon_{s,j}} \qquad \mathrm{B}_{\tilde{u}_s,0} = 1$$

$$\mathrm{C}_{\tilde{u}_s,j} = 0 \qquad \mathrm{C}_{\tilde{u}_s,0} = 0$$

$$\mathrm{D}_{\tilde{u}_s,j} = -\mathbf{A}\mathrm{Py} \cdot e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta^*_{s,j}}\right)} \qquad \mathrm{D}_{\tilde{u}_s,0} = 0.$$

Above, the solid phase mass boundary condition has been applied by specifying special coefficients for the 0-th volume (i.e. the particle volume).

3.4.3 Discretization of initial, boundary and interface conditions

Numerical implementation of initial conditions is straightforward. The variable arrays for θ_s, $\tilde{u}_s$, θ_g, $\tilde{Y}_f$ and $\tilde{Y}_o$ are initialized with the values described in subsection 3.3.8. Then, the particle initial condition is implemented by setting $\theta_{s,0}$ and $\theta_{g,0}$ equal to 1. To discretize the boundary and interface conditions, the rest of the subsection follows the same procedure of

the previous two subsections: a condition is treated with finite volume approximations and then cast in the form of TDM coefficients that can be combined with the coefficients from the governing equations.

Starting with the gas phase species inner boundary conditions, one estimates the mass fraction gradient at the particle as

$$\left(\frac{\partial \tilde{Y}_i}{\partial \varepsilon}\right)_{\varepsilon_p} = \frac{2\left(\tilde{Y}_{i,1} - \tilde{Y}_{i,0}\right)}{\Delta\varepsilon_g}.$$

where $\tilde{Y}_{i,0}$ represents the mass fraction of i right at the particle surface. The boundary conditions for $\tilde{Y}_f$ and $\tilde{Y}_o$ then become

$$\frac{2\left(\tilde{Y}_{f,1} - \tilde{Y}_{f,0}\right)}{\Delta\varepsilon_g} = \tilde{u}_g(\varepsilon_p)\left(\tilde{Y}_{f,0} - 1\right), \tag{3.4.11}$$

$$\frac{2\left(\tilde{Y}_{o,1} - \tilde{Y}_{o,0}\right)}{\Delta\varepsilon_g} = \tilde{u}_g(\varepsilon_p)\tilde{Y}_{o,0}, \tag{3.4.12}$$

and can be cast in the form of equation 3.4.8 by defining

$$\mathrm{A}_{\tilde{Y}_f,0} = 0 \qquad \mathrm{A}_{\tilde{Y}_o,0} = 0$$

$$\mathrm{B}_{\tilde{Y}_f,0} = 1 + \Delta\varepsilon_g\tilde{u}_g(\varepsilon_p) \qquad \mathrm{B}_{\tilde{Y}_o,0} = 1 + \Delta\varepsilon_g\tilde{u}_g(\varepsilon_p)$$

$$\mathrm{C}_{\tilde{Y}_f,0} = -1 \qquad \mathrm{C}_{\tilde{Y}_o,0} = -1$$

$$\mathrm{D}_{\tilde{Y}_f,0} = \Delta\varepsilon_g\tilde{u}_g(\varepsilon_p) \qquad \mathrm{D}_{\tilde{Y}_o,0} = 0.$$

The particle interface conditions can be approximated in a similar fashion as

$$\gamma_p \frac{\theta_0 - \theta_0^\circ}{\Delta\tau} = P\mathbf{A}\left[b_s\tilde{\lambda}_{s,1}\frac{2\left(\theta_{s,1} - \theta_{s,0}\right)}{\Delta\varepsilon_s} + b_g R\mathrm{Py}\sum_{j=1}^{N_s}\tilde{h}_g(\theta^*_{s,j})e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta^*_{s,j}}\right)}\right]$$

$$+ (1-P)\left[b_g\tilde{\lambda}_{g,1}\frac{2\left(\theta_{g,1} - \theta_{g,0}\right)}{\Delta\varepsilon_g} + \psi{\theta_\infty}^4\right]$$

$$- \left[P\mathbf{A}b_g R\mathrm{Py}\sum_{j=1}^{N_s}\left(\frac{\tilde{h}_g(\theta_0^*)}{\theta_0^*}\right)e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta^*_{s,j}}\right)} + (1-P)\psi(\theta_0^*)^3\right]\theta_0. \quad (3.4.13)$$

Note that the negative source terms in equation 3.3.52 have been grouped together and linearized per the discussion in previous subsections. The model first uses θ_g^* to implement equation 3.4.13 as a boundary condition for θ_s. Then the updated values for θ_s are used to apply it as a boundary condition for θ_g. Finally, agreement is forced between $\theta_{s,0}$ and $\theta_{g,0}$ to

satisfy condition 3.3.53. This results in the following coefficients for θ_s:

$$\mathrm{A}_{\theta_s,0} = 0$$

$$\mathrm{B}_{\theta_s,0} = 1 + \frac{2\Delta\tau P\mathbf{A}b_s\tilde{\lambda}_{s,1}}{\gamma_p\Delta\varepsilon_s} + \frac{\Delta\tau(1-P)\psi(\theta_0^*)^3}{\gamma_p} + \frac{\Delta\tau}{\gamma_p}P\mathbf{A}b_gR\mathrm{Py}\sum_{j=1}^{N_s}\left(\frac{\tilde{h}_g(\theta_{s,0}^*)}{\theta_{s,0}^*}\right)e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_{s,j}^*}\right)}$$

$$\mathrm{C}_{\theta_s,0} = -\frac{2\Delta\tau P\mathbf{A}b_s\tilde{\lambda}_{s,1}}{\gamma_p\Delta\varepsilon_s}$$

$$\mathrm{D}_{\theta_s,0} = \theta_{s,0}^\circ + \frac{\Delta\tau(1-P)}{\gamma_p}\left[\frac{2b_g\tilde{\lambda}_{g,1}\left(\theta_{g,1}^* - \theta_{g,0}^*\right)}{\Delta\varepsilon_g} + \psi{\theta_\infty}^4\right] + \frac{\Delta\tau}{\gamma_p}P\mathbf{A}b_gR\mathrm{Py}\sum_{j=1}^{N_s}\tilde{h}_g(\theta_{s,j}^*)e^{\mathrm{Ar}_s\left(1-\frac{1}{\theta_{s,j}^*}\right)},$$

and the following coefficients for θ_g:

$$\mathrm{A}_{\theta_s,0} = 0$$

$$\mathrm{B}_{\theta_s,0} = 1 + \frac{2\Delta\tau(1-P)b_g\tilde{\lambda}_{g,1}}{\gamma_p\Delta\varepsilon_g}$$

$$\mathrm{C}_{\theta_s,0} = -\frac{2\Delta\tau(1-P)b_g\tilde{\lambda}_{g,1}}{\gamma_p\Delta\varepsilon_g}$$

$$\mathrm{D}_{\theta_s,0} = \theta_{g,0}^\circ + \left(\theta_{s,0} - \theta_{s,0}^\circ\right) - \frac{2\Delta\tau(1-P)b_g\tilde{\lambda}_{g,1}\left(\theta_{g,1}^* - \theta_{g,0}^*\right)}{\gamma_p\Delta\varepsilon_g}.$$

Numerical implementation of the far field boundary conditions is achieved by solving the governing equations on an approximately semi-infinite domain (i.e. one that is sufficiently large). Whether or not the domain is adequately sized is tested at each iteration by applying the boundary condition

$$\frac{\phi_N - \phi_{N-1}}{\Delta\varepsilon} = 0 \tag{3.4.14}$$

and checking that ϕ_N has not deviated from its initial value by more than some specified tolerance. The tolerances used are the same as those used as convergence criteria; they are discussed in a later subsection. As a reminder, in the above equation ϕ is a generic scalar that represents θ_s, θ_g, $\tilde{Y}_f$ and $\tilde{Y}_o$. If at any point the solid or gas phase domain is found to be too small, the model simulation is restarted with a larger domain. When recast in the form of the TDM algorithm, equation 3.4.14 results in coefficient values $A_N = -1$, $B_N = 1$ and $C_N = D_N = 0$.

3.4.4 Tri-diagonal matrix algorithm

The Tri-diagonal matrix algorithm, also known at the Thomas algorithm, is an efficient method for solving systems of equations that result in tri-diagonal coefficient matrices. Consider the system of equations

$$\mathbf{F} \cdot \phi = \mathrm{d},$$

where $\mathbf{F}$ is a square matrix of coefficients, d is a vector of constants, and ϕ is the unknown solution. $\mathbf{F}$ is tri-diagonal if the elements along its diagonal and on either side of the diagonal are non-zero and all the other elements are zero. Such a system results from the present model because the value of ϕ in each cell depends only on the values of one or both of its neighbors. In this case, Gaussian elimination can be carried out with minimal steps by organizing the non-zero elements of $\mathbf{F}$ into three equal-length one-dimensional arrays: A contains elements to the left of the diagonal, B contains the elements along the diagonal, and C contains the elements to the right of the diagonal [1]. This requires that $\mathrm{A}_0 = \mathrm{C}_N = 0$, where subscripts 0 and N denote the first and last elements, respectively. The system of equations can now be written as equation 3.4.8,

$$\mathrm{A}_j\phi_{j-1} + \mathrm{B}_j\phi_j - \mathrm{C}_j\phi_{j+1} = \mathrm{D}_j,$$

restated here for convenience.

The solution algorithm starts by defining two new arrays, C' and D', where

$$C'_0 = \frac{C_0}{B_0} \qquad\qquad D'_0 = \frac{D_0}{B_0}.$$

It finds the rest of the elements of C' by successively calculating

$$C'_j = \frac{C_j}{B_j - A_j C'_{j-1}}.$$

for elements 1 through $N-1$. Similarly, the remaining elements of D' are found by sweeping elements 1 through N and calculating

$$D'_j = \frac{D_j - A_j D_{j-1}}{B_j - A_j C'_{j-1}}.$$

Finally, the algorithm finds the solution array ϕ by defining

$$\phi_N = D'_N,$$

and then sweeping backwards (from $N-1 \rightarrow 0$) and calculating

$$\phi_j = D'_j - C'_j \phi_{j+1}.$$

3.4.5 Iteration and convergence criteria

As mentioned previously, the governing equations are solved used a fully-implicit iterative scheme. At each time step, the scheme starts by guessing that the solution will be the same as at the previous time step. Based on this guess, transport properties and nonlinear terms are estimated and the fully implicit finite volume equations are solved to find an improved guess. This improved guess yields improved transport properties and nonlinear terms and thus an improved guess. This iterative procedure continues until the deviation in every cell from iteration to iteration is less than some specified tolerance. To reduce computation time, the convergence criteria used for this model is 10^{-4}. Due to the potential to have small but significant values of mass fraction, convergence for $\tilde{Y}_f$ and $\tilde{Y}_o$ is based on a relative tolerance,

as in

$$\frac{|\tilde{Y}_i - \tilde{Y}_i^*|}{\text{maximum}\left[\tilde{Y}_i^*, 10^{-7}\right]} < 10^{-4},$$

where the superscript $*$ denotes the guess from the previous iteration and the form of the denominator avoids division by zero. For θ_s and θ_g, an absolute tolerance is used to define convergence,

$$\theta - \theta^* < 10^{-4}.$$

The stiff nonlinear governing equations often require additional techniques to assure that the solution continually converges from iteration to iteration [2]. This model achieves that goal through adaptive time-stepping, i.e. the model makes an initial guess for the time step and halves the time step if required for convergence. When a solution is converging, the error between iterations continually decreases and convergence is achieved in a relatively small number of iterations. Thus, the time step is halved if the error between iterations happens to increase (suggesting divergence) or if the number of iterations required is greater than 50.

3.5 Ignition of powdered cellulose by stainless steel particles

As stated in section 3.1, the goal of the present work is to use the idealized model developed in the last two sections to explore the physics underlying the ignition process. To keep the investigation focused on plausible real-world ignition scenarios, the model requires reasonable input values for its numerous parameters. Thus, it is useful to choose a test case to simulate, such as the experimental data discussed in Chapter 2. Specifically, the results discussed in this section focus on the case of stainless steel spheres igniting a packed bed of powdered cellulose. Particle diameters of interest ranged from 10 mm to $<$ 1 mm. However, the reader is reminded that the objective of the study is not necessarily to duplicate quantitative experimental results, but instead to deduce trends in the ignition phenomena. This approach also provides an opportunity to compare the model and experimental results and determine the ways in which the greatly simplified model is still lacking.

The following subsections describe the various parameter values chosen and the reaction schemes used to model volatilization and combustion. The remainder of the section presents and discusses the model results, particularly in the context of the experimental data.

3.5.1 Model parameter values

Table 3.1 outlines the solid fuel and particle thermal properties used in the model. The particle properties used are for 302 stainless steel, and the solid properties are for cellulose or biomass. The value for the bulk fuel density comes directly from the author's experimental work. A brief survey of the literature will show that there are large variations in the properties

Table 3.1: Solid fuel and particle thermal property values.

Property	Symbol	Value
Particle density (kg/m^3)	ρ_p	7860 [7]
Particle heat capacity (J/kgK)	c_p	500 [7]
Particle emissivity	ϵ	0.9[a] [8]
Solid fuel density (kg/m^3)	ρ_s	300
Solid fuel specific heat (J/kgK)	c_s	2510 [9]
Solid fuel thermal conductivity (W/mK)	λ_s	12.6×10^{-2} [9]
Solid fuel pore emissivity	ϵ_{pore}	1.0 [4]
Solid fuel pore diameter (m)	d_{pore}	2×10^{-5} [4]

[a]Conservative estimate based on expected particle temperatures and data for oxidized alloy 303.

used in cellulose and biomass pyrolysis models, and as such the values chosen for this study should be thought of simply as rough estimates [4, 9–11]. The gas phase is assumed to have the thermal properties of nitrogen according to the NASA property tables, as mentioned in a previous section. Finally, model results were found to be quite sensitive to grid spacing, so 0.01 mm and 0.1 mm grid sizes were used in the solid and gas domains, respectively. As part of future work, The author will conduct an analysis of model sensitivity to grid spacing and the convergence criteria (mentioned in the previous section).

3.5.2 Cellulose pyrolysis and combustion mechanisms

When discussing the chosen pyrolysis mechanism, justifications must be made concerning several neglected aspects. As mentioned previously, it is assumed that oxidative decomposition reactions are unimportant. This assumption is based on the fact that the pyrolysis front is located underneath the particle, such that there is limited surface area through which oxygen might diffuse to the reaction front. Neglecting oxidation reactions also allows one to take advantage of the large body of research concerning non-oxidative biomass pyrolysis for energy conversion. Another aspect that is not explicitly covered is the secondary cracking of pyrolysis products in the gas phase. Given the oxygen present in air and the rough approximation of a global combustion reaction, some cracking can be accounted for by the combustion mechanism. This is not applicable however in regions of the domain where oxy-

gen has not diffused; in these regions, cracking is likely to occur, but for simplicity is not considered here.

The most widely-known mechanism for cellulose pyrolysis is the one proposed by Bradbury et al., also known as the Broido-Shafizadeh mechanism [12]. The first step in this scheme is an energy-neutral transition to a molten intermediate known as active cellulose. This active cellulose then decomposes along two competing first-order pathways: one that produces only high molecular-weight condensible vapors (known as tars), and another that produces a solid char and lighter gases. Depending on residence times, the tars may decompose further due to vapor-phase cracking. A diagram of this scheme is shown in Figure 3.12. Reaction 3 is favored at higher temperatures, while reaction 2 is favored at lower tempera-

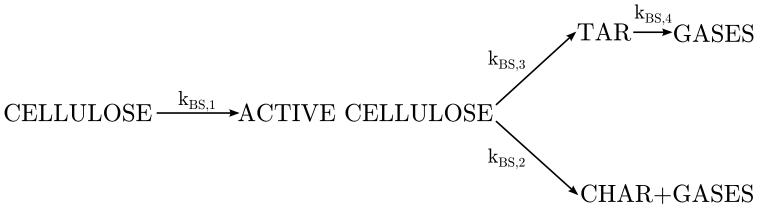

Figure 3.12: Cellulose pyrolysis mechanism proposed by Bradbury et al. [12].

tures. Recently, Cho et al. used improved experimental techniques to estimate the kinetic coefficients $k_{BS,1}$, $k_{BS,2}$ and $k_{BS,3}$ and the heats of reaction for reactions 2 and 3 [13]. They found the char-and-gases pathway to be exothermic and the tar pathway to be endothermic. Their results are summarized in Table 3.2. Recent research has shown that the mechanism

Table 3.2: Kinetic parameter values for Broido-Shafizadeh scheme reprinted from [13].

Parameter	Present work			Reference[a]	
	$\log_{10}A$ (A in s^{-1})	E_A [kJ mol^{-1}]	ΔH[b] (kJ equiv mol^{-1})	$\log_{10}A$ (A in s^{-1})	E_A [kJ mol^{-1}]
$k_{bs,1}$[s^{-1}]	21.62 ± 1.02	257.72 ± 3.23	$-1.5 < \Delta H_1 < 0$	19.45	242.67
$k_{bs,2}$[s^{-1}]	5.74 ± 0.98	102.94 ± 4.56	−170.17 ± 7.27	10.12	153.13
$k_{bs,3}$[c] [s^{-1}]	14.81 ± 0.01	198.91 ± 0.02	121.38 ± 2.42	14.50	197.90

[a] See Bradbury et al. [b] Heat of reaction was determined by DSC experiments. [c] The kinetic parameters were taken from a previous study.

proposed by Bradbury et al. is less accurate at high temperatures, when additional products are formed (most notably hydroxyacetaldehyde). More complex mechanisms have been proposed by Piskorz et al. and Banyasz et al.; although the two models are similar, the model of Banyasz et al. is used here due to its relative simplicity [14]. A diagram of the mechanism is shown in Figure 3.13. According to this scheme, a low temperature char pathway competes with an intermediate-to-high temperature pathway. The latter pathway begins with the formation of active cellulose but then splits again depending on the reaction temperature. At

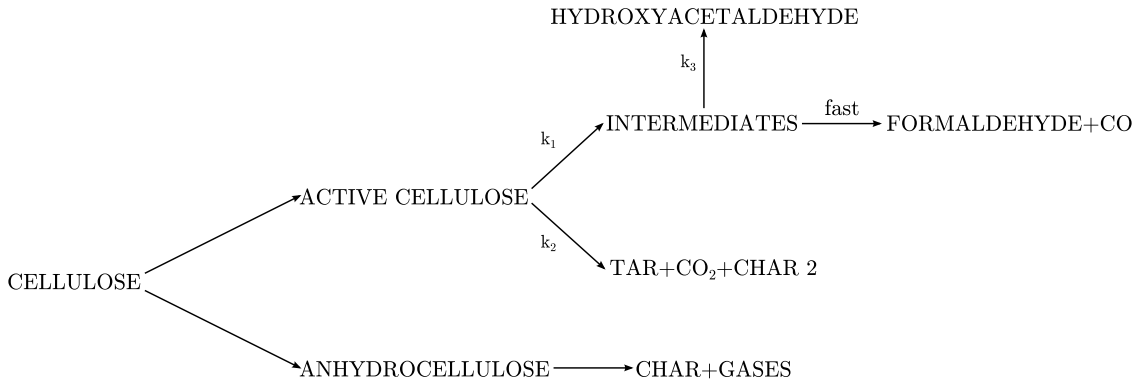

Figure 3.13: Cellulose pyrolysis mechanism proposed by Banyasz et al. [15].

intermediate temperatures, tar forms along with CO_2 and a potential secondary char. At high temperatures, intermediates are formed; these intermediates either quickly decompose to formaldehyde and CO or react further to form hydroxyacetaldehyde.

In all of these cellulose pyrolysis mechanisms, high temperatures and heating rates favor pathways that produce only volatiles, and this is the basis for the no-char assumption in the model. Additionally, at higher temperatures the transition to active cellulose can be ignored, resulting in a single reaction step that goes from cellulose to volatiles [15]. Here, 'volatiles' refer to some combination of tar, formaldehyde, CO, CO_2, and hydroxyacetaldehyde. Implicit in the model formulation is the assumption that other biomass components can also be treated with this reduced mechanism.

Relative to industrial pyrolysis processes, even fast-pyrolysis, the temperatures and heating rates that occur during hot particle ignition are very high. Therefore, the author chose $Z_s = 10^{14}$ s^{-1} and $E_{a,s} = 195.8$ kJ/mol as a starting point for the model. These are the kinetic parameters associated with the high-temperature pathway 1 in Figure 3.13 [15]. However, at the temperatures of interest (500°C or higher), these values made pyrolysis so easy that over very short time scales (< 0.1 s) the diffusion front between the volatiles and air was pushed 4-5 cm away from the particle surface. Given the uncertainty in the kinetics, the activation energy of the volatilization reaction was altered slightly to produce behavior somewhat closer to that observed in the Schlieren videos of Chapter 2. A final value of $E_{a,s} = 200$ kJ/mol was chosen, which is in excellent agreement with the activation energy of the tar reaction found by Cho et al. [13]. To the author's knowledge, there is no data regarding heats of reaction for the high temperature pathways proposed by Banyasz et al. Thus, the heat of reaction for the volatilization reaction was taken to be that of the tar reaction 3 given in Table 3.2 (i.e. $\Delta H_{vol} = 750$ kJ/kg).

Very little data exists concerning the combustion of cellulose pyrolysis products, likely due to the fact that the composition of such products are unknown, variable and dependent on

a large number of factors. A number of modeling studies have focused on ignition and flame spread over thin cellulose sheets, but the kinetic constants in those models were arbitrarily selected or loosely fit to experimental data [16, 17]. Even so, the parameter values used in those studies yielded physically reasonable model results, and their use of a single step first-order combustion reaction dovetails nicely with the current level of model complexity. As a starting point, the author selected the kinetic parameters used by Nakabe et al.: $Z_g = 8 \times 10^5$ $\mathrm{m^3/kgs}$, $E_{a,g} = 67$ kJ/mol, $\nu_o = 3.57$ and $Q = 30$ MJ/kg. Interestingly, these values resulted in ignition at all temperatures of interest; the only way to achieve a non-ignition event was to simulate particle temperatures that were so low that little to no volatiles were produced. This behavior does not align with the Schlieren videos mentioned previously, which indicate that large particles are capable of producing large amounts of pyrolyzate without an ignition event. Much like in the solid phase, the gas phase activation energy was augmented slightly from the values of Nakabe et al., this time such that the ignition limit for large particles very roughly matched the experimental data. This procedure resulted in a value of 100 kJ/mol for $E_{a,g}$, which is significantly higher than the activation energy used by Nakabe et al., but in the neighborhood of other hydrocarbon fuels [18].

3.5.3 Results

Simulation results for an 8 mm particle initially at 923K are shown in Figure 3.14. As a reminder, the dimensionless variables of interest are defined as

$$\varepsilon = \frac{r}{r_p}, \qquad \tau = \frac{\alpha_g t}{{r_p}^2}, \qquad \theta = \frac{T}{T_i}, \qquad \tilde{h}_g = \frac{h_g}{c_g T_i},$$

$$\tilde{u} = \frac{r_p u}{\alpha_g}, \qquad \tilde{Y}_f = \frac{Y_f}{Y_{f,B}}, \qquad \tilde{Y}_o = \frac{Y_o}{\nu_o Y_{f,B}},$$

Since $\varepsilon = {}^{r}/_{r_p}$, a value of zero for the x-axis quantity $\varepsilon - 1$ represents the particle surface for all of the plots in Figure 3.14. The θ_g profiles show that ignition occurs around $\tau = 1.5$ and that by $\tau = 2.5$ the reaction front is transitioning to a diffusion flame. Ignition begins in the extremely hot and fuel-rich mixing region nearest to the particle, but as the reaction begins to take off, the reaction front propagates as a pre-mixed flame through the rest of the mixing layer, as discussed by Liñan and Crespo [19]. This is the cause of the shift in the peak gas phase temperature between $\tau = 1.98$ and $\tau = 2.47$. It is also the reason for the shapes of the last $\tilde{Y}_o$ and $\tilde{Y}_f$ profiles; the oxygen is completely consumed because it is the limiting reactant, while the fuel mass fraction is only beginning to be reduced where the temperatures are highest. At some time after $\tau = 2.5$, the front settles where the equivalence ratio is 1, forming a stable diffusion flame.

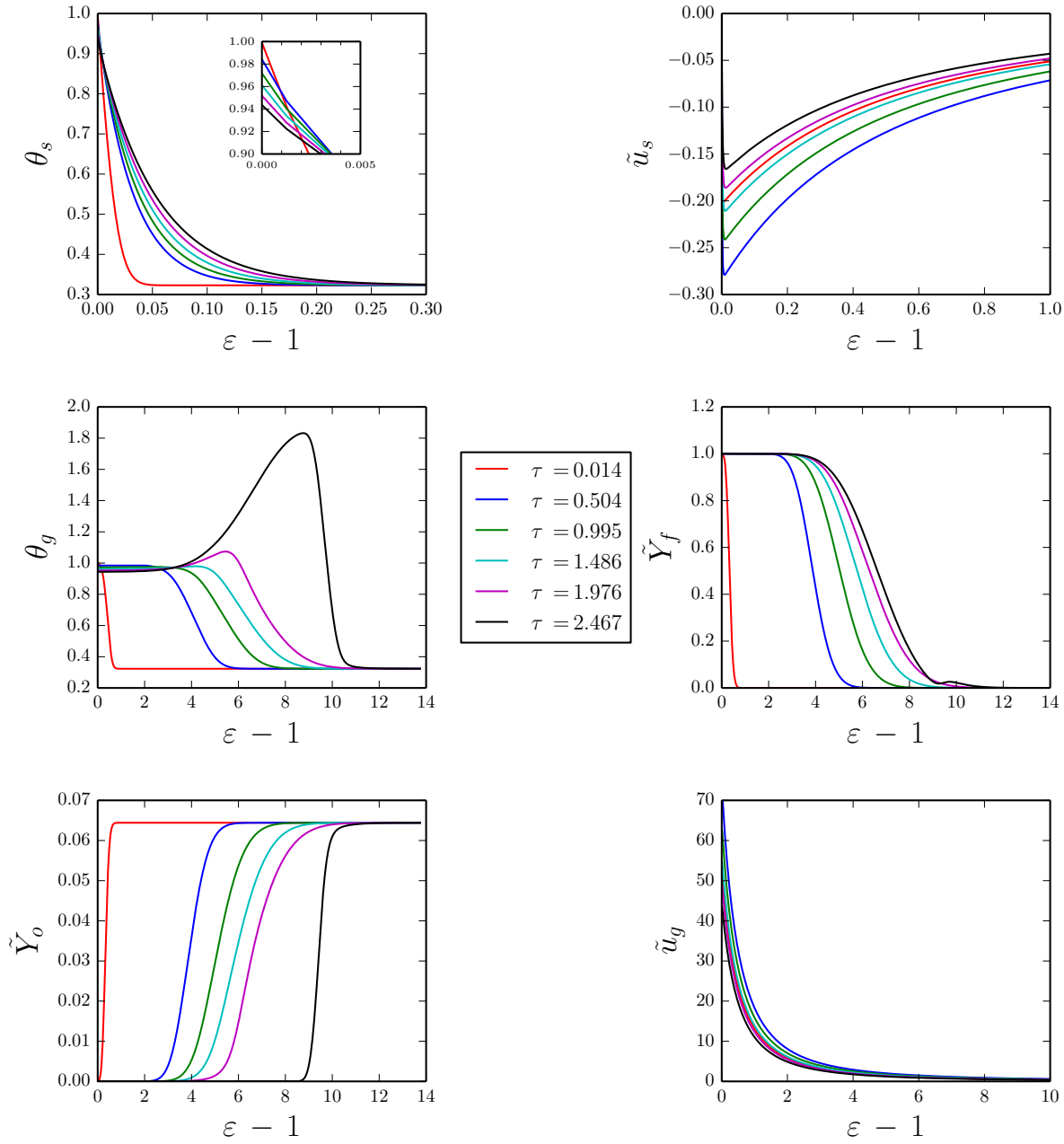

Figure 3.14: Modeling results for $d_p = 8$ mm and $T_i = 923$K.

The plot of θ_s has several interesting features. First, the particle does not cool significantly due to its size, dropping in temperature by only around 4% prior to ignition (see inset plot). Second, prior to $\tau = 0.5$ the temperature profile within the cellulose is changing rapidly, but for $\tau > 0.5$ the changes slow. This indicates that heat diffusion into the fuel bed

is dominant until temperatures become high enough that pyrolysis occurs and the particle sinks into the fuel bed at a fairly steady rate. The minor changes that occur after $\tau = 0.5$ are due to the particle cooling; this reduces the temperature within the cellulose, reducing pyrolysis and allowing heat to diffuse slightly deeper into the virgin fuel. This behavior is mirrored in the plots for $\tilde{u}_s$ and $\tilde{u}_s$. At early times, the virgin fuel velocity and volatiles mass flow are increasing, but for $\tau > 0.5$ they begin to slowly decrease. Of note with regards to $\tilde{u}_s$: one might expect the solid phase velocity to increase with ε because the mass flow at any radial location is equal to the amount of material being pyrolyzed between that location and the particle surface. The further out the location is, the more mass is being pyrolyzed and the higher the required mass flow. However, the surface area across which the mass flows increases with ε^2. The graph clearly shows that this increasing-area effect is dominant, such that the dimensionless mass flux $\tilde{u}_s$ decreases with ε. This effect is also responsible for the sharp curvature near $\varepsilon = 0$ because the highest relative velocities occur next to the closed boundary of the particle surface.

Based on the experimental results of the previous chapter, an 8 mm particle falls within the large particle ignition regime, and as such the simulation results in Figure 3.14 can be considered as representative of cellulosic fuel ignition by large particles. According to the model, the particle does not cool significantly over the timescales of interest, implying that ignition is not sensitive to particle properties. After an initial period of rapid change, fuel volatiles are also produced at a fairly steady rate. This results in a mixing layer between the hot volatiles and surrounding air that is moving outward away for the particle. If one assumes that particle surface temperature is approximately constant, the thermal gradients within the expanding volatile region are very small. Ignition by large particles then occurs when temperatures in the mixing layer remain high enough for thermal runaway to occur even as the hot volatiles are diluted by the cool surrounding air. The proximity of the mixing layer to the particle is unimportant, and the ignition time τ_{ig} is the time required for balance to be achieved between thermal gradients in the mixing layer and the finite rate chemistry. In this simplified large particle case, ignition can occur even for relatively low particle temperatures, albeit at very long times when the mixing layer is far away from the particle surface. In the real system captured in the Schlieren videos of Chapter 2, buoyant convection and mixing become important as the mixing layer moves farther away from the particle. This increased transport would increase losses and explain why, experimentally, ignition occurred very quickly ($t_{ig} < 1$ s) or not at all.

As particle size decreases, particle cooling becomes important and the reaction zone loses heat to both the particle and the surrounding air. As an example, simulation results for θ_s, $\tilde{u}_s$, θ_g and $\tilde{Y}_f$ for a 2 mm particle initially at 994K are shown in Figure 3.15. Plots of $\tilde{Y}_o$ and $\tilde{u}_g$ are not shown because they are similar to the plots in Figure 3.14 and don't yield any additional insight. The profiles of θ_s clearly show that the 2 mm particle cools significantly, and relative to the 8 mm case heat diffuses deeper into the virgin cellulose with time. As a result of this cooling and the particle size generally, $\tilde{u}_s$ and the flow of volatiles into the

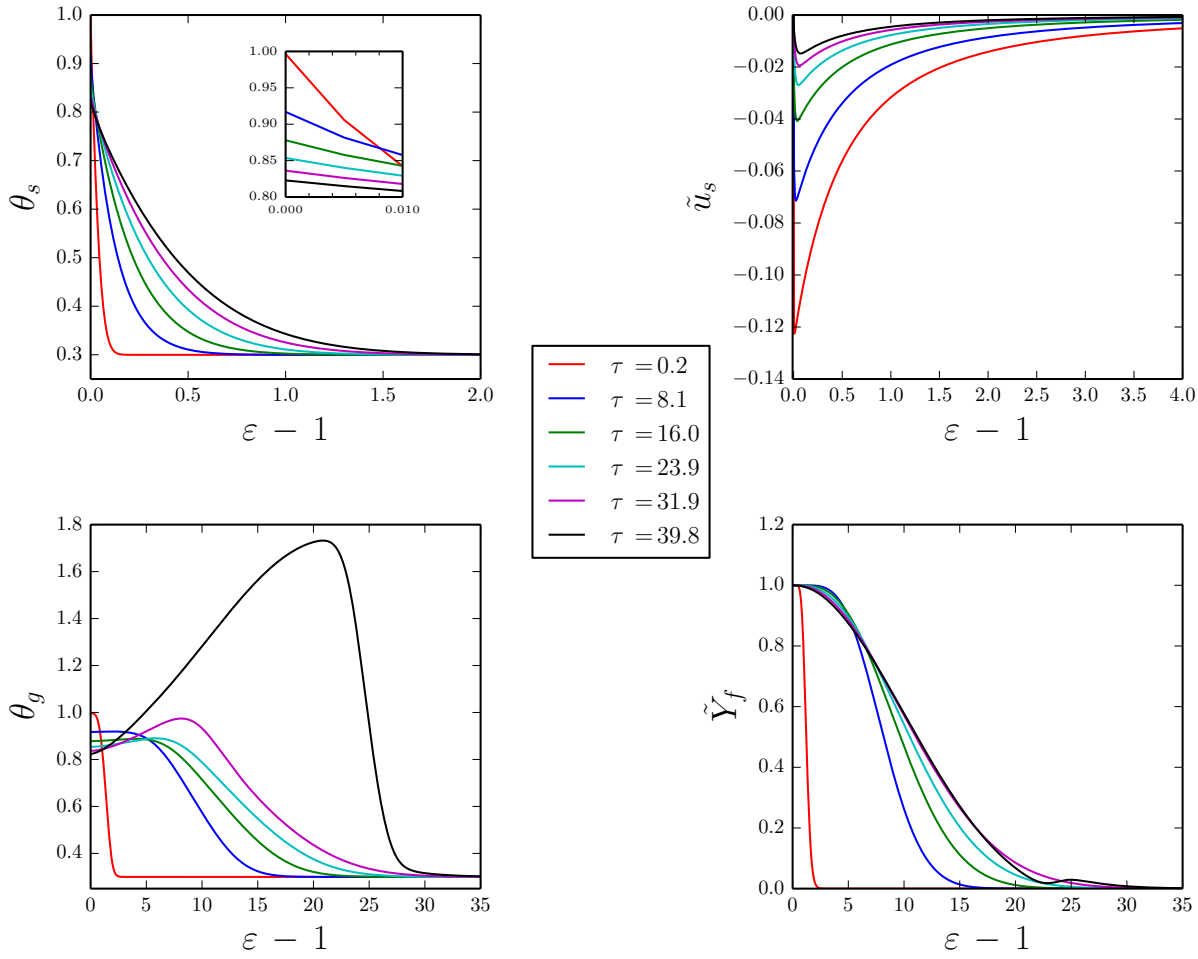

Figure 3.15: Modeling results for $d_p = 2$ mm and $T_i = 994$K.

gas phase also decrease more significantly with time. The plot of θ_g indicates that particle cooling has a dual effect on gas phase temperatures. Not only is the boundary temperature of the gas domain decreasing, but the reduced production of volatiles also keeps the mixing layer closer to the particle. This dual effect creates noticeable thermal gradients between the mixing layer and the particle and prompts heat diffusion back towards the relatively cool particle. Heat generation in the reaction zone must now overcome these losses in addition to losses to the cold surrounding air. In a way, this is similar to the canonical hot surface ignition problem, in that the thermal gradient normal to the surface is a controlling factor for ignition [3]. However in the particle ignition case, the normal gradient changes directions, which might more appropriately be called 'cooling-surface ignition'. A similar phenomena was observed by Takano et al. in their study of overloaded electric wires [20].

Decreasing the particle diameter further intensifies this cooling-surface effect, as demonstrated by the simulation results for $d_p = 0.625$ mm and $T_i = 1291$K in Figure 3.16. The

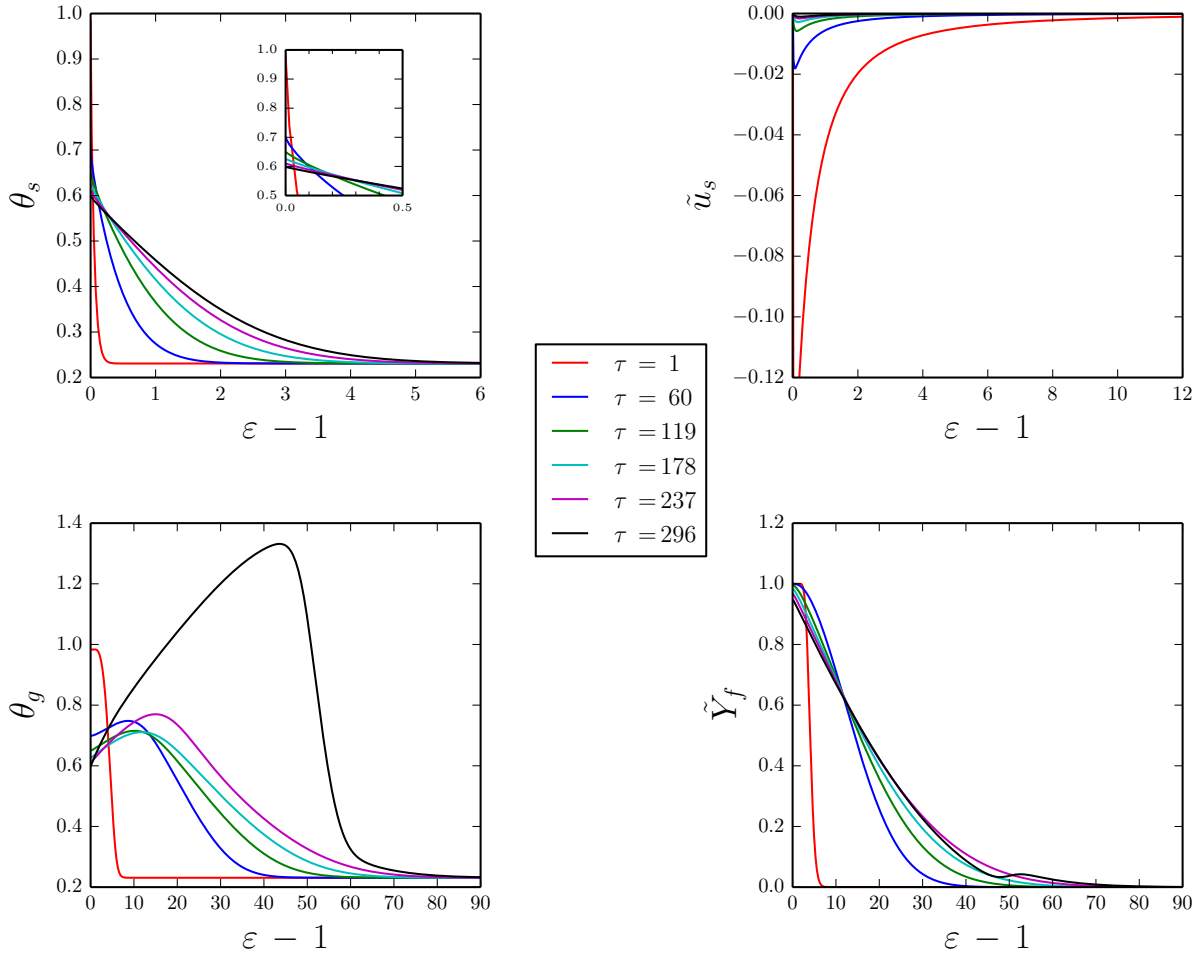

Figure 3.16: Modeling results for $d_p = 0.625$ mm and $T_i = 1291$K.

profiles of θ_s show that the particle cools very rapidly to 70% its original temperature before cooling even further. The result is that after $\tau = 1$ very little pyrolyzate is produced and the mixing layer remains very close to the particle surface. At $\tau = 60$, significant thermal gradients are already present between the particle and the reaction zone, and these gradients remain throughout ignition and even as the diffusion flame begins to form. Overcoming these sharp gradients requires high heat release rates in the gas phase and thus exceedingly high temperatures. In fact, ignition by the 0.625 mm particle is so sensitive to this heat release rate that a reduction of just 2K in the initial particle temperature prohibits ignition (i.e. $\tau_{ig} \to \infty$). As discussed in the previous chapter in section 2.6, a similar scenario was

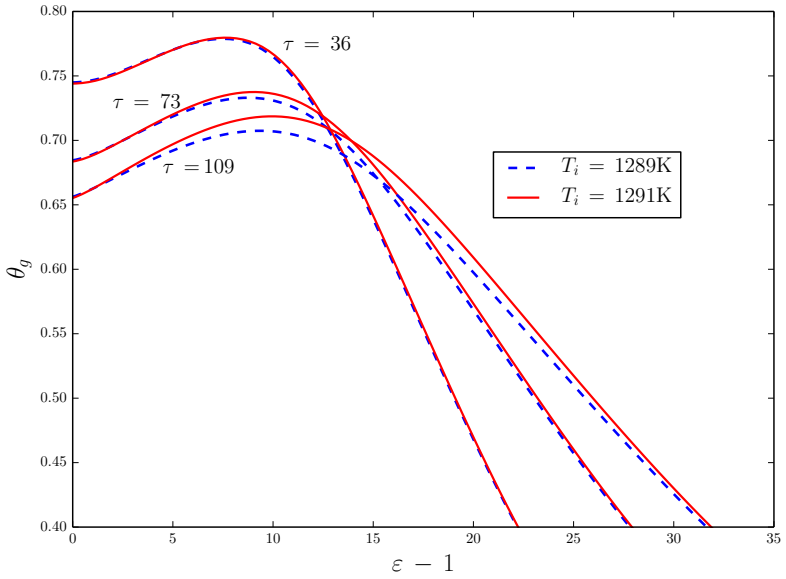

Figure 3.17: Gas phase temperature profiles at early times for $d_p = 0.625$ mm.

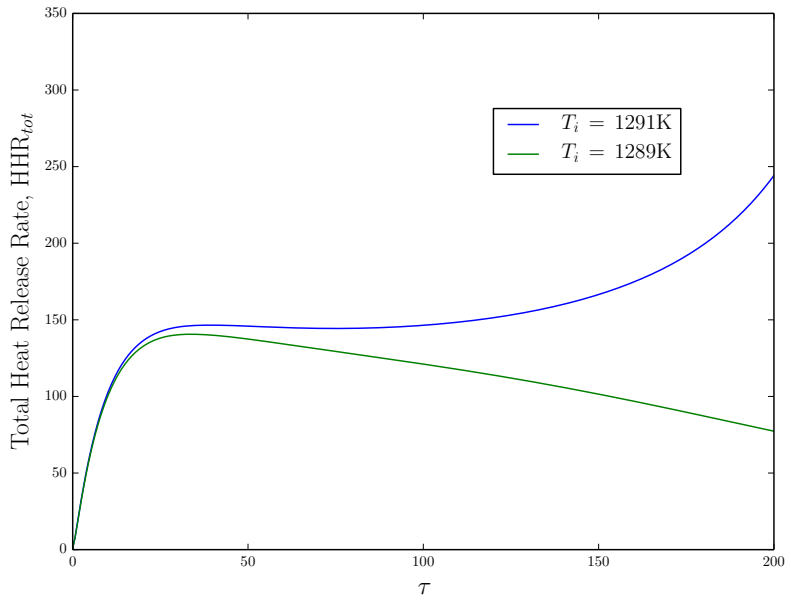

Figure 3.18: Total gas phase heat release rate as a function of time for $d_p = 0.625$ mm.

likely observed experimentally: random variations in the experiment changed particle losses slightly, resulting in different ignition outcomes for what appeared to be identical particles. This behavior differs from that of large particles where the ignition time varied smoothly with temperature, at least according to the simplified model.

Figures 3.17 and 3.18 demonstrate this sensitivity more clearly. Figure 3.17 shows early temperature profiles for a 0.625 mm particle at $T_i = 1289$K and $T_i = 1291$K. Although the profiles are initially equivalent, the peak temperatures for the two cases begin to differ, despite near identical temperature gradients bordering the reaction zone. Figure 3.18 shows the total heat release rate in the gas phase for the two cases, calculated as

$$\mathrm{HRR}_{tot} = \int_{\varepsilon_p}^{\infty} 4\pi\tilde{Q}\mathrm{Da}\,\tilde{Y}_f\,\tilde{Y}_o\,e^{\mathrm{Ar}_g\left(1-\frac{1}{\theta_g}\right)\varepsilon'^2}\,\mathrm{d}\varepsilon'.$$

Again, the HRR profiles are the same at early times, but due to the slightly higher initial temperature, the heat release rate for the $T_i = 1291$K is higher and eventually takes off.

In sum, for the smallest particles, reducing the particle size steepens the thermal gradient between the particle and the reaction front, requiring higher heat release rates and as such much higher temperatures. Even at these high temperatures (e.g. 1289K), ignition will still not occur unless some heat release rate threshold is overcome (e.g. at 1291K); ignition then occurs relatively quickly. Continuing to decrease particle diameter would eventually result in a case where the mixing layer was so thin and so close to the particle that the required temperature would asymptote towards the infinite. As evidence of this behavior, simulation results with a 0.25 mm particle never resulted in ignition, even for temperatures up to 2300 K. The importance of small particle cooling explains the sensitivity to particle bulk energy observed experimentally; the bulk energy of a small particle must be high enough such that, even after cooling, the particle temperature remains high (and the gas phase gradients small). It also explains the greater variation in the experimental ignition data for small spheres. Minor variations in the experiment result in increased particle cooling and increased gradients between the reaction zone and the particle.

3.5.4 Comparison with experimental data

To compare the model results with the experimental data, the author used the model to find ignition temperature as a function of diameter, in keeping with the form of the experimental results of Chapter 2. As mentioned in the previous subsection, the current model predicts ignition time to vary continuously with initial particle temperature, especially for large particles. For the purposes of finding a physically relevant ignition limit, ignition is said to 'occur' if the model predicts an ignition time less than 1 sec. While this cutoff is somewhat arbitrary, it is chosen based on the experimental results, which indicate that ignition occurs

very rapidly ($t_{ig} \ll 1$ sec) or not at all. Programmatically, the ignition time is defined as the point at which the maximum temperature in the gas phase increases from one time step to the next; such a condition indicates a temperature spike that is only possible if heat generation is greater than heat losses at the reaction front. Future work will investigate alternative definitions of the ignition limit. Based on these criteria, ignition temperatures were found using two algorithms: the first bounded the ignition limit within a 50K temperature interval, while the second repeatedly bisected this interval until the ignition temperature was obtained within 1K. As an example, the Python code employing these algorithms to find the model ignition limit for different particle volumetric heat capacities can be found in Appendix B.

Figure 3.19 shows the model ignition limit along with the curve for 5% flaming ignition probability based on experimental results. Note that the three diameter-temperature pairs discussed in the previous subsection were taken from points along the model ignition limit. The two curves are very similar qualitatively, suggesting that the conclusions drawn in the

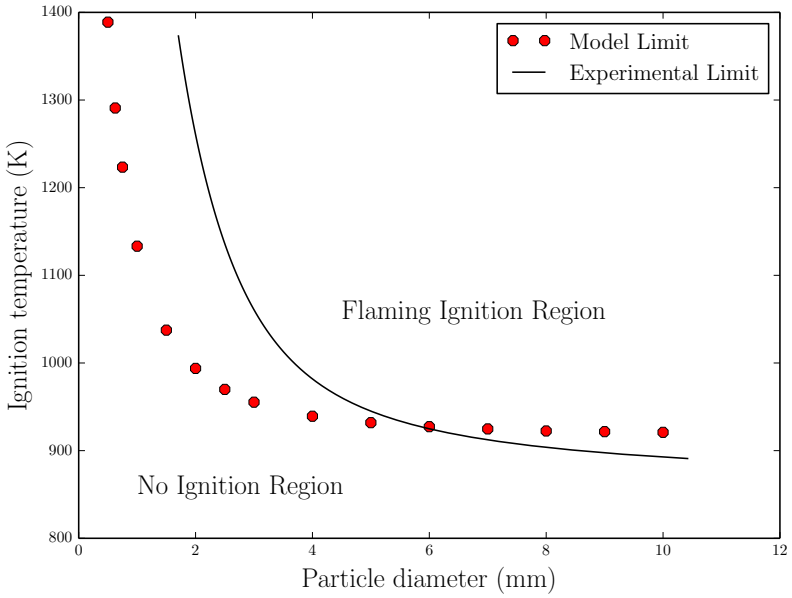

Figure 3.19: Ignition temperature as a function of particle diameter according to model and experiments.

preceding subsection are applicable to ignition experiments despite the model's simplicity. Quantitatively, there is decent agreement for large diameters, but this is a result of the model parameters as discussed in subsection 3.5.2. However, the model predicts ignition by very small particles at temperatures less than 1100°C, something not observed in experiments. There are a number of possible reasons for this discrepancy, some related to the gas phase and others related to the solid phase. As discussed in the next subsection, the lumped particle

assumption is most accurate for smaller particles, such that it would not be the cause of the differences in the ignition limits.

In the gas phase, it is possible that the combustion mechanism is overly simplified, resulting in an inaccurate representation of the heat release rate that has been shown to be important. Even if the simplified combustion scheme is sufficient, the gas phase chemistry constants may be chosen poorly. The model's neglect of density changes and buoyancy also means it does not account for increased convective losses. If these losses are most important at the reaction zone one would expect them to affect ignition behavior at all particle sizes. If the dominant effect is to increase particle cooling, that would shift the cooling-surface ignition regime to larger diameters. Further study is required to determine the relative magnitude of these losses.

A notable assumption made in the model is that the solid is dry, whereas the experimental cellulose fuel bed was lab conditioned with a moisture content of around 7%. Compared to a dry fuel bed, a moist fuel bed would require greater heat transfer from the particle to evaporate the water in addition to decomposing the solid. This additional cooling effect would be most important for small particles since heat losses are small relative to the thermal mass of larger particles. As such, the dry-fuel-bed assumption may be the reason for the differences shown in Figure 3.19. Inaccurate solid phase parameters could also be responsible for the discrepancy, since they would result in higher volatile flow and less particle cooling. Improved solid-phase kinetics may even be able to account for moisture effects [2]. In other studies of solid phase decomposition, sensitivity analyses often identify the thermal conductivity and the heat of decomposition reaction as the two most sensitive parameters, and future work will focus on how they affect the model results [9]. Another possibility is that the grid size in the solid phase is still not small enough to accurately resolve the temperature gradients that result from very small, extremely hot particles. As mentioned previously, future work will also include a grid sensitivity analysis of the model.

3.5.5 Validity of lumped particle assumption

Additional insight can be gained about hot-particle ignition by assessing the validity of the lumped particle assumption, particularly near the flaming/non-flaming limit. To truly test its validity would require complete knowledge of ignition times and particle heat losses in the experimental system, but one has greater confidence in the current model if it is at least internally consistent (i.e. model predictions agree with the lumped assumption).

For a solid body to be considered lumped, temperature gradients within the body must be much less than temperature gradients outside the body. The Biot number, Bi, is the ratio

of external to internal thermal resistance and is defined as

$$\mathrm{Bi} = \frac{hl}{\lambda}.$$

Here, h is the external heat transfer coefficient, l is the characteristic length of the body and λ is the thermal conductivity of the body. l is frequently taken as the ratio of the body's volume to its surface area. When $\mathrm{Bi} \ll 1$, the lumped condition is met. Often, one assumes that for $\mathrm{Bi} \leq 0.1$ the lumped assumption is valid. For a spherical stainless-steel particle, $l = {r_p}/{3}$ and $\lambda_p = 21.5$ W/mK. An estimate of h is more difficult to find since the particle losses heat through all three modes of heat transfer. An alternative approach is to compare the temperature profile of a lumped particle with $\mathrm{Bi} = 0.1$ to the particle temperature profiles according to the model. If the model particles are cooling more quickly than the 'critically lumped' particle, then their Biot numbers are greater than 0.1 and the model is not internally consistent. The temperature of a cooling lumped body as a function of time is

$$T(t) = (T_i - T_\infty)\, e^{-Aht/V\rho c} + T_\infty, \tag{3.5.1}$$

where A is the body surface area, V is the body volume, ρc is the body volumetric heat capacity and T_i is the initial temperature of the body. Nondimensionalizing equation 3.5.1 as in subsection 3.3.8 yields

$$\Theta_p(\tau) = \frac{\theta_p - \theta_\infty}{1 - \theta_\infty} = e^{-9\mathrm{Bi}\mathbf{A}_p\tau}.$$

where $\mathbf{A}_p = {\alpha_p}/{\alpha_g}$ is the particle-to-gas ratio of thermal diffusivity. Figure 3.20 plots $\Theta_p(\tau)$ for the $\mathrm{Bi} = 0.1$ case along with the three ignition scenarios discussed in subsection 3.5.3. For all values of τ the model particles are cooling less significantly than the critically-lumped particle, indicating that their approximate Biot numbers are less than 0.1 and they meet the lumped criterion.

However, because the heat flux at the particle surface affects the solid and gas domains, an additional requirement is that the timescales of interest are large relative to the diffusion time in the body. When this is not the case, temperature gradients over the whole of the particle may be small, but the gradients near the surface may be large. The Fourier number, Fo, is a measure of the timescale of interest relative to the characteristic diffusion time of a

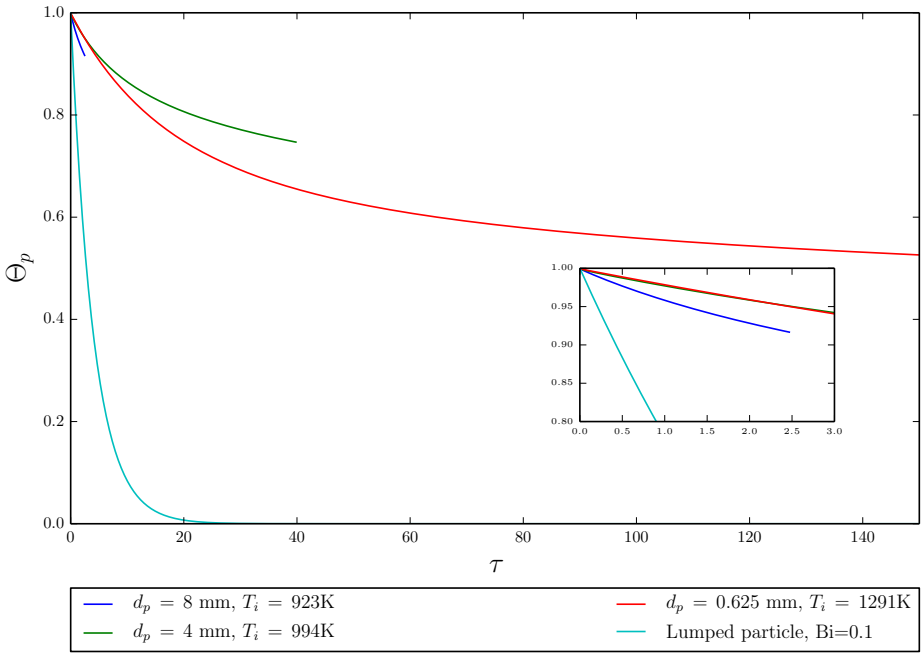

Figure 3.20: Particle temperature as a function of time for different particle sizes and Bi=0.1.

solid body, defined as

$$\text{Fo} = \frac{\alpha t}{l^2},$$

where α is the thermal diffusivity of the solid, t is the timescale of interest and l is the characteristic length of the body. When Fo $\gg$ 1, one is assured that overall temperature gradients and surface temperature gradients are similar, and one often approximates this with the criterion that Fo $\geq$ 10. The most obvious timescale associated with the ignition process is the ignition time, t_{ig}. While the physics at much earlier times are likely important, this serves as a reasonable starting point for this assessment. The ignition time has a corresponding ignition Fourier number, Fo_{ig}. Table 3.3 shows t_{ig} and Fo_{ig} for the three representative limit scenarios from subsection 3.5.3.

Clearly, the ignition times for the 8 mm and 2 mm particles are due to the definition of the ignition limit used for this study. Still, the Schlieren images indicate that t_{ig} is on the order of 0.1-1 s near the ignition limit, such that small metal particles completely meet the lumped criteria according to the model. It is also clear that large particles do not meet the Fo criterion. Internal gradients near the surface of these particles are significant, and one expects particle surface temperatures to be lower than predicted by the model. In this case,

Table 3.3: Ignition times and Fourier numbers for 3 representative scenarios.

Particle diameter (mm)	**Particle temperature (K)**	t_{ig} (s)	$\mathrm{Fo_{ig}}$
8	923	0.99	3
2	994	0.99	49
0.625	1291	0.78	392

losses between the combustion reaction front and the particle might be important even for large particles.

Even so, if one considers the extreme of a semi-infinite particle where only a thin outer layer is in thermal contact with the gas and solid domains (i.e. $\mathrm{Fo_{ig}} \ll 1$), the difference between the particle surface temperature and its original temperature may be small depending on the thermal properties of the system. As an example, consider the classic solution for contact between two semi-infinite bodies. According to this solution, the temperature at the contact surface between the two bodies is

$$T_{surf} = \frac{\sqrt{(\lambda\rho c)_A}T_{A,i} + \sqrt{(\lambda\rho c)_B}T_{B,i}}{\sqrt{(\lambda\rho c)_A} + \sqrt{(\lambda\rho c)_B}},$$

where the subscripts A and B denote the two bodies. Treating body A as stainless steel with an initial temperature of 923K and body B as powdered cellulose at room temperature, the resulting surface temperature is $T_{surf} = 902.7$K, just 2.3% less than the initial stainless steel temperature. Considering body B as air results in an even smaller difference.

Of course, this simple example does not account for convective or radiative heat transfer or increased losses due to the volatilization reaction. Regardless, it demonstrates that if the particle thermal inertia $(\lambda\rho c)_p$ is sufficiently large relative to the thermal inertias of the fuel bed and gas phase, the particle surface temperature will remain near the impact temperature. Interestingly, a large thermal conductivity would increase both the thermal inertia *and* the thermal diffusivity of the particle. For a given ignition time, a higher thermal diffusivity would make the lumped assumption valid for larger particles. And, even if the Fo criterion is not met for large particles, the model results may still be quite accurate if particle thermal inertia is sufficiently high. Therefore, one can clarify the conclusions of subsection 3.5.3 by stating that large-particle temperature is more likely to remain approximately constant over the ignition time if the particle has a high thermal conductivity. As particle size decreases,

the lumped assumption is more valid, heat losses from the reaction zone to the particle are important and particle size and energy play a more prominent role.

A final caveat to this analysis: the deviation between the experimental and model results suggests that particle heat losses are severe for particles less than 4 mm in size, such that in real-world ignition the Bi criterion may only met by the smallest particles. If the experimentally studied particles were not effectively lumped, then thermal conductivity would play a role in heat transfer from the particle surface. In this case, copper's large relative thermal conductivity may explain the deviation of the copper ignition limit observed in Chapter 2.

3.6 Effect of varying particle properties

After using the model to explore the ignition physics for the base case of stainless steel spheres igniting cellulose, the author used the model to investigate the effect of variable particle properties on the ignition limit, mirroring the experimental investigation of particle metal type. With the assumption of a lumped particle, the particle properties that one may change are the volumetric heat capacity $(\rho c)_p$ and the emissivity ϵ. By assuming an arbitrary melting range and holding other properties constant, one can also determine the effect of particle melting. In terms of the dimensionless governing equations, changing the volumetric heat capacity changes the ratios b_g and b_s. Changing the emissivity changes the value of ψ, and particle melting manifests itself as a non constant γ_p. The fuel bed was cellulose for all the simulations discussed in the following subsections, and the ignition limit for stainless steel particles was treated as a base case against which comparisons could be made. For reference, the experimental no-ignition limits for the different particle metals are reprinted in Figure 3.21.

3.6.1 Volumetric heat capacity and emissivity

A brief survey of possible hot particle materials indicates that volumetric heat capacities can vary from around 2-4 MJ/m^3 for metals down to around 1 MJ/m^3 for carbonaceous materials [21, 22]. Thus, ignition temperature was found as a function of particle diameter for $(\rho c)_p$ from 3.9 MJ/m^3 (representing stainless steel) down to 1 MJ/m^3. The results are shown in Figure 3.22. Qualitatively, all of the curves are similar, indicating that the general physics described in the previous section apply across a large range of volumetric heat capacities. As expected, the smaller the particle, the more prominent the effect of changing $(\rho c)_p$ because particles with lower thermal mass cool faster. In contrast with the experimental ignition limits, which agreed across metal types for smaller diameters, the model results indicate that ignition temperatures can increase by more than 100K for a decrease from $(\rho c)_p = 3.9$ MJ/m^3 to $(\rho c)_p = 2$ MJ/m^3. This difference may be explained by the factors discussed in subsection 3.5.4 or they simply may not be observable due to random variation in the experiments. For $(\rho c)_p = 1$ MJ/m^3, ignition temperatures are noticeably higher and it

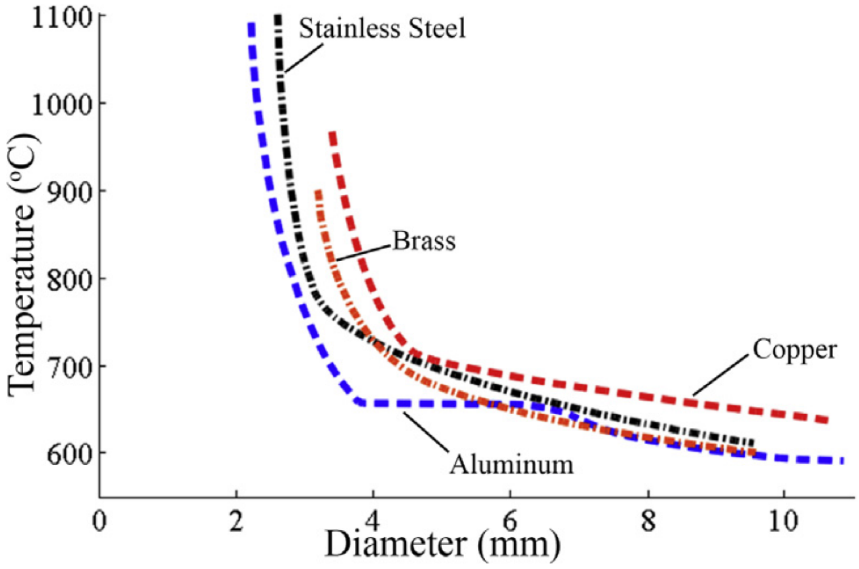

Figure 3.21: No-ignition limits of the four metals tested, reprinted from Chapter 2

is clear that even larger particles experience significant cooling. This suggests that, at a given temperature and neglecting oxidation reactions, hot carbonaceous particles and even hot abrasive fragments (for which $(\rho c)_p < 2$ MJ/m^3) are less hazardous than their metal counterparts.

Emissivity was varied between 0.1 and 0.9 while keeping all other properties as those of stainless steel. As one might expect, decreasing emissivity decreases the particle heat losses due to radiation and allows ignition at lower temperatures. However, this decrease in ignition temperature relative to the base case is extremely small, with a max temperature drop of less thank 4K between the $\epsilon = 0.9$ and the $\epsilon = 0.1$ cases. This suggests that over the timescales important for ignition, radiation is overshadowed by the other modes of particle heat loss. Given the low thermal conductivity of air, heat conduction into the solid is likely the largest source of particle heat loss, but assessing the relative magnitudes of heat losses due to conduction and convection of volatiles requires further investigation.

3.6.2 Melting

A melting range of $\Delta T_m = 925 - 940$K and a heat of melting of $\Delta H_m = 400$kJ/kg were chosen so as to approximate the melting characteristics of aluminum alloy 1100. However, it should be noted that all thermal properties were equivalent between the solid and molten phases and taken as the values for stainless steel. In this way, the effect of the additional heat of melting alone can be assessed. The ignition limits for the melting and non-melting

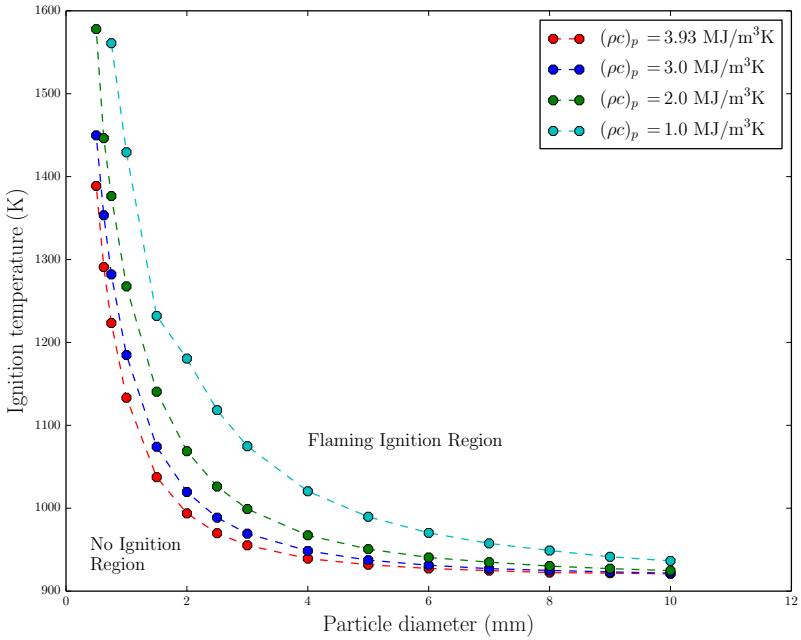

Figure 3.22: Ignition temperature as a function of particle diameter for different volumetric heat capacities.

base case are shown in Figure 3.23. The bulk sphere energy along the ignition limit for the two cases is shown in Figure 3.24. Clearly, the model predicts that melting reduces the temperature required for hot particle ignition. As expected from the experimental results, the energy of particles along the ignition limit is he same for the melting and non-melting cases. This agrees with the experimental results for aluminum and indicates that the heat of melting is the dominant reason for the shape of the aluminum ignition limit (as opposed to changes in molten particle shape or differing solid/molten properties). Another point of agreement is that both model and experimental results show a reduced effect for the smallest particles. Quantitatively, the curvature of the melting ignition limit is much sharper in the model case, but some discrepancy is expected due to the differences between experimental and model results for the base stainless steel case.

The large differences between the ignition limits in Figure 3.23 indicate that sub-3 mm particles, whether solid or molten, rapidly cool to temperatures in the melting range. At that point, the high effective heat capacity in the melting range resists further cooling of the molten particle. In comparison to the solid particle, gas phase temperature gradients near the molten particle are reduced and lower gas phase temperatures are required for ignition. If for some reason a small particle never cooled down to the melting range, then

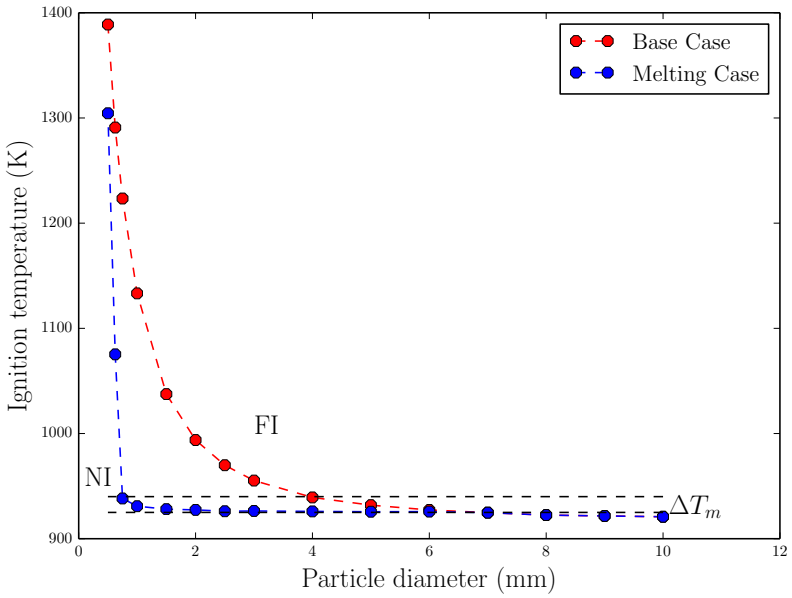

Figure 3.23: Ignition temperature as a function of particle diameter for melting and non-melting particles.

it would be irrelevant whether the particle was solid or molten (particle properties are the same for both phases). This explains why the differences in the ignition limits decrease for very small particles. Ignition by such small particles requires temperatures well above the melting range, and as such the sensible enthalpy loss above the liquidus temperature is more important than latent heat loss. Thus, the solid and molten ignition temperatures are more similar for very small (< 0.75 mm) particles. In sum, melting increases the effective energy available to a particle, but only in the event that the particle temperature drops below the liquidus temperature.

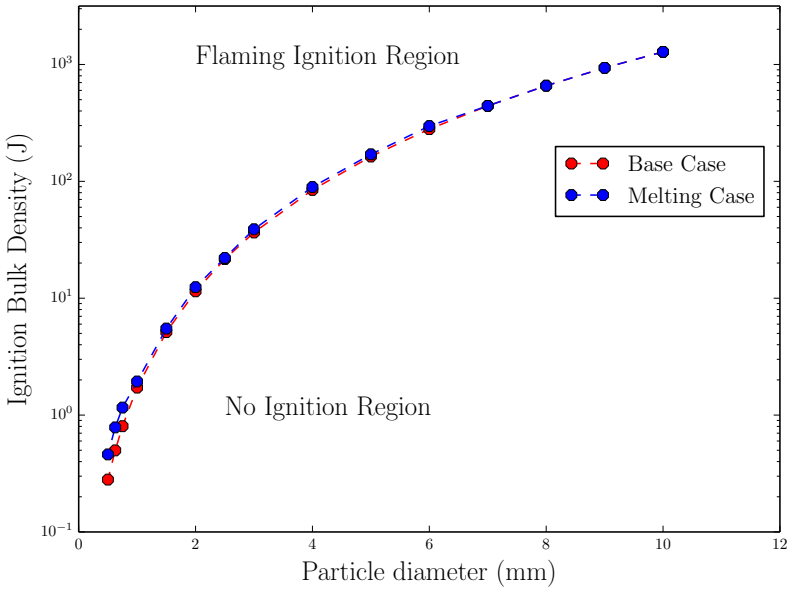

Figure 3.24: Bulk ignition energy as a function of particle diameter for melting and non-melting particles.

3.7 References

[1] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Mcgraw-Hill, 1980.

[2] Christopher Lautenberger. "A Generalized Pyrolysis Model for Combustible Solids". PhD thesis. University of California, Berkeley, 2007.

[3] C.K. Law. *Combustion Physics*. Cambridge University Press, 2006.

[4] C. Di Blasi. "HEAT, MOMENTUM, AND MASS TRANSPORT THROUGH A SHRINKING BIOMASS PARTICLE EXPOSED TO THERMAL RADIATION". In: *Chem. Eng. Sci.* 51 (1995), pp. 1121–1132.

[5] B.J. McBride, S. Gordon, and M.A. Reno. *Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species*. Tech. rep. 4513. Cleveland, Ohio: Lewis Research Center, National Aeronatics and Space Administration, 1993.

[6] G. Agarwal and B. Lattimer. "Method for measuring the standard heat of decomposition of materials". In: *Thermo. Acta* 545 (2012), pp. 34–47.

[7] MatWeb. *302 Stainless Steel, annealed strip*. May 2015. URL: `http://www.matweb.com/search/DataSheet.aspx?MatGUID=f290e46425d648f6963d16a9a80e63c4&ckck=1`.

[8] Omega. *Emissivity of Common Materials*. May 2015. URL: `http://www.omega.com/literature/transactions/volume1/emissivitya.html`.

[9] H.C. Kung. "A Mathematical Model of Wood Pyrolysis". In: *Comb. Flame* 18 (1972), pp. 185–195.

[10] C. Lautenberger and A.C. Fernandez-Pello. "Modeling Ignition of Combustible Fuel Beds by Embers and Heated Particles". In: *Forest Fires 2008*. 2008.

[11] H.C. Kung. "Modeling the propagation of drying and decomposition fronts in wood". In: *Comb. Flame* 139 (2004), pp. 16–27.

[12] G.W. Bradbury, Y. Sakai, and F. Shafizadeh. "A Kinetic Model for Pyrolysis of Cellulose". In: *J. App.Ploy. Sci.* 23 (1979), pp. 3271–3280.

[13] J. Cho, J.M. Davis, and G.W. Huber. "The Intrensic Kinetics and Heats of Reaction for Cellulose and Char Formation". In: *ChemSusChem* 3 (2010), pp. 1162–1165.

[14] C. Di Blasi. "Modeling chemical and physical processes of wood and biomass pyrolysis". In: *Prog. Comb. Sci.* 34 (2008), pp. 47–90.

[15] J.L. Banyasz et al. "Gas evolution and the mechanism of cellulose pyrolysis". In: *Fuel* 80 (2001), pp. 1757–1763.

[16] K. Nakabe et al. "Ignition and Transition to Flame Spread Over a Thermally Thin Cellulosic Sheet in a Microgravity environment". In: *Comb. Flame* 98 (1994), pp. 361–374.

[17] K.B. McGrattan, T. Kashiwagi, and H.R. Baum. "Effect of Ignition and Wind on the Transition to Flame Spread in a Microgravity environment". In: *Comb. Flame* 106 (1996), pp. 377–391.

[18] S. McAllister, J.Y. Chen, and C.A. Fernandez-Pello. *Fundamentals of Combustion Processes*. Springer, 2011.

[19] A. Liñan and A. Crespo. "An Asymptotic Analysis of Unsteady Diffusion Flames for Large Activation Energies". In: *Comb. Sci. Tech.* 14 (1976), pp. 95–117.

[20] Y. Takano et al. "Ignition limits of short-term overloaded electric wires in microgravity". In: *Proc. Comb. Inst.* 34 (2013), pp. 2665–2673.

[21] The Engineering Toolbox. *Classification of Coal*. May 2015. URL: `http://www.engineeringtoolbox.com/classification-coal-d_164.html`.

[22] The Engineering Toolbox. *Solids- Specific Heats*. May 2015. URL: `http://www.engineeringtoolbox.com/specific-heat-solids-d_154.html`.

Chapter 4

CONCLUSION

Up to this point, research on hot particle spot fire ignition has largely focused on particle generation and transport. A small number of studies have examined what occurs after a hot particle contacts a natural fuel bed, but until recently the process remained poorly understood. Past experimental studies have considered several effects: particle size, particle temperature, fuel bed composition and fuel bed moisture content [1–6]. Finney et al. also discussed the effects of particle metal type on ignition, as did Stokes, but neither work investigated these effects rigorously [3, 6]. In the study by Finney et al., metal-type effects were confounded by size effects, while Stokes only observed that aluminum and steel particles present a greater hazard because they can combust during flight. While this is true, up to this point no study has thoroughly investigated the effect of changing particle properties in the absence of particle combustion. Overall, the focus of these studies has been on assessing apparent hazard and not on understanding the ignition process. Theoretical studies have either applied hot spot theory or considered ignition by self-heated particles and firebrands; none have explicitly addressed the case of a hot body that cools upon impacting a fuel bed [7–13].

Researchers in the author's laboratory were the first to study the effects of particle size and temperature systematically [14]. Even so, they did not address the random variation that is apparent in their results and the results of previous experimental studies. Characterizing this random variation is important for defining conservative ignition limits, which can be used as the basis for intelligent work-site and clear-cutting regulations. As an example, one might consider conducting an experiment where two equal-sized hot particles are dropped onto a fuel bed, one at a slightly lower temperature than the other. If the hotter particle ignited the fuel bed and the other particle did not, one might assume that they have bounded the ignition temperature for particles of that size. However, conducting a larger number of ignition tests and treating the experimental data statistically would likely reveal that there is a small but finite chance of ignition at lower temperatures. Given the large number of particles that can be produced by processes like power line arcing and hot work, this small

ignition probability can still represent a real fire hazard.

In addition to its relevancy to fire safety, hot particle spot fire ignition is a fundamentally interesting problem that combines all modes of heat and mass transfer with solid phase pyrolysis chemistry and gas phase combustion. The goals of the work presented in this dissertation were to: 1) increase the understanding of what happens when a hot particle contacts a cellulosic fuel in an oxidizing environment, and 2) determine how different particle properties affect this process. To achieve these goals, a large number of ignition experiments varying particle size, temperature and metal type were conducted in a controlled laboratory setting. The results of these experiments were statistically analyzed to find approximate ignition boundaries and identify boundary trends with respect to the particle parameters of interest. Schlieren images recorded during the ignition experiments were also used to more accurately describe the ignition process. Based on these images, a theoretical model of hot particle spot fire ignition was developed and used to explore the experimental trends further. Model simulations identified the important physics controlling ignition for different sized particles and clarified many of the trends observed in the experimental data. The major findings of these efforts are summarized in the next section. The final section outlines future work to be undertaken by the author and identifies remaining work still to be conducted on the topic of hot particle spot fire ignition.

In addition to the findings described below, this work presents a few techniques and approaches that are novel, at least to the best of the author's knowledge. The use of logistic regression has seen very limited use in gas phase ignition studies, and prior to this work it had not been applied to solid phase ignition data [15]. Since publication of this work, several studies have applied it to solid fuel ignition, but none have used it in conjunction with maximum likelihood estimation to find regression parameters [16, 17]. In terms of modeling, the choice to treat the system as two hemispherical domains connected by a lumped particle is also new. Despite its simplicity, this approach has yielded valuable information about the phenomena controlling hot particle spot fire ignition.

4.1 Summary of major findings

As one would expect, smaller particles require higher minimum impact temperatures to ignite. Flaming ignition of powdered cellulose by particles $\leq$ 11 mm in size requires particle temperatures of at least 600°C. Even accounting for the melting effects of aluminum, particles 4 mm and smaller require temperatures of at least 650°C. Experimentally, ignition was never observed for 2 mm particles at temperatures up to 1100°C. Even so, the statistical analysis indicates that ignition by particles 2 mm and smaller may be possible at temperatures above 950°. Recent work conducted in the author's laboratory suggests that powdered cellulose fuel beds with bulk densities in the range of this study are more easily ignited than natural fuels such as dry grass and pine needles [18, 19]. That being the case, the ignition temperatures presented here can be considered conservative benchmarks, and they are summarized below

in Table 4.1. However, they should be treated with caution, as they do not account for variations in a number of important (and as yet unstudied) parameters that are discussed in section 4.2. High-speed Schlieren images taken during the ignition experiments show that

Table 4.1: Guidelines for hot particle spot fire ignition of powdered cellulose.

Particle size range (mm)	**Conservative minimum ignition temperature (°C)**
$4 - 11$	600
$2 - 4$	650
< 2	950

once particles land, they volatilize the powdered cellulose and the fuel vapor diffuses out into the surrounding air. Larger particle pyrolyze significantly more fuel than smaller particles, resulting in more violent ignitions. Ignition occurs in the mixing layer between the vapor and the air, either during the initial expansion of the pyrolyzate away from the particle, or after a stable plume of volatiles has formed. Flash-point type ignition is possible, and for a stable flame to form the particle has to pyrolyze fuel even after ignition occurs. Random variations in the experimental data appear to be due to variations in impact dynamics (particularly bouncing), as well as the degree of fuel bed penetration.

For very small spheres, the temperature required for ignition is very sensitive to particle size, while for very large spheres, ignition temperature shows only a weak dependence on that variable. Between these two extremes, there is a transition region where the size-sensitivity of ignition temperature varies smoothly with the particle diameter. Modeling results indicate that in the large-particle, high-conductivity limit, the particle's surface temperature remains close to its impact temperature over the timescales of ignition. As a result, particle thermal properties are unimportant and the ignition occurs when heat generation in the mixing layer is sufficient to overcome losses to the surrounding air. This seems to explain why minimum ignition temperature is less sensitive to particle size for larger spheres: because particle temperature is constant, the thermal mass of the particle and the amount of pyrolyzate produced are less important than the temperature within the mixing layer.

When the large-particle limit does not apply, the particle cools upon impact with the fuel bed. As a result of this cooling, the particle volatilizes a finite amount of fuel and the mixing layer remains close to the particle. Due to its proximity to the relatively cool particle, the reaction zone experiences greater losses and must generate a larger amount of heat for ignition to occur. The physics that are important in the large particle limit are still present, but they are complicated by the losses to the particle. This effect becomes more drastic as particle size decreases, and seems to explain why ignition by small spheres is more sensitive

to random variation in the ignition experiments. Random variations can affect how quickly a particle cools, but would have less effect on a 'thermally massive' large particle.

Because cooling is so important, the initial bulk energy is more useful than impact temperature for predicting ignition by smaller particles. Along those lines, the additional heat of melting available to molten particles helps to resist particle cooling; as such, molten particles can ignite at lower temperatures than solid particles of the same size with similar thermal properties. This phenomena was observed experimentally for aluminum particles $3.5 - 7$ mm in diameter. When particle size is small enough, the particle cools extremely rapidly and the gas phase temperatures required for ignition are well above the liquidus temperature of the particle. For these particles, the additional heat of melting is less important. It is important to note that a particle's diameter affects the small-particle ignition process beyond its role in defining bulk energy; it also affects the rate of particle losses, the amount of fuel volatilized, and the physics described by the large-particle limit.

Experimental results suggest that varying particle volumetric heat capacity between 2.4 and 3.9 MJ/m^{3}K does not greatly affect the minimum ignition temperature. Model results corroborate this observation, and indicate that decreasing the volumetric heat capacity in this range increases the minimum ignition temperature somewhat for small spheres, and minimally for large spheres. In comparison, lower values of $(\rho c)_p$ (e.g. 1 MJ/m^{3}K) increase the importance of particle cooling, and thus the minimum ignition temperature, for all but the largest particles of interest. Experimental results do not show a clear trend with particle emissivity, and model results indicate that decreasing ϵ from 0.9 to 0.1 decreases the minimum ignition temperature by a maximum of 4K. This implies that, over the timescales of ignition, radiation heat transfer is small relative to other modes of particle heat loss. The final notable experimental observation is that copper particles require slightly higher ignition temperatures and seem more sensitive to experimental variation compared to aluminum, steel or brass particles. A likely explanation for this phenomena is that copper's significantly higher thermal conductivity causes it to cool more quickly, but the exact cause remains unclear. It may also be associated with the oxide layer that forms on copper particles during heating.

Finally, the results of the simple 1D model agreed qualitatively with the experimental results, but predicted drastically low ignition temperatures for particles $\leq$ 4 mm in size. There are a number of potential reasons for this discrepancy, and it is not clear whether insufficient model complexity or poorly chosen kinetic and transport constants are to blame.

4.2 Future work

The scope of this and previous work should make it clear that there is an enormous number of questions still to be answered concerning hot particle spot fire ignition. Even with respect to the simple model described in this study, there are a number of additional steps that can

be taken to improve and strengthen the model results. First, the author will investigate the sensitivity of the model to a number of parameters, both numerical and theoretical. As mentioned previously, the high heat fluxes created by small hot particles may not be adequately resolved by the current solid phase grid spacing, and a grid sensitivity study will be conducted to determine how fine a mesh is needed. Given the sensitivity of pyrolysis models to solid phase thermal conductivity and the heat of volatilization, these parameters will be varied within reasonable limits to determine if they can correct some of the discrepancy between the model and experimental results. The author will also determine the degree to which model accuracy can be improved by reducing the tolerance used in the convergence criteria. Similarly, the physically motivated but somewhat arbitrary ignition criteria used to find the model ignition limits will be reconsidered. Finally, the different modes of particle heat loss will be compared using the model to determine which mode is the largest and if there are any modes that can be neglected.

In addition to these improvements to the current model, researchers in the author's laboratory are working on other important aspects of the problem. The least studied of these aspects is smoldering ignition by hot particles. Once a smoldering reaction has been initiated, it can transition into a spot fire, and this transition can occur well after the initial particle impact. As a result, smoldering and its potential transition to flaming represent a serious fire hazard. The initiation of self-sustained smoldering fronts by hot particles is currently under investigation in the author's laboratory. From this fire safety perspective, it is also crucial to study the small and extremely hot particles that are produced by real-world processes like power line interaction and hot work. One can simulate these processes in the laboratory environment, but characterizing the particles they produce is difficult. In particular, impact temperature is difficult to measure, and knowledge of this variable is important for understanding the ignition process. When one uses particle heating methods that allow more accurate particle temperature measurements (such as an electric tube furnace), the maximum temperatures one can achieve are limited. Future work in the author's laboratory will focus on measuring a particle's temperature remotely using optical systems, and using these systems to investigate ignition by more realistic particle sources. For future experiments beyond these efforts, there are a large number of experimentally unstudied parameters that are potentially important to smoldering and flaming ignition, including: fuel bed moisture content, fuel bed temperature, powdered fuel bed density, cross-flow wind speed, ambient temperature, ambient humidity, and particle shape.

As demonstrated by the current study, numerical modeling of the hot particle ignition problem can give invaluable insight into the physics controlling physics and allow one to investigate aspects that are difficult experimentally. Along these lines, there are a number of improvements which can be made to the current model approach. First among these is treatment of the problem as a three-dimensional system such that one can accurately resolve temperature gradients in the particle and volatile flow near the particle. Efforts to create such a model are currently underway in the author's laboratory. Accounting for

buoyancy and a variable density in the gas phase would also be important. From there, more complexity could be added to the solid phase, including more complex composition, moisture loss, volume changes and more complex decomposition chemistry. Particle oxidation and combustion could also be considered. With all this additional complexity, the model would require accurate estimations of chemistry and transport constants, and this in itself would require a significant research effort. As a starting point, one could determine heats of reaction of high temperature/high heat flux pyrolysis pathways for cellulose and other components of biomass such as lignin, xylan and hemicellulose.

4.3 References

[1] G.E. Pleasance. *An Examination of Particles from Conductor Clashes as Possible Sources of Bushfire Ignition, App. 5*. Tech. rep. unpublished. Victoria, Australia: State Electricity Commission of Victoria, 1977.

[2] A.D. Stokes. "Fire Ignition by Electrically Produced Incandescent Particles". In: 10 (1990), pp. 175–187.

[3] A.D. Stokes. "Fire Ignition by Copper Particles of Controlled Size". In: *J. Elec. Electronics Engr., Australia* 10 (1990).

[4] G.W.G. Rowntree and A.D. Stokes. "Fire Ignition by Aluminum particles of Controlled Size". In: *J. of Elec. Electronics Engr., Australia* 14 (1994), pp. 117–123.

[5] T. Tanaka. "On the Flammability of Combustible Materials by Welding Splatter". In: *Reports of the Natl. Res. Inst. of Police Sci.* 30 (1977), pp. 151–158.

[6] M.A. Finney et al. *A Study of Ignition by Rifle Bullets*. Tech. rep. Fort Collins, Colorado: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2013.

[7] J.C. Jones. "Predictive Calculations of the Effect of an Accidental Heat Source on a Bed of Forest Litter". In: *J. Fire Sci.* 11 (1993), pp. 80–86.

[8] J.C. Jones. "Further Calculations Concerning the Accidental Supply of Heat to a Bed of Forest Material". In: *J. Fire Sci.* 12 (1994), pp. 502–505.

[9] J.C. Jones. "Improved Calculations Concerning the Ignition of Forest Litter by Hot Particle Ingress". In: *J. Fire Sci.* 13 (1995), pp. 350–356.

[10] A.N. Subbotin. "Study of the critical conditions of initiation of a surface fire". In: *Phys. Mech. Eco., Abstracts of the Int. Sci. Conf.* Tomsk, Russia, 1994, pp. 149–150.

[11] A.I. Zvyagils'kaya and A.N. Subbotin. "Influence of Moisture Content and Heat and Mass Exchange with the Surrounding Medium on the Critical Conditions of Initiation of Surface Fire". In: *Comb. Exp. Shock Waves* 32 (1996), pp. 558–564.

[12] A.M. Grishin et al. "Ignition of a Layer of Combustible Forest Materials". In: *Comb. Exp. Shock Waves* 34 (1998), pp. 613–620.

[13] C. Lautenberger and A.C. Fernandez-Pello. "Modeling Ignition of Combustible Fuel Beds by Embers and Heated Particles". In: *Forest Fires 2008*. 2008.

[14] R.M. Hadden et al. "Ignition of Combustible Fuel Beds by Hot Particles: An Experimental and Theoretical Study". In: *Fire Tech,* 47 (2011). http://dx.doi.org/10.1007/s10694-010-0181-x, pp. 341–355.

[15] S.P.M. Bane et al. "Statistical analysis of electrostatic spark ignition of lean H2/O2/Ar mixtures". In: *Int. J. Hydrogen En.* 36 (2011), pp. 2344–2350.

[16] D. Coldham, A. Czerwinski, and T. Marxsen. *Probability of Bushfire Ignition from Electric Arc Faults*. Tech. rep. Victoria, Australia: Energy Safe Victoria, 2011.

[17] S. Wang, H. Chen, and N. Liu. "Ignition of expandable polystyrene foam by a hot particle: An experimental and numerical study". In: *J. Haz. Mat.* 283 (2015), pp. 536–543.

[18] J.L. Urban, C.D. Zak, and C. Fernandez-Pello. "SPOT FIRE IGNITION OF NATURAL FUEL BEDS BY HOT ALUMINUM PARTICLES". In: *Fire and Materials 2015*. 2015.

[19] J.L. Urban, C.D. Zak, and C. Fernandez-Pello. "The Effect of Fuel Bed Composition on the Spot Fire Ignition of Natural Fuels by Hot Metal Particles". In: *The 10th Asia-Oceania Symposium on Fire Science and Technology,* 2015.

Appendix A

Python Code for 1D Model of Hot Particle Spot Fire Ignition

```
#import required modules
import numpy as np
from scipy.integrate import odeint
import pickle
import matplotlib.pyplot as plt
import time
from numba import jit

#NON-VARYING PARTICLE AND AMBIENT PARAMETERS
#*********************************************************************

c_p = 500 #use base units, this is the solid specific heat
c_pl = 500 #base units, liquid specific heat
deltaHmelt = 400*10**3 #heat of melting, J/kg
Ts = 925 #solidus temperature, in Kelvin
P = 0.5 #portion of surface area submerged in fuel bed
Tinf = 298 #in K

#NON-VARYING FUEL BED PARAMETERS
#*********************************************************************

#thermophysical
rho_s,l_s,c_s = 300,12.6*10**-2,2510 #base units, density from exp, but
#other two from Kung
#here l is short for lambda, symbol for thermal conductivity
a_s = l_s/(rho_s*c_s) #Thermal diff. in base units
```

```
w_p = 1 #pore emissivity
d_pore = 2*10**-5 #pore diameter, from DiBalsi pyrolysis paper
#thermochemical
Zs = 1*10**14 #acc. to Banyasz
# Zs = 6.5*10**14 #pre-exponential factor for tar reactionac. to chemsuschem
# Ea_s = 199*10**3 #activation energy for tar reaction acc. to CSC
# Ea_s = 195.8*10**3 #acc. to Banyasz
Ea_s = 200*10**3 #just making it a bit harder to py. Ea from tar with Z
#from formaldehyde
Hvol = 750*10**3 #heat of reaction for tar formation, endothermicity acc.
#for in eqns

#NON-VARYING GAS PHASE PARAMETERS
#*********************************************************************

#thermophysical
Ru = 8.314 #universal gas constant
rho_inf = 101.3*29/(Ru*Tinf)
l_inf = 10**-4*np.exp(0.94*np.log(Tinf) + 0.12*10**3/Tinf - 0.12*10**5/\
        (Tinf**2) - 0.12)
c_inf = (10**3/28)*Ru*(3.53 - 1.24*10**(-4)*Tinf - 5.03*10**(-7)*Tinf**2\
        + 2.44*10**(-9)*Tinf**3 - 1.41*10**(-12)*Tinf**4)
a_g = l_inf/(rho_inf*c_inf)

#create two arrays, one for temp and one for enthalpy, for switching back and
#forth
Tref = np.linspace(298,3000,1500)
v1 = 3.53*np.ones_like(Tref)
v1[Tref >= 1000] = 2.95
v2 = -1.24*10**(-4)*np.ones_like(Tref)
v2[Tref >= 1000] = 1.40*10**-3
v3 = -5.03*10**(-7)*np.ones_like(Tref)
v3[Tref >= 1000] = -4.93*10**(-7)
v4 = 2.44*10**(-9)*np.ones_like(Tref)
v4[Tref >= 1000] = 7.86*10**(-11)
v5 = -1.41*10**(-12)*np.ones_like(Tref)
v5[Tref >= 1000] = -4.61*10**(-15)
href = (10**3/28)*Ru*(v1*Tref + 0.5*v2*Tref**2 + v3*Tref**3/3 +\
        0.25*v4*Tref**4 + 0.2*v5*Tref**5)

#thermochemical
# Zg = 1.55*10**10 #units are [(kmol/m^3)^(1-a-b2)]/s for eth
```

```
Zg = 8*10**5 #units are m^3/(kg*s) for cellulose
Ea_g = 100000 #units are J/mol, just trying to match data
# Ea_g = 67000 #units are J/mol for cellulose
# Ea_g = 125520 #units J/mol for eth
m = 1 #fuel exponent for cell
#m = 0.15 #fuel exponent for eth
n = 1 #fuel exponent for cell
#n = 1.6 #oxidizer exponent for eth
# Q = 29*10**6 #J/kg heat of combustion for eth
# Q = 14.8*10**6 #J/kg heat of combustion for tar (assumed as levoglucosan)
Q = 30*10**6 #J/kg heat of combustion for cell
# nuO2 = 3 #coeff for oxygen when eth is fuel (molar)
nuO2 = 3.57 #coeff for oxy when cell is fuel (mass)
Mf = 46.07 #kg/kmol
Mo2 = 32 #kg/kmol
Yo2inf = 0.23 #initial mass fraction of oxygen in air
Yp_inf = 0.77 #initial mass fraction of inert(N2) in air


#NON-VARYING DIMENSIONLESS PARAMETERS
#*************************************************************************
sigO = nuO2 #for when combustion reaction is on mass basis
# sigO = Mo2*nuO2/Mf #for when combustion reaction is on molar basis
Ahat = a_s/a_g
Rhat = rho_s/rho_inf
ep_p = 1

#NON-VARYING NUMERICAL SCHEME INPUTS
#*************************************************************************

Gs = 0 #growth rate of non-uniform mesh widths in solid phase
Gg = 0 #growth rate of non-uniform mesh widths in gas phase
A = 1 # guess for relaxation constant

#BEGIN FUNCTION LIBRARY
#*************************************************************************

#Function that produces non-uniform mesh vectors
@jit
def meshcreator(N,domainsize,ep_p,G):

        #Create array of growth coefficients for cell size
```

```
grower = (1 + G)**np.array(list(range(N-1)))

#Determine smallest cell width
epstepmin = domainsize/sum(grower)

#Create epsilon array
ep = np.array([10**-28,ep_p]) #avoids divide by zero
for i in range(N-1):
        ep = np.hstack([ep,ep_p+epstepmin*sum(grower[0:i+1])])

#Create array of cell center locations
centers = np.zeros_like(ep)
for i in range(1,N):
        centers[i] = (ep[i] + ep[i+1])/2

#Trim epsilon and center arrays to proper length
ep = ep[0:-1]
centers = centers[0:-1]

#Create array of cell center locations relative to edge of particle
ep_plot = centers - ep_p
ep_plot[0] = 0

#Create array of cell widths
epstep = np.hstack([ep_p,epstepmin*grower])

#Create arrays of cell off-widths. L is (j-) and R is (j+)
epstepL = np.zeros_like(epstep)
epstepR = np.zeros_like(epstep)

epstepL[1] = centers[1]-ep[1]
epstepR[0] = centers[1]-ep[1]
epstepR[-1] = ep[-1]+epstep[-1]-centers[-1]

for i in range(2,N):
        epstepL[i] = epstepR[i-1] = centers[i] - centers[i-1]

#Determine ratios used in spatial averaging in fin. vol. formulation
fL = (centers[2]-ep[2])/epstepL[2]
fR = (ep[3] - centers[2])/epstepR[2]

return [ep,ep_plot,epstep,epstepL,epstepR,fL,fR]
```

```
#Function for lhat_s (dim'less solid thermal diffusivity) at cell faces
@jit
def lhat_s(theta_s,Ti,fLs,Ns):

        #preallocate
        lhatsL = np.ones_like(theta_s)*10**-28 #avoids divide by zero
        lhatsR = np.zeros_like(theta_s)

        #Calculate conductivity based on mass weighting in cell and with
        #a radiation correction term
        l =  l_s + 13.5*5.67\
        *10**-8*d_pore*Ti**3*theta_s**3/w_p

        lhatsL[1] = (1/l_s)*l[1]
        lhatsR[0] = (1/l_s)*l[1]
        lhatsR[-1] = (1/l_s)*l[-1]

        for i in range(2,Ns):
                lhatsL[i] = lhatsR[i-1] = (1/l_s)*((1-fLs)/l[i-1]\
                 + fLs/l[i])**(-1)

        return [lhatsL,lhatsR]

#Function for lhat_g (dim'less gas thermal diffusivity) at cell faces
@jit
def lhat_g(theta_g,Ti,fLg,Ng):

        #preallocate
        lhatgL = np.ones_like(theta_g)*10**-28 #avoids divide by zero
        lhatgR = np.zeros_like(theta_g)

        #define vectors of constants according to NASA polynomials. Values
        #change above and below 1000K
        v1 = 0.94*np.ones_like(theta_g)
        v1[theta_g >= 1000/Ti] = 0.65
        v2 = 0.12*10**3*np.ones_like(theta_g)
        v2[theta_g >= 1000/Ti] = -0.15*10**3
        v3 = -0.12*10**5*np.ones_like(theta_g)
        v3[theta_g >= 1000/Ti] = -0.14*10**5
        v4 = -0.12*np.ones_like(theta_g)
        v4[theta_g >= 1000/Ti] = 2.2
```

```
        #Calculate values at cell centers according to NASA prop. data. Fit
        #is of the form ln(l) = v1*ln(T) + v2/T + v3/T^2 + D
        l = 10**-4*np.exp(v1*np.log(Ti*theta_g) + v2/(Ti*theta_g) +\
         v3/((Ti*theta_g)**2) + v4)

        lhatgL[1] = (1/l_inf)*l[1]
        lhatgR[0] = (1/l_inf)*l[1]
        lhatgR[-1] = (1/l_inf)*l[-1]

        for i in range(2,Ng):
                lhatgL[i] = lhatgR[i-1] = (1/l_inf)*((1-fLg)/l[i-1] +\
                 fLg/l[i])**(-1)

        return [lhatgL,lhatgR]

#Function for gamG (dim'less gas heat capacity)
@jit
def gamG_(theta_g,Ti):

        #preallocate
        gamG = np.ones_like(theta_g)*10**-28 #avoids divide by zero

        #define vectors of constants according to NASA polynomials. Values
        #change above and below 1000K
        v1 = 3.53*np.ones_like(theta_g)
        v1[theta_g >= 1000/Ti] = 2.95
        v2 = -1.24*10**(-4)*np.ones_like(theta_g)
        v2[theta_g >= 1000/Ti] = 1.40*10**-3
        v3 = -5.03*10**(-7)*np.ones_like(theta_g)
        v3[theta_g >= 1000/Ti] = -4.93*10**(-7)
        v4 = 2.44*10**(-9)*np.ones_like(theta_g)
        v4[theta_g >= 1000/Ti] = 7.86*10**(-11)
        v5 = -1.41*10**(-12)*np.ones_like(theta_g)
        v5[theta_g >= 1000/Ti] = -4.61*10**(-15)

        #Calculate values at cell centers according to NASA prop. data. Fit
        #is of the form c = Ru*(v1 + v2*T + v3*T^2 + v4*T^3 + v5*T^3)
        c = (10**3/28)*Ru*(v1 + v2*(Ti*theta_g) + v3*(Ti*theta_g)**2 +\
         v4*(Ti*theta_g)**3 + v5*(Ti*theta_g)**4)

        gamG = (1/c_inf)*c
```

```
        return gamG

#Function that finds dimless particle specific heat
@jit
def gamP_(theta_s,Ti,Tl):

        if theta_s*Ti < Ts:

                gamS = 1

        elif theta_s*Ti >= Ts and theta_s*Ti < Tl:

                gamS = deltaHmelt/(c_p*(Tl-Ts))

        else:

                gamS = c_pl/c_p

        return gamS

#Function that converts theta_g into hHat_g
@jit
def hHat_g_(theta_g,Ti):

        T_g = theta_g*Ti
        h_g = np.interp(T_g,Tref,href)

        return h_g/(c_inf*Ti)

#Function that finds effective specific heat,
#as in hHat_g = gamGstar*theta_g
@jit
def gamGstar(theta_g,Ti):

        #note that in this case theta_g is a single temperature, not an array

        T = theta_g*Ti
        if T >= 1000:
                v1 = 2.95
                v2 = 1.40*10**-3
                v3 = -4.93*10**(-7)
```

```
                v4 = 7.86*10**(-11)
                v5 = -4.61*10**(-15)
        else:
                v1 = 3.53
                v2 = -1.24*10**(-4)
                v3 = -5.03*10**(-7)
                v4 = 2.44*10**(-9)
                v5 = -1.41*10**(-12)

        cGstar = (10**3/28)*Ru*(v1 + 0.5*v2*T + v3*T**2/3 + 0.25*v4*T**3 +\
        0.2*v5*T**4)

        return cGstar/c_inf

#Function that sets up convection-diffusion equations for TDM solve
@jit
def CDsetup(phi_o,C,Fo,Dr,Dl,u,S1,S2,taustep,fL,fR,epstepL,epstepR,\
        epstep,ep,N):

        #preallocate
        a = np.zeros_like(phi_o)
        b = np.zeros_like(phi_o)
        c = np.zeros_like(phi_o)
        d = np.zeros_like(phi_o)

        for j in range(1,N-1):#Doing the middle first

                        ajR = max(-u[j], Fo*(Dr[j]/epstepR[j]\
                         + 2*Dr[j]*epstep[j]/ep[j]*epstepR[j]) - fR*u[j], 0)

                        ajL = max(u[j], Fo*Dl[j]/epstepL[j]\
                         + fL*u[j], 0)

                        aj = -(ajL + ajR)

                        a[j] = -(taustep/C[j])*ajL/epstep[j]

                        b[j] =  1 - (taustep/C[j])*aj/epstep[j] - (taustep/C[j])*S1[j]

                        c[j] = -(taustep/C[j])*ajR/epstep[j]

                        d[j] = phi_o[j] + (taustep/C[j])*S2[j]
```

```
        return [a,b,c,d]

#Function that sets up solid phase mass conservation for TDM solve
@jit
def uHat_sSetup(theta_sn,S1,epstep_s,ep_s,Ns):

        #Preallocate some variables to be used for calculating uHat_s
        aP = np.zeros_like(theta_sn)
        bP = np.zeros_like(theta_sn)
        cP = np.zeros_like(theta_sn)
        dP = np.zeros_like(theta_sn)

        #Populate the middle of arrays. cP is all zeros for calc.
        #of uHat_s
        for j in range(1,Ns-1):#Doing the middle first
                aP[j] = -1/epstep_s[j]
                bP[j] = 2/ep_s[j] + 1/epstep_s[j]
                dP[j] = S1[j]

        return [aP,bP,cP,dP]

#Tri Diagonal Matrix Algorithm(a.k.a Thomas algorithm) solver for the lin.
#systems of equations produced by the iterative solve loop
# @jit
def TDMAsolver(a, b, c, d):

    nf = len(a)     # number of equations
    ac, bc, cc, dc = map(np.array, (a, b, c, d))     # copy the array
    for it in range(1, nf):
        mc = ac[it]/bc[it-1]
        bc[it] = bc[it] - mc*cc[it-1]
        dc[it] = dc[it] - mc*dc[it-1]

    xc = ac
    xc[-1] = dc[-1]/bc[-1]

    for il in range(nf-2, -1, -1):
        xc[il] = (dc[il]-cc[il]*xc[il+1])/bc[il]

    del bc, cc, dc  # delete variables from memory
```

```
    return xc

def OneDHFCI(r_p,Ti,ds_s,ds_g,runmode,rho_p=7860,w=0.9,Tl=925):

        #Define input-dependent dimensionless parameters
        #*********************************************************************

        theta_inf = Tinf/Ti #Initial and far boundary condition value for theta
        Ar_s = Ea_s/(Ru*Ti)
        Ar_g = Ea_g/(Ru*Ti)
        Py = r_p**2*Zs*np.exp(-Ar_s)/a_s
        Hhat_v = Hvol/(c_s*Ti)
        Qhat = Q/(c_inf*Ti)
        # Da = (r_p**2/a_g)*Zg*np.exp(-Ar_g)*rho_inf**(m+n-1)*Mf**(1-m)\
        #*(sigO/Mo2)**n #for when combustion reaction is on molar basis
        Da = (r_p**2/a_g)*Zg*np.exp(-Ar_g)*sigO*rho_inf #for when combustion
        #reaction is on mass basis
        b_s = rho_s*c_s/(rho_p*c_p)
        b_g = rho_inf*c_inf/(rho_p*c_p)
        psi = r_p*w*5.67*10**-8*Ti**3/(rho_p*c_p*a_g)

        #Define input-dependent numerical scheme inputs
        #*********************************************************************
        # Ns = 400 #number of spatial grids in solid phase
        Ns = int(ds_s/0.00001) #number of spatial grids in solid phase
        domainsize_s = ds_s/r_p #fuelbed domain in units of dim'less distance
        #(ep_s)
        # Ng = 400 #number of spatial grids in gas phase
        Ng = int(ds_g/0.0001)
        domainsize_g = ds_g/r_p #gas domain in units of dim'less distance (ep_g)
        timetot = 1*a_g/(r_p**2) #in units of dim'less time (tau)
        taustep_guess = 10**-4*a_g/(r_p**2) #guess of time step in units of
        #dim'less time (tau)
        sampleRes = 10**-4*a_g/(r_p**2) #temporal sample resolution in units of
        #dim'less time
        Nt = timetot/sampleRes #number of temporal samples

        #Iterating function that solves nonlinear system at each time step. Finite
        #volume formulation that is fully implicit in time with opt. numerical
        #relaxation
        def solver(cs,ep_plot_g,ep_plot_s):
                # t1 = time.perf_counter()
```

```
#Use values at current step (cs) as solution at i (old)
uHat_so = cs[0]
theta_so = cs[1]
uHat_go = cs[2]
Yhat_Fo = cs[3]
Yhat_Oo = cs[4]
theta_go = cs[5]
# t2 = time.perf_counter()
#Preallocate for the guesses, multiply by one to avoid double assgn.
uHat_sg = 1.0*uHat_so
theta_sg = 1.0*theta_so
uHat_gg = 1.0*uHat_go
Yhat_Fg = 1.0*Yhat_Fo
Yhat_Og = 1.0*Yhat_Oo
theta_gg = 1.0*theta_go
# t3 = time.perf_counter()
# t4 = time.perf_counter()
#Setup while loop, timestep, iteration counter and dom-error switch
Error = 10**12
OldError = 10**12
numit = 0
taustep = taustep_guess
killI = 'off'
MaxTempDiffI = 0
# t5 = time.perf_counter()
# t6 = time.perf_counter()
while Error > 10**-4:

        #Calculate diffusivities at volume faces and specific heat base
        #on guess temps for every 5th iteration
        # t7 = time.perf_counter()
        if np.mod(numit,5) == 0:

                [lhat_sL,lhat_sR] = lhat_s(theta_sg,Ti,fLs,Ns)
                [lhat_gL,lhat_gR] = lhat_g(theta_gg,Ti,fLg,Ng)
                gamG = gamG_(theta_gg,Ti)
        # t8 = time.perf_counter()
        #First use value uHat_sg and theta_gg to solve
        #for solid phase temperatures.
        # t9 = time.perf_counter()
        #Fill arrays
        [ats,bts,cts,dts] = CDsetup(theta_so,np.ones_like(theta_so),\
```

```
                Ahat,lhat_sR,lhat_sL,uHat_sg,\
                -Hhat_v*Ahat*Py*np.exp(Ar_s*(1-1/theta_sg))/thet
                 np.zeros_like(theta_so),taustep,\
                fLs,fRs,epstepLs,epstepRs,epstep_s,ep_s,Ns)
# t10 = time.perf_counter()
#Apply BCs
bts[0] = 1 + (taustep/gamP_(theta_sg[0],Ti,Tl))*P*Ahat*b_s*\
        lhat_sR[0]/epstepRs[0] +\
        (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*psi*theta_sg[0]
         + (taustep/gamP_(theta_sg[0],Ti,Tl))*P*b_g*Rhat*Ahat*Py
         gamGstar(theta_sg[0],Ti)*\
         np.sum(np.exp(Ar_s*(1-1/theta_sg[1:]))\
        *ep_s[1:]**2*epstep_s[1:])
cts[0] = -(taustep/gamP_(theta_sg[0],Ti,Tl))*P*Ahat*b_s*\
                        lhat_sR[0]/epstepRs[0]
dts[0] = theta_so[0] + (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)\
                        *b_g*lhat_gR[0]*(theta_gg[1]\
        - theta_gg[0])/epstepRg[0] +\
                        (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)
                        psi*theta_inf**4\
        +(taustep/gamP_(theta_sg[0],Ti,Tl))*P*b_g*Rhat*Ahat*Py*\
        np.sum(hHat_g_(theta_sg[1:],Ti)*\
                np.exp(Ar_s*(1-1/theta_sg[1:]))*\
                ep_s[1:]**2*epstep_s[1:])

# bts[-1] = 1
# dts[-1] = theta_inf
bts[-1] = 1
ats[-1] = -1
# t11 = time.perf_counter()
#Apply numerical relaxation
bts = bts/A
dts = dts + (1-A)*bts*theta_sg
# t12 = time.perf_counter()
#Solve the system
theta_sn = TDMAsolver(ats,bts,cts,dts)
# t13 = time.perf_counter()
#Then use theta_sn, Yhat_Fg, Yhat_Og and uHat_gg to solve for g
#phase temperatures
# t14 = time.perf_counter()
#Fill arrays
[atg,btg,ctg,dtg] = CDsetup(theta_go,gamG,\
```

```
        1,lhat_gR,lhat_gL,uHat_gg*gamG,np.zeros_like(theta_go),\
        Qhat*Da*Yhat_Fg**m*Yhat_Og**n*np.exp(Ar_g*(1-1/theta_gg)
        taustep,fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t15 = time.perf_counter()
# #Fill arrays
# [atg,btg,ctg,dtg] = CDsetup(theta_go,gamG,\
#         1,lhat_gR,lhat_gL,uHat_gg*gamG,np.zeros_like(theta_g
#         np.zeros_like(theta_go),\
#         taustep,fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)

#Apply BCs
btg[0] = 1 + (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*b_g*\
               lhat_gR[0]/epstepRg[0]
ctg[0] = -(taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*b_g*\
               lhat_gR[0]/epstepRg[0]
dtg[0] = theta_go[0] + (theta_sn[0] - theta_so[0])\
                       - (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-
                       b_g*lhat_gR[0]*\
                       (theta_gg[1] - theta_gg[0])/epstepRg[0]

# btg[-1] = 1
# dtg[-1] = theta_inf
btg[-1] = 1
atg[-1] = -1
# t16 = time.perf_counter()

#Apply numerical relaxation
btg = btg/A
dtg = dtg + (1-A)*btg*theta_gg
# t17 = time.perf_counter()

#Solve the system
theta_gn = TDMAsolver(atg,btg,ctg,dtg)
# t18 = time.perf_counter()

#Update the solid phase p-temp to the gas phase p-temp
theta_sn[0] = theta_gn[0]

#Next, use guess of theta_sn to get new value for uHat_sn

[aP,bP,cP,dP] = uHat_sSetup(theta_sn,\
        - Ahat*Py*np.exp(Ar_s*(1-1/theta_sn)),epstep_s,ep_s,Ns)
```

```
#Apply boundary conditions. Ends of dP were specified during
#initialization
aP[-1] = -1
bP[-1] = 1
bP[0] = 1
# t19 = time.perf_counter()

# #Apply numerical relaxation
# bP = bP/A
# dP = dP + (1-A)*bP*uHat_sg

#Solve the linear system W1*uHat_s = b1 and apply num. relax.
uHat_sn = TDMAsolver(aP,bP,cP,dP)
# t20 = time.perf_counter()
#Determine gas phase velocity

uHat_gn = P/(1-P)*(Rhat/ep_g**2)*Ahat*Py*\
np.sum(np.exp(Ar_s*(1-1/theta_sn[1:]))*ep_s[1:]**2*epstep_s[1:])

uHat_gn[0] = uHat_gn[1]
# t21 = time.perf_counter()
#Solve fuel species conservation

#Fill arrays
[af,bf,cf,df] = CDsetup(Yhat_Fo,np.ones_like(Yhat_Fo),1,\
        np.ones_like(Yhat_Fo),np.ones_like(Yhat_Fo),\
        uHat_gn,-Da*Yhat_Fg**m*Yhat_Og**n*\
        np.exp(Ar_g*(1-1/theta_gn))/\
        np.maximum(Yhat_Fg,10**-8*np.ones_like(Yhat_Fg)),\
        np.zeros_like(Yhat_Fo),taustep,\
        fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# [af,bf,cf,df] = CDsetup(Yhat_Fo,np.ones_like(Yhat_Fo),1,\
#         np.ones_like(Yhat_Fo),np.ones_like(Yhat_Fo),\
#         uHat_gn,np.zeros_like(Yhat_Fo),\
#         np.zeros_like(Yhat_Fo),taustep,\
#         fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t22 = time.perf_counter()
#Apply BCs
bf[0] = (1 + epstepRg[0]*uHat_gn[1])
cf[0] = -1
df[0] = epstepRg[0]*uHat_gn[1]
```

```
# bf[-1] = 1
# df[-1] = 0
bf[-1] = 1
af[-1] = -1

# #Apply BCs
# bf[0] = 1
# cf[0] = 0
# df[0] = 1
# bf[-1] = 1
# df[-1] = 0

#Apply numerical relaxation
bf = bf/A
df = df + (1-A)*bf*Yhat_Fg

#Solve the system
Yhat_Fn = TDMAsolver(af,bf,cf,df)

#Solve oxy species conservation
# t23 = time.perf_counter()
#Fill array
[ao2,bo2,co2,do2] = CDsetup(Yhat_Oo,np.ones_like(Yhat_Oo),1,\
        np.ones_like(Yhat_Oo),np.ones_like(Yhat_Oo),uHat_gn,\
        -Da*Yhat_Fg**m*Yhat_Og**n*np.exp(Ar_g*(1-1/theta_gn))/\
        np.maximum(Yhat_Og,10**-8*np.ones_like(Yhat_Og)),\
        np.zeros_like(Yhat_Oo),taustep,fLg,\
        fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# [ao2,bo2,co2,do2] = CDsetup(Yhat_Oo,np.ones_like(Yhat_Oo),1,\
#         np.ones_like(Yhat_Oo),np.ones_like(Yhat_Oo),uHat_gn,\
#         np.zeros_like(Yhat_Oo),\
#         np.zeros_like(Yhat_Oo),taustep,fLg,\
#         fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t24 = time.perf_counter()
#Apply BCs
bo2[0] = 1 + epstepRg[0]*uHat_gn[1]
co2[0] = -1
do2[0] = 0

# bo2[-1] = 1
# do2[-1] = Yo2inf/sigO
```

```
bo2[-1] = 1
ao2[-1] = -1

#Apply numerical relaxation
bo2 = bo2/A
do2 = do2 + (1-A)*bo2*Yhat_Og

#Solve the system
Yhat_On = TDMAsolver(ao2,bo2,co2,do2)

#Set values at particle equal to value next to particle for
#plotting
Yhat_Fn[0] = Yhat_Fn[1]
Yhat_On[0] = Yhat_On[1]
# t25 = time.perf_counter()
#Calculate error and determine if time step needs to be dec.
YhatF_Error = np.max(np.abs(Yhat_Fn-Yhat_Fg)/\
        np.maximum(Yhat_Fg,10**-7*np.ones_like(Yhat_Fg)))
YhatO_Error = np.max(np.abs(Yhat_On-Yhat_Og)/\
        np.maximum(Yhat_Og,10**-7*np.ones_like(Yhat_Og)))

theta_sError = np.max(np.abs(theta_sn-theta_sg))
theta_gError = np.max(np.abs(theta_gn-theta_gg))

NewError = np.max([YhatF_Error,YhatO_Error,theta_sError,\
        theta_gError])
# t26 = time.perf_counter()
if NewError > Error or numit > 50:

        taustep = 0.5*taustep
        numit = 0
        uHat_sg = 1.0*uHat_so
        theta_sg = 1.0*theta_so
        uHat_gg = 1.0*uHat_go
        Yhat_Fg = 1.0*Yhat_Fo
        Yhat_Og = 1.0*Yhat_Oo
        theta_gg = 1.0*theta_go
        # if NewError > Error:
        #         print('Diverging')
        # else:
        #         print('max interations exceeded')
```

```
                # if NewError == YhatF_Error:
                #         print('Error in fuel species')
                # elif NewError ==YhatO_Error:
                #         print('Error in oxygen species')
                # elif NewError == theta_sError:
                #         print('Error in solid temp')
                # else:
                #         print('Error in gas temp')

                Error = 10**12
                OldError = 10**12
        else:

                OldError = Error
                Error = NewError
                numit = numit + 1
                uHat_sg = 1.0*uHat_sn
                theta_sg = 1.0*theta_sn
                uHat_gg = 1.0*uHat_gn
                Yhat_Fg = 1.0*Yhat_Fn
                Yhat_Og = 1.0*Yhat_On
                theta_gg = 1.0*theta_gn
                # print('Good Run, error' + str(Error))
        # t27 = time.perf_counter()

# t28 = time.perf_counter()
newvalues = [uHat_sn,theta_sn,uHat_gn,Yhat_Fn,Yhat_On,theta_gn]
# t29 = time.perf_counter()
BoundaryDeltaG = np.max([np.abs(theta_gn[-2]-theta_inf),\
        np.abs(Yhat_Fn[-2]),\
        np.abs(Yhat_On[-2] - Yo2inf/sigO)])
BoundaryDeltaS = np.abs(theta_sn[-2]-theta_inf)

if BoundaryDeltaG > 10**-4:

        print('Edge of numerical gas-phase domain has been reached.'+\
                ' Need to rerun case with larger domain.')
        killI = 'onG'
if BoundaryDeltaS > 10**-4:
        print('Edge of numerical solid-phase domain has been reached.'+\
                ' Need to rerun case with larger domain.')
        killI = 'onS'
```

```
        MaxTempDiffI = np.max(theta_gn)-np.max(theta_go)
        # t30 = time.perf_counter()
        return [newvalues,taustep,killI,MaxTempDiffI]

#Initialize time step, mesh and dim'less variables
#*********************************************************************
[ep_s,ep_plot_s,epstep_s,epstepLs,epstepRs,fLs,fRs] =\
meshcreator(Ns,domainsize_s,ep_p,Gs)
[ep_g,ep_plot_g,epstep_g,epstepLg,epstepRg,fLg,fRg] =\
meshcreator(Ng,domainsize_g,ep_p,Gg)
tauarray = np.zeros((Nt+2,1),float) #establish time vector
tau = 0 #initialize tau counter
i = 1 #counter for time stepping
kill = 'off' #indicator for when edge of domain is reached
MaxTempDiff = 0 #indicator of ignition
igprintswitch = 'on' #forces to print ignition msg only once
tau_ig = 0 #will hold ignition time
uHat_s = np.zeros((Nt+2,Ns),float) #dim'less solid phase velocity
theta_s = np.ones((Nt+2,Ns),float)*theta_inf #dim'less sol phase temp
theta_s[0,0] = 1 #Initial condition at boundary for theta_s
uHat_g = np.zeros((Nt+2,Ng),float) #dim'less gas phase velocity for diagn.
# Yhat_F = 10**-28*np.ones((Nt+2,Ng),float)#init. mass fraction of fuel,
#avoid divide by zero
Yhat_F = np.zeros((Nt+2,Ng),float)#init. mass fraction of fuel, avoid

Yhat_O = np.ones((Nt+2,Ng),float)*Yo2inf/sigO #Initialize mass frac of O2
#Yp = np.ones((Nt,Ng),float)*Yp_inf #Initialize mass frac of inert spec.
theta_g = np.ones((Nt+2,Ng),float)*theta_inf #dim'less gas phase temp
theta_g[0,0] = 1 #Initial condition at boundary for theta_g

#Loop that marches through time, solving non-lin system (see above) at
#each time step
#*********************************************************************

cs =[uHat_s[0,],theta_s[0,],uHat_g[0,],Yhat_F[0,],Yhat_O[0,],theta_g[0,]]

if runmode == 'bisect':
        while tau <= timetot and MaxTempDiff <= 0 and kill == 'off':
        # while tau <= timetot and np.max(cs[5]) <= 1.05 and kill == 'off':
        # while tau <= timetot and kill == 'off':
```

```
                #Solve non-lin system for next time step using current step and
                #pack variable arrays at current step for passing (globally) to
                #solver
                [cs,taustep,kill,MaxTempDiff] = solver(cs,ep_plot_g,ep_plot_s)

                #keep track of current time
                tau = tau + taustep
                # print(tau)

                if tau >= i*sampleRes:

                        i += 1 #proceed to next diff res step
                        # if np.mod(i,100)==0:
                        #         print(tau)

                if MaxTempDiff > 0 and igprintswitch == 'on':
                        tau_ig = tau
                        print('Ignition occurred.')
                        igprintswitch = 'off'


        if tau >= timetot:
                print('max time reached.')

        thing2return =  [tau_ig,kill]

else:
        # while tau <= timetot and MaxTempDiff <= 0 and kill == 'off':
        # while tau <= timetot and np.max(cs[5]) <= 1.05 and kill == 'off':
        while tau <= timetot and kill == 'off':

                #Solve non-lin system for next time step using current step and
                #pack variable arrays at current step for passing (globally) to
                #solver
                [cs,taustep,kill,MaxTempDiff] = solver(cs,ep_plot_g,ep_plot_s)

                #keep track of current time
                tau = tau + taustep
                # print(tau)

                if tau >= i*sampleRes:
```

```
                        uHat_s[i,] = cs[0]
                        theta_s[i,] = cs[1]
                        uHat_g[i,] = cs[2]
                        Yhat_F[i,] = cs[3]
                        Yhat_O[i,] = cs[4]
                        theta_g[i,] = cs[5]

                        tauarray[i] = tau
                        # print('--------' + str(tau))
                        i += 1 #proceed to next diff res step
                        if np.mod(i,100)==0:
                                print(tau)
                        # print(i)

                if i == Nt-2:
                        add = 100
                        #add = round((1- tau[i]/timetot)*i/(tau[i]/timetot))
                        uHat_s = np.vstack([uHat_s,np.ones((add,Ns),float)])
                        theta_s = np.vstack([theta_s,theta_inf*\
                                np.ones((add,Ns),float)])
                        uHat_g = np.vstack([uHat_g,np.zeros((add,Ng),float)])
                        Yhat_F = np.vstack([Yhat_F,10**-28*\
                                np.ones((add,Ng),float)])
                        Yhat_O = np.vstack([Yhat_O,(Yo2inf/sigO)*\
                                np.ones((add,Ng),float)])
                        theta_g = np.vstack([theta_g,theta_inf*\
                                np.ones((add,Ng),float)])
                        tauarray = np.vstack([tauarray,np.zeros((add,1),float)])
                        Nt += add

                if MaxTempDiff > 0 and igprintswitch == 'on':
                        tau_ig = tau
                        print('Ignition occurred.')
                        igprintswitch = 'off'


        if tau >= timetot:
                print('max time reached.')

        uHat_s = uHat_s[0:i,]
        theta_s = theta_s[0:i,]
        uHat_g = uHat_g[0:i,]
```

```
                Yhat_F = Yhat_F[0:i,]
                Yhat_O = Yhat_O[0:i,]
                theta_g = theta_g[0:i,]
                tauarray = tauarray[0:i,]

                thing2return = [ep_s,ep_plot_s,ep_g,ep_plot_g,tauarray,uHat_s,\
                        theta_s,uHat_g,Yhat_F,Yhat_O,theta_g,tau_ig,kill]
        return thing2return

#*************************************************************************
#END FUNCTION LIBRARY

#SETUP SWEEPING PROCESSES
#*************************************************************************

#First create array for diameters of interest
d_p = np.array([0.010,0.009,0.008,0.007,0.006,0.005,0.004,0.003,0.0025,\
        0.002,0.0015,0.001,0.00075,0.000625,0.0005])

#Convert to radius
r_p = d_p/2

#Then create array for variable to be swept, in this case volumetric
#heat capacity.
Sweeper = np.array([940])

#Give a starting guess for temperatures where the bounding algorithm
#starts looking, 'new' and 'old' subscript are used in bounding
Tnew = np.array([920.875,921.65625,922.4375,924.78125,927.34375,931.90625,\
                                        939.25,955.28125,969.875,993.8125,1037.46875,113
                                          1223.4375,1290.90625,1388.84375])
Tnew = np.tile(Tnew,(len(Sweeper),1))
Told = 1.0*Tnew

#Specify a temperature interval over which to bound for the bounding
#algorithm
delTbound = 50

#Preallocate arrays for the ignition limit temperatures
Titop = np.zeros((len(Sweeper),len(d_p)),float)
Tibot = np.zeros((len(Sweeper),len(d_p)),float)
```

```
#CARRY OUT SWEEP AND SAVE VARIABLES
#*************************************************************************

#Sweep through variable of interest and diameters and store ignition
#temperatures
for j in range(0,len(d_p)):

        print('working on d = ' + str(d_p[j]*10**3) + 'mm.')

        for k in range(0,len(Sweeper)):

                ds_s = 0.004
                ds_g = 0.04

                print('working on Sweeper = ' + str(Sweeper[k]))

                while Titop[k,j] == 0 and Tibot[k,j] == 0:

                        [tau_ig,kill] = OneDHFCI(r_p[j],Tnew[k,j],\
                        ds_s,ds_g,'bisect',Tl = Sweeper[k])

                        if kill == 'onS':

                                ds_s = ds_s + 0.001

                        elif kill == 'onG':

                                ds_g = ds_g + 0.010

                        elif tau_ig > 0:

                                print('Ignited at ' + str(Tnew[k,j]) +  'K.')

                                if Tnew[k,j]-Told[k,j] > 0:
                                        Titop[k,j] = Tnew[k,j]
                                        Tibot[k,j] = Told[k,j]
                                else:
                                        Told[k,j] = Tnew[k,j]
                                        Tnew[k,j] = Told[k,j] - delTbound
                        else:

                                print('Did not ignite at ' + str(Tnew[k,j]) +  'K.')
```

```
                                if Tnew[k,j]-Told[k,j] >= 0:
                                        Told[k,j] = Tnew[k,j]
                                        Tnew[k,j] = Told[k,j] + delTbound
                                else:
                                        Tibot[k,j] = Tnew[k,j]
                                        Titop[k,j] = Told[k,j]

                        while Titop[k,j] - Tibot[k,j] > 1:

                                [tau_ig,kill] = OneDHFCI(r_p[j],(Titop[k,j] + Tibot[k,j])/2,\
                                ds_s,ds_g,'bisect',Tl = Sweeper[k])

                                if kill == 'onS':

                                        ds_s = ds_s + 0.001

                                elif kill == 'onG':

                                        ds_g = ds_g + 0.010

                                elif tau_ig > 0:

                                        Titop[k,j] = (Titop[k,j] + Tibot[k,j])/2
                                        print('Ignited at ' + str(Titop[k,j]) +  'K.')

                                else:

                                        Tibot[k,j] = (Titop[k,j] + Tibot[k,j])/2
                                        print('Did not ignite at ' + str(Tibot[k,j]) +  'K.')

                        print('Completed Sweeper = '+ str(Sweeper[k]))

                print('Completed d = '+ str(d_p[j]*10**3) + 'mm')

        #Store variable of interest, diameters, and corresponding ignition temps

        # pickle.dump([Titop,Tibot],open('InitialTemps2.p','wb'))
        pickle.dump([d_p,Sweeper,Titop,Tibot],open('Limit_Melting.p','wb'))
```

Appendix B

Python Code Used to Find Model Ignition Limit for Different Particle Volumetric Heat Capacities

```
#import required modules
import numpy as np
from scipy.integrate import odeint
import pickle
import matplotlib.pyplot as plt
import time
from numba import jit

#NON-VARYING PARTICLE AND AMBIENT PARAMETERS
#*********************************************************************

c_p = 500 #use base units, this is the solid specific heat
c_pl = 500 #base units, liquid specific heat
deltaHmelt = 400*10**3 #heat of melting, J/kg
Ts = 925 #solidus temperature, in Kelvin
P = 0.5 #portion of surface area submerged in fuel bed
Tinf = 298 #in K

#NON-VARYING FUEL BED PARAMETERS
#*********************************************************************

#thermophysical
rho_s,l_s,c_s = 300,12.6*10**-2,2510 #base units, density from exp, but
#other two from Kung
```

```
#here l is short for lambda, symbol for thermal conductivity
a_s = l_s/(rho_s*c_s) #Thermal diff. in base units
w_p = 1 #pore emissivity
d_pore = 2*10**-5 #pore diameter, from DiBalsi pyrolysis paper
#thermochemical
Zs = 1*10**14 #acc. to Banyasz
# Zs = 6.5*10**14 #pre-exponential factor for tar reactionac. to chemsuschem
# Ea_s = 199*10**3 #activation energy for tar reaction acc. to CSC
# Ea_s = 195.8*10**3 #acc. to Banyasz
Ea_s = 200*10**3 #just making it a bit harder to py. Ea from tar with Z
#from formaldehyde
Hvol = 750*10**3 #heat of reaction for tar formation, endothermicity acc.
#for in eqns

#NON-VARYING GAS PHASE PARAMETERS
#*********************************************************************

#thermophysical
Ru = 8.314 #universal gas constant
rho_inf = 101.3*29/(Ru*Tinf)
l_inf = 10**-4*np.exp(0.94*np.log(Tinf) + 0.12*10**3/Tinf - 0.12*10**5/\
        (Tinf**2) - 0.12)
c_inf = (10**3/28)*Ru*(3.53 - 1.24*10**(-4)*Tinf - 5.03*10**(-7)*Tinf**2\
        + 2.44*10**(-9)*Tinf**3 - 1.41*10**(-12)*Tinf**4)
a_g = l_inf/(rho_inf*c_inf)

#create two arrays, one for temp and one for enthalpy, for switching back and
#forth
Tref = np.linspace(298,3000,1500)
v1 = 3.53*np.ones_like(Tref)
v1[Tref >= 1000] = 2.95
v2 = -1.24*10**(-4)*np.ones_like(Tref)
v2[Tref >= 1000] = 1.40*10**-3
v3 = -5.03*10**(-7)*np.ones_like(Tref)
v3[Tref >= 1000] = -4.93*10**(-7)
v4 = 2.44*10**(-9)*np.ones_like(Tref)
v4[Tref >= 1000] = 7.86*10**(-11)
v5 = -1.41*10**(-12)*np.ones_like(Tref)
v5[Tref >= 1000] = -4.61*10**(-15)
href = (10**3/28)*Ru*(v1*Tref + 0.5*v2*Tref**2 + v3*Tref**3/3 +\
        0.25*v4*Tref**4 + 0.2*v5*Tref**5)
```

```
#thermochemical
# Zg = 1.55*10**10 #units are [(kmol/m^3)^(1-a-b2)]/s for eth
Zg = 8*10**5 #units are m^3/(kg*s) for cellulose
Ea_g = 100000 #units are J/mol, just trying to match data
# Ea_g = 67000 #units are J/mol for cellulose
# Ea_g = 125520 #units J/mol for eth
m = 1 #fuel exponent for cell
#m = 0.15 #fuel exponent for eth
n = 1 #fuel exponent for cell
#n = 1.6 #oxidizer exponent for eth
# Q = 29*10**6 #J/kg heat of combustion for eth
# Q = 14.8*10**6 #J/kg heat of combustion for tar (assumed as levoglucosan)
Q = 30*10**6 #J/kg heat of combustion for cell
# nuO2 = 3 #coeff for oxygen when eth is fuel (molar)
nuO2 = 3.57 #coeff for oxy when cell is fuel (mass)
Mf = 46.07 #kg/kmol
Mo2 = 32 #kg/kmol
Yo2inf = 0.23 #initial mass fraction of oxygen in air
Yp_inf = 0.77 #initial mass fraction of inert(N2) in air


#NON-VARYING DIMENSIONLESS PARAMETERS
#*************************************************************************
sigO = nuO2 #for when combustion reaction is on mass basis
# sigO = Mo2*nuO2/Mf #for when combustion reaction is on molar basis
Ahat = a_s/a_g
Rhat = rho_s/rho_inf
ep_p = 1

#NON-VARYING NUMERICAL SCHEME INPUTS
#*************************************************************************

Gs = 0 #growth rate of non-uniform mesh widths in solid phase
Gg = 0 #growth rate of non-uniform mesh widths in gas phase
A = 1 # guess for relaxation constant

#BEGIN FUNCTION LIBRARY
#*************************************************************************

#Function that produces non-uniform mesh vectors
@jit
def meshcreator(N,domainsize,ep_p,G):
```

```
#Create array of growth coefficients for cell size
grower = (1 + G)**np.array(list(range(N-1)))

#Determine smallest cell width
epstepmin = domainsize/sum(grower)

#Create epsilon array
ep = np.array([10**-28,ep_p]) #avoids divide by zero
for i in range(N-1):
        ep = np.hstack([ep,ep_p+epstepmin*sum(grower[0:i+1])])

#Create array of cell center locations
centers = np.zeros_like(ep)
for i in range(1,N):
        centers[i] = (ep[i] + ep[i+1])/2

#Trim epsilon and center arrays to proper length
ep = ep[0:-1]
centers = centers[0:-1]

#Create array of cell center locations relative to edge of particle
ep_plot = centers - ep_p
ep_plot[0] = 0

#Create array of cell widths
epstep = np.hstack([ep_p,epstepmin*grower])

#Create arrays of cell off-widths. L is (j-) and R is (j+)
epstepL = np.zeros_like(epstep)
epstepR = np.zeros_like(epstep)

epstepL[1] = centers[1]-ep[1]
epstepR[0] = centers[1]-ep[1]
epstepR[-1] = ep[-1]+epstep[-1]-centers[-1]

for i in range(2,N):
        epstepL[i] = epstepR[i-1] = centers[i] - centers[i-1]

#Determine ratios used in spatial averaging in fin. vol. formulation
fL = (centers[2]-ep[2])/epstepL[2]
fR = (ep[3] - centers[2])/epstepR[2]
```

```
        return [ep,ep_plot,epstep,epstepL,epstepR,fL,fR]

#Function for lhat_s (dim'less solid thermal diffusivity) at cell faces
@jit
def lhat_s(theta_s,Ti,fLs,Ns):

        #preallocate
        lhatsL = np.ones_like(theta_s)*10**-28 #avoids divide by zero
        lhatsR = np.zeros_like(theta_s)

        #Calculate conductivity based on mass weighting in cell and with
        #a radiation correction term
        l =  l_s + 13.5*5.67\
        *10**-8*d_pore*Ti**3*theta_s**3/w_p

        lhatsL[1] = (1/l_s)*l[1]
        lhatsR[0] = (1/l_s)*l[1]
        lhatsR[-1] = (1/l_s)*l[-1]

        for i in range(2,Ns):
                lhatsL[i] = lhatsR[i-1] = (1/l_s)*((1-fLs)/l[i-1]\
                 + fLs/l[i])**(-1)

        return [lhatsL,lhatsR]

#Function for lhat_g (dim'less gas thermal diffusivity) at cell faces
@jit
def lhat_g(theta_g,Ti,fLg,Ng):

        #preallocate
        lhatgL = np.ones_like(theta_g)*10**-28 #avoids divide by zero
        lhatgR = np.zeros_like(theta_g)

        #define vectors of constants according to NASA polynomials. Values
        #change above and below 1000K
        v1 = 0.94*np.ones_like(theta_g)
        v1[theta_g >= 1000/Ti] = 0.65
        v2 = 0.12*10**3*np.ones_like(theta_g)
        v2[theta_g >= 1000/Ti] = -0.15*10**3
        v3 = -0.12*10**5*np.ones_like(theta_g)
        v3[theta_g >= 1000/Ti] = -0.14*10**5
```

```
        v4 = -0.12*np.ones_like(theta_g)
        v4[theta_g >= 1000/Ti] = 2.2

        #Calculate values at cell centers according to NASA prop. data. Fit
        #is of the form ln(l) = v1*ln(T) + v2/T + v3/T^2 + D
        l = 10**-4*np.exp(v1*np.log(Ti*theta_g) + v2/(Ti*theta_g) +\
         v3/((Ti*theta_g)**2) + v4)

        lhatgL[1] = (1/l_inf)*l[1]
        lhatgR[0] = (1/l_inf)*l[1]
        lhatgR[-1] = (1/l_inf)*l[-1]

        for i in range(2,Ng):
                lhatgL[i] = lhatgR[i-1] = (1/l_inf)*((1-fLg)/l[i-1] +\
                 fLg/l[i])**(-1)

        return [lhatgL,lhatgR]

#Function for gamG (dim'less gas heat capacity)
@jit
def gamG_(theta_g,Ti):

        #preallocate
        gamG = np.ones_like(theta_g)*10**-28 #avoids divide by zero

        #define vectors of constants according to NASA polynomials. Values
        #change above and below 1000K
        v1 = 3.53*np.ones_like(theta_g)
        v1[theta_g >= 1000/Ti] = 2.95
        v2 = -1.24*10**(-4)*np.ones_like(theta_g)
        v2[theta_g >= 1000/Ti] = 1.40*10**-3
        v3 = -5.03*10**(-7)*np.ones_like(theta_g)
        v3[theta_g >= 1000/Ti] = -4.93*10**(-7)
        v4 = 2.44*10**(-9)*np.ones_like(theta_g)
        v4[theta_g >= 1000/Ti] = 7.86*10**(-11)
        v5 = -1.41*10**(-12)*np.ones_like(theta_g)
        v5[theta_g >= 1000/Ti] = -4.61*10**(-15)

        #Calculate values at cell centers according to NASA prop. data. Fit
        #is of the form c = Ru*(v1 + v2*T + v3*T^2 + v4*T^3 + v5*T^3)
        c = (10**3/28)*Ru*(v1 + v2*(Ti*theta_g) + v3*(Ti*theta_g)**2 +\
         v4*(Ti*theta_g)**3 + v5*(Ti*theta_g)**4)
```

```
        gamG = (1/c_inf)*c

        return gamG

#Function that finds dimless particle specific heat
@jit
def gamP_(theta_s,Ti,Tl):

        if theta_s*Ti < Ts:

                gamS = 1

        elif theta_s*Ti >= Ts and theta_s*Ti < Tl:

                gamS = deltaHmelt/(c_p*(Tl-Ts))

        else:

                gamS = c_pl/c_p

        return gamS

#Function that converts theta_g into hHat_g
@jit
def hHat_g_(theta_g,Ti):

        T_g = theta_g*Ti
        h_g = np.interp(T_g,Tref,href)

        return h_g/(c_inf*Ti)

#Function that finds effective specific heat,
#as in hHat_g = gamGstar*theta_g
@jit
def gamGstar(theta_g,Ti):

        #note that in this case theta_g is a single temperature, not an array

        T = theta_g*Ti
        if T >= 1000:
                v1 = 2.95
```

```
                v2 = 1.40*10**-3
                v3 = -4.93*10**(-7)
                v4 = 7.86*10**(-11)
                v5 = -4.61*10**(-15)
        else:
                v1 = 3.53
                v2 = -1.24*10**(-4)
                v3 = -5.03*10**(-7)
                v4 = 2.44*10**(-9)
                v5 = -1.41*10**(-12)

        cGstar = (10**3/28)*Ru*(v1 + 0.5*v2*T + v3*T**2/3 + 0.25*v4*T**3 +\
        0.2*v5*T**4)

        return cGstar/c_inf

#Function that sets up convection-diffusion equations for TDM solve
@jit
def CDsetup(phi_o,C,Fo,Dr,Dl,u,S1,S2,taustep,fL,fR,epstepL,epstepR,\
        epstep,ep,N):

        #preallocate
        a = np.zeros_like(phi_o)
        b = np.zeros_like(phi_o)
        c = np.zeros_like(phi_o)
        d = np.zeros_like(phi_o)

        for j in range(1,N-1):#Doing the middle first

                        ajR = max(-u[j], Fo*(Dr[j]/epstepR[j]\
                         + 2*Dr[j]*epstep[j]/ep[j]*epstepR[j]) - fR*u[j], 0)

                        ajL = max(u[j], Fo*Dl[j]/epstepL[j]\
                         + fL*u[j], 0)

                        aj = -(ajL + ajR)

                        a[j] = -(taustep/C[j])*ajL/epstep[j]

                        b[j] =  1 - (taustep/C[j])*aj/epstep[j] - (taustep/C[j])*S1[j]

                        c[j] = -(taustep/C[j])*ajR/epstep[j]
```

```
                        d[j] = phi_o[j] + (taustep/C[j])*S2[j]

        return [a,b,c,d]

#Function that sets up solid phase mass conservation for TDM solve
@jit
def uHat_sSetup(theta_sn,S1,epstep_s,ep_s,Ns):

        #Preallocate some variables to be used for calculating uHat_s
        aP = np.zeros_like(theta_sn)
        bP = np.zeros_like(theta_sn)
        cP = np.zeros_like(theta_sn)
        dP = np.zeros_like(theta_sn)

        #Populate the middle of arrays. cP is all zeros for calc.
        #of uHat_s
        for j in range(1,Ns-1):#Doing the middle first
                aP[j] = -1/epstep_s[j]
                bP[j] = 2/ep_s[j] + 1/epstep_s[j]
                dP[j] = S1[j]

        return [aP,bP,cP,dP]

#Tri Diagonal Matrix Algorithm(a.k.a Thomas algorithm) solver for the lin.
#systems of equations produced by the iterative solve loop
# @jit
def TDMAsolver(a, b, c, d):

    nf = len(a)     # number of equations
    ac, bc, cc, dc = map(np.array, (a, b, c, d))     # copy the array
    for it in range(1, nf):
        mc = ac[it]/bc[it-1]
        bc[it] = bc[it] - mc*cc[it-1]
        dc[it] = dc[it] - mc*dc[it-1]

    xc = ac
    xc[-1] = dc[-1]/bc[-1]

    for il in range(nf-2, -1, -1):
        xc[il] = (dc[il]-cc[il]*xc[il+1])/bc[il]
```

```
    del bc, cc, dc  # delete variables from memory

    return xc

def OneDHFCI(r_p,Ti,ds_s,ds_g,runmode,rho_p=7860,w=0.9,Tl=925):

        #Define input-dependent dimensionless parameters
        #*********************************************************************

        theta_inf = Tinf/Ti #Initial and far boundary condition value for theta
        Ar_s = Ea_s/(Ru*Ti)
        Ar_g = Ea_g/(Ru*Ti)
        Py = r_p**2*Zs*np.exp(-Ar_s)/a_s
        Hhat_v = Hvol/(c_s*Ti)
        Qhat = Q/(c_inf*Ti)
        # Da = (r_p**2/a_g)*Zg*np.exp(-Ar_g)*rho_inf**(m+n-1)*Mf**(1-m)\
        #*(sigO/Mo2)**n #for when combustion reaction is on molar basis
        Da = (r_p**2/a_g)*Zg*np.exp(-Ar_g)*sigO*rho_inf #for when combustion
        #reaction is on mass basis
        b_s = rho_s*c_s/(rho_p*c_p)
        b_g = rho_inf*c_inf/(rho_p*c_p)
        psi = r_p*w*5.67*10**-8*Ti**3/(rho_p*c_p*a_g)

        #Define input-dependent numerical scheme inputs
        #*********************************************************************
        # Ns = 400 #number of spatial grids in solid phase
        Ns = int(ds_s/0.00001) #number of spatial grids in solid phase
        domainsize_s = ds_s/r_p #fuelbed domain in units of dim'less distance
        #(ep_s)
        # Ng = 400 #number of spatial grids in gas phase
        Ng = int(ds_g/0.0001)
        domainsize_g = ds_g/r_p #gas domain in units of dim'less distance (ep_g)
        timetot = 1*a_g/(r_p**2) #in units of dim'less time (tau)
        taustep_guess = 10**-4*a_g/(r_p**2) #guess of time step in units of
        #dim'less time (tau)
        sampleRes = 10**-4*a_g/(r_p**2) #temporal sample resolution in units of
        #dim'less time
        Nt = timetot/sampleRes #number of temporal samples

        #Iterating function that solves nonlinear system at each time step. Finite
        #volume formulation that is fully implicit in time with opt. numerical
        #relaxation
```

```
def solver(cs,ep_plot_g,ep_plot_s):
        # t1 = time.perf_counter()
        #Use values at current step (cs) as solution at i (old)
        uHat_so = cs[0]
        theta_so = cs[1]
        uHat_go = cs[2]
        Yhat_Fo = cs[3]
        Yhat_Oo = cs[4]
        theta_go = cs[5]
        # t2 = time.perf_counter()
        #Preallocate for the guesses, multiply by one to avoid double assgn.
        uHat_sg = 1.0*uHat_so
        theta_sg = 1.0*theta_so
        uHat_gg = 1.0*uHat_go
        Yhat_Fg = 1.0*Yhat_Fo
        Yhat_Og = 1.0*Yhat_Oo
        theta_gg = 1.0*theta_go
        # t3 = time.perf_counter()
        # t4 = time.perf_counter()
        #Setup while loop, timestep, iteration counter and dom-error switch
        Error = 10**12
        OldError = 10**12
        numit = 0
        taustep = taustep_guess
        killI = 'off'
        MaxTempDiffI = 0
        # t5 = time.perf_counter()
        # t6 = time.perf_counter()
        while Error > 10**-4:

                #Calculate diffusivities at volume faces and specific heat base
                #on guess temps for every 5th iteration
                # t7 = time.perf_counter()
                if np.mod(numit,5) == 0:

                        [lhat_sL,lhat_sR] = lhat_s(theta_sg,Ti,fLs,Ns)
                        [lhat_gL,lhat_gR] = lhat_g(theta_gg,Ti,fLg,Ng)
                        gamG = gamG_(theta_gg,Ti)
                # t8 = time.perf_counter()
                #First use value uHat_sg and theta_gg to solve
                #for solid phase temperatures.
                # t9 = time.perf_counter()
```

```
#Fill arrays
[ats,bts,cts,dts] = CDsetup(theta_so,np.ones_like(theta_so),\
                Ahat,lhat_sR,lhat_sL,uHat_sg,\
                -Hhat_v*Ahat*Py*np.exp(Ar_s*(1-1/theta_sg))/thet
                 np.zeros_like(theta_so),taustep,\
                fLs,fRs,epstepLs,epstepRs,epstep_s,ep_s,Ns)
# t10 = time.perf_counter()
#Apply BCs
bts[0] = 1 + (taustep/gamP_(theta_sg[0],Ti,Tl))*P*Ahat*b_s*\
        lhat_sR[0]/epstepRs[0] +\
        (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*psi*theta_sg[0]
         + (taustep/gamP_(theta_sg[0],Ti,Tl))*P*b_g*Rhat*Ahat*Py
         gamGstar(theta_sg[0],Ti)*\
         np.sum(np.exp(Ar_s*(1-1/theta_sg[1:]))\
        *ep_s[1:]**2*epstep_s[1:])
cts[0] = -(taustep/gamP_(theta_sg[0],Ti,Tl))*P*Ahat*b_s*\
                        lhat_sR[0]/epstepRs[0]
dts[0] = theta_so[0] + (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)\
                        *b_g*lhat_gR[0]*(theta_gg[1]\
        - theta_gg[0])/epstepRg[0] +\
                        (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)
                        psi*theta_inf**4\
        +(taustep/gamP_(theta_sg[0],Ti,Tl))*P*b_g*Rhat*Ahat*Py*\
        np.sum(hHat_g_(theta_sg[1:],Ti)*\
                np.exp(Ar_s*(1-1/theta_sg[1:]))*\
                ep_s[1:]**2*epstep_s[1:])

# bts[-1] = 1
# dts[-1] = theta_inf
bts[-1] = 1
ats[-1] = -1
# t11 = time.perf_counter()
#Apply numerical relaxation
bts = bts/A
dts = dts + (1-A)*bts*theta_sg
# t12 = time.perf_counter()
#Solve the system
theta_sn = TDMAsolver(ats,bts,cts,dts)
# t13 = time.perf_counter()
#Then use theta_sn, Yhat_Fg, Yhat_Og and uHat_gg to solve for g
#phase temperatures
# t14 = time.perf_counter()
```

```
#Fill arrays
[atg,btg,ctg,dtg] = CDsetup(theta_go,gamG,\
        1,lhat_gR,lhat_gL,uHat_gg*gamG,np.zeros_like(theta_go),\
        Qhat*Da*Yhat_Fg**m*Yhat_Og**n*np.exp(Ar_g*(1-1/theta_gg)
        taustep,fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t15 = time.perf_counter()
# #Fill arrays
# [atg,btg,ctg,dtg] = CDsetup(theta_go,gamG,\
#         1,lhat_gR,lhat_gL,uHat_gg*gamG,np.zeros_like(theta_g
#         np.zeros_like(theta_go),\
#         taustep,fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)

#Apply BCs
btg[0] = 1 + (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*b_g*\
                lhat_gR[0]/epstepRg[0]
ctg[0] = -(taustep/gamP_(theta_sg[0],Ti,Tl))*(1-P)*b_g*\
                lhat_gR[0]/epstepRg[0]
dtg[0] = theta_go[0] + (theta_sn[0] - theta_so[0])\
                        - (taustep/gamP_(theta_sg[0],Ti,Tl))*(1-
                        b_g*lhat_gR[0]*\
                        (theta_gg[1] - theta_gg[0])/epstepRg[0]

# btg[-1] = 1
# dtg[-1] = theta_inf
btg[-1] = 1
atg[-1] = -1
# t16 = time.perf_counter()

#Apply numerical relaxation
btg = btg/A
dtg = dtg + (1-A)*btg*theta_gg
# t17 = time.perf_counter()

#Solve the system
theta_gn = TDMAsolver(atg,btg,ctg,dtg)
# t18 = time.perf_counter()

#Update the solid phase p-temp to the gas phase p-temp
theta_sn[0] = theta_gn[0]

#Next, use guess of theta_sn to get new value for uHat_sn
```

```
[aP,bP,cP,dP] = uHat_sSetup(theta_sn,\
        - Ahat*Py*np.exp(Ar_s*(1-1/theta_sn)),epstep_s,ep_s,Ns)

#Apply boundary conditions. Ends of dP were specified during
#initialization
aP[-1] = -1
bP[-1] = 1
bP[0] = 1
# t19 = time.perf_counter()

# #Apply numerical relaxation
# bP = bP/A
# dP = dP + (1-A)*bP*uHat_sg

#Solve the linear system W1*uHat_s = b1 and apply num. relax.
uHat_sn = TDMAsolver(aP,bP,cP,dP)
# t20 = time.perf_counter()
#Determine gas phase velocity

uHat_gn = P/(1-P)*(Rhat/ep_g**2)*Ahat*Py*\
np.sum(np.exp(Ar_s*(1-1/theta_sn[1:]))*ep_s[1:]**2*epstep_s[1:])

uHat_gn[0] = uHat_gn[1]
# t21 = time.perf_counter()
#Solve fuel species conservation

#Fill arrays
[af,bf,cf,df] = CDsetup(Yhat_Fo,np.ones_like(Yhat_Fo),1,\
        np.ones_like(Yhat_Fo),np.ones_like(Yhat_Fo),\
        uHat_gn,-Da*Yhat_Fg**m*Yhat_Og**n*\
        np.exp(Ar_g*(1-1/theta_gn))/\
        np.maximum(Yhat_Fg,10**-8*np.ones_like(Yhat_Fg)),\
        np.zeros_like(Yhat_Fo),taustep,\
        fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# [af,bf,cf,df] = CDsetup(Yhat_Fo,np.ones_like(Yhat_Fo),1,\
#         np.ones_like(Yhat_Fo),np.ones_like(Yhat_Fo),\
#         uHat_gn,np.zeros_like(Yhat_Fo),\
#         np.zeros_like(Yhat_Fo),taustep,\
#         fLg,fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t22 = time.perf_counter()
#Apply BCs
bf[0] = (1 + epstepRg[0]*uHat_gn[1])
```

```
cf[0] = -1
df[0] = epstepRg[0]*uHat_gn[1]

# bf[-1] = 1
# df[-1] = 0
bf[-1] = 1
af[-1] = -1

# #Apply BCs
# bf[0] = 1
# cf[0] = 0
# df[0] = 1
# bf[-1] = 1
# df[-1] = 0

#Apply numerical relaxation
bf = bf/A
df = df + (1-A)*bf*Yhat_Fg

#Solve the system
Yhat_Fn = TDMAsolver(af,bf,cf,df)

#Solve oxy species conservation
# t23 = time.perf_counter()
#Fill array
[ao2,bo2,co2,do2] = CDsetup(Yhat_Oo,np.ones_like(Yhat_Oo),1,\
        np.ones_like(Yhat_Oo),np.ones_like(Yhat_Oo),uHat_gn,\
        -Da*Yhat_Fg**m*Yhat_Og**n*np.exp(Ar_g*(1-1/theta_gn))/\
        np.maximum(Yhat_Og,10**-8*np.ones_like(Yhat_Og)),\
        np.zeros_like(Yhat_Oo),taustep,fLg,\
        fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# [ao2,bo2,co2,do2] = CDsetup(Yhat_Oo,np.ones_like(Yhat_Oo),1,\
#         np.ones_like(Yhat_Oo),np.ones_like(Yhat_Oo),uHat_gn,\
#         np.zeros_like(Yhat_Oo),\
#         np.zeros_like(Yhat_Oo),taustep,fLg,\
#         fRg,epstepLg,epstepRg,epstep_g,ep_g,Ng)
# t24 = time.perf_counter()
#Apply BCs
bo2[0] = 1 + epstepRg[0]*uHat_gn[1]
co2[0] = -1
do2[0] = 0
```

```
# bo2[-1] = 1
# do2[-1] = Yo2inf/sigO
bo2[-1] = 1
ao2[-1] = -1

#Apply numerical relaxation
bo2 = bo2/A
do2 = do2 + (1-A)*bo2*Yhat_Og

#Solve the system
Yhat_On = TDMAsolver(ao2,bo2,co2,do2)

#Set values at particle equal to value next to particle for
#plotting
Yhat_Fn[0] = Yhat_Fn[1]
Yhat_On[0] = Yhat_On[1]
# t25 = time.perf_counter()
#Calculate error and determine if time step needs to be dec.
YhatF_Error = np.max(np.abs(Yhat_Fn-Yhat_Fg)/\
        np.maximum(Yhat_Fg,10**-7*np.ones_like(Yhat_Fg)))
YhatO_Error = np.max(np.abs(Yhat_On-Yhat_Og)/\
        np.maximum(Yhat_Og,10**-7*np.ones_like(Yhat_Og)))

theta_sError = np.max(np.abs(theta_sn-theta_sg))
theta_gError = np.max(np.abs(theta_gn-theta_gg))

NewError = np.max([YhatF_Error,YhatO_Error,theta_sError,\
        theta_gError])
# t26 = time.perf_counter()
if NewError > Error or numit > 50:

        taustep = 0.5*taustep
        numit = 0
        uHat_sg = 1.0*uHat_so
        theta_sg = 1.0*theta_so
        uHat_gg = 1.0*uHat_go
        Yhat_Fg = 1.0*Yhat_Fo
        Yhat_Og = 1.0*Yhat_Oo
        theta_gg = 1.0*theta_go
        # if NewError > Error:
        #         print('Diverging')
        # else:
```

```
                #         print('max interations exceeded')

                # if NewError == YhatF_Error:
                #         print('Error in fuel species')
                # elif NewError ==YhatO_Error:
                #         print('Error in oxygen species')
                # elif NewError == theta_sError:
                #         print('Error in solid temp')
                # else:
                #         print('Error in gas temp')

                Error = 10**12
                OldError = 10**12
        else:

                OldError = Error
                Error = NewError
                numit = numit + 1
                uHat_sg = 1.0*uHat_sn
                theta_sg = 1.0*theta_sn
                uHat_gg = 1.0*uHat_gn
                Yhat_Fg = 1.0*Yhat_Fn
                Yhat_Og = 1.0*Yhat_On
                theta_gg = 1.0*theta_gn
                # print('Good Run, error' + str(Error))
        # t27 = time.perf_counter()

# t28 = time.perf_counter()
newvalues = [uHat_sn,theta_sn,uHat_gn,Yhat_Fn,Yhat_On,theta_gn]
# t29 = time.perf_counter()
BoundaryDeltaG = np.max([np.abs(theta_gn[-2]-theta_inf),\
        np.abs(Yhat_Fn[-2]),\
        np.abs(Yhat_On[-2] - Yo2inf/sigO)])
BoundaryDeltaS = np.abs(theta_sn[-2]-theta_inf)

if BoundaryDeltaG > 10**-4:

        print('Edge of numerical gas-phase domain has been reached.'+\
                ' Need to rerun case with larger domain.')
        killI = 'onG'
if BoundaryDeltaS > 10**-4:
        print('Edge of numerical solid-phase domain has been reached.'+\
```

```
                        ' Need to rerun case with larger domain.')
                killI = 'onS'

        MaxTempDiffI = np.max(theta_gn)-np.max(theta_go)
        # t30 = time.perf_counter()
        return [newvalues,taustep,killI,MaxTempDiffI]

#Initialize time step, mesh and dim'less variables
#*************************************************************************
[ep_s,ep_plot_s,epstep_s,epstepLs,epstepRs,fLs,fRs] =\
meshcreator(Ns,domainsize_s,ep_p,Gs)
[ep_g,ep_plot_g,epstep_g,epstepLg,epstepRg,fLg,fRg] =\
meshcreator(Ng,domainsize_g,ep_p,Gg)
tauarray = np.zeros((Nt+2,1),float) #establish time vector
tau = 0 #initialize tau counter
i = 1 #counter for time stepping
kill = 'off' #indicator for when edge of domain is reached
MaxTempDiff = 0 #indicator of ignition
igprintswitch = 'on' #forces to print ignition msg only once
tau_ig = 0 #will hold ignition time
uHat_s = np.zeros((Nt+2,Ns),float) #dim'less solid phase velocity
theta_s = np.ones((Nt+2,Ns),float)*theta_inf #dim'less sol phase temp
theta_s[0,0] = 1 #Initial condition at boundary for theta_s
uHat_g = np.zeros((Nt+2,Ng),float) #dim'less gas phase velocity for diagn.
# Yhat_F = 10**-28*np.ones((Nt+2,Ng),float)#init. mass fraction of fuel,
#avoid divide by zero
Yhat_F = np.zeros((Nt+2,Ng),float)#init. mass fraction of fuel, avoid

Yhat_O = np.ones((Nt+2,Ng),float)*Yo2inf/sigO #Initialize mass frac of O2
#Yp = np.ones((Nt,Ng),float)*Yp_inf #Initialize mass frac of inert spec.
theta_g = np.ones((Nt+2,Ng),float)*theta_inf #dim'less gas phase temp
theta_g[0,0] = 1 #Initial condition at boundary for theta_g

#Loop that marches through time, solving non-lin system (see above) at
#each time step
#*************************************************************************

cs =[uHat_s[0,],theta_s[0,],uHat_g[0,],Yhat_F[0,],Yhat_O[0,],theta_g[0,]]

if runmode == 'bisect':
        while tau <= timetot and MaxTempDiff <= 0 and kill == 'off':
        # while tau <= timetot and np.max(cs[5]) <= 1.05 and kill == 'off':
```

```
        # while tau <= timetot and kill == 'off':

                #Solve non-lin system for next time step using current step and
                #pack variable arrays at current step for passing (globally) to
                #solver
                [cs,taustep,kill,MaxTempDiff] = solver(cs,ep_plot_g,ep_plot_s)

                #keep track of current time
                tau = tau + taustep
                # print(tau)

                if tau >= i*sampleRes:

                        i += 1 #proceed to next diff res step
                        # if np.mod(i,100)==0:
                        #         print(tau)

                if MaxTempDiff > 0 and igprintswitch == 'on':
                        tau_ig = tau
                        print('Ignition occurred.')
                        igprintswitch = 'off'


        if tau >= timetot:
                print('max time reached.')

        thing2return =  [tau_ig,kill]

else:
        # while tau <= timetot and MaxTempDiff <= 0 and kill == 'off':
        # while tau <= timetot and np.max(cs[5]) <= 1.05 and kill == 'off':
        while tau <= timetot and kill == 'off':

                #Solve non-lin system for next time step using current step and
                #pack variable arrays at current step for passing (globally) to
                #solver
                [cs,taustep,kill,MaxTempDiff] = solver(cs,ep_plot_g,ep_plot_s)

                #keep track of current time
                tau = tau + taustep
                # print(tau)
```

```
        if tau >= i*sampleRes:

                uHat_s[i,] = cs[0]
                theta_s[i,] = cs[1]
                uHat_g[i,] = cs[2]
                Yhat_F[i,] = cs[3]
                Yhat_O[i,] = cs[4]
                theta_g[i,] = cs[5]

                tauarray[i] = tau
                # print('--------' + str(tau))
                i += 1 #proceed to next diff res step
                if np.mod(i,100)==0:
                        print(tau)
                # print(i)

        if i == Nt-2:
                add = 100
                #add = round((1- tau[i]/timetot)*i/(tau[i]/timetot))
                uHat_s = np.vstack([uHat_s,np.ones((add,Ns),float)])
                theta_s = np.vstack([theta_s,theta_inf*\
                        np.ones((add,Ns),float)])
                uHat_g = np.vstack([uHat_g,np.zeros((add,Ng),float)])
                Yhat_F = np.vstack([Yhat_F,10**-28*\
                        np.ones((add,Ng),float)])
                Yhat_O = np.vstack([Yhat_O,(Yo2inf/sigO)*\
                        np.ones((add,Ng),float)])
                theta_g = np.vstack([theta_g,theta_inf*\
                        np.ones((add,Ng),float)])
                tauarray = np.vstack([tauarray,np.zeros((add,1),float)])
                Nt += add

        if MaxTempDiff > 0 and igprintswitch == 'on':
                tau_ig = tau
                print('Ignition occurred.')
                igprintswitch = 'off'


if tau >= timetot:
        print('max time reached.')

uHat_s = uHat_s[0:i,]
```

```
                theta_s = theta_s[0:i,]
                uHat_g = uHat_g[0:i,]
                Yhat_F = Yhat_F[0:i,]
                Yhat_O = Yhat_O[0:i,]
                theta_g = theta_g[0:i,]
                tauarray = tauarray[0:i,]

                thing2return = [ep_s,ep_plot_s,ep_g,ep_plot_g,tauarray,uHat_s,\
                        theta_s,uHat_g,Yhat_F,Yhat_O,theta_g,tau_ig,kill]
        return thing2return

#*************************************************************************
#END FUNCTION LIBRARY

#SETUP SWEEPING PROCESSES
#*************************************************************************

#First create array for diameters of interest
d_p = np.array([0.010,0.009,0.008,0.007,0.006,0.005,0.004,0.003,0.0025,\
        0.002,0.0015,0.001,0.00075,0.000625,0.0005])

#Convert to radius
r_p = d_p/2

#Then create array for variable to be swept, in this case volumetric
#heat capacity.
Sweeper = np.array([940])

#Give a starting guess for temperatures where the bounding algorithm
#starts looking, 'new' and 'old' subscript are used in bounding
Tnew = np.array([920.875,921.65625,922.4375,924.78125,927.34375,931.90625,\
                                        939.25,955.28125,969.875,993.8125,1037.46875,113
                                          1223.4375,1290.90625,1388.84375])
Tnew = np.tile(Tnew,(len(Sweeper),1))
Told = 1.0*Tnew

#Specify a temperature interval over which to bound for the bounding
#algorithm
delTbound = 50

#Preallocate arrays for the ignition limit temperatures
Titop = np.zeros((len(Sweeper),len(d_p)),float)
```

```
Tibot = np.zeros((len(Sweeper),len(d_p)),float)

#CARRY OUT SWEEP AND SAVE VARIABLES
#*****************************************************************************

#Sweep through variable of interest and diameters and store ignition
#temperatures
for j in range(0,len(d_p)):

        print('working on d = ' + str(d_p[j]*10**3) + 'mm.')

        for k in range(0,len(Sweeper)):

                ds_s = 0.004
                ds_g = 0.04

                print('working on Sweeper = ' + str(Sweeper[k]))

                while Titop[k,j] == 0 and Tibot[k,j] == 0:

                        [tau_ig,kill] = OneDHFCI(r_p[j],Tnew[k,j],\
                        ds_s,ds_g,'bisect',Tl = Sweeper[k])

                        if kill == 'onS':

                                ds_s = ds_s + 0.001

                        elif kill == 'onG':

                                ds_g = ds_g + 0.010

                        elif tau_ig > 0:

                                print('Ignited at ' + str(Tnew[k,j]) +  'K.')

                                if Tnew[k,j]-Told[k,j] > 0:
                                        Titop[k,j] = Tnew[k,j]
                                        Tibot[k,j] = Told[k,j]
                                else:
                                        Told[k,j] = Tnew[k,j]
                                        Tnew[k,j] = Told[k,j] - delTbound
                        else:
```

```
                                print('Did not ignite at ' + str(Tnew[k,j]) +  'K.')

                                if Tnew[k,j]-Told[k,j] >= 0:
                                        Told[k,j] = Tnew[k,j]
                                        Tnew[k,j] = Told[k,j] + delTbound
                                else:
                                        Tibot[k,j] = Tnew[k,j]
                                        Titop[k,j] = Told[k,j]

                        while Titop[k,j] - Tibot[k,j] > 1:

                                [tau_ig,kill] = OneDHFCI(r_p[j],(Titop[k,j] + Tibot[k,j])/2,\
                                ds_s,ds_g,'bisect',Tl = Sweeper[k])

                                if kill == 'onS':

                                        ds_s = ds_s + 0.001

                                elif kill == 'onG':

                                        ds_g = ds_g + 0.010

                                elif tau_ig > 0:

                                        Titop[k,j] = (Titop[k,j] + Tibot[k,j])/2
                                        print('Ignited at ' + str(Titop[k,j]) +  'K.')

                                else:

                                        Tibot[k,j] = (Titop[k,j] + Tibot[k,j])/2
                                        print('Did not ignite at ' + str(Tibot[k,j]) +  'K.')

                        print('Completed Sweeper = '+ str(Sweeper[k]))

                print('Completed d = '+ str(d_p[j]*10**3) + 'mm')

        #Store variable of interest, diameters, and corresponding ignition temps

        # pickle.dump([Titop,Tibot],open('InitialTemps2.p','wb'))
        pickle.dump([d_p,Sweeper,Titop,Tibot],open('Limit_Melting.p','wb'))
```