Co-Milling and Cofiring of Woody Biomass with Coal in Utility Boilers: Enabling Technology Through Experiments and Modelling

Seyedhassan Fakourian
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation

This Dissertation is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
Co-Milling and Cofiring of Woody Biomass with Coal in Utility Boilers: Enabling Technology Through Experiments and Modelling

Seyedhassan Fakourian

A dissertation submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Andrew R. Fry, Chair
David O. Lignell
Larry Lin Baxter
Jost O.L. Wendt
Stella Day Nickerson

Department of Chemical Engineering
Brigham Young University

Copyright © 2020 Seyedhassan Fakourian
All Rights Reserved
ABSTRACT

Co-Milling and Cofiring of Woody Biomass with Coal in Utility Boilers: Enabling Technology Through Experiments and Modelling

Seyedhassan Fakourian
Department of Chemical Engineering, BYU
Doctor of Philosophy

Beetle-killed trees and woody residues degenerate and may lead to wildfires and uncontrolled CO2 emission. Woody biomass is known as a neutral CO2 solid fuel since it generates the same amount of CO2 that takes from atmosphere during its growing up. Cofiring woody biomass with coal in existing coal power plants is a reasonable solution to reduce the net amount of CO2 emission and decrease the risk of wildfires. However, there are some challenges ranging from providing and handling the woody biomass to the operation of cofiring woody biomass with coal. Co-milling of the fuels and ash deposition on the heat exchanger surfaces during cofiring are among the most critical challenges. A CFD model simulated the behavior of the pulverized particles and evaluate the impact of geometry and operational changes on mill performance. In addition, we measured the ash deposit rate derived from cofiring woody biomass with coal in a pilot combustor (1500 kW) and full-scale furnace. Moreover, we developed a model to predict ash deposit rate during combustion of coal and its blend with a variety of biomass.

The post-processing analysis of CFD modelling of co-milling woody biomass with coal shows that the entrained large woody biomass particles exit the pulverizer along with the fine coal particles due to their lower density than that of coal particles. Some simple geometry and operational changes can optimize mill performance by reducing the number of large biomass particles in the product stream. Therefore, it makes the particle size distribution (PSD) of the product stream of co-milling more like that of coal.

The collected data set of fly ash particles and ash deposit samples shows that the ash formation and deposit rates were not impacted significantly by cofiring woody biomass with coal. The concentration of alkali metals in the ash aerosol during cofiring was slightly higher than that of coal. Cofiring in pilot scale combustor made a tri-modal PSD of ash aerosol particles; however, the distribution was bimodal in the full-scale boiler. The ash deposit rates during cofiring in 1500 kW combustor were higher (30 to 70%) at locations closer to the burner at short operation times.

Our developed model of ash deposit rate investigated two types of stickiness models of fly ash particles to the surface of heat exchanger: melt fraction stickiness model (MFSM) and kinetic energy stickiness model (KESM). The developed model suggested that the MFSM, which is based on the melt fraction of ash and our novel approach to condensation of alkali vapor species, was more accurate in predicting ash deposit rate of a variety of fuel combustion of a 100-kW combustor. The model calculated four mechanisms: inertial impaction, thermophoresis, condensation, and eddy impaction.

Keywords: CFD, co-milling, pulverizer, coal, woody biomass, PSD, ash deposit rate, stickiness model, mechanism, inertial impaction, thermophoresis, condensation, eddy.
ACKNOWLEDGEMENTS

Rocky Mountain Power’s Sustainable Transportation and Energy Plan (STEP), administered through the University of Utah, funded this work. I appreciate their support.

I would like to express my sincere appreciation to my advisor, Dr. Andrew Fry, for his technical support, motivation, patience, and knowledge. He has helped me kindly anytime that I needed and encouraged me to figure out the hard problems throughout this research. I would like to give my appreciation to my graduate committee members Dr. Baxter, Dr. Lignell, Dr. Wendt, and Dr. Nickerson for their time, and support to guide me throughout this degree. In addition, I would like to thank Dr. Pitt for his help in mass and heat transport problems of this study.

I would like to thank Dr. Zsolt Dobó for helping me collecting the dimension data of the pilot-scale pulverizer. I will not forget the collaboration of Tanner Jasperson and Jinze Dai in programming by C# and simulating by FactSage package, respectively. I strongly appreciate Dr. Zhonghua Zhan for providing some hints in modeling of ash deposit on the heat exchanger surface. He was very helpful. I would like to thank Dr. Yueming Wang and Xiaolong Li who collaborated in some experimental works of collecting ash deposit samples and analyzing fly ash aerosol. I greatly appreciate their advisor, Dr. Wendt, for his help to let me collaborate in some of their tests during summer 2017.

Finally, I would like to thank to my parents, wife, kids for their persistent support and encouragement. They are the most important asset in my life. Particularly, I have to thank my wife, for her pure support during these years and providing a calm environment for me to pursue my goals.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>GLOSSARY OF TERMS</td>
<td>xiv</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Literature survey</td>
</tr>
<tr>
<td>2.1</td>
<td>Background</td>
</tr>
<tr>
<td>2.2</td>
<td>Techniques for biomass preprocessing</td>
</tr>
<tr>
<td>2.3</td>
<td>Milling</td>
</tr>
<tr>
<td>2.4</td>
<td>Ash content and ash composition of coal and woody biomass fuels</td>
</tr>
<tr>
<td>2.5</td>
<td>Ash transformations</td>
</tr>
<tr>
<td>2.6</td>
<td>Ash deposition behavior</td>
</tr>
<tr>
<td>2.7</td>
<td>Measuring ash deposition rates and properties</td>
</tr>
<tr>
<td>2.8</td>
<td>Modeling ash deposition rates</td>
</tr>
<tr>
<td>2.9</td>
<td>Summary of literature review</td>
</tr>
<tr>
<td>3</td>
<td>Objective and tasks</td>
</tr>
<tr>
<td>3.1</td>
<td>Objective</td>
</tr>
<tr>
<td>3.2</td>
<td>Tasks</td>
</tr>
<tr>
<td>4</td>
<td>Analysis of particle behavior inside the classifier of a Raymond Bowl Mill while co-milling woody biomass with coal</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Materials and methods</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Experimental work</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Model description</td>
</tr>
<tr>
<td>4.3</td>
<td>Results and discussion</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Hardware manipulation scenarios</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Scenarios of operating parameters</td>
</tr>
<tr>
<td>4.4</td>
<td>Conclusion</td>
</tr>
<tr>
<td>5</td>
<td>Design and construction of an air-cooled ash deposit probe and a fly ash aerosol probe to collect and measure ash deposit in a pilot and full-scale boiler</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Ash deposit probe</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Ash deposit probe, tested in a 1.5 MW combustor</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Modified ash deposit probe, tested in a full-scale boiler (Hunter, unit-3)</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Control system of the ash deposit probe</td>
</tr>
</tbody>
</table>
Modelling of ash deposition rate of coal combustion and its blend with woody biomass in a 1.5 MW combustor. ........................................................................................................................................... 133

9.1 Introduction ........................................................................................................................................... 133

9.2 Results and discussion ...................................................................................................................... 133

9.2.1 Thermodynamic analysis ...................................................................................................................... 134

9.2.2 Ash deposition model ............................................................................................................................ 138

9.3 Conclusion ........................................................................................................................................... 143

10 Summary and conclusion ..................................................................................................................... 144

10.1 Analysis of the trajectories of the pulverized particles of woody biomass and coal during co-milling........................................................................................................................................... 144

10.2 Cofiring the blend of woody biomass and coal and compare the samples of ash deposit and ash aerosol to those of coal combustion ........................................................................................................................................... 145

10.3 Ash deposit model of the combustion of a range of solid fuels in a 100-kW combustor. ........................................................................................................................................... 147

11 Recommended future work ..................................................................................................................... 150

References ........................................................................................................................................... 152

Appendix A: The developed code for analysis of particle trajectories .............................................................. 166

Appendix B: Fabrication drawings of ash deposit probe ................................................................................ 273

Appendix C: Summary of the PI data of Hunter#3 ............................................................................................ 288

Appendix D: Numerical code of ash deposit rate ........................................................................................... 290
# LIST OF TABLES

Table 2-1: Ash elemental analysis of woody biomass and bituminous coal [52] ..................... 12
Table 2-2: Proximate analysis and ultimate analysis of wood and coal, reported in reference [56] ........................................................................................................................................................................... 12
Table 4-1: Summary of model boundary conditions identified in Figure 4-3. ......................... 35
Table 4-2: Final values of parameters that were adjusted to match the baseline model with experimental measurements. ................................................................................................................................................................................................................................. 37
Table 4-3: Primary results of analysis of running hardware scenarios. ................................. 47
Table 6-1: Ultimate and proximate analysis of fuels from the past study [118]. Values are % mass as received. HHV is kJ/kg as received ................................................................. 67
Table 6-2. Mineral matter composition, % mass, from the past study [118]. Difference between sum and 100% is undetermined ................................................................. 67
Table 6-3. Target operating parameters designed to match coal-fired utility boiler conditions, within the capabilities of the L1500 ................................. 68
Table 6-4. Average measured operating conditions for each fuel ........................................ 69
Table 6-5. Operating parameters (measured and calculated) relevant to the particle deposition behavior ................................................................. 69
Table 7-1: Ultimate and proximate analysis of fuels. Values are % mass as received before normalization. HHV is kJ/kg as received ................................. 86
Table 7-2. Mineral matter composition, % mass. Difference between sum and 100% is undetermined ................................................................. 86
Table 7-3: Target operating parameters of Hunter boiler, Unit-3 ........................................ 86
Table 7-4: Actual schedule of ash deposit and ash aerosol sampling .................................. 87
Table 7-5: Average measured operating conditions for each fuel ....................................... 88
Table 7-6: Operating parameters (measured and calculated) relevant to the particle deposition behavior ................................................................. 88
Table 8-1: Ultimate and Proximate Fuel Analysis ................................................................. 103
Table 8-2: Mineral ash analysis. .................................................................................................. 103
Table 8-3: The parameters of overall Nusselt number cited in [181] ........................................ 114
Table 8-4: Distinct and Common Parameters used in the model. The fuels include: a) RH, b) Sufco#1, c) Sufco#2, d) 20RH-80Sufco#1, e) PRB, f) 13RH-87PRB, g) Pet coke, h) Illinois, i) 60Illinois-40PRB, j) Torrefied, k) 50Torrefied-50Sufco#2 .......................................................... 124
Table 8-5: The experimental ash deposition data for three operation times of 30, 60, and 120 minutes. The fuels include a) RH, b) Sufco#1, c) Sufco#2, d) 20RH-80Sufco#1, e) PRB, f) 13RH-87PRB, g) Pet coke, h) Illinois, i) 60Illinois-40PRB, j) Torrefied, k) Torrefied-50Sufco#2. Only one test was performed (60-minute test) for fuels (e), (h), and (i). ........................................................................................................................................ 125
Table 9-1: Temperature ranges of alkali vapor condensation .................................................. 136
Table 9-2: Temperature ranges of ash melt fractions. ............................................................... 138
Table 9-3: Operating parameters of the 1.5 MW combustor ...................................................... 138
LIST OF FIGURES

Figure 1-1: Coal-fired utility power plant schematic [9]. ............................................................... 2

Figure 2-1: Typical mass and energy balance of the torrefaction process. E=energy, M=Mass [36]. ................................................................................................................................................. 8

Figure 2-2: Schematic side view of the physical milling system.................................................... 9

Figure 2-3: Two commercial classifiers: (a) static classifier (b) dynamic classifier [47] .......... 10

Figure 2-4: Included and excluded minerals in coal [67]. ............................................................ 14

Figure 2-5: Main processes in ash aerosol formation in biomass combustion [74]................. 15

Figure 2-6: a) Heterogeneous nucleation on the external surface; b) Homogeneous
nucleation to form a particle [60].................................................................................................. 15

Figure 2-7: (a) Picture of the ash deposition probe; (b) The details of the probe [104]. .......... 20

Figure 2-8: Schematic view of the ash deposition probe used by Y. Shao et al. [107]. .......... 22

Figure 2-9: The relative accumulated deposition versus deposition time in different ash
deposition mechanism [83]. .......................................................................................................... 24

Figure 4-1: A schematic diagram of a static classifier indicating typical movement of
particles ........................................................................................................................................ 32

Figure 4-2: Beckman Coulter analysis of particle size distribution for coal-woody biomass
blend [118]. ................................................................................................................................... 33

Figure 4-3: (a) Translucent representation of meshed model in Barracuda VR,
(b) SolidWorks model of void space. .............................................................................................. 34

Figure 4-4: PSD of blend product stream in simulation and experiment. ................................. 39

Figure 4-5: Average mass flow rate of product stream (kg/s). ..................................................... 40

Figure 4-6: PSD of coal particles at input (injectors) and product stream............................. 40

Figure 4-7: PSD of woody biomass particles at input (injectors, which presented in
Figure 4-3) and product stream ..................................................................................................... 41

Figure 4-8: PSD of product stream, bottom of classifier, and injectors. ................................. 41
Figure 6-6. Comparison of ash aerosol PSD measured at port 7 and port 10 while firing coal.

Figure 6-7. Size segregated compositions of ash aerosols sampled at port 7 for each of the different fuels.

Figure 6-8. Images of deposit build-up on coupon surfaces for each deposit test interval and fuel. By row: 1) Coal 2) Torrefied 3) Steam X. The picture on the bottom right is representative of a 68-minute sample from Port 10.

Figure 6-9. Mass of collected deposit and rate of deposition for the three fuel conditions in both Port 7 and Port 10.

Figure 6-10. Ash deposit compositions for the 30 and 90-minute samples taken in port 7 for all three fuel conditions.

Figure 6-11. Ash deposit compositions for the 30 and 90-minute samples taken in port 10 for all three fuel conditions.

Figure 6-12. SEM images of ash deposit samples taken in port 7 with a deposition time of 90 minutes for all three conditions.

Figure 6-13. The PSD of 30 and 90 minutes deposit samples collected in port 7 for all three fuel conditions.

Figure 7-1: Schematic view of the boiler of Hunter-3 (Ref.: US DOE Cooperative Agreement No. DE-NT0005288).

Figure 7-2: The schematic view of Hunter, Unit-3, and the location of test port for ash deposit and ash aerosol probes (Image from Babcock & Wilcox [9]).

Figure 7-3: Ash deposit sampling probe at the test port.

Figure 7-4: Selected PI data during torrefied blend and coal combustion; fuel flow (top left), pulverizer motor current (top right), NOx (bottom left), and SO2 concentration (bottom right).

Figure 7-5: PSDs of ash aerosols in baseline and cofiring cases.

Figure 7-6: Size-segregated compositions of ash aerosols in the two cases.

Figure 7-7: Compositions of ash aerosols (above 15.7 μm) collected by cyclone.
Figure 9-7: Predicted ash deposit shape after 90 minutes of Sufco coal operation at port 7...... 142
### GLOSSARY OF TERMS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Abbreviation</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_c$</td>
<td>Stokes-Cunningham factor</td>
<td>$St_{eff}$</td>
<td>Effective Stokes number</td>
<td></td>
</tr>
<tr>
<td>$C_D$</td>
<td>Drag coefficient</td>
<td>$t$</td>
<td>Time (s)</td>
<td></td>
</tr>
<tr>
<td>$C_{eddy}$</td>
<td>Mass concentration of eddy particles (kg·m⁻³)</td>
<td>$T$</td>
<td>Temperature (K)</td>
<td></td>
</tr>
<tr>
<td>$C_p$</td>
<td>Mass concentration of fly ash particles surrounding the probe (kg·m⁻³)</td>
<td>$T_g$</td>
<td>Flue gas temperature (K)</td>
<td></td>
</tr>
<tr>
<td>$d_p$</td>
<td>Particle diameter (m)</td>
<td>$T_s$</td>
<td>Deposit surface temperature (K)</td>
<td></td>
</tr>
<tr>
<td>$d_{p-eddy}$</td>
<td>Particle diameter of eddy impaction (m)</td>
<td>$T_{coupon,s}$</td>
<td>Coupon surface Temperature (K)</td>
<td></td>
</tr>
<tr>
<td>$D_v$</td>
<td>Diffusion coefficient of vapor (m²·s⁻¹)</td>
<td>$T_X$</td>
<td>The temperature at which melt fraction of ash particle or ash deposit is X/100</td>
<td></td>
</tr>
<tr>
<td>$D_h$</td>
<td>Hydraulic diameter of probe (m)</td>
<td>$\nabla T_{g-s}$</td>
<td>temperature gradient between flue gas and ash deposit surface temperature (K·m⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$E_p$</td>
<td>Young Modulus of particle (pa)</td>
<td>$\bar{U}$</td>
<td>Magnitude gas velocity (flue gas velocity) (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$E_s$</td>
<td>Young Modulus of surface (pa)</td>
<td>$u_{in}$</td>
<td>Inertial impaction velocity (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$F$</td>
<td>Weighting factor</td>
<td>$u_g$</td>
<td>Flue gas velocity (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$F_{th}$</td>
<td>Thermophoretic force (N)</td>
<td>$u_p$</td>
<td>Ash particle velocity (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$f_p$</td>
<td>Probability of impacting and sticking the π side of probe surface in eddy impaction</td>
<td>$u_{p,crit}$</td>
<td>Particle critical velocity (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>Gravity (m·s⁻²)</td>
<td>$u_r$</td>
<td>Maximum reverse flue gas velocity (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$h_g$</td>
<td>Forced convective heat transfer coefficient (W·m⁻²·K⁻¹)</td>
<td>$u_{rms}$</td>
<td>Root-mean-square speed (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$I_{cond}$</td>
<td>Mass flux of vapor condensation (kg·m⁻²·s⁻¹)</td>
<td>$u_{c,crit}$</td>
<td>Critical velocity of surface capture (m·s⁻¹)</td>
<td></td>
</tr>
<tr>
<td>$I_{eddy}$</td>
<td>Mass flux of eddy impaction (kg·m⁻²·s⁻¹)</td>
<td>$u_{th}$</td>
<td>Thermophoretic velocity (m·s⁻¹)</td>
<td></td>
</tr>
</tbody>
</table>
KESM  Kinetic energy stickiness model  \( V_l \)  Volume of liquid phase of ash deposit (m³)

\( Kn \)  Knudsen number  \( V_s \)  Volume of solid phase of ash deposit (m³)

\( k_e \)  Lumped empirical parameter  \( \chi_{\text{ash}} \)  Ash mass fraction

\( k_{\text{deposit}} \)  Thermal conductivity of ash deposit (W·m⁻¹·K⁻¹)

\( k_g \)  Thermal conductivity of flue gas (W·m⁻¹·K⁻¹)

\( \eta_{\text{stick}} \)  Sticking efficiency

\( k_{g-N_2} \)  Thermal conductivity of nitrogen gas (W·m⁻¹·K⁻¹)

\( \eta_{\text{imp}} \)  Impaction efficiency

\( k_p \)  Thermal conductivity of ash particle (W·m⁻¹·K⁻¹)

\( \varepsilon_{\text{deposit}} \)  Ash deposit porosity

\( k_s \)  Thermal conductivity of solid phase (W·m⁻¹·K⁻¹)

\( \varepsilon_0 \)  Initial ash deposit porosity

\( L_{\text{deposit}} \)  Ash deposit thickness (m)

\( \psi \)  Stokes correction factor

MFSM  Melt fraction stickiness model  \( \rho_p \)  Ash particle density (kg·m⁻³)

\( n \)  Empirical number  \( \rho_g \)  Flue gas density (kg·m⁻³)

\( \overline{Nu} \)  Nusselt number  \( \mu_g \)  Flue gas viscosity (kg·m⁻¹·s⁻¹)

PSD  Particle Size Distribution  \( \mu_s \)  Ash deposit surface viscosity (kg·m⁻¹·s⁻¹)

\( P(T_p) \)  Sticking probability of the impacting particles at the particle temperature for melt fraction stickiness model  \( \Lambda \)  Thermal conductivity ratio of ash particle over flue gas (\( \Lambda = k_p/k_g \))

\( P(T_s) \)  Sticking probability of the ash deposit at the deposit surface temperature for melt fraction stickiness model  \( \phi \)  Coefficient of thermophoretic force

\( P_p \)  Sticking probability of the impacting particle for Kinetic energy stickiness model  \( \lambda \)  Mean free path length (m)

\( P_s \)  Sticking probability caused by the surface for Kinetic energy stickiness model  \( \sigma_r \)  Stefan-Boltzmann constant
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_v(T_g) )</td>
<td>Partial vapor pressure of the condensing components (Pa)</td>
</tr>
<tr>
<td>( p_{v,s}(T_s) )</td>
<td>Saturation vapor pressure at the deposit surface temperature (Pa)</td>
</tr>
<tr>
<td>( Re )</td>
<td>Reynolds number around the probe</td>
</tr>
<tr>
<td>( Re_p )</td>
<td>Ash particle Reynolds number</td>
</tr>
<tr>
<td>( R_g )</td>
<td>Universal gas constant (( J \cdot kg^{-1} \cdot K^{-1} ))</td>
</tr>
<tr>
<td>( Sc )</td>
<td>Schmitt number</td>
</tr>
<tr>
<td>( Sh )</td>
<td>Sherwood number</td>
</tr>
<tr>
<td>( Sr )</td>
<td>Strouhal number</td>
</tr>
<tr>
<td>( St )</td>
<td>Stokes number</td>
</tr>
<tr>
<td>( \alpha_r )</td>
<td>Absorbance thermal radiation</td>
</tr>
<tr>
<td>( \epsilon_r )</td>
<td>Emittance thermal radiation</td>
</tr>
<tr>
<td>( \pi )</td>
<td>Pi number</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Surface tension of ash deposit (( N \cdot m^{-1} ))</td>
</tr>
<tr>
<td>( \theta )</td>
<td>Angle of impacting ash particle</td>
</tr>
<tr>
<td>( \nu_p )</td>
<td>Poisson’s ratio of particle</td>
</tr>
<tr>
<td>( \nu_s )</td>
<td>Poisson’s ratio of surface</td>
</tr>
<tr>
<td>( \dot{m}_{fuel} )</td>
<td>Fuel rate; defined as the ratio of the mass of fuel to time</td>
</tr>
<tr>
<td>( \dot{m}_g )</td>
<td>Flue gas flow rate; defined as the ratio of the mass of flue gas to time</td>
</tr>
</tbody>
</table>
1 Introduction

Woody biomass co-firing with coal in utility boilers is an attractive solution to both reduce greenhouse gas emissions and mitigate wildland fire hazard in our national forests. There are two factors that make woody biomass attractive for cofiring with coal: woody biomass is a CO₂ neutral and renewable material and standing dead trees are an indicator of unhealthy forest and increase the length and severity of wildfires. As biomass grows, it consumes similar amounts of CO₂ from the atmosphere as it produces during combustion. Using biomass to replace coal as fuel will offset the CO₂ emission from coal combustion, rendering woody biomass a CO₂ neutral and renewable fuel [1]. It is known that biomass is a relatively plentiful and a significant source of energy for mitigating and alleviating greenhouse gas emissions [2-8]. Its cofiring with coal in existing coal utility boilers leverages both of these topics: reducing the net CO₂ emission and the risk of wildfire [4].

Understanding the basic process and hardware producing electricity from coal is necessary in order to discuss the impacts of adding woody biomass to the fuel mixture. In a typical coal-fired power plant, as presented in Figure 1-1, coal is combusted to produce steam in order to drive a turbine and generate electricity. Coal supply is sent to a building where equipment is used to crush the coal (the “crusher house”), reducing its size from the “as received” distribution, usually 4” minus, down to approximately ½” minus (Label A in Figure 1-1). The crushed coal, which is stored in silos (Label B in Figure 1-1), is fed to the pulverizer (or mill), where it is again reduced in size from approximately ½” minus to a chalk dust consistency with a mean particle size of
approximately 50 microns. The pulverizer is labeled “C” in Figure 1-1. Hot air is also introduced into the mill in order to dry the coal and convey the pulverized material to the boiler (Label D in Figure 1-1). The entrained coal particles from the pulverizer are injected into the boiler through several burners. The boiler is generally in the shape of an upside down “U” and the yellow ovals in the boiler of Figure 1-1 designate flames attached to burners. Heat from the combustion process transfers through radiation and convection to the walls of the boiler where water is converted to steam. Upon combustion, the coal forms flue gas, which is a gaseous byproduct with the major components being N₂, CO₂, water and residual oxygen. The flue gas contains particulate or ash (called fly ash), which is composed of the non-combustible components from the coal.

Figure 1-1: Coal-fired utility power plant schematic [9].

There are challenges in using a blend of woody biomass and coal in feeding utility boilers, especially when changes to equipment designed for pure coal operation are to be avoided. Such challenges include biomass material handling (ranging from collection to storage), milling, heat
release profile in the boiler, change in ash behavior and deposition, maintenance of the boiler and operation costs and changes to boiler efficiency and capacity. Milling of the blended fuel (co-milling) and deposition on the heat exchanger surfaces of the boiler (ash deposition) are among the most critical challenges.

As discussed above, milling the fuel particles, which is the blend of biomass and coal in this study, before feeding the boiler through burners is necessary. Generally, woody biomass particles are much larger than that of coal particles and they cannot be reduced to the same size as coal particles [10]. One of the key technological hurdles is the behavior of the mill when operating on a blend of woody biomass and coal. Prior studies have shown that unprepared woody biomass will build up over time in a coal mill, increase the power requirement, and eventually plug the mill [11]. This study evaluates the impacts of prepared (thermally treated) woody biomass on mill performance.

Ash deposit formation is a major consideration in boiler performance and occurs in both the radiative and convective selections of the boiler [10]. Ash deposition on the heat exchanger surfaces is attributed to problems such as premature shutdown, low efficiency, and high corrosion [12-14]. Previous experiments show that cofiring of woody biomass with coal changes the ash deposition behavior including ash formation mechanism, tenacity, and its deposit location significantly [15]. Hence, studying ash deposition mechanisms and developing a predictive model for deposition rates will help facilitate coal replacement with biomass in utility boilers.

The objective of this thesis is to develop data sets and models that help facilitate cofiring of woody biomass and coal in existing coal-fired boilers without hardware modifications. Specifically, we should understand the behavior of co-milling in a pulverizer, develop a data set of mineral matter transformation and ash deposition on heat transfer surfaces; and to develop and
validate tools to predict ash deposit rate while cofiring woody biomass and coal. In order to achieve these objectives a computation fluid mechanics (CFD) model was used to simulate the behavior of entrained particles in a pulverizer and its relation to product particle size distribution (Chapter 4). A controlled-temperature ash deposit probe and an ash aerosol probe were designed and constructed which were to be applied at multiple combustion scales in order to collect and measure entrained particles and ash deposit on the coupon surface of the probe (Chapter 5). Ash deposit and ash aerosol generated during the cofiring of woody biomass with coal versus pure coal combustion were compared in a 1.5 MW combustor (Chapter 6) and a 500 MW utility boiler (Chapter 7). A mechanistic model was developed and validated to predict the ash deposit formation on the coupon surface and to determine the dominant ash deposit mechanism (Chapter 8). In addition, our developed model predicts the ash deposit rate of the collected ash deposit samples from the 1.5 MW experiments (Chapter 9). The summary and conclusions from this work are then presented (Chapter 10). The recommendations for the future research are provided in Chapter 11.
2 Literature survey

2.1 Background

Existing coal power plants, which generate electricity, lead to global warming. Gas emissions of coal combustion such as oxides of sulfur (SO\textsubscript{x}) and nitrogen (NO\textsubscript{x}) are important issues, which lead to acid rain and ozone depletion [3]. More importantly, coal is the second major source of electricity generation, after natural gas, in the United States, and coal demand is likely to remain an important source of energy for the next few decades [16]. Cofiring woody biomass and coal has attracted a great deal of attention among the low-cost techniques and alternatives to reduce gaseous emissions of NO\textsubscript{x}, SO\textsubscript{2}, and CO\textsubscript{2} [3]. Combusting wood in a coal combustor is considered a carbon neutral process because the CO\textsubscript{2} emitted is offset by CO\textsubscript{2} consumption during the wood growing cycle [17]. In addition, sulfur content of biomass is less than most types of coal [18]. Therefore, biomass cofiring is widely known as a useful technology to help reduce oxides of sulfur (SO\textsubscript{x}) and sometimes nitrogen (NO\textsubscript{x}) emissions due to its lower combustion temperature and larger devolatilization compared to pure coal combustion [18-20].

Woody biomass properties differ from coal in some important ways. Woody biomass includes trees (e.g., removed or “thinned” from forests to reduce fire hazard or stimulate growth of remaining stands), forest residues (e.g., limbs, tree tops, and other materials generally left on-site after logging), fast-growing tree species cultivated in plantation-like settings, and mill or plugging plant residues [21]. Woody biomass is an attractive energy source because of its widespread availability [21]. In addition, removing forest residues could reduce the threat of
catastrophic wildfires, and it can provide feedstock for energy generation [21]. Therefore, eliminating unhealthy trees and clearing areas for fresh growth are important for forest health.

The economics of cofiring biomass with coal are important and have been addressed in recent years [22-24]. The electricity cost from cofiring is strongly dependent upon the feedstock's origin, type, composition, and cofiring technology [25]. In addition, the cost of handling, preparing, transporting the feedstock relative to the plant's geographic location are very important factors [26]. Cofiring capital cost for installation in many power plants is around $50/(kW of installed capacity) for cyclone boilers, and range from $150–$300/(kW of installed capacity) for pulverized coal boilers due to upgrades for fuel handling and preparation equipment [27]. Leveraging existing infrastructure of a coal power plant helps to reduce the capital costs of biomass utilization by minimizing necessary modifications [4, 18]. However, operating costs of a cofiring power plant are generally higher compared with those of coal power plants [4, 28]. Fuel preparation and transportation cost are the greatest contributor to high operating costs, even when the biomass is free at the production point [28]. Even with all these expenses, biomass cofiring can be cheaper than other renewable alternatives, and it is widely accessible [28].

2.2 Techniques for biomass preprocessing

Utilizing wood as a coal replacement requires preprocessing to make the chemical and physical properties of the biomass more coal-like. One of the major differences between wood and coal is that woody biomass has a lower bulk density and heating value and a higher volatile yield and moisture content in comparison with coal [29]. Some researchers show that the non-friable behavior of biomass leads to difficulties in grinding and pulverization [8]. In addition, biomass particles have a wide range of sizes and aspect ratios that make the material very difficult to handle.
and feed. Therefore, pretreatment processes such as torrefaction, pelletization, and steam explosion can substantially improve biomass milling and combustion characteristics [30]. In general, these processes improve the grindability of woody biomass and result in more coal-like heating values [8].

Torrefaction is a thermal pretreatment process that occurs externally heated retort reactor, or rotary kiln to make biomass more like coal so that it is easy to grind. Torrefaction removes moisture, some of the volatiles and oxygen from biomass and breaks down the fibrous hemicellulose fraction to produce a more grindable and energy dense fuel [31-34]. Torrefied wood can subsequently be compressed to pellets to facilitate transport and reduce storage costs [35]. Pellets combust similar to coal; but their heating value is less than that of coal and their handling is more expensive and difficult. Nevertheless, pellets make a good coal replacement and provides the valuable opportunity to generate clean energy without a costly conversion.

Figure 2-1 shows a schematic of a general mass and energy balance on torrefaction [36]. Approximately 70% of the mass remains in a solid form that contains 90% of the initial amount of energy. Therefore, 30% of the initial mass turning into torrefaction gases contain around 10% of the initial energy content of the biomass. As a result, torrefaction keeps the majority of the initial energy content of the biomass, and this process improves the fuel properties [36].

Steam explosion (SE) is a widely used, low-cost and efficient pretreatment technology for wood; and steam-exploded woody biomass does not contribute significantly to emission because its pollutant elements such as Sulphur are removed in the process [37, 38]. This process exposes the biomass to high pressures using saturated steam in a pressure vessel. Once the pressure inside the wood cells is equilibrated, the vessel pressure is suddenly released to burst the cell wall and break down the material structure [39]. After steam explosion, biomass is removed, rinsed, dried
and pelletized [40]. The main advantages of this method include: “the increase of accessible surface area; higher substrate digestibility; depolymerization of lignin, and solubilization of hemicellulose” [41].

![Diagram of torrefaction process]

**Figure 2-1: Typical mass and energy balance of the torrefaction process. E=energy, M=Mass [36].**

Pelletization densifies and physically compacts woody biomass into hydrophobic and low-eroding pellets [30]. Such a process is one of the most common densification technologies for woody biomass to increase the bulk density of the raw material [42, 43]. P. Gilbert et al. [44] studied pelletization methods for different switchgrass and they evaluated the effects of pressure and temperature on the quality of pellets in terms of density, mechanical strength and durability. They found that the effect of temperature is more significant than pressure on pellet quality. Further studies are needed for improving pelleting technology for biomass, however it is outside the scope of this research [45].

### 2.3 Milling

A milling system pulverizes dry coal before it is transmitted to the boiler. Coal with a size distribution of 2” or less is continuously loaded into a bowl or onto a plate, in the most commonly used mills. A diagram detailing the components of a mill is included in Figure 2-2. The as received
coal is loaded into the mill, using a gravimetric coal feeder. Upon entering the mill, the coal falls into a large steel bowl which is mounted to a drive system causing it to rotate. Large steel rollers, or rolls, mounted on arms, press against the bowl wall. As the bowl rotates, the rolls crush the coal against the bowl wall. Simultaneously, warm air, swirls through a tangential port below the bowl and entrains coal as it rises and exits through the classifier at the top of the mill. Coal particles remain in the bowl until they become small enough to become entrained in the air.

![Figure 2-2: Schematic side view of the physical milling system.](image)

However, co-milling of woody biomass and coal can cause some complications, which are avoided by proper pretreatment processes. These include buildup of un-milled pieces around the rolls, which leads to high power requirements and a potential clog. The change of the particle size distribution of the fine particles is another issue that is caused by co-milling.

The particle size distribution of fuel in entrained combustion systems is an important parameter relating to combustion behavior. Large coal particles (larger than 50 mesh, which is equivalent to 297 microns in diameter) generally lead to incomplete burnout. Kinetics for burnout
of large wood particles are much faster than coal particles, due to volatile yield, porosity and density. However, reducing the fuel to smaller particle sizes is desirable for combustion performance. Some studies have shown that reducing the size of woody biomass particles to the small sizes of coal particles is difficult due to the different physical properties of biomass from coal (density and aspect ratio) [19]. Usually the result is that large biomass particles exit the mill early, with a larger particle size distribution.

Classifiers help prevent coarse particles from exiting the milling system alongside the fine particles and they are widely used in industry coupled with the mill [46-48]. There are two major types of industrial classifiers, as shown in Figure 2-3: static and dynamic. The majority of power plants use static classifiers, which are less efficient than dynamic classifiers, leaving significant room for modifications to improve performance [47]. Dynamic classifiers utilize rotating vane blades to control the cut size, whereas adjustable and stationary vanes control the cut size in static classifiers [47]. The work in this research deals with a static classifier.

Figure 2-3: Two commercial classifiers: (a) static classifier (b) dynamic classifier [47]
Some important parameters such as blade angle affect the classifier performance [47]. Blades, which are also called vanes, induce tangential velocity to gases entering the classifier. Generally, classifier performance and efficiency are determined by three criteria: cut size, sharpness of cut, and grade efficiency. The aforementioned parameters are defined to compare efficiencies of the steady state classifying processes. Changing to a more aggressive vane angle increases the pressure drop across the classifier, causing it to produce higher grade efficiencies, and higher classification sharpness with lower cut size [47, 49].

2.4 Ash content and ash composition of coal and woody biomass fuels

Ash content (or ash yield) is the quantity of the incombustible inorganic residue resulting from complete combustion and is usually measured as part of a proximate analysis [50-52]. Ash content and composition, which can be determined by proximate analysis and ash elemental analysis, varies from fuel to fuel [53]. In general, the amount of ash content for woody biomass is much less than that for coal [54]. The ash content in woody biomass, which mostly stems from the materials that the plant uses during its growth, such as water and soil, is about the lowest among the biomass fuels [55, 56]. Coal ash content includes the components of six main oxides (Al₂O₃, SiO₂, K₂O, CaO, Fe₂O₃, and TiO₂), which are detected in an ash chemistry analysis, commonly using x-ray fluorescence or inductively coupled plasma mass spectrometry (ICP MS) [53]. Table 2-1 shows the chemical ash composition of typical woody biomass and bituminous coal in the oxide form, which is not necessarily its form in the parent fuel [52]. The representative proximate and ultimate analyses of wood and coal are presented in Table 2-2.

Ash content of wood residues, including branches, bark, etc., is generally 3.0 wt.%, which is higher than that of heartwood of beetle-killed trees at 0.1-0.5 wt.% ash. The reference that is
provided in Table 2-2 likely refers to heartwood. In any case, heartwood ash contents are less than that of bituminous coal, which is about 8-15.7 wt.%.

**Table 2-1: Ash elemental analysis of woody biomass and bituminous coal [52].**

<table>
<thead>
<tr>
<th>Element</th>
<th>Mean value of Wood and Woody Biomass (wt. %)</th>
<th>Mean value of Bituminous Coal (wt. %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiO₂</td>
<td>2.22</td>
<td>56.14</td>
</tr>
<tr>
<td>CaO</td>
<td>43.03</td>
<td>4.90</td>
</tr>
<tr>
<td>K₂O</td>
<td>10.75</td>
<td>1.61</td>
</tr>
<tr>
<td>P₂O₅</td>
<td>3.48</td>
<td>0.22</td>
</tr>
<tr>
<td>AL₂O₃</td>
<td>5.09</td>
<td>24.82</td>
</tr>
<tr>
<td>MgO</td>
<td>6.07</td>
<td>1.55</td>
</tr>
<tr>
<td>Fe₂O₃</td>
<td>3.44</td>
<td>6.68</td>
</tr>
<tr>
<td>SO₃</td>
<td>2.78</td>
<td>2.16</td>
</tr>
<tr>
<td>Na₂O</td>
<td>2.85</td>
<td>0.77</td>
</tr>
<tr>
<td>TiO₂</td>
<td>0.29</td>
<td>1.15</td>
</tr>
</tbody>
</table>

**Table 2-2: Proximate analysis and ultimate analysis of wood and coal, reported in reference [56].**

<table>
<thead>
<tr>
<th>Fuel</th>
<th>Proximate analysis (wt.% by dry basis)</th>
<th>Ultimate analysis (wt.% by dry, ash free basis)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fixed Carbon</td>
<td>Volatile Matter</td>
</tr>
<tr>
<td>Wood</td>
<td>15.7-21.4</td>
<td>78.1-84.1</td>
</tr>
<tr>
<td>Coal</td>
<td>20.0-71.8</td>
<td>12.4-51.8</td>
</tr>
</tbody>
</table>

Biomass fuels can be categorized in three groups in terms of their ash composition [57]: (I) Rich in Ca, K but lean in Si, (II) Rich in Si but lean in Ca, K, and (III) Rich in Ca, K, P. Woody biomass, which is used in this PhD study, belongs to group (I) and it has low content of nitrogen, sulfur, and ash, compared to many fuels especially coal [26, 57]. Lower sulfur content in biomass compared to coal leads to a reduction of SO₃ emissions [58]. In addition, low nitrogen content may lead to low NOₓ emission. Meanwhile, NOₓ formation has a very complex dependency on the nitrogen content [59].

Elements that are especially important for deposit behavior can vary significantly between coal and Woody biomass. Although woody biomass generally has lower ash content compared to coal, it contains higher percentages of alkaline and alkali elements on ash basis, but not necessarily on a fuel basis, which determines the amount of such material flowing through a boiler [56, 60].
The main alkali metal in biomass is potassium whereas sodium is the dominant alkali species in coal [14, 61]. The amount of potassium, calcium, and chlorine in biomass is more than that in coal; the interaction between such elements and the coal minerals may lead to deposits which are sticky to the heat exchanger surfaces, causing corrosion and ash deposit on heat exchanger surface [58, 62-64]. Chlorine, which is a main element in ash formation, transports the alkali-containing compounds, particularly potassium, to make potassium chloride stick to the surfaces. Potassium chloride usually reacts with sulfur oxides to form potassium sulfate and causes severe deposit formations [14, 62].

2.5 Ash transformations

Measuring aerosol formation (particles smaller than 1.0 micron) during cofiring of woody biomass with coal is not in the scope of this research (performed by University of Utah) and the procedure of its measurements is presented in the work of Z. Zhan et. al. [13]. However, presenting a brief literature review about such an issue is necessary. Since ash aerosol is a precursor of deposition, the investigation of ash partitioning mechanisms while cofiring biomass with coal is important.

Inorganic materials in pulverized coal can be characterized as being included, excluded, or organically bound. Included and excluded minerals exist in mineral form, but some appear imbedded in the coal particle (included) and others are extraneous (excluded). By contrast, organically bound material is part of the organic matrix and is not in mineral form as shown in Figure 2-4 [65]. Biomass inorganic material always includes organic materials and some included minerals. The excluded minerals of biomass, which are a lower fraction than that of coals, usually stem from soil contamination during the harvest [66].
Generally speaking, the inorganic components of the fuel convert to gas species, submicron aerosol particles, and coarse particles during combustion [68-70], which each contributes to deposition through different mechanisms. Figure 2-5 summarizes the main processes of ash aerosol formation that can be classified into two main categories: solid-to-particle and solid-vapor-particles processes [71]. Ash formation in biomass combustion is complex because it consists of homogenous and heterogeneous mechanisms to form ash aerosol in gas-to-particle conversion [72]. In heterogeneous nucleation, the vaporized ash condenses onto a surface of the same species, such as an ash particle or onto a surface of a foreign species such as heat exchanger surface. However, homogenous nucleation forms a new phase as shown in Figure 2-6. If the saturation ratio, which is the ratio of the partial pressure of the condensing gas to its vapor pressure, is much larger than one, homogeneous condensation dominates. If the saturation ratio is only a little larger than one, heterogeneous condensation occurs [60]. When the molten particles contact each other they may coalesce, forming a united particle [71]. However, particle fragmentation reduces the effects of mineral coalescence [73].
Figure 2-5: Main processes in ash aerosol formation in biomass combustion [74].

Figure 2-6: a) Heterogeneous nucleation on the external surface; b) Homogeneous nucleation to form a particle [60].

Reaction of elements of coal and biomass during cofiring can exacerbate ash deposition. Sulfate compounds, which are higher in ultimate analysis of coal than woody biomass [56], have higher melting temperature [75, 76] and it leads to be solid at the same temperature that chlorine compounds might be melty and sticky. Higher concentration of potassium of woody biomass [52] decreases the melting point temperature of ash aerosol particles, increases the melt fraction, and enhances the stickiness of ash particles to the surface [77]. The concentration of KCl may have a dominant influence on deposit formation [78]. However, the existence of sulfur may lead to
different results. Potassium is the main alkali metal in most biomass solid fuels whereas sodium is the dominant alkali species in coal [14, 61, 79]. Acid components, \( \text{Al}_2\text{O}_3 \) and \( \text{SiO}_2 \), may capture and react with Na and K vapor species and decrease the condensation potential [80]. Therefore, aluminosilicate (\( \text{Al}_x\text{Si}_y\text{O}_z \)) can retain the majority of K at ash [79]. Vaporized alkali metals from biomass such as potassium may react with coal off gases like sulfur to chemically convert to a different species (\( \text{K}_2\text{SO}_4 \)), which is a solid at the temperature of the particle prior its impact on a surface. This differs from high-rank coal combustion, where alkali concentrations are typically low, and from biomass combustion, where sulfur content is low. Low-rank coals contain substantial amounts of Na in the form of organically distributed material.

### 2.6 Ash deposition behavior

Ash deposition on surface is known in two major forms: slagging (deposition of liquid-like ash to the boiler walls in radiation zone) and fouling (deposition of solid ash particles to heat exchanger tubes in convective zone) [81], which is a function of the flue gas and particle temperatures at different locations in the boiler. The focus of this research is on fouling. Ash deposition in the form of fouling generally consists of two major layers, the sticky inner layer and the loose outer layer [82]. The inner ash deposition layer is mostly formed by the combination of chemical reactions, condensation, and thermophoresis, and the outer layer is formed by the inertial impaction of larger particles [67]. However, this observation cannot be inferred in all cases of ash deposition because it may only be valid at some specific conditions. Although some researchers have investigated the ash deposition mechanism, this phenomenon is not well understood. Further investigation is necessary for prediction of ash deposit formation rate, size distribution, and composition when cofiring biomass and coal [83].
A combination of five mechanisms are possible for ash deposition, which include inertial impaction, thermophoresis, condensation, chemical reaction, and eddy impaction [14, 51, 67, 84-86]. Since ash deposition accumulates over time, the dominant deposition mechanism may change with time [87]. The followings will provide a detailed description of each of these mechanisms.

(I) Inertial impaction: Particles above 10 µm have enough force to go through the flue gas boundary layer and may contact with the heat exchanger surface [85, 86]. However, particles that make contact with the surface may or may not stick [67]. The sticking efficiency, which is defined as the ratio of particles that stick to the surface to the total particles that hit at the surface, depends upon the geometry of the heat exchanger surface, particle size, density, impaction angle, and gas flow properties, as well as the particle viscosity [85, 88-91]. The impaction efficiency, which is the ratio of the number of particles that impact the heat exchanger surface to the number of particles directed to the surface by flue gas, is a function of the Stokes number, which is given by [51, 85, 92]:

$$St \equiv \frac{\rho d_P^2 u_P}{9 \mu g d_c \psi},$$  \hspace{1cm} (2-1)$$

where:

$$\rho_P, d_P, \text{ and } u_P \text{ represent particle density, diameter, and mean velocity, respectively;}$$

$$\mu_g \text{ and } d_c \text{ represent gas viscosity and tube diameter, respectively;}$$

and $\psi$ is a correction factor that is only important when the particles do not obey Stokes law [51]. Generally, an increase in Stokes number leads to an increase in impaction efficiency because the particle is not likely to follow the fluid around the surface and instead impacts the surface.
(II) Thermophoresis: The suffix -phoresis means migration, so thermophoresis is the phenomenon of subjecting movable particles to a force induced by a temperature gradient; the resulting force is called a thermophoretic force [93, 94]. The suspended particles in flue gas with a high temperature gradient experience a net force that is dominated by gas molecules from the hot side of the gradient because they have higher kinetic energy [85, 95]. The temperature gradient decreases with growing ash deposition layer on the heat exchanger surface, reducing the thermophoretic force with time [85]. Experimental results show that thermophoresis is the major ash deposit mechanism where the size of fly ash particles range between 0.5-5 µm [86]. The thermophoretic force depends upon the Knudsen number, particle diameter, and the material properties [85, 96]. In the past, some researchers have investigated thermophoretic force in both laminar and turbulent conditions [94-97].

(III) Condensation: This is the mechanism of condensing vapors on the cold surfaces whose temperature is at the dew point of species in the flue gas [85]. Condensation is a minor participant to deposit formation for bituminous coal, whereas it is an important mechanism for biomass combustion because of more volatile inorganic matter in biomass [85]. Condensation occurs mostly at the early stages of ash deposition on the heat exchanger surfaces [92]. X. Wang et al. [98] investigated the ash deposition mechanism of a high sodium and calcium concentration coal in a full-scale boiler. They showed that condensation of sodium and calcium sulfates is significant for fouling to occur. The vaporized potassium may condense on cooled tube surfaces, leading to the creation of sticky surfaces that accelerate the rate of ash deposition [99]. P. M. Walsh et al. [91] conducted research on the ash deposition resulting from firing lignitic coal. They showed that sodium sulfate condenses on suspended particles as well as the heat exchanger surface when the gas has cooled to the sodium sulfate dew point. However, when the gas temperature is below the
dew point, the sodium sulfate condenses mostly on the particles instead of the tube because of the high mass transfer coefficient and surface area of the particles.

(IV) **Chemical reaction:** Heterogeneous chemical reactions occur mostly between gases and the materials in the deposit [85]. Sulfation, alkali absorption, and oxidation on the tube heat exchanger surface are the most important chemical reactions that lead to ash deposition or the growth of existing deposits [67, 100]. The mechanism of Chemical reaction is not investigated in this research.

(V) **Eddy impaction:** In addition to the above mechanisms, eddy impaction can be an ash deposition mechanism by which eddies provide enough momentum to the fine ash particles to impact tube surfaces, although they are too small to impact based on the average gas velocity [67, 100]. Eddy impaction can change the location where deposits occur. Often the result is that deposition occurs on the “downwind” side of a tube due to eddy impaction. M. Li [101] presented an analytical eddy impaction model which deals with ideal, isothermal flows and monodispersed particles to show that eddy impaction is the only deposition mechanism for these conditions. The mechanism of eddy impaction can be understood well by large-eddy simulation (LES), which considers the turbulent energy, momentum transfer, and turbulent mixing [102].

### 2.7 Measuring ash deposition rates and properties

This section presents a short summary of ash deposition probes. An ash deposit probe is necessary to provide a surface to collect the deposited ash for analyzing its composition and formation rate. Generally, deposit probes are made of stainless steel or ceramic and are cooled using air or water to control the surface temperature, which represents the surface temperature of a heat exchanger.
M. S. Bashir et al. [103] investigated the probe exposure time and its surface temperature on ash deposit rate in a full-scale boiler of 350 MW, which is fed by straw and wood. They designed a 3 m-long probe that hung on a flange-connected hinge and a load cell was used to calculate the deposit mass. In addition, a CCD (charge-coupled device) camera was applied to record ash behavior on the surface, especially during natural and soot blowing deposit shedding. Each measurement took between 2 and 18 days. As shown in Figure 2-7, S. Zheng et al. [104, 105] used an ash deposition probe in a 660 MW furnace to collect ash deposits in order to develop a model for fly ash deposition. They used an S-type thermocouple to measure the temperature of different positions of the probe. Although their probe is easily installed, its deposition-sampling surface is only a vertical surface without a temperature control system, which is not applicable for fouling deposit collection of this thesis.

![Figure 2-7: (a) Picture of the ash deposition probe; (b) The details of the probe [104].](image)

L. Baxter et al. [14] conducted a comprehensive investigation on the ash deposition in biomass-fired boilers. They constructed a continuously rotating horizontal probe made out of stainless steel, which is located across the flue gas exit of a pilot-scale reactor to get the emission spectra from the deposit for Fourier Transform Infrared (FTIR) spectroscopy analysis. In addition,
a stationary probe was installed alongside the rotating probe to collect extra deposit. It is a good idea to use a rotating probe because it is accurate in generating axisymmetric deposits. X. Jin et al. [106] inserted a vertical condensation probe at the bottom of the combustor to study the condensation behaviors of potassium salts formed from biomass combustion. Its vertical position avoided inertial impaction of the ash deposit on the probe surface.

Y. Shao et al. [107] studied the ash deposition behavior of three-fuel blends in a pilot-scale reactor. As shown in Figure 2-8, they made an ash deposition probe of stainless steel 316L, which maintains its surface temperature using cooling air. Their probe design is typical of many air-cooled deposition probes regardless of its position in either vertical or horizontal. H. Zhou et al. [108] presented a technique to measure the effective heat conductivity of ash deposit of a high sodium coal in a pilot-scale furnace. They used two K-type thermocouples to measure the outer and inner surface temperature of the ash deposition probe and an oil circulating system at the constant temperature of 503 K. In addition, they applied a CCD monitoring system to monitor the ash deposition growth on the probe surface [109]. CCD connects to the image processing system to calculate the deposit thickness based on the edge detection.

A.M. Beckmann et al. [110] investigated ash deposition in a pilot scale furnace and assumed that inertial impaction and thermophoresis are the major mechanisms of ash deposition. They showed that the deposition rate on the air-cooled probe is larger than that on the uncooled probe because of the thermophoretic effect. Their research demonstrates that the surface temperature is a very important parameter on ash deposit rate. In addition, controlling surface temperature of the probe contributes to the condensation mechanism that we will show to be very important. Z. Zhan et al. [82] presented a pilot-scale and temperature-controlled ash deposition probe to collect deposits on both its horizontal and vertical surfaces. They showed a higher ash
deposition rate on the vertical surface at lower probe surface temperature because of high thermophoretic force.

![Schematic view of the ash deposition probe used by Y. Shao et al. [107].](image)

**Figure 2-8: Schematic view of the ash deposition probe used by Y. Shao et al. [107].**

### 2.8 Modeling ash deposition rates

This section provides a concise review about the ash deposition models of coal combustion or its cofiring with biomass. Current research generally focuses on using computational fluid dynamics (CFD) calculations, modified with deposition mechanisms to predict the ash deposition rates in both steady state and time resolved simulations [83]. T. J. Taha et al. [111] constructed a model using Ansys-CFX to simulate ash deposition behavior of cofiring of meat and bone meal (MBM) with coal in a tangentially-fired full-scale boiler. They used the FactSage program to perform thermodynamic calculations which predicted the composition of ash particles [112]. They showed that the amount of fouling increases in higher percentages of MBM during cofiring. Meanwhile, they considered only inertial impaction as the main ash deposit mechanism and ignored other mechanisms. A. Leppänen et al. [113] used the Fine Particle Model (FPM) in Fluent
to develop a steady-state model to predict the ash deposition rate and its composition in a full-scale boiler burning dry solid black liquor. The main limitation in their study is that they assumed an average value of ash deposit thickness instead of time dependency. Zheng et al. [104, 105] used direct simulation Monte Carlo (DSMC) to model the fly ash deposition in a coal boiler and calculated the ash deposition rate for both the inner and outer layers using a critical velocity model and a viscosity model. They showed that the inner layer of the deposit consists of fine particles less than 10 µm in diameter. Their model focused on the probe surface instead of whole boiler, so their approach is similar to the current research, although their probe surface includes only the vertical surface (Figure 2-7). In addition, it seems that the effects of eddy turbulence and condensation were not considered in their model. Good reviews on modelling in industrial furnaces are presented elsewhere [114, 115]. X. Yang et al. [83, 84] developed a CFD model using Ansys (Fluent) to predict the ash deposition formation in a lignite combustion using inertial impaction, thermophoresis, and condensation mechanisms. As shown in Figure 2-9, the relative accumulated deposited mass, which is the ratio of the accumulated deposited mass to the total deposited mass, increases gradually over time, although the condensation mechanism increases only in the initial stage. S. S. Lokare et al. [100] developed an ash deposition model in coal-straw co-firing using Fluent and C++ to predict the rates and mechanisms. They assumed that the total deposition rate forms mostly through inertial impaction, condensation, and eddy impaction. However, their ash deposition rate is not time dependent.
2.9 Summary of literature review

Many studies have been conducted about the co-milling and cofiring of biomass and coal during the past decades. However, there is still a significant gap in understanding the issues of co-milling of woody biomass with coal and its ash deposit rate in both pilot and full-scale combustors. This thesis provides deeper understanding in the following areas:

1. The effect of co-milling on the performance of a static classifier pulverizer by elucidating the behavior of the particles.
2. Changes of mineral matter and deposition of ash on the heat exchangers around super heaters and reheaters of utility boilers.
3. Develop a model to predict the ash deposit rate and the dominant mechanisms of ash deposit formation for a variety of fuels.
3 Objective and tasks

3.1 Objective

The objective of this PhD thesis is to develop data sets, models and general understanding that allow cofiring of woody biomass and coal in existing coal-fired boilers without hardware modifications. Decreasing the net emission of CO\(_2\) and reducing the threat of forest fires are the motivating factors for this objective. Partial replacement of coal by woody biomass in existing coal utility boilers is a reasonable solution to reduce net emission of CO\(_2\) and potential threat of wildfires. Meanwhile, simple hardware manipulation of the pulverizers might be significant in order to produce milled fuels like coal. In addition, ash deposition on the heat exchanger surfaces should not be changed significantly by cofiring of small portions of woody biomass with coal while maintaining the same operating parameters. Therefore, the following tasks are defined to address these challenges: “co-milling biomass and coal” and “ash deposit on the heat exchanger surfaces during cofiring”.

3.2 Tasks

In order to fulfill the objective of this thesis, we should:

- understand milled particle behavior in industrial pulverizers during co-milling of woody biomass and coal and the fundamental drivers for that behavior;
develop data sets describing mineral matter transformations and ash deposition on
the heat exchangers around the super heater and re heater (fouling) during cofiring;
and

develop robust but simple numerical models to predict ash deposit rate during
cofiring of fuels with diverse chemistry in order to help operators of utility boilers.

Therefore, the Scope of Work of the presented thesis includes three tasks:

1) Computational Fluid Dynamics (CFD) modeling of the pulverized woody biomass and
ccoal particle inside a milling system. The objective of this task is to understand the differences
between coal and biomass particle behavior in the classifier of an industrial pulverizer, so that
subtle modifications to hardware or changing operating conditions can be considered. This will
result in more coal-like operation for the coal and biomass blend.

- We implemented Barracuda VR 17.1.0, which is the software package developed
  by Computational Particle Fluid Dynamics (CPFD), to simulate the particles
  trajectory in the pulverizer.

- We developed and validated boundary conditions for moving parts (bowl and rolls)
in order to represent realistic particle behavior in a geometrically static model. The
operating conditions of the model include, but are not limited to, the quantity and
location of the particle injectors, the rate and angle of injection, PSD, and pressure
drop of air from inlet to outlet. These operating conditions were shown to result in
proper outcomes that validate against the experimental data, which are the product
stream flowrates and their PSD.
• We developed a C# code to post process the output files from the Barracuda software in order to track individual particle trajectories which was necessary information for understanding biomass vs coal particle behavior.

• We leveraged the validated model and post processing tools to perform parametric studies to demonstrate the effectiveness of simple modifications to geometry and operating conditions in order to obtain a more coal-like PSD for the product fuel.

2) Measuring the ash deposition rate derived from combustion of coal biomass blends in a utility boiler. The objective of this task is to develop a data set of ash behavior during cofiring of woody biomass and coal in a utility boiler. To fulfill this objective, we measured the ash deposition behavior and composition on representative heat exchanger surfaces while firing a 15% biomass, 85% coal blend by mass in a pilot combustor and a full-scale furnace. For these tests biomass was prepared by both torrefaction and by steam explosion.

• We designed and construct a temperature-controlled tube-in-crossflow deposition probe to measure the rate of particle deposit on coupon surfaces. Key design parameters of the probe included: control system to maintain coupon surface temperature at a setpoint representative of the vertical reheater and smooth extraction from the furnace port to maintain deposit integrity.

• We designed and constructed an extractive aerosol particle probe, which would allow the analysis of entrained ash concentration, PSD (from 10 nm to 20 µm) and composition.

• We operated both probes in a pilot combustor (1500 kW_{TH}) and full-scale furnace (Hunter, Unit 3, 500 MW_{e}) to collect, weigh and further analyze the deposit while firing pure coal and the prepared biomass blends.
• We used these data to develop an expectation of the ash behavior when firing woody biomass from the Manti La Sal National Forrest in PacifiCorp’s Hunter, Unit 3 and to perform future model validations.

3) Develop a Model to predict ash deposit rate while firing blends of coal and a variety of biomass with various chemical and physical properties. The objective of this task is to generate a relatively simple model to help the operators of a utility boiler predict ash deposit rate (in a fouling mechanism) based on fuel properties, which does not rely on CFD analysis.

• We performed an exhaustive literature survey of ash deposition mechanisms including: inertial impaction, thermophoresis, condensation, and eddy impaction. In addition, various model formulations and algorithms were reviewed.
• We developed a deposition model relying on algorithms in the literature but incorporating a novel approach in determining sticking efficiency, which results in more accurate ash deposit rate prediction than the previous research.
• We validated our developed model against multiple existing data sets from a 100 kW\textsubscript{TH} down-fired combustor while firing a variety of biomass and coal blends;
• We applied our model using data sets from previous tasks (data of 1500 kW\textsubscript{TH} combustor);
• We made observations concerning deposition mechanisms and fuel properties.
4 Analysis of particle behavior inside the classifier of a Raymond Bowl Mill while co-milling woody biomass with coal

4.1 Introduction

One of the key technological hurdles for cofiring woody biomass with coal is the behavior of the pre-combustion fuel milling equipment when operating on a blend of prepared woody biomass and coal. Therefore, cofiring of woody biomass with coal in electricity generating utility boilers requires either significant hardware upgrades for fuel handling and combustion or a pre-treatment process to convert the biomass into a more “coal-like” fuel. Even though the resulting pre-treated biomass fuel is more “coal-like” there are differences in milled particle density and aspect ratio resulting in asymmetric behavior in the mill classifier. Combustion kinetics of woody biomass and coal are different, and large biomass particles would be expected to burn out in utility boiler conditions, meaning it is not necessary to grind biomass as fine as coal [117]. Large woody biomass particles can still burn out because of much higher volatile yields and lower densities. However, operators of utility boilers are more comfortable with a standard particle size distribution. Therefore, simple operational and hardware configuration changes can make the grind similar to that of pure coal should be pursued.

The objective of this research is to model the behavior of a blend of prepared biomass and coal particles in a Raymond Bowl Mill and to evaluate the expected performance change of the mill classifier when implementing simple hardware modifications. There have been few studies where milling systems have been modeled. Afolabi et al. [47] designed a laboratory scale static classifier and a Computational Fluid Dynamics (CFD) model of that hardware. They showed the importance of using proper cyclone dimensions to allow large particles residence time in the classifier cone where they are captured and returned to the bowl for size reduction. Foster et al. [49] studied modifications to the classifier to reduce the cut size, resulting in a smaller particle size distribution (PSD) from the milled product. Neither of these studies investigated multiple particle types with differing physical properties, representative of cofiring of prepared woody biomass with coal. Because the specific grinding energy can be determined elsewhere [118], we feel it is outside the scope of this manuscript.

4.2 Materials and methods

Coal is typically received from the mine with a size distribution with a large diameter limit of about 2 inches. Before it is combusted, the coal particles must be reduced in size so that 70 mass % of the particles are smaller than 200 mesh (74 microns) with less than 1 mass % larger than 50 mesh (297 microns). This usually results in a mass mean particle size distribution of around 50 microns. A mill is a device that pulverizes coal into the desired size distribution (see Figure 2-2). A strong motor rotates the bowl, which causes the coal to be sandwiched in between the bowl and the rolls. The feed materials are crushed and ground due to the heavy force imposed by the rolls. Simultaneously, warm air, which evaporates the moisture of the particles, comes into the system, swirls around the bowl, and entrains the small-pulverized particles towards the classifier. The entrained particles and the air enter the classifier from the top and swirl inside the classifier. Coarse
particles fall into the bowl through the bottom of the classifier for re-grinding, and the fine particles exit the system through the top of the classifier. These fine particles are then carried to the boiler.

Coarse particles cause problems in the combustion system. They may not have enough residence time to completely combust, reducing combustion efficiency, and they can promote deposition near the burner. Classifiers are a hardware component of the mill which separates large particles from the entrained flow and returns them to the milling system so that they can be reduced in size [46-48]. This objective is achieved using a cyclone. The entrained particles enter the cyclone at the top through adjustable vanes which induce a tangential component to the air velocity. The tangential velocity causes the particles to migrate in an outward radial direction toward the cyclone walls. Larger particles move more quickly due to this centrifugal force and are more likely to impact the surface where they are pulled out of entrainment. The large particles then move down the wall of the cyclone and are reintroduced into the bowl for further milling. The outlet for the air and small particles is at the top of the classifier through a cylindrical tube, or vortex finder, which extends down into the cyclone. The geometry of the vortex finder relative to the cyclone forces the air and entrained particles to make rotations down through the cyclone before entering the vortex finder. This provides enough residence time for large particles to be captured by the surface of the cyclone. The fine particles which remain entrained in the air are carried to the boiler through the burners. The majority of power plants use static classifiers, which are less efficient than dynamic classifiers, leaving significant room for modifications to improve performance [47]. The work in this research deals with a static classifier. A simple diagram indicating the movement of particles through a static classifier is included as Figure 4-1 with: (1) entrained particles move from the bowl toward the cyclone, (2) entering the cyclone through vanes,
(3) exiting the classifier through the vortex finder as the product stream and (4) coarse particles returning to the bowl.

![Figure 4-1: A schematic diagram of a static classifier indicating typical movement of particles.](image)

**4.2.1 Experimental work**

In a study by Zsolt [118], the performance of a CE 312 Raymond Bowl Mill was investigated while milling a Bituminous Coal and blends of prepared woody biomass with Bituminous Coal. Results from this study provide hardware configuration, operational data and performance data for model validation and parametric studies. Of particular interest are the experimentally measured product PSD, as shown in Figure 4-2, and product flowrate (0.189 kg/s) when milling a 15% wt.% blend of torrefied pellets produced at a process temperature of 325 °C with 85% Bituminous Coal. Figure 4-2 indicates that the blended fuel produces a bimodal distribution with the mode centered at about 400 microns unique to the blend when compared with pure coal. It is reasonable to assume that the particles in this mode are dominated by the prepared
biomass. This suggests that physical properties of the biomass, presumably density and aspect ratio, allow larger biomass particles to escape from the classifier than corresponding coal particles.

![Graph showing particle size distribution](image)

Figure 4-2: Beckman Coulter analysis of particle size distribution for coal-woody biomass blend [118].

### 4.2.2 Model description

I produced a computational fluid dynamic (CFD) simulation of the CE 312 Raymond Bowl mill using Computational Particle Fluid Dynamics (CPFD) Barracuda Virtual Reactor 17.1.0 software package. This computational package was developed to represent dense particle phase systems and is based on the Multiphase Particle-in-Cell (MP-PIC) method which uses a particle probability distribution function [119].

Complete engineering drawings were not available for creation of the geometry and mesh. Missing dimensions were scaled from available drawings, assumed, or directly measured. Figure 4-3 shows the Solidworks model of the mill void space as it was imported into Barracuda VR as a
stereolithographic (STL) file and the fully meshed geometry in Barracuda VR with locations of interest identified. It should be noted that I made this Solidwork model.

Since it is not possible to represent the rotation of the bowl and rolls or to represent the mechanical process of particle size reduction in the CFD model, and because experimental data are not available, we assume the location, size distribution, and rate that particles are generated for entrainment near the bowl. These parameters were discussed with experts in the field including engineers at: General Electric (formerly Alstom), Arvos Group (the holder of the Raymond bowl mill technology) and at a utility operating these mills. Reasonable ranges of values were determined for each parameter and their modeled values were varied in order to accurately reproduce the validation data with the model.

Figure 4-3: (a) Translucent representation of meshed model in Barracuda VR, (b) SolidWorks model of void space.
Locations for the model inlet and outlet boundaries have been identified in Figure 4-3 as:

1) Warm air inlet, 2) milled particle injection locations and 3) milled product and air outlet. The conditions at these boundaries are summarized in Table 4-1.

<table>
<thead>
<tr>
<th>Boundary</th>
<th>Description</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Temperature (K)</td>
<td>394</td>
<td>366</td>
<td>338</td>
</tr>
<tr>
<td></td>
<td>Pressure (kPa)</td>
<td>95</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flow (m/s)</td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The main assumptions considered in the Barracuda base simulation are:

- The physical process of particle size reduction occurring in the bowl cannot be represented. Instead, the resulting size distribution of ground particles (both coal and wood) and the location of their injection into the model are assumed. The size distribution is a parameter that was adjusted in order to match the product size distribution to measured values. The injection locations were placed along the rim of the bowl, concentrated near the rolls.

- In the mill, large particles collected by the classifier are recycled to the bowl for further size reduction. In the model, large particles captured by the classifier are removed from the simulation by capturing them in a cavity at the bottom of the classifier cyclone. The assumed ground particle size distribution and rate of injection account for these recycled particles. The injection rate is the sum of the rates of milled product, particles captured by the classifier and particles captured at
other locations in the system. The injection rate is a parameter adjusted until the rate of milled product matches the value measured in experiment.

- The recycle ratio (RR) is defined as the ratio of the flowrate of particles entering the classifier to the flowrate of the product stream. Experts tell us that RR should be close to 4, although this is a value that has not been reportedly measured. The RR can also be influenced by the assumed PSD of ground particles.

- Values of unknown or unmeasured operating conditions are assumed and adjusted within a reasonable range of values until the behavior of the model closely matches observed experimental behavior. For example, pressure at the inlet and the outlet of the mill have been assumed.

- Experimental PSD is volume base whereas the generated PSD by Barracuda is in terms of mass. Particle density and spherical particles have been used to convert mass-based PSD to volume-based PSD for comparison.

- The thickness of pulverizer surfaces has been slightly altered in the model to reduce stair stepping caused by meshing.

In order to tune the baseline model to match experimental data the following model parameters were adjusted: the PSD of milled particles, the number and the position of the injection locations of milled particles, the angle of expansion and the velocity of the injected particles, the drag model, particle density, particle sphericity, the position of the cone at the bottom of the classifier, air pressure, and air velocity. The value used for these parameters in the tuned baseline model are found in Table 4-2.
4.3 Results and discussion

The PSD of the mill product stream in the simulation is in good agreement with that of the experiment, which is shown in Figure 4-4. In addition, the time averaged prediction of mass flow rate of the product stream matches well with the experiment at around 0.189 kg/s, which is presented in Figure 4-5. It is worthy to note that this time dependent model has only come to a pseudo-steady state where the behavior such as particle exit rates fluctuate around a mean value which will be detailed later in Figure 4-5. A portion of the large particles injected near the rim of the bowl are not entrained and carried into the classifier. Instead, they deposit in locations within the cavity near the bowl. It is expected that eventually the rate of this deposition will become equal to the rate of re-entrainment resulting in steady state operation. It is not reasonable to extend a simulation out to times where this would occur. The recycle ratio in the baseline simulation is 4.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density (kg/m³)</td>
<td>1250 [120]</td>
</tr>
<tr>
<td>Density (Woody Biomass)</td>
<td>985</td>
</tr>
<tr>
<td>Sphericity*</td>
<td>0.85</td>
</tr>
<tr>
<td>Emissivity</td>
<td>0.75</td>
</tr>
<tr>
<td>Drag model</td>
<td>Ergun</td>
</tr>
<tr>
<td>Multiplier constant of drag model</td>
<td>1.0</td>
</tr>
<tr>
<td>Close pack volume fraction</td>
<td>0.66</td>
</tr>
<tr>
<td>Maximum momentum redirection from collision</td>
<td>40%</td>
</tr>
<tr>
<td>Normal-to-wall momentum retention</td>
<td>0.8</td>
</tr>
<tr>
<td>Tangent-to-wall momentum retention</td>
<td>0.85</td>
</tr>
<tr>
<td>Diffuse bounce</td>
<td>2.0</td>
</tr>
</tbody>
</table>

*Shape factor

A comparison of the measured PSD of the pure coal product stream to the blended coal and biomass product stream shows that the blend produces a bimodal size distribution with a new mode of large particles with an average size of ~400 microns. This suggests that milling of the
woody biomass produces many more large particles which exit the system than milling of coal, likely due to differences in particle density. To capture this behavior, the assumed PSD of each of these materials have been adjusted accordingly in the model. The assumed PSD of input pulverized coal particles, which contains around nine mass percent particles larger than >300 µm, and PSD of coal particles in the product stream are compared in Figure 4-6. Similarly, the assumed PSD of input pulverized woody biomass, which contains around 33 mass percent particles larger than >300 µm, and the PSD of woody biomass in the product stream are compared in Figure 4-7. Figure 4-8 is a comparison of the PSDs of the product material, the material collected in the classifier and the injected particles. This plot indicates that size distribution of the particles collected in the classifier is very similar to the injected material and the product stream includes only the small fraction of particles. This behavior suggests that there are many assumed PSDs for the injected milled particles that would satisfy the measured product PSD. Including the constraint of a RR of 4 limits the range of possible values for this assumption significantly. However, the assumptions made here for the baseline simulation are good enough to generate some understanding of the relative behavior of coal and biomass particles in the system and evaluate the impact of simple modifications to the classifier.

Barracuda software generates data files called general mesh viewers (GMV), which include information about the particles and cells at snapshots in time during the simulation; every particle in the simulation is tracked individually and has a unique identity number. For this simulation Barracuda was configured to generate a GMV file every 0.2 seconds. For our simulation of 10 seconds, 51 GMV files were generated. It was necessary to analyze the data of all 51 GMV files simultaneously to determine the pathway of individual wood and coal particles in order to understand the differences in their behavior in the classifier. To fulfill this purpose, all particles
that exited the simulation in the product stream were identified and then their pathways were tracked and plotted using information in previous GMV files. A code was developed using C# (see “Appendix A: The developed code for analysis of particle trajectories” by Tanner Jasperson). The results of this analysis show that most large particles (> than 300 microns) of coal and woody biomass that exit from the system bypass the cyclone in the classifier by moving directly from the vanes at the top of the classifier into the vortex finder. To demonstrate this behavior the pathway of a large (2 mm) woody biomass particle throughout the mill is plotted in Figure 4-9 as: (1) Injection of the particle into the system and subsequent swirling around the bowl, (2) Entrainment in the gases moving towards the classifier, and (3) Exiting from the system. It is possible that refining the simulation to smaller time steps would show that the particle entered the cyclone of the classifier, however it is unlikely that the particle traveled to an appreciable depth in the cyclone.

*Figure 4-4: PSD of blend product stream in simulation and experiment.*
Figure 4-5: Average mass flow rate of product stream (kg/s).

Figure 4-6: PSD of coal particles at input (injectors) and product stream.
Figure 4-7: PSD of woody biomass particles at input (injectors, which presented in Figure 4-3) and product stream.

Figure 4-8: PSD of product stream, bottom of classifier, and injectors.

Quantitatively this analysis shows that less than 5% of particles exiting as product make rotations inside cyclone of the classifier before entering the vortex finder and exiting the system, regardless of their type and size. This is an important result which may suggest 1) that particles which make rotations in the cyclone of the classifier are effectively captured and return to the
bowl and 2) that there are an appreciable number of large woody biomass particles which bypass the cyclone of the classifier completely. Small and simple modifications to the system operation and hardware were evaluated to determine if the latter observation may easily be impacted or resolved.

Figure 4-9: An example pathway of a large woody biomass particle before exiting.

4.3.1 Hardware manipulation scenarios

Three scenarios were defined and simulated to evaluate the effect of hardware manipulation on the classifier performance. The scenarios entitled $S_{\text{vane}}$, $S_{\text{vortex}}$, and $S_{\text{vane-vortex}}$ represent modifications of tightening the angle of the classifier inlet vanes, lengthening the vortex finder and a combination of changing the vane angle and lengthening the vortex finder, respectively. Figure 4-10 shows the geometry configurations along with the modifications to the vane angle and vortex finder shape, which form our hardware scenarios.
Figure 4-10: Top view of (a) base vane angle (b) closer vane angle; cut side view of (c) base vortex (d) deep vortex.

Figure 4-11 compares PSDs of three hardware scenarios with experimental PSD. It is apparent that each of these scenarios reduces significantly the amount of coarse particles in the product stream. Lengthening the vortex finder demonstrates a more favorable impact on PSD than adjustment of the vane angle. However, the impacts appear to be additive where the best results are obtained by lengthening the vortex and adjusting the vane angle.

Figure 4-11: Experimental PSD and generated PSD of scenarios Sc\textsubscript{vane}, Sc\textsubscript{vortex}, and Sc\textsubscript{vane-vortex}. 
The C# code was used to analyze the GMVs of the hardware scenarios by tracking how deep particles rotate in the classifier of the cyclone before entering the vortex finder and exiting with the product stream. Figure 4-12 presents the geometry of the classifier in its baseline configuration with an indication of location of each component in the Y-axis (depth). Entrained particles enter the classifier through the vanes, at the height of 1.28-1.43 m. The depth to which the exiting particles penetrated into the cyclone were evaluated and discretized into four regions: between 1.1 and 1.0, between 1.0 and 0.9, between 0.9 and 0.8 m and finally less than 0.8 m. Since the length of the vortex in the Sc\textsubscript{vortex} and Sc\textsubscript{vane-vortex} scenarios extends down to 1.07 m the behavior of particles above 1.1 m was not evaluated.

Figure 4-13 presents the ratio (in %) of the number of the product particles that swirl inside the classifier down to the indicated depth before exiting over the total number of particles exiting as product from the classifier. As discussed previously a low percentage of particles (<5%) in the baseline simulation swirl inside the classifier before exiting, regardless of their type. In the case of Sc\textsubscript{vortex}, there is no significant difference between coal and biomass, but the percentage of swirling particles increases up to around 13% at the 1.0 m. depth. In Sc\textsubscript{vane} and Sc\textsubscript{vane-vortex}, the percentages of swirling coals are higher than that of swirling woody biomass. The tighter vane angle induces high tangential velocity and ultimately impacts the swirl depth of particles more than the length of the vortex finder. In addition, the higher tangential velocity is expected to impact particles with a higher density more as demonstrated in the difference in behavior between the coal and biomass particles in the simulations. The combination of a modification of the vortex finder and the vane angle, Sc\textsubscript{vane-vortex}, pushes the exiting particles deepest into the classifier, resulting in the highest likelihood that large particles will be captured. The percentages of product particles that penetrate
down below 0.8 m is zero in all scenarios. This is reasonable as it is not expected that particles reaching the bottom of the cyclone cone will capture by the classifier.

Figure 4-12: Height of the pulverizer focusing on the classifier measurements in Y-axis (m).

Figure 4-13: Count percentages of exiting swirling particles over total exiting particles at various classifier depth.
Figure 4-14 shows the ratio (in %) of the number of particles greater than 300 microns that swirl inside the classifier to the indicated depth before exiting as product over the total number of particles greater than 300 micron that exit as product. It is demonstrated that no large exiting particles swirl inside the classifier before exiting in the base simulation and the condition where only the vane angle is adjusted. This shows that modification of the vane angle is not expected to have much effect on reducing the number of large biomass particles leaving with the product, which is suggested by the data presented in Figure 4-11. Conversely, lengthening the vortex finder is expected to have a significant impact on pushing the large particles deep into the cyclone of the classifier where they are more likely to be captured. When combined with an adjustment of the vane angle, up to 80% of the particles reach a depth of Y=1.0 m.

*Figure 4-14: Count percentages of large exiting swirling particles over total large exiting particles of >300 microns (count %)*
A secondary result of making small modifications to the vane angle and the length of the vortex finder is that the overall pressure drop of the system increases and the flowrate of carrier gas may decrease, resulting in a decrease in product flowrate and an increase in RR. The impact of these modifications on product flowrate and RR without adjusting any other parameters was quantified in this study and is presented in Table 4-3. Maintaining the product flowrate at desired levels would require the adjustment of other operating parameters. This optimization was not included in this study as it would be more beneficial to perform on a mill that was being optimized for demonstration of biomass cofiring. However the sensitivity of performance to operating parameters was investigated and is presented in the next section.

**Table 4-3: Primary results of analysis of running hardware scenarios.**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Flow rate of product stream (kg/s)</th>
<th>Recycle Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scvane: Effect of Closer Vane Angle</td>
<td>0.0834</td>
<td>9.9</td>
</tr>
<tr>
<td>Scvortex: Effect of Deep Vortex</td>
<td>0.122</td>
<td>7.0</td>
</tr>
<tr>
<td>Scvane-vortex: Effect of Closer Vane Angle &amp; Deep Vortex</td>
<td>0.0756</td>
<td>11.0</td>
</tr>
<tr>
<td>Base simulation</td>
<td>0.190</td>
<td>4.2</td>
</tr>
</tbody>
</table>

### 4.3.2 Scenarios of operating parameters

The effect of air velocity and pressure drop on classifier performance was investigated and the results are presented in Figure 4-15. Two scenarios with air velocities of 12.0 m/s (for Sc\textsubscript{low-velocity}) and 18.0 m/s (for Sc\textsubscript{high-velocity}) were simulated, where air velocity of the base simulation is 15.0 m/s and two scenarios of low pressure drop (Sc\textsubscript{low-PD}=92-85 kPa) and high pressure drop (Sc\textsubscript{high-PD}=100-80 kPa) of air flow were simulated, where the air pressure drop in the base
simulation is 95-82 kPa. The results of these simulations suggest that operating parameters have a much smaller impact on PSD than modifications to the mill geometry. PSDs of all of these four scenarios match relatively well with the experimental PSD. However the flow rate of the product stream is very sensitive to the velocity of the air, and it increases nearly linearly with increased air velocity as shown in Figure 4-16. These results suggest that the air velocity can be used to maintain the product flowrate after making small modifications to the classifier geometry without significantly impacting the increased performance in particle size distribution.

Figure 4-15: Experimental PSD and generated PSD of scenarios S_{low-velocity}, S_{high-velocity}, S_{low-PD}, and S_{high-PD}.

Figure 4-16: Changes of flowrate of product stream versus air velocity.
4.4 Conclusion

Results from an experimental study on the behavior of a CE 312 Raymond Bowl Mill while processing pure coal and blends of coal and biomass were simulated using CFD to understand the differences in behavior between coal and biomass particles in the classifier and to evaluate the impact of geometry and operational changes on mill performance. The pulverizer was simulated using Barracuda VR 17.1.0 software package. Assumed parameters were varied within reasonable constraints in order to match PSD and the mass flow rate of the product stream with the experimental data. A C# code was produced to post process the results of the simulation in such a way to elucidate single particle behavior in the system. The main results of this study can be summarized as follows:

- Small changes in hardware configuration can improve PSD of the product stream, or in other words can reduce the number of large particles exiting the system with the product stream. Evaluated in this study were changes to the angle of the classifier inlet vanes and lengthening of the vortex finder in the cyclone. Each of these modifications decreased the amount of large particles exiting the system as product and the effects of the two modifications were additive. For example, the volume percentage of large particles (>300 µm) in the product stream decreased from five percent to around four percent when comparing the vane case to the baseline simulation and reduced to less than 0.5% for the other two hardware manipulations. However, hardware manipulations are expected to require additional operational changes to maintain product flowrate.

- Tightening the vane angle will force particles to penetrate deeper into the classifier, where they are more likely to be collected and sent back to the bowl. However, the
effect is expected to be greater on denser particles. Therefore, coal particles will be more impacted than biomass particles.

- Lengthening the vortex finder in the cyclone of the classifier was necessary to make large coal and biomass particles penetrate deep into the classifier before exiting. This may indicate that the classifier in the system that we evaluated was poorly designed. This also indicates that the length of the vortex finder should be evaluated when optimizing mill operation for coal and biomass cofiring.

Changes in operating conditions do not significantly impact the PSD of the product stream but changing the air velocity strongly influences the rate of product particle production. This suggests that a combination of simple modifications to hardware configuration and modifications to operating conditions can be used to optimize mill performance by reducing the number of large particles in the product stream while maintaining product flow rate and recycle ratio. This project improved the understanding about the trajectories of the particle inside the pulverizer and provided a novel approach on CFD simulations of the pulverizers.
5 Design and construction of an air-cooled ash deposit probe and a fly ash aerosol probe to collect and measure ash deposit in a pilot and full-scale boiler.

5.1 Introduction

Determination of the mineral matter transformations is a key component of this project. Of particular interest is the behavior of ash deposition at conditions where fouling occurs in full-scale coal-fired utility boilers in the region of the secondary superheater and reheater. To fulfill this objective, ash aerosol and ash deposit probes were designed and constructed to collect entrained ash and ash deposit samples, respectively. This chapter will discuss the design and construction of those devices.

5.2 Ash deposit probe

The ash deposit probe provides a temperature-controlled tube-in-crossflow surface to collect ash deposit in order to measure the ash deposit rate and provide samples for morphological analysis. Cooling air is the flowing medium to maintain the coupon surface temperature that represents the reheater in utility boilers. In addition, the geometry and surface conditions of the probe coupon had to reasonably represent the heat transfer surfaces in a real utility boiler. The ash deposit probe is to be able to insert and retract smoothly from the furnace. Therefore, the samples could be collected without causing sluffing of the collected deposit. It was designed to operate at both pilot scale (1.5 MW) and full-scale boiler. It was modified for the full-scale test, which is discussed below.
5.2.1 Ash deposit probe, tested in a 1.5 MW combustor

I designed the ash deposit probe using Solidworks and had it constructed by Precision Machining Lab (PML) of BYU. The probe was constructed of two stainless-steel concentric pipes and was detailed in Figure 5-1 (see Appendix B: Fabrication drawings of ash deposit probe). The inner pipe (1” SCH 40) with an OD of 3.34 cm and an ID of 2.66 cm directed cooling air down the length of the probe and dispersed it through a perforated manifold onto two deposit coupons, each 10.16 cm in length, in order to maintain them at a desired temperature. The coupons and the outer pipe were fabricated from 2” SCH 80 and SCH 10 pipes, respectively, with an OD of 6.03 cm and IDs of 4.92 and 5.48 cm, respectively. After the cooling air impinges on the two coupons, it is directed back out of the combustion system using the outer pipe and ejected through a second perforated manifold. The constructed probe measured approximately 5.8 m so that it could also be applied in a utility boiler in the future. A type-k thermocouple (1/8”) was inserted through the center of the probe with the tip imbedded in a thermowell drilled into the inside surface of the first deposit coupon. The rate of cooling air flow was controlled using an actuated control valve connected to logic operating a PID control scheme to control the temperature at a given set point, which was 811 K. This system is similar to the probes used in previous studies [82].

The probe was installed on the furnace hanging from a steel frame supporting an I-beam with a beam trolley to easily insert and retract the probe without disturbing the deposit, which is presented in Figure 5-2. I designed, constructed and operated the steel frame. The probe, which was hung by a turnbuckle and clamp for leveling, can move backward and forward alongside of the conveyor (beam) of the frame to insert into a sampling hole (port tap) of the combustor. The valve-actuator connected at the beginning of the probe, which was a safe place in order to avoid devoting an extra room for the actuator on the floor or the other places.
5.2.2 Modified ash deposit probe, tested in a full-scale boiler (Hunter, unit-3)

Although the cooling air was sufficient to control the coupons at the desired temperature, it was determined during the 1.5 MW$_{TH}$ testing that it was insufficient to cool the length of the probe body sufficiently. During that testing, the probe body reached a temperature on the side facing the flame that the probe became curved and was difficult to retract from the furnace. Such curved shape made difficulties for pulling out the probe from the sampling hole of the combustor.
However, such difficulties did not cause the ash deposit loss. Therefore, the ash deposit probe, which was described in section 5.2.1, was modified to operate in a full-scale boiler (Hunter, Unit 3). I designed and constructed the modified ash deposit probe.

![Figure 5-3: Configuration of the modified ash deposit probe, including the cross view of modified cooling water.](image)

To improve cooling, we modified the probe by adding a cooling water jacket to the probe body that extended to within 66 cm of the probe tip while the cooling system for the coupons remained unaltered. The outer pipe (2 ½” SCH 10) with an OD 7.3 cm and an ID of 6.69 cm; and the inner pipe (1 ½” SCH 10) with an OD 4.82 cm and an ID of 4.27 cm were added to the base
probe. Therefore, probe was cooled by both air and water to maintain a stable temperature for the coupon surface and keep the probe straight during the operation at the boiler. In addition, two tension-type load cells (200 kgf and 5 kgf) were added to the probe trolley mounts in order to measure the total force and the imposed force by the ash deposit, which the 200 kgf load cell was mounted at nearby of the mass center of the probe. The other load cell was mounted around the coupons to measure the ash deposit rate. Therefore, the turnbuckles replaced by the load cells. In addition to the existing 1/8” K-type thermocouple, another thermocouple (1/16” K-type) was added to measure the coupon surface temperature to get assurance that the thermocouples remained imbedded in the thermowell. Figure 5-3 shows the configuration of the modified probe system, including the cooling water.

5.2.3 Control system of the ash deposit probe

As discussed, the coupon surface temperature of our ash deposit probe was controlled by cooling air. SNAP PAC (Programmable Automation Controller) system, which is a popular control system of OPTO22, was used in this project. SNAP-PAC system contains four main components: software, controller, brain, and I/O (input/output). The software including PAC control and PAC Display was downloaded from www.opto22.com. Our laptop was devoted as the controller. In addition, SNAP-PAC-EB2 was added to the system as the brain. The brain connected to the laptop by an Ethernet cable. Proportional-Integral-Derivative (PID) controller data includes 0.03, 2.0, and 0.0 for gain, integral, and derivative, respectively. Moreover, we used the following SNAPs, which were mounted on a SNAP-PAC rack, to set up the control system:

- SNAP-AILC, which is for load cells;
- SNAP-AITM-8, which is for the thermocouples;
- SNAP-AOA-23, which is for control valve,
- SNAP-PS24, which is a SNAP supply power to convert 110 VAC to 24 VDC.

Figure 5-4 displays the algorithm of the SNAP-PAC components connected to the probe. Our team helped me assemble this system.

![Diagram of SNAP-PAC components](image)


Figure 5-4: The algorithm of SNAP-PAC to carry out the control system of this study.

### 5.3 Fly ash aerosol probe

The ash aerosol probe was designed and constructed by Brigham Young University (BYU) and operated by University of Utah in conjunction with their proven particle sampling equipment and methodologies which will be detailed below. Ash aerosol in the flue gas, which is the precursor of ash deposit was extracted from the furnace using a water-cooled isokinetic sampling probe.
which is detailed in Figure 5-5. The probe is isokinetic because the velocity of the gas sample entering the probe is equal to the flue gas velocity. This probe was constructed of 4 concentric stainless-steel tubes. The outermost three tubes were used to direct cooling water from outside of the combustion system through the inner annulus to the probe tip and eject cooling water through the outer annulus. These three tubes had an OD of 3.81, 2.54 and 1.27 cm with wall thicknesses of 0.34, 0.21 and 0.17 cm, respectively. The particle laden gas was removed from the combustion system through the inner cavity of these three tubes. The sample was quenched and diluted at the probe tip by injecting gas through a 0.64 cm OD, 0.38 cm ID central tube. The sample from this system was further diluted and then supplied to particle analysis equipment. This probe was similar in length to the deposit probe for later application in a full-scale utility boiler.

![Water-cooled entrained ash sampling probe.](image)

*A Figure 5-5: Water-cooled entrained ash sampling probe.*

A real time determination of particle size was performed using a combination of a Scanning Mobility Particle Sizer (SMPS) and an Aerodynamic Particle Sizer (APS). Alternatively, the sample could be collected in size segregated bins using a Berner Low-Pressure Impactor (BLPI) which are later analyzed for composition and morphology analysis through scanning electron microscopy (SEM) and energy-dispersive X-ray spectroscopy (EDS). The methods for this analysis have been well documented in other studies [82, 121, 122]. The SMPS/APS are installed
on a two-stage dilution sampling system, in which the flue gas is diluted by nitrogen gas (N\textsubscript{2}) with the dilution ratio about 15:1; and then diluted by filtered air in manifold with dilution ratio about 25:1. Thus, the total dilution is about 375:1. The size range measured by SMPS is between 0.0143 and 0.6732 µm. The sample flow rate going into SMPS is 0.3 L/min and every measurement takes about 2 minutes. 10-20 repeatable measurements are conducted and averaged for each case to ensure reliable results. The size range in APS is from 0.532 to 20 µm. The sample flow rate is 5 L/min and one single sampling measurement takes about 20 seconds. For each sampling, over 100 measurements are used, and an averaged result is obtained. The measurements of SMPS and APS are simultaneous, using the same sample flow. Therefore, combining PSD data from the two instruments produces a result ranging from 0.01 to 20 µm. The sample flow of SMPS/APS and BLPI are the same to save time. Therefore, it is a two-stage dilution. The size range in BLPI is between 0.0324 and 15.7 µm. The sample flow rate is 23 L/min and every single sampling measurement takes about 30 minutes. It takes about two hours for three repeatable measurements in each sampling case.
6 Ash aerosol and deposit formation from combustion of coal and its blend with woody biomass in a 1.5 MW pilot scale combustor

6.1 Introduction

Of particular interest in this study is the mineral matter behavior during combustion conditions representative of utility boiler operation. Ash deposition on heat exchange surfaces can induce physical phenomenon that reduce efficiency and availability of the power system. It is expected that due to differences in the elemental content of mineral matter in coal and biomass that their cofiring may lead to changes in ash transformations and deposition [62, 122].

The elemental composition of mineral matter as a mixture from the coal and biomass can have profound effects on the downstream behavior of ash. The alkali content of woody biomass fuels such as potassium (K) volatilizes during combustion and subsequently condenses in a K-rich ash deposit, lowering the melting point temperature and generating a sticky deposit layer on the heat exchange surface [14, 68, 123]. However, alkali chlorides, formed from combustion of some biomass may react with sulfur from coal to form alkali sulfate which can decrease the stickiness of the impacting fly ash particles reducing ash deposit growth [124].

Many researchers have investigated the effect of mixing mineral elemental compositions on deposit grown characteristics while firing fuel blends. D. Nordgren [125] cofired straw/wood and straw/bark in a 150 kW boiler at different percentages and showed that fouling ash deposition rates of all mixtures were lower than that of pure straw combustion. This indicates that the dilution of straw ash with mineral matter from wood leads to a reduction in deposit rate. It has been shown
in a lab-scale combustor that the deposition rate while firing a blend of peat/bark increases as the fraction of bark increases in the range of 0.4-0.7 wt% [126]. Y. Liu [127] demonstrated that ash deposition rate decreases by increasing the ratio of a bituminous coal in a blend with a high-alkali coal in a 30 kW circulating fluidized bed. K. Qiu [128] investigated the ash deposition rate while firing a blend of coal and rice hull in a pilot furnace. Qiu observed the ash deposit thickness increased when firing a higher concentration of rice hull in the mixture. A.L. Robinson [129] performed pilot-scale combustion tests firing blends of three different types of coals and four types of biomass and they show that the interactions between the alkali chlorides from straw and the sulfur from coal reduce the stickiness of fly ash particles. D. Zhang [130] cites that some researchers reported that cofiring of coal with wood wastes do not change significantly the ash deposit rate due to the low ash content of wood; however, this result may be specific to the wood composition. C. Ndibe [131] investigated the ash deposit characteristics of cofiring torrefied spruce and bituminous coal in a 500 kW boiler. They showed that the tendency of alkali salt condensation, which results from the significant amount of potassium in such biomass, reduces while cofiring compared with the pure combustion of biomass. Cofiring low concentrations of a calcium-rich wood blended with coal has been studied by H. Zhou [15] in a 300 kW boiler and it was observed that calcium-sulfate-bonded ash deposit promotes the deposit growth. X.G. Xu [124] showed that the ash deposition resulting from cofiring wheat straw with two distinct high- and low-sulfur coals in a 25 kW down-fired combustor increased as the potassium concentration increased.

The objective of this paper is to present a detailed data set describing the pilot-scale combustion of pulverized coal and blends of pulverized coal and prepared woody biomass, which is suitable for the validation of predictive tools. To fulfill this objective, blends of Utah Bituminous
coal and torrefied and steam exploded wood that were prepared in a previous study [116] were
fired in a 1500 kW\textsubscript{TH} entrained flow combustor. Measurements were made of particle aerosol and
deposition behavior, furnace heat balance, flame intensity and combustion compositions.

6.2 Materials and methods

6.2.1 Overview of L1500 combustor

The L1500 is a 1500 kW\textsubscript{TH} entrained flow combustor located at the University of Utah. It
was designed for NO\textsubscript{x} emission studies during pulverized coal combustion and therefore has a
realistic burner turbulent mixing scale resulting in flame behavior and fuel burnout relevant for
full-scale utility boilers. The schematic view of L1500 furnace is shown in Figure 6-1. The furnace
includes a dual register low-NO\textsubscript{x} burner, a radiation section and a connective section.

![Figure 6-1: Schematic view of L1500 furnace with the radiative and convective sections indicated and the burner geometry detailed.](image)

A model of the dual-register low-NOx burner on the L1500 is presented as Figure 6-2. It
is constructed of 4 annular pipes and a cylindrical opening in a refractory-lined ‘burner plate’. The
innermost annulus is a bluff body and was not included for this investigation. The next larger annulus is the Primary or coal carrying pipe. The next larger annulus is for natural gas used only during heat up and overnight operation. The Inner and outer secondary air registers are the outermost annulus. The dimensions of each of these flow paths is also included in Figure 6-2. Each of the Secondary registers is supplied air through an adjustable swirl block capable of inducing a tangential component to the air velocity.

![Figure 6-2: Schematic and dimensions of the low-NOx burner registers on the L1500.](image)

The burner is mounted on a plate cast in refractory. The exit of the annular pipe cluster terminates 6.985 cm before the face of the refractory. At the planar surface where the burner pipes terminate, the cylindrical hole through the refractory is 21.59 cm in diameter (consistent with the OD of the Outer Secondary air register) and then widens at a 38° angle to form a quarl, terminating at a diameter of 31.87 cm at the refractory face surface and the beginning of the radiation section. The radiation section of the furnace comprises 12 sections. The first 4 sections have internal
dimensions of 1.0414 m wide by 1.1684 m tall and are 1.2192 m in length. Sections 5 through 10 have a square cross section of 1.0414 m and are 1.2192 in length. Section 11 has a square cross section of 1.0414 m and is 0.6096 m in length. Section 12 has a square cross section of 1.0414 m and is 1.9413 m in length and its outlet is tapered into a 0.6604 m diameter round outlet into the transition section into the convective section. Components downstream of this point are outside the interest of this study. Each of the sections in the radiation section are lined with layers of refractory manufactured by Harbison Walker International (HWI). The inside surface is 21.59 cm thick of Ultra-Green SR, followed by a 5.08 cm thick layer of Insboard 3000 and then two layers totaling 7.62 cm of Insboard 2600. In each of the first 11 sections, there is a sample port in the center of the section length. The centerline of the sample port in the first section is 60.960 cm from the quarl outlet and each of the subsequent sample port centerlines are an additional 1.219 m from the previous port centerline.

Across sectional view of the first four sections of the L1500 is presented in Figure 6-3. The first two sections contain water cooled plates on two walls installed coplanar with the inside refractory surface and centered on the sample port which are 90.17 cm in length and 13.97 cm tall. Sections 3 and 4 contain water cooled heat exchangers made of ½" SCH 40 pipe, bent to make 4 full passes and two half passes up and down the furnace wall spanning a length of 91.44 cm and reaching to within 4.013 cm of the ceiling and 16.71 cm of the floor. Sections 5 through 11 do not have active heat exchange surfaces, but do contain water cooled liners in the center sampling port on one side of the furnace, which also remove heat. Each of these devices is equipped with flow meters on the water supply and K thermocouples measuring the water temperature in and out.

A sample probe removes gas from the furnace, after the transition section at the inlet to the convective section detailed in Figure 6-1. This sample is pulled through a particle filter and then
is quenched and chilled to remove moisture and is sent to a bank of analyzers whose readings are continuously recorded in the DCS system. The bank of analyzers includes Yokogawa AV8C O₂ (0-25%), California Analytical ZRH CO/CO₂ (0-2000 ppm for CO, 0-20% for CO₂), Thermo Environmental 42C NOₓ (0-10,000 ppm) and California Analytical 601 SO₂ (0-5000 ppm).

![Cross-sectional view of the burner and first four sections of the L1500.](image)

**Figure 6-3. Cross-sectional view of the burner and first four sections of the L1500.**

### 6.2.2 Fly ash particle and deposit sampling

Determination of the mineral matter transformations were a key component of this project. Of particular interest was the behavior of ash deposition at conditions where fouling occurs in full-scale coal-fired utility boilers in the region of the primary superheater and reheater. To fulfill this objective, probes were developed to extract samples of entrained ash and to measure deposition rate on a surface representative of heat transfer tubing associated with the aforementioned systems. The details of the probes were discussed in in Chapter 5.

Both the deposition and the aerosol sampling probes were installed in the furnace in port 7 and port 10 of the furnace, which allowed sampling at flue gas temperatures that are representative
of the region of the boiler containing the primary superheater and the vertical reheater. These locations are 7.92 and 11.58 m from the planar face of the burner quarl exit.

The two probes of ash deposit and ash aerosol samplings are inserted alternately in the flue gas in south ports of 7 and 10. These locations are 7.92 and 11.58 m from the planar face of the burner quarl exit. As discussed, the probes were designed long enough because of their implementations in full-scale boilers. Figure 6-4 is a photo showing the aerosol sampling system connected to port 10 with the deposit probe at port 7. The ash deposit samples and fly ash particles were sent to the University of Utah for the lab analysis.

![Figure 6-4: Ash deposit and ash aerosol sampling at port 7 and 10, respectively.](image)

### 6.2.3 Fuel preparation and analysis

The purpose of this study was to evaluate the combustion behavior of blends of prepared woody biomass and coal for use in unmodified coal-fired utility boilers. Raw biomass material
was harvested from US Forest Service land, under their direction, near Alta Ski Resort in Little Cottonwood Canyon, Utah. Wood from this area was chosen because of its availability and its similarity in species and composition to biomass from the Manti La-Sal National Forest. The raw material was prepared using two pretreatment processes: torrefaction and steam explosion. Torrefaction is a thermal process to make biomass more like coal so that it is easy to grind. Torrefaction removes moisture and oxygen from biomass and breaks down the fibrous hemicellulose fraction to produce a more grindable and energy dense fuel [31-34]. Torrefied wood can subsequently be compressed to pellets to facilitate transport and reduce storage costs [35]. Its combustion is similar to coal; but its heating value is less than that of coal and the handling is more expensive. Steam explosion (Steam X) exposes the biomass to high pressures using saturated steam in a pressure vessel. Once the pressure inside the wood cells is equilibrated, the vessel pressure is suddenly released to burst the cell wall and break down the material structure [39]. After steam explosion, biomass is removed, rinsed, dried and pelletized [40].

For the purpose of this study, each of the prepared biomass were blended with a Utah Bituminous Coal from the Sufco mine in a 15% biomass, 85% coal by weight mixture. Large samples of these blends and the pure parent coal were milled to a 70% through 200 mesh grind using a CE 312 Raymond Bowl Mill. The observed milling behavior and resulting particle size distribution (PSD) are detailed elsewhere [116, 118].

In this paper, the pure Sufco coal will be subsequently referred to as “Coal”, the coal/torrefied biomass blend will be referred to as “Torrefied” and the coal/steam exploded blend will be referred to as “Steam X”. The ultimate and proximate analysis of these fuels are presented in Table 6-1 and their mineral matter compositions are presented in Table 6-2.
Of particular concern when firing biomass fuels is the content of sodium and potassium. Their interaction with other elements such as sulfur can contribute significantly to fouling. Table 6-2 shows that these biomass materials are fairly benign contributors of alkali metals. In fact, the steam exploded blend shows a notable reduction in sodium content. This observation is supported by the expected impacts of the steam exploding process. After the cell walls have been ruptured by the rapid pressure reduction, the material is rinsed, removing any sodium that resided within the solution in the cells. It is expected that blending coal with biomass should provide a lower overall ash content than pure coal because woody biomass contains less ash. However, the fuel analysis presented in Table 6-1 show the ash content is greater for the blended fuels. This is likely due to earthy materials that were included as part of the harvesting and upgrading processes.

| Table 6-1: Ultimate and proximate analysis of fuels from the past study [118]. Values are % mass as received. HHV is kJ/kg as received. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ash C H N S O Cl H2O Vol. FC HHV (kJ/kg) |
| Coal | 12.74 | 63.05 | 4.35 | 1.13 | 0.45 | 11.61 | 0.145 | 6.52 | 38.26 | 42.47 | 25959 |
| Torrefied | 14.99 | 59.66 | 4.36 | 1.00 | 0.47 | 14.12 | 0.180 | 5.21 | 41.69 | 38.10 | 24676 |
| Steam X | 14.16 | 60.57 | 4.49 | 1.06 | 0.48 | 14.16 | 0.178 | 4.89 | 42.32 | 38.63 | 24746 |

| Table 6-2. Mineral matter composition, % mass, from the past study [118]. Difference between sum and 100% is undetermined. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SiO2 Al2O3 TiO2 CaO Fe2O3 K2O MgO Na2O SO3 P2O5 BaO MnO2 SrO |
| Coal | 54.07 | 11.03 | 4.17 | 15.25 | 4.17 | 1.20 | 3.99 | 1.26 | 6.35 | 0.29 | 0.08 | 0.08 |
| Torrefied | 55.55 | 10.48 | 0.53 | 15.88 | 4.07 | 1.49 | 4.08 | 1.33 | 5.98 | 0.31 | 0.08 | 0.05 | 0.08 |
| Steam X | 57.53 | 12.40 | 0.62 | 12.82 | 4.09 | 1.05 | 3.77 | 0.97 | 6.06 | 0.27 | 0.06 | 0.02 | 0.08 |

The increase in ash content for the two biomass blend materials compared to the pure coal in Table 6-1 are surprising. This increase is likely due to the mixing process, which occurred on a concrete floor and may not have been cleaned well enough.
6.2.4 Experimental conditions

The test plan was developed for this program to generate a data set that represents as closely as possible the operating conditions of a full-scale wall-fired pulverized coal utility boiler. The key parameters we were determined to match are consistent with low-NO$_x$ burner operation with lower furnace staging and over-fire air configuration. They are listed in Table 6-3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firing Rate</td>
<td>kW</td>
<td>880</td>
</tr>
<tr>
<td>Primary Gas/Coal Ratio</td>
<td>Unitless (mass)</td>
<td>2.0</td>
</tr>
<tr>
<td>Secondary Air Distribution</td>
<td>Unitless (mass)</td>
<td>30/70</td>
</tr>
<tr>
<td>(Inner/Outer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower Furnace Stoichiometric Ratio</td>
<td>Unitless</td>
<td>0.9</td>
</tr>
<tr>
<td>Excess O$_2$</td>
<td>Vol %, dry</td>
<td>3 - 4</td>
</tr>
</tbody>
</table>

Table 6-3. Target operating parameters designed to match coal-fired utility boiler conditions, within the capabilities of the L1500.

6.3 Results and discussion

6.3.1 Actual operating conditions

The L1500 was operated for approximately 8 hours for each of the three fuels. During this period, deposit samples were collected for 30, 60 and 90 minute accumulation times in ports 7 and 10 along with the aerosol sampling. During that period, the furnace was operated at constant conditions. The sampling commenced when it was determined that the refractory walls had approached a thermal equilibrium for the given condition heat distribution. Following the experiments, the experimental conditions (including reactant flowrates, fuel compositions and measured gas compositions) were averaged over the steady-state operating period analyzed to determine the consistency of the data and air leakage into the furnace and fuel feeding system. It should be understood that the calculated air leakage rate is the result of mass balance calculations.
and contains error from all of the measured flow rates and compositions. The resulting operating conditions are summarized in Table 6-4.

### Table 6-4. Average measured operating conditions for each fuel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Coal</th>
<th>Torrefied</th>
<th>Steam X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firing Rate</td>
<td>kW</td>
<td>922.7</td>
<td>873.5</td>
<td>881.9</td>
</tr>
<tr>
<td>Fuel Rate</td>
<td>kg/hr</td>
<td>122.0</td>
<td>127.4</td>
<td>128.3</td>
</tr>
<tr>
<td>Primary Air Rate</td>
<td>kg/hr</td>
<td>235.4</td>
<td>245.7</td>
<td>229.8</td>
</tr>
<tr>
<td>Primary Air Temp</td>
<td>K</td>
<td>305.8</td>
<td>303.2</td>
<td>303.3</td>
</tr>
<tr>
<td>Inner Secondary Air Rate</td>
<td>kg/hr</td>
<td>198.7</td>
<td>191.4</td>
<td>197.2</td>
</tr>
<tr>
<td>Inner Secondary Air Temp</td>
<td>K</td>
<td>525.0</td>
<td>524.4</td>
<td>523.9</td>
</tr>
<tr>
<td>Inner Secondary Air Swirl</td>
<td>%</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Outer Secondary Air Rate</td>
<td>kg/hr</td>
<td>418.4</td>
<td>378.5</td>
<td>414.7</td>
</tr>
<tr>
<td>Outer Secondary Air Temp</td>
<td>K</td>
<td>532.6</td>
<td>527.5</td>
<td>532.4</td>
</tr>
<tr>
<td>Outer Secondary Air Swirl</td>
<td>%</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Air Leakage (Calculated)</td>
<td>kg/hr</td>
<td>163.3</td>
<td>155.6</td>
<td>186.0</td>
</tr>
<tr>
<td>Staging Air Rate</td>
<td>kg/hr</td>
<td>267.4</td>
<td>268.2</td>
<td>261.7</td>
</tr>
<tr>
<td>Staging Air Temp</td>
<td>K</td>
<td>320.7</td>
<td>319.5</td>
<td>319.2</td>
</tr>
<tr>
<td>Flue Gas O₂</td>
<td>Vol %, Dry</td>
<td>4.52</td>
<td>4.23</td>
<td>4.47</td>
</tr>
<tr>
<td>Flue Gas CO₂</td>
<td>Vol %, Dry</td>
<td>14.73</td>
<td>15.26</td>
<td>15.20</td>
</tr>
</tbody>
</table>

### Table 6-5. Operating parameters (measured and calculated) relevant to the particle deposition behavior

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Port</th>
<th>Coal</th>
<th>Torrefied</th>
<th>Steam X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flue Gas Velocity</td>
<td>m/s</td>
<td>7</td>
<td>1.50</td>
<td>1.45</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>1.40</td>
<td>1.31</td>
<td>1.33</td>
</tr>
<tr>
<td>Flue Gas Temperature</td>
<td>K</td>
<td>7</td>
<td>1363.7</td>
<td>1318.4</td>
<td>1340.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>1278.2</td>
<td>1194.3</td>
<td>1254.3</td>
</tr>
<tr>
<td>Coupon Surface Temperature</td>
<td>K</td>
<td>7 &amp; 10</td>
<td>811±10</td>
<td>811±10</td>
<td>811±10</td>
</tr>
<tr>
<td>Calculated Flue Gas Density [132]</td>
<td>kg/m³</td>
<td>7</td>
<td>0.262</td>
<td>0.270</td>
<td>0.266</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.279</td>
<td>0.299</td>
<td>0.285</td>
</tr>
</tbody>
</table>

The measured and calculated operating parameters that are relevant for aerosol and deposit sampling are summarized in Table 6-5. The air flow rate includes the sum of primary air, inner
secondary air, outer secondary air, staging air and air leakage. A suction pyrometer probe measures the flue gas temperature at the two sample locations.

In the following sections, whenever data are compared graphically for the three fuel conditions, the coal data will be represented in grey, Torrefied data will be represented using blue and Steam X data will be shown in orange. Comparisons of behavior due to location or other conditions will be represented by other colors.

6.3.2 Properties of ash aerosol

The aerosol PSDs measured at ports 7 and 10 are shown in Figure 6-5, where the ash concentrations on the left plots are based on flue gas volume at standard state, while the right plots are the total ash mass basis. The mineral compositions in the fuels and combustion temperature determine the formation of sub-micron ash particles, which show similar PSDs in the three cases. The ultra-fine mode at size < 0.1 µm is possibly formed through nucleation and condensation of vaporized inorganic species in the sampling probe [124]. Char fragmentation is the main mechanism for the super-micron mode (>1.0 µm) and is not affected by combustion temperature. The possible third mode, which is the central mode at the range of 0.1-1.0 µm, is formed through the coagulation of nuclei. The central mode is called the coagulation mode (usually around 0.3 µm). This mode for the Torrefied conditions is slightly higher than that of the other fuels. The flue gas temperature is the key parameter in vaporization and coagulation rate, which is relatively lower for the Torrefied conditions. The main processes of ash aerosol formation are described elsewhere [71]. In general, the ash aerosol sampling results suggest that cofiring prepared woody biomass with coal plays a minor role in altering the aerosol formation from the pure coal condition.
Figure 6-5. Comparison of ash aerosol PSDs sampled from port 10 (top) and port 7 (bottom) for the three fuel conditions

Figure 6-6 compares the ash aerosol PSDs for the Coal condition at ports 7 and 10. These data show reduced concentration of sub-micron particles in the range of 0.1-1.0 µm and increased concentration of particle sizes in the range of 1.0-5.0 µm at port 10 compared with port 7. Typically, the ash particles < 5 µm follow well with the flue gas stream. However, the concentration of fly ash particles > 10 µm, which are not measured as a component of the entrained ash, reduce along the horizontal flow path in L1500 furnace due to low velocities and particle settling. The difference in ash aerosol concentrations at the two locations may be due to coagulation and subsequently may lead to the different growth of local ash deposition.
The compositions of ash particles sampled using the BLPI at port 7 are shown in Figure 6-7. The results of the BLPI analysis on samples from port 10 are similar and are not presented here. The sub-micron size range of ash aerosol enriched slightly with alkali species of sodium (Na) and potassium (K) for the Torrefied and Steam X cases compared to Coal conditions. This could be caused by increased vaporization of alkali species due to the introduction of woody biomass, in which potassium and other minerals are more likely to be organically-bonded. The less volatile species such as Ca and Si, which contribute the most mass, have similar distributions in the three cases. Mass fraction changes of Si and Al follow each other, suggesting their common existence as alumina-silicates in fly ash. The size-segregated compositions of the three cases present slight variations; therefore, the transformation of minerals during cofiring is hardly affected by the wood content. Accordingly, the three cases with different fuels produce aerosols with similar size distributions. Higher mass fractions of calcium and sulfur are found in the fine particles smaller than 0.1 um, which is likely caused by the condensation of CaSO₄. This is consistent with the occurrence of ultra-fine mode around 0.04 µm in aerosol concentrations (see Figure 6-5).
6.3.3 Properties of ash deposits

The deposit sampling was performed according to plan with some small alterations due to constraints on fuel supply and general sampling schedule. At the end of each sampling period, the deposit probe was carefully removed from the reactor using the overhead beam trolley. The deposit was photographed and then the ash built up on the temperature-controlled coupons was removed by scraping the surface clean into a sample pan. These samples were then weighed and stored for further analysis. Some selected images of the coupons after the operations are shown in Figure 6-8. In these images, the end cap of the deposition probe is glowing because it was not a cooled surface of the probe.

The collected ash deposit samples were weighed using a balance with 0.1-gram accuracy. It is assumed that the visual positive and negative errors of collecting ash deposit are estimated at +1% and -5%, respectively. It says that 1% of the collected ash deposit samples is from outside of the coupon surfaces; and 5% of the collected ash deposit samples on the coupons were lost. Ash deposition rates were determined by dividing the ash deposit mass by the total coupon surface area.
and the implementation time. Ash deposit masses and rates are presented in Figure 6-9 with port 7 data presented in the left plot and port 10 data presented in the right plot. The ash deposit rate is higher for the two blended fuels at 30 minute sampling times in port 7. At this location we would expect higher particle concentration of particles larger than 5 microns than in port 10. The port 10 samples show that at long build-up times, there is no difference between the deposit rate of the Torrefied and Coal conditions. However, for the 30 minute sample time, the deposition rates may be slightly lower for the blended fuels. Comparison between the deposit rates for port 7 and port 10 indicate the expected behavior, with larger rates for positions closer to the burner. Generally, it is safe to say that the deposition rates are similar while firing these blended fuels or coal.

<table>
<thead>
<tr>
<th>Port 7, 30 min</th>
<th>Port 7, 90 min</th>
<th>Port 10, 30 min</th>
<th>Port 10, 90 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-8. Images of deposit build-up on coupon surfaces for each deposit test interval and fuel. By row: 1) Coal 2) Torrefied 3) Steam X. The picture on the bottom right is representative of a 68-minute sample from Port 10.

Figure 6-10 shows the composition of ash deposits sampled in port 7 for 30 and 90 minute retention times and Figure 6-11 shows the same for port 10. Silicon (Si), Aluminum (Al), Calcium (Ca), and Iron (Fe), which are typically found in the outer layers of ash deposits [123], have the highest enrichments for all of the tests. The composition of the deposits do not differ significantly as a function of fuel. The sodium content of the Coal deposits are slightly lower than for the
biomass blends. The qualitative analysis of the thermodynamic equilibrium of vapor species simulated using FactSage 7.3 suggests that the sulfur content in the condensed phase while firing the Torrefied fuel should be higher than the other fuels. This observation is in agreement with the higher sulfur concentration of ash aerosol (see Figure 6-7) and ash deposit samples (Figure 6-10) for the Torrefied condition compared with the other fuels. Sulfate compounds have higher melting temperature [75, 76]; and may be solid at the same temperature where Cl compounds might be sticky. However, the differences in sulfur concentration of the fuels, ash aerosol, and ash deposit samples are not large. Sulfur retention efficiency, which is defined as the fraction of the total fuel sulfur that is retained in ash, is not investigated here. Dunxi Yu [133] showed that the sulfur retention efficiency can increase with the molar ratio of alkali and alkaline earth metallic species to sulfur. Higher concentration of K declines the melting point temperature of ash aerosol particles and increases the melt fraction; and leads to enhance ash deposit growth [77]. Variation of potassium concentration of the three fuels is not large but is highest for the Torrefied fuel.

![Figure 6-9. Mass of collected deposit and rate of deposition for the three fuel conditions in both Port 7 and Port 10.](image)
Figure 6-10. Ash deposit compositions for the 30 and 90-minute samples taken in port 7 for all three fuel conditions.

Figure 6-11. Ash deposit compositions for the 30 and 90-minute samples taken in port 10 for all three fuel conditions.

SEM images of ash deposits collected in port 7 for 90 minute retention time are shown in Figure 6-12. They are presented in three scales of 500 µm, 300 µm, and 100 µm. These images show very little difference in the morphology of the deposits for the three different fuels. The ash deposits are composed of many large spherical particles. The fine particles accumulated near the surface of
the coarse particles may represent the nucleated alkali species which would likely enhance the stickiness of coarse particles [124].

Figure 6-12. SEM images of ash deposit samples taken in port 7 with a deposition time of 90 minutes for all three conditions.
The PSD of the collected ash deposit samples where measured by a laser diffraction particle size analyzer (Beckman Coulter LS230). The results of this analysis are shown in Figure 6-13 for 30 and 90 minute samples collected in port 7 for all three fuels. The PSD is shifted to slightly larger particles in the 90 minute samples compared with the shorter sampling times. This could be a result of coagulation, or larger particles begin to be retained on the surface after a sticky layer has developed. Figure 6-13 also suggests that the two blended fuels result in larger particle sizes in the deposit when compared to the Coal condition. More significantly, there is a marked shift in deposited particle size for the blended fuels at the long retention times in the particle sizes between 300 and 400 microns. It is also interesting that there is little difference in the Coal PSDs for short and long retention times, indicating a difference in behavior between the Coal and the biomass blends.

Figure 6-13. The PSD of 30 and 90 minutes deposit samples collected in port 7 for all three fuel conditions.
6.4 Conclusions

A 1500 kW pulverized coal combustor was used to elucidate the differences in mineral matter transformations and deposition between pure coal and blends of coal with torrefied wood and steam exploded wood. The blends were prepared at a 15 mass % composition in Utah Bituminous coal and fired using combustion conditions relevant for a full-scale coal-fired utility boiler. Detailed measurements of particle size distribution, composition and morphology were performed for both the entrained ash aerosol and deposits. The rate of deposition was also determined at two locations in the furnace for the pure coal and the two blends. Overall these data provided confidence that blends of prepared woody biomass can be fired in an unmodified boiler without significant impacts on ash behavior.

Ash aerosol measurements showed that the particle size distributions were nearly the same for the three fuels tested. However, there was a deviation where the torrefied blend showed an increase in concentration in the 0.1 to 1 µm size range. Analysis of pure coal samples showed that aerosol at longer furnace residence times the PSD shifted from fine particles 0.1 to 1 µm to larger particles in the 1 to 5 µm range. This is likely due to coagulation of the aerosol particles. Composition measurements of the collected aerosol indicated that the blended fuels displayed slightly higher concentrations of alkali metals in the submicron range, possibly due to increased vaporization of organically bound metals. Calcium concentrations were also higher for the blended fuels. However, transformations are only slightly impacted by the wood blending.

Deposition measurements showed that the rates were higher at the location closer to the burner, likely due to saltation of large particles in low velocity flue gas, resulting in lower overall particle concentrations in ports further from the burner. The deposit rates were higher (30-70%) for the biomass blends at locations closer to the burner and at short collection times, but nearly the
same for all fuels at long collection times. At locations further from the burner and at short collection times the deposit rates were slightly lower (11-22%) for the biomass blends, but there was no difference at long collection times. The composition of the deposits did not vary significantly as a function of fuel. The deposits were slightly enriched in sodium for the biomass blend tests. SEM imaging of the deposit materials showed essentially no difference in morphology due to the biomass blending. Particle size distributions of the deposits indicated that there was a measurable change towards larger particles in the 300 to 400 µm range when firing the biomass blends for all retention times. The shift towards larger particles was greater for longer collection times. However, the coal PSD remained constant for long retention times.
7 Characterization of ash deposit and ash aerosol in a 500 MW full-scale furnace

7.1 Introduction

Deposition related to biomass combustion has been widely studied in recent years. G. Akar [134] analyzed the collected ash deposit of lignite high-calcium coal at several elevations of a 2×210 MW boiler in Turkey and they suggest that its blend with raw coal can reduce ash deposition rate. K.H. Anderson [135] investigated cofiring coal with straw in a 150 MW pulverized-fuel and collected ash deposit samples in several locations of the utility boiler. They showed that the visual ash deposit amount on the upstream side of the probe increases by cofiring of coal and straw at maximum share of 20% (energy base) of straw due to the increased amount of K and S during cofiring; and the chemistry and structure of upstream ash deposit changes when cofiring. H. Wu et al. [136] investigated the ash deposition behavior while suspension-firing wood with injection of coal ash in an 800 MWth unit in Denmark. They showed that the ash deposition changes by adding coal fly ash at different locations representing high and low flue gas temperatures. They reported that, at the locations where the flue gas temperature was low (1023 to 1073 K), the ash deposit propensity reduced by adding coal fly ash, which is likely due to less K compositions (KCl and KOH/K₂CO₃) in the ash deposit.

In this work, the raw woody biomass collected from a National Forest in Utah was pre-treated by torrefaction, a process to improve energy density, grindability and hydrophobic behavior of the raw wood [32], then cofired with pulverized Utah coal in a local plant. The focus of this
project was primarily on the effects from the wood minerals upon mechanisms of ash formation and fouling deposition, but the balance of plant impacts was also investigated.

The objective of this work was to develop a data set of ash aerosol concentrations and compositions and fouling deposit rates and samples in cofiring torrefied woody biomass with pulverized coal in existing full-scale coal-fired boiler. To achieve such an objective, the modified ash deposit probe, which is described in Chapter 5.2.2, was installed in a full-scale boiler (Hunter unit 3) in order to collect ash deposit on the coupon. In addition, ash aerosol sampling was performed by University of Utah. The baseline coal combustion test was also conducted for comparison with the cofiring case. This work is a part of a joint project, which includes the following tasks:

- Deposit and entrained ash sampling (Brigham Young University and the University of Utah);
- Radiation intensity measurement (University of Utah and Chalmers University);
- FTIR gas trace and major species analysis (Chalmers University);
- Continuous gas analysis (PacifiCorp and Brigham Young University).

7.2 Materials and methods

7.2.1 Overview of Hunter-3 boiler

A schematic view of the full-scale boiler, Hunter Unit 3, with the dimensions of the radiant section is shown in Figure 7-1. This unit is a Babcock & Wilcox opposed wall-fired unit with 20 burners in four rows and five over fire air (OFA) ports in one row on each wall. Fouling, which is
the ash deposit in convection zone, is mostly formed near the secondary super heater and vertical reheater; and makes significant problems for the boiler performance. The full capacity of the boiler is approximately 500 MW<sub>e</sub>. The target load condition for this project was 90% of full load with typical fuel and air distribution resulting in a lower furnace stoichiometric ratio of about 0.9.

### 7.2.2 Fly ash particle and deposit sampling

Ash deposit formation on heat exchanger surfaces of a boiler leads to deficient heat transfer and an increase maintenance costs. Therefore, determination of mineral matter transformations of fly ash aerosol and ash deposits are essential for this project. To fulfill this objective, probes were developed to extract samples of entrained ash and to measure deposition rate on a surface representing tube conditions in the adjacent vertical reheater heat exchange section. The details of each of these probes were discussed in Chapter 5.

![Figure 7-1: Schematic view of the boiler of Hunter-3 (Ref.: US DOE Cooperative Agreement No. DE-NT0005288).](image-url)
Both the deposition and the aerosol sampling probes were installed in the furnace in port 1, the center of reheater, rear side of floor 14\textsuperscript{th} of Hunter Unit 3, as shown in Figure 7-2. The two probes (ash deposit and ash aerosol probes) are inserted alternately in the flue gas. They are mounted from an I-beam, which was wired to the mezzanine upper floor. Figure 7-3 is a photo showing the ash deposit sampling system installed in the test port.

As discussed earlier in chapter 5.2.2, the deposit probe was modified by adding cooling water, load cells and an additional K-type thermocouple. We tested the probe several times during shake down testing for assurance of its operability. During these tests the weld between the water- and air-cooled sections failed, likely due to different thermal expansion in the two sections. The water jacket was drained for the remainder of the tests, resulting in out of spec probe balance with respect to the load cells. However, it was determined that the probe was still capable of controlling the coupon temperature at the desired setpoints using the air system, which allowed us to meet our objectives. For the remainder of the testing, the load cells were removed.

\textit{Figure 7-2: The schematic view of Hunter, Unit-3, and the location of test port for ash deposit and ash aerosol probes (Image from Babcock & Wilcox [9]).}
Figure 7-3: Ash deposit sampling probe at the test port.

7.2.3 Fuel preparation and analysis

Raw woody biomass was collected near ALTA Ski Resort, because of its availability and similar composition to biomass of Manti La-Sal National Forest. The material was torrefied by Amaron Energy at a temperature of 325 °C to produce 724 tons of pellets. A brief description of the torrefaction process was presented in 6.2.3. In this project, the pelletized torrefied biomass was blended with Sufco coal (a Utah Bituminous coal from Sufco mine) in a 15% biomass, 85% coal by weight mixture. Samples of the fuels were collected during the testing from the redler conveyor, which fills the bunker and silos. It takes approximately eight hours for the fuel to travel through the silos to arrive the mills which is in turn immediately fed to the burners. The ultimate and proximate analysis of these fuel samples are presented in Table 7-1 and their mineral matter compositions are presented in Table 7-2.

Table 7-2 shows that the mineral composition of ash was similar for the coal and for the torrefied blend although potassium content of the torrefied blend were slightly higher than that of coal. Generally, since woody biomass contains less ash than coal, cofiring torrefied biomass generates less fly ash than coal. It is expected that the interaction of sodium and potassium contents
of the fuels with the other elements such as sulfur will influence the rate of fouling. However, the
difference in alkali content observed here are minimal.

Table 7-1: Ultimate and proximate analysis of fuels. Values are % mass as received before
normalization. HHV is kJ/kg as received

<table>
<thead>
<tr>
<th>Redler Samples</th>
<th>Fuel</th>
<th>Ash</th>
<th>C</th>
<th>H</th>
<th>N</th>
<th>S</th>
<th>O</th>
<th>H2O</th>
<th>Vol.</th>
<th>FC</th>
<th>HHV (kJ/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/22/19</td>
<td>20:16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24614</td>
</tr>
<tr>
<td>Torrefied blend</td>
<td>11.5</td>
<td>60.1</td>
<td>4.4</td>
<td>1.1</td>
<td>0.4</td>
<td>12.9</td>
<td>9.7</td>
<td>38.9</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/23/19</td>
<td>2:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24300</td>
</tr>
<tr>
<td>Torrefied blend</td>
<td>10.5</td>
<td>60.1</td>
<td>4.5</td>
<td>1</td>
<td>0.4</td>
<td>12.3</td>
<td>11.4</td>
<td>39.4</td>
<td>38.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/23/19</td>
<td>22:59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26409</td>
</tr>
<tr>
<td>Coal</td>
<td>11.5</td>
<td>63.8</td>
<td>4.5</td>
<td>1.3</td>
<td>0.6</td>
<td>9.5</td>
<td>8.8</td>
<td>36</td>
<td>43.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7-2. Mineral matter composition, % mass. Difference between sum and 100% is
undetermined.

<table>
<thead>
<tr>
<th>Redler Samples</th>
<th>Fuel</th>
<th>SiO2</th>
<th>Al2O3</th>
<th>TiO2</th>
<th>CaO</th>
<th>Fe2O3</th>
<th>K2O</th>
<th>MgO</th>
<th>Na2O</th>
<th>SO3</th>
<th>P2O5</th>
<th>BaO</th>
<th>MnO</th>
<th>SrO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/22/19</td>
<td>20:16</td>
<td></td>
<td>56.6</td>
<td>14.1</td>
<td>0.7</td>
<td>10.5</td>
<td>4</td>
<td>1.3</td>
<td>4.9</td>
<td>2.2</td>
<td>6.7</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Torrefied blend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/23/19</td>
<td>2:00</td>
<td></td>
<td>56.7</td>
<td>14.1</td>
<td>0.7</td>
<td>10.1</td>
<td>3.8</td>
<td>1.4</td>
<td>4.2</td>
<td>2.4</td>
<td>6.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Torrefied blend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8/23/19</td>
<td>22:59</td>
<td></td>
<td>56</td>
<td>14.9</td>
<td>0.7</td>
<td>8.5</td>
<td>4.3</td>
<td>1.1</td>
<td>3.8</td>
<td>2.2</td>
<td>8.8</td>
<td>0.4</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Coal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7-3: Target operating parameters of Hunter boiler, Unit-3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firing Rate</td>
<td>MW</td>
<td>430</td>
</tr>
<tr>
<td>Primary Gas/Coal Ratio</td>
<td>Unitless (mass)</td>
<td>1.97</td>
</tr>
<tr>
<td>Secondary Air Distribution</td>
<td>Unitless (mass)</td>
<td>30/70</td>
</tr>
<tr>
<td>(Inner/Outer)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower Furnace Stoichiometric Ratio</td>
<td>Unitless</td>
<td>0.9</td>
</tr>
<tr>
<td>Excess O2</td>
<td>Vol %, dry</td>
<td>3 - 4</td>
</tr>
</tbody>
</table>

7.2.4 Experimental conditions

The boiler operating conditions were assigned for typical low-NOx operation and
controlled to the best of the operators’ ability. The key parameters of the test plan of this program
are listed in Table 7-3.
Table 7-4: Actual schedule of ash deposit and ash aerosol sampling.

<table>
<thead>
<tr>
<th>Date in</th>
<th>Time in</th>
<th>Date Out</th>
<th>Time out</th>
<th>Sample</th>
<th>Fuel</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/22/2019</td>
<td>20:10</td>
<td>8/22/2019</td>
<td>20:50</td>
<td>30 Min Deposit Sample</td>
<td>Blend</td>
</tr>
<tr>
<td></td>
<td>20:50</td>
<td>8/22/2019</td>
<td>21:50</td>
<td>60 Min Deposit Sample</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td></td>
<td>22:51</td>
<td>8/23/2019</td>
<td>00:21</td>
<td>SMPS APS</td>
<td>Blend</td>
</tr>
<tr>
<td></td>
<td>00:37</td>
<td>8/23/2019</td>
<td>03:32</td>
<td>BLPI</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td>8/23/2019</td>
<td>05:13</td>
<td>8/23/2019</td>
<td>06:43</td>
<td>90 Min Deposit Sample</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td></td>
<td>17:53</td>
<td>8/23/2019</td>
<td>18:23</td>
<td>30 Min Deposit Sample</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td></td>
<td>18:38</td>
<td>8/23/2019</td>
<td>19:38</td>
<td>60 Min Deposit Sample</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td></td>
<td>23:15</td>
<td>8/24/2019</td>
<td>01:43</td>
<td>BLPI</td>
<td>Sootblower operation</td>
</tr>
<tr>
<td>8/24/2019</td>
<td>02:31</td>
<td>8/24/2019</td>
<td>04:01</td>
<td>90 Min Deposit Sample</td>
<td>Sootblower operation</td>
</tr>
</tbody>
</table>

7.3 Results and discussion

7.3.1 Actual operating conditions

It was arranged through dispatch that Unit 3 of Hunter power plant operated at the test conditions for approximately 24 hours for each of the two fuels. While operating at these steady-state conditions, ash deposit samples along with fly ash aerosol sampling were collected for 30, 60- and 90-minute periods in the location previously described. Sootblowers, which are lances with high velocity steam jets used to clean deposits off of heat transfer surfaces, were switched off during the sampling, but where operated in between sampling cycles. The actual schedule for deposit and aerosol sampling is presented in Table 7-4.
Table 7-5: Average measured operating conditions for each fuel.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Torrefied blend</th>
<th>Coal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net Firing Rate</td>
<td>MW</td>
<td>428.1</td>
<td>428.9</td>
</tr>
<tr>
<td>Fuel Rate</td>
<td>kg/hr</td>
<td>184,000</td>
<td>168,000</td>
</tr>
<tr>
<td>Primary Air Rate</td>
<td>kg/hr</td>
<td>401,000</td>
<td>395,000</td>
</tr>
<tr>
<td>Primary Air Temp</td>
<td>K</td>
<td>481</td>
<td>451</td>
</tr>
<tr>
<td>Secondary Air Rate</td>
<td>kg/hr</td>
<td>235,000</td>
<td>235,000</td>
</tr>
<tr>
<td>Secondary Air Temp</td>
<td>K</td>
<td>543</td>
<td>543</td>
</tr>
<tr>
<td>Secondary Air Swirl</td>
<td>%</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>OFA Flow Rate</td>
<td>kg/hr</td>
<td>536,000</td>
<td>528,000</td>
</tr>
<tr>
<td>OFA Temperature</td>
<td>K</td>
<td>543</td>
<td>543</td>
</tr>
<tr>
<td>Flue Gas O₂</td>
<td>Vol %, Dry</td>
<td>3.3</td>
<td>3.3</td>
</tr>
</tbody>
</table>

Table 7-6: Operating parameters (measured and calculated) relevant to the particle deposition behavior.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Torrefied blend</th>
<th>Coal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured Flue Gas Temperature*</td>
<td>K</td>
<td>1355</td>
<td>1307</td>
</tr>
<tr>
<td>Coupon Surface Temperature</td>
<td>K</td>
<td>811±10</td>
<td>811±10</td>
</tr>
<tr>
<td>Calculated Flue Gas Density [132]</td>
<td>kg/m³</td>
<td>0.263</td>
<td>0.273</td>
</tr>
</tbody>
</table>

* The flue gas was measured after 60-minute tests of ash deposit probes

Some of the operating conditions, including reactant flowrates and measured gas compositions were averaged during the 90-minute tests of torrefied blend and coal, which are summarized in Table 7-5. The measured and calculated operating parameters that are relevant for aerosol and deposit sampling are summarized in Table 7-6. A suction pyrometer probe was used to measure the flue gas temperature at the two sample locations. A summary of the plant operating data of Hunter Unit-3 is presented in Appendix C: Summary of the PI data of Hunter#3.
7.3.2 The plant instrumentation (PI) data analysis

The plant instrumentation (PI) data provides a data collection particularly in terms of temperature, pressure, flowrate, pulverizer performance, boiler load, and gas species concentration. Figure 7-4 presents selections of the PI data concerning fuel flowrate, pulverizer motor current, NOx, and SO2 in the flue gas. As displayed in Figure 7-4 (top left), the fuel flowrates during the torrefied blend tests were about 6.6% higher than those of coal combustion. This issue is expected since the high heating value of torrefied blend is less than that of coal (see Table 7-1); and the higher fuel flowrate is required to meet 430 MW. In addition, the PI data of pulverizer motor current (top right) shows that the torrefied blend increased the required Amps for milling the fuel particles. This issue supports our discussion in Chapter 4 that the co-milling woody biomass and coal impacts the pulverizer performance and leads to higher power requirements compared to coal milling. Moreover, as displayed in Figure 7-4 (bottom), the generated NOx and SO2 in the flue gas during torrefied blend were reduced, which were 88.6% and 71.9% of coal combustion, respectively. The reduction of NOx and SO2 in the flue gas during cofiring torrefied biomass with coal is partially due to lower contents of S and N in the fuel analysis of torrefied blend compared to coal (see Table 7-1). In addition, lower combustion temperature and larger devolatilization of torrefied blend combustion compared to pure coal combustion can lead to the reduction of NOx emissions [18-20]. Additionally, there is likely sulfur adsorption on to the blended ash particles.
Fly ash aerosols

Figure 7-5 shows the comparison of particle size distributions (PSDs) of the entrained ash between the Coal and Torrefied blend conditions. The results presented on the left are based on standard flue gas volume while results presented on the right are based on input ash mass which accounts for the gas volume change. Although the particle sizers and the impactor produced inconsistent concentrations on particles smaller than 0.1 μm, no significant variation is found between coal and torrefied blend within the same analyzing method. Coal combustion has slightly higher ash concentration in flue gas than the pure torrefied woody biomass, as we expected, but the PSD remained the same for each of the conditions fuel conditions.
The PSDs shown in Figure 7-5 display two modes. The smaller mode peaks at ~0.6 μm and exists for both fuel conditions and is likely generated through the nucleation and coagulation of inorganic species that were vaporized and later condensed. The larger mode peaking at about 4 μm, is formed by super-micron particles released from char fragmentation. Results based on the input ash mass suggest that this mechanism in torrefied blend is similar to coal combustion.

![Figure 7-5: PSDs of ash aerosols in baseline and cofiring cases.](image)

The size-segregated particulate samples collected by the BLPI where analyzed by Scanning Electron Microscopy-Energy Dispersive Spectroscopy (SEM-EDS) and the results are presented in Figure 7-6. In general, all the elements display a similar correlation between size and composition for coal and the torrefied blend. Torrefied wood has more magnesium and calcium in original ash analysis, which are likely available in the form of organically bound ions. These metals are released from biomass matrix during volatilization to form vaporized ash followed by condensation during cooling. This mechanism produces ultra-fine particles with coagulation to each other and generates the sub-micron ash aerosols enriched with such metals. As shown in Figure 7-6, Si and Al, which are the less volatile species, formed the super-micron mode aluminosilicate compounds in the PSDs. In addition, the measured chlorine is negligible compared to the other elements. It is expected that the alkali chlorine compositions (NaCl and KCl) in the
vapor phase can condense on the cold surfaces such as the deposit probe coupons to form new compositions with sulfur, which may lead to a sticky layer causing ash deposit growth. In the next chapter, which is about modelling the ash deposit rate, it is demonstrated that the condensation of K and Na as well as S and Cl on the cooler surfaces can form a sticky layer and increase the ash deposit growth.

Figure 7-6: Size-segregated compositions of ash aerosols in the two cases.

A cyclone with the cut-off size around 15.7 μm collects the bulk ash aerosols prior to the low-pressure impactor. Compositions of cyclone ash in the two cases are compared in log scale in Figure 7-7. It shows that the mass percentages of the fly ash particles of torrefied fuel and coal are almost identical. Meanwhile, sulfur concentration of the torrefied blend was higher than that of coal, which leads to increase the melting point of ash particles; and decrease their probabilities to stick to the heat exchanger surface (or coupon surface of ash deposit probe).
Figure 7-7: Compositions of ash aerosols (above 15.7 μm) collected by cyclone.

7.3.4 Ash deposition

Ash deposit samples were collected on our probe coupons, of which the surface temperature was controlled at 811 K by internal cooling air, with holding times of 30, 60 and 90 minutes in both coal combustion and cofiring torrefied blend. Figure 7-8 shows images of the collected ash deposit on the probe as it was removed from the furnace. The end cap was not cooled like the coupons and is therefore glowing. The chunk ashes on the cap end suggest that they are likely originated from the upstream heat exchanger surfaces during shedding events and seem to only adhere to the hot probe tip. It is important to observe that there was carryover of this deposit from the probe tip onto the coupon for the torrefied 60 min condition.

Figure 7-9 reports the total weights and growth rates of the ash deposits collected from the coupon surface, which does not include the end cap. The samples were carefully scrapped off and then weighed using a balance of 0.01-gram accuracy. Due to the significant ambient air convection outside the furnace and air leakage toward it, air entrained a small portion of the scrapped ash deposit from the coupon surface during scrapping. Therefore, it is assumed that the visual negative error of collecting ash deposit is estimated at -10%, meaning that 10% of the collected ash deposit
samples on the coupons were lost. The deposition rate was calculated as the total deposition mass on the coupon per the coupon surface area per time.

For the 30-minute holding time samples, the mass of ash deposits from the fuels are nearly identical. The ash deposit mass for the 60-minute holding time was for the torrefied blend was around twice that of coal, which was likely due to a chunk ash from ash shedding of upstream heat exchangers that carried over from the probe tip. This data point complicates interpretation of the data with the expected result that deposition rates while firing the torrefied blend should be similar to, or lower than, when firing the pure coal. However, the discrepancy is almost certainly due to the chunk of deposit growing from the hot probe tip. Obviously, there are different mechanisms at play for deposit in that area. The ash deposit mass for the 90-minute holding time leads to lower ash deposit for Torrefied blend than that of coal. Regarding Table 7-1, which presents the ash content of the fuels, Torrefied#2 blend, which represents the torrefied blend of 90-minute period, has lower ash content than that of coal. Therefore, it should lead to a lower ash deposition rate for 90-minute period as displayed in Figure 7-9. Higher Sulfur wt% of ash aerosol particles of torrefied blend, which is shown in Figure 7-7, leads to decrease in stickiness and supports this argument.
Figure 7-9: Ash deposition rates and deposit weights at different times

Figure 7-10: Ash deposit compositions sampled at different periods (from left to right: Torrefied 30, 60, 90 minutes, Coal 30, 60, 90 minutes).

Compositions of ash deposit samples are presented in Figure 7-10. The ash deposit sample of the torrefied blend of 60-minute holding was divided into two parts, the chunk part, and the powdery part on coupons, which were separated using a screen. This sample has higher silicon and lower calcium contents than the others, which supports the observation that the chunk deposit was likely not formed during the cofiring test, but a result from ash shedding from the upstream heat.
exchangers or a different mechanism that is impacted by the high temperature of the probe tip. Ash deposit compositions at 30 and 90 minutes only show slight differences in composition from the coal combustion to cofiring with torrefied biomass. Ca, Si, and Al are the main elements in ash deposit samples, meaning that they form aluminosilicate and calcium compounds.

Figure 7-11 shows SEM images of the ash deposit samples at 30 and 90 minute holding times in two scales of 300 µm and 500 µm. Both fuels produce super-micron spherical particles aggregated with micron-sized fine particles. The fine particles on the surface of the larger particles may represent the condensed alkali vapor species following nucleation to form sticky layer on the surface to catch the coming particles [124]. It is also found that the 30-minute deposits of the torrefied blend, 300 µm, contain more coarse particles in irregular shapes, while such sizes cannot be found in 90-minute deposit of the same scale (300 µm). This issue suggests that the larger particles within the ash deposit sample of 30-minute period of torrefied blend might be originated from the shakedown tests that were performed before the 30-minute Torrefied.

7.4 Conclusions

Ash aerosol and deposits were sampled on a full-scale boiler during Sufco coal combustion and cofiring of Sufco coal with torrefied biomass (85/15 wt. % coal/ torrefied woody biomass). Results from the two tests are compared. In general, no significant difference is found in fly ash aerosol size and composition as well as the ash deposit composition between the two fuel conditions. Some conclusions on the major findings are:

- The analysis of ash aerosol particles shows that there is no significant difference between coal and torrefied blend combustion. The mode of sub-microns is due to nucleation and
coagulation of condensed vapor species; and the mode of super microns is due to the fragmentation of the coarse particles.

Figure 7-11: SEM images of 30- and 90-minute deposits (top: Torrefied blend; bottom: Coal).

- The size-segregated compositions of ash aerosols and compositions of ash aerosols show that the coal and torrefied blend display a similar relationship between size and composition. Al and Si, which are the main elements of ash aerosols, contribute to aluminosilicate compositions. In addition, higher Sulfur contents of ash particles of above 15.7 µm may lead to increase the melting point of ash particles and decrease their sticking efficiency to the heat exchanger surfaces.
• The shape and weight of the ash deposit sample of 60-minute period is odd. It suggests that the shedding ash deposit from an upstream heat exchanger might impact the coupon surface of the ash deposit probe or grow from the hot probe tip.

• In general, the fuel types of this project do not influence significantly the ash deposit rate at low periods. However, since the ash content of torrefied blend is less than that of coal, and sulfur percentage of their coarse fly ash particles are higher than coal, they result in lower ash deposit rate for torrefied biomass than coal combustion.

• The fuel types that were used in this project do not change significantly the morphology of ash deposit samples. Since the morphology of 30-minute period of torrefied blend (300 µm) includes larger particles than that of coal, it suggests that such large particles either remained on the coupon surface from the shaking down tests or added to the ash deposit sample during sampling from outside.
8 A novel method to model ash deposit growth of a wide range of solid fuels in a 100 kW combustor.

8.1 Introduction

Inorganic elements of solid fuels, contribute mostly to ash and may deposit on the surfaces with a lower temperature than the flue gas [137]. Alkali contents of woody biomass, like sodium (Na) or potassium (K), evaporate during combustion and later condense with the other elements such as chlorine (Cl) and sulfur (S) to make alkali rich ash deposits. Such deposits have a low melting point and create a sticky layer on heat exchanger surfaces [14, 68]. Alkali elements can additionally react with silicon (Si) to produce other compounds with low-melting points, further exacerbating ash deposition [138].

In addition to chemical composition of the ash, many physical properties affect the deposit mechanism and the ash deposition rate. These include ash particle properties such as particle size distribution (PSD), composition, velocity, density, and temperature as well as flue gas properties including temperature, viscosity, density, and velocity [139]. Slag deposition, which is a liquid-like ash deposition to the boiler walls in radiation zone [81], is beyond the scope of this research. Our focus is on fouling, which occurs at gas and particle temperatures below the ash fusion temperature.

Modelling may be utilized to further our understanding of how the factors discussed above may affect systems where woody biomass is cofired with coal [114]. There are many publications concerning the modeling of the ash deposition rate, although there is not yet a complete
understanding of ash transformations and deposition [140]. Most of the published models are based upon Computational Fluid Dynamics (CFD) and include complex sub-models of gas phase and particle phase [114, 141-148]. In addition, some of them ignore the contributions of condensation and eddy impaction, which is addressed in this study. Other researchers have developed models to predict the fly ash formation [149, 150]; however, this is beyond of the scope of this study. S. S. Lokare et al. [100] developed an ash deposition model using Fluent and C++ to predict the rates and mechanisms of ash deposition while cofiring straw and coal. Their model assumed that the total deposition rate is dependent on inertial impaction, condensation, and eddy impaction; however, their ash deposition rate was not time dependent. Further reviews of ash deposition models have also been published [114, 115, 141, 151, 152].

Despite these advances, there are few modelling studies that address the effect of biomass type and its ratio in cofiring with coal [130]. The objective of this paper is to generate a model to help the operators of utility boilers readily predict the ash deposit rate on the heat exchanger surfaces for a wide range of solid fuels including coal, biomass, and their blends. Our developed model is not based on CFD and accounts for geometry only in the region of a heat transfer surface. To fulfill such objective, a model is developed which includes four ash deposit mechanisms: inertial impaction, thermophoresis, condensation, and eddy impaction [14, 51, 67, 84-86]. The stickiness of impacting ash particles in the model is evaluated using two separate stickiness models. The predicted ash deposition rates of a 100-kW boiler are then compared with the corresponding experimental data and the dominant ash deposit mechanisms for all surfaces of our probe are determined.
8.2 Materials and methods

8.2.1. Experimental work

The following section details the various materials necessary to conduct the trials in this study. This includes the milling system, combustor, fuels, and probe.

*Milling System Description:* Before entering the combustor, a milling system pulverizes the dry solid fuels. Some studies have shown that reducing the size of woody biomass particles to the small sizes of coal particles is difficult due to the different physical properties of biomass from coal, such as density and aspect ratio [19]. Usually, the result is that large biomass particles exit the mill early, shifting the PSD to larger sizes. The critical processing steps of one such mill, a Raymond Bowl Mill, were investigated elsewhere by milling a pure bituminous coal and blends of processed woody biomass with coal to simulate the behavior of the fine particles before exiting the pulverizer [116, 118, 153, 154].

*Combustor Description:* The experimental work was conducted in a down-fired 100 kW boiler, which is called an oxy-fuel combustor (OFC). The combustor, shown in Figure 8-1, was designed to have three zones: ignition, radiation, and convection. There are nine pairs of ports in the vertical section of the OFC for sampling and observation. The ignition zone extends from ports 1 to 3 of the OFC and has an inner diameter of 0.37m, an outer diameter of 0.76m and a total length of 1.22m. After a transition zone, the radiation zone extends from ports 5 to 9 (0.27m x 0.61m x 2.60m/I.D. x O.D. x L). The radiation zone is followed by a horizontal convection zone (0.15m x 0.15m x 3.66m/I.D. x O.D. x L) which is comprised of eight heat exchangers [122]. The burner used for the experiments was a one register swirl burner which facilitates the mixing of the fuel and oxidant [155]. Data was collected with an ash deposit probe through port 6. It is also important
to note that the peak temperatures for the reactor occur in the ignition zone occurs around port 2, which is difficult to measure [122]. More details about OFC can be found elsewhere [67, 122].

![Figure 8-1: Configuration of 100 kW oxy-fuel combustor (OFC).](image)

**Fuel Description:** Eleven solid fuels were combusted in OFC including a) Rice husk (RH), b) Bituminous Utah Sufco coal #1 (Sufco#1), c) Bituminous Utah Sufco coal #2 (Sufco#2), a d) blend of 20 wt. % Rice husk with 80 wt. % Sufco coal #1 (20RH-80Sufco#1), e) Powder River Basin sub-bituminous coal (PRB), f) a blend of 13 wt. % Rice husk with 87 wt. % PRB (13RH-87PRB), g) Petroleum Coke (Petcoke), h) Illinois bituminous coal (Illinois), i) a blend of 60 wt. % Illinois bituminous coal with 40 wt. % PRB (60Illinois-40PRB), j) Torrefied woody biomass (Torrefied), and k) a blend of 50 wt. % Torrefied woody biomass with 50 wt. % Sufco coal #2 (50Torrefied-50Sufco#2). The fuel analysis and mineral ash analysis are presented in Table 8-1 and Table 8-2, respectively.
Table 8-1: Ultimate and Proximate Fuel Analysis.

<table>
<thead>
<tr>
<th>Sample</th>
<th>ASH (%)</th>
<th>C (%)</th>
<th>H (%)</th>
<th>N (%)</th>
<th>S (%)</th>
<th>O (diff) (%)</th>
<th>H2O (%)</th>
<th>Volatile (%)</th>
<th>FC (%)</th>
<th>HHV (kJ/kg) (%)</th>
<th>Cl (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) RH</td>
<td>33.67</td>
<td>28.47</td>
<td>4.15</td>
<td>1.05</td>
<td>0.10</td>
<td>24.42</td>
<td>8.16</td>
<td>48.96</td>
<td>9.22</td>
<td>11,551</td>
<td>0.071</td>
</tr>
<tr>
<td>b) Sufco#1</td>
<td>8.36</td>
<td>67.87</td>
<td>4.77</td>
<td>1.09</td>
<td>0.36</td>
<td>11.44</td>
<td>6.11</td>
<td>38.49</td>
<td>47.04</td>
<td>27,677</td>
<td>0.047</td>
</tr>
<tr>
<td>c) Sufco#2</td>
<td>13.96</td>
<td>62.41</td>
<td>4.52</td>
<td>1.10</td>
<td>0.46</td>
<td>11.04</td>
<td>6.52</td>
<td>37.36</td>
<td>42.16</td>
<td>27,319</td>
<td>*---</td>
</tr>
<tr>
<td>d) 20RH-80Sufco#1</td>
<td>13.42</td>
<td>59.99</td>
<td>4.67</td>
<td>1.08</td>
<td>0.31</td>
<td>14.04</td>
<td>6.52</td>
<td>40.58</td>
<td>39.48</td>
<td>24,451</td>
<td>*---</td>
</tr>
<tr>
<td>e) PRB</td>
<td>4.94</td>
<td>53.72</td>
<td>3.59</td>
<td>0.78</td>
<td>0.23</td>
<td>13.05</td>
<td>23.69</td>
<td>33.36</td>
<td>38.01</td>
<td>21,115</td>
<td>*---</td>
</tr>
<tr>
<td>f) 13RH-87PRB</td>
<td>8.67</td>
<td>50.44</td>
<td>3.66</td>
<td>0.82</td>
<td>0.21</td>
<td>14.53</td>
<td>21.67</td>
<td>35.39</td>
<td>34.27</td>
<td>19,871</td>
<td>*---</td>
</tr>
<tr>
<td>g) Petcoke</td>
<td>2.99</td>
<td>82.51</td>
<td>6.02</td>
<td>1.71</td>
<td>0.56</td>
<td>0.49</td>
<td>0.57</td>
<td>10.18</td>
<td>86.26</td>
<td>35,720</td>
<td>*---</td>
</tr>
<tr>
<td>h) Illinois</td>
<td>9.42</td>
<td>63.47</td>
<td>4.36</td>
<td>1.24</td>
<td>3.12</td>
<td>8.76</td>
<td>6.96</td>
<td>36.04</td>
<td>44.90</td>
<td>26,870</td>
<td>*---</td>
</tr>
<tr>
<td>i) 60Illinois-40PRB</td>
<td>7.63</td>
<td>59.57</td>
<td>4.05</td>
<td>1.06</td>
<td>1.96</td>
<td>10.48</td>
<td>15.26</td>
<td>34.97</td>
<td>42.14</td>
<td>24,567</td>
<td>*---</td>
</tr>
<tr>
<td>j) Torrefied</td>
<td>0.19</td>
<td>51.75</td>
<td>5.29</td>
<td>0.14</td>
<td>0.02</td>
<td>36.29</td>
<td>6.32</td>
<td>74.20</td>
<td>19.29</td>
<td>21,534</td>
<td>0.03</td>
</tr>
<tr>
<td>k) 50Torrefied-50Sufco#2</td>
<td>7.08</td>
<td>57.08</td>
<td>4.91</td>
<td>0.62</td>
<td>0.24</td>
<td>23.67</td>
<td>6.42</td>
<td>55.78</td>
<td>30.73</td>
<td>24,427</td>
<td>*---</td>
</tr>
</tbody>
</table>

* Cl either was not measured or was under the detection limit.

Table 8-2: Mineral ash analysis.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Al2O3 (%)</th>
<th>CaO (%)</th>
<th>Fe2O3 (%)</th>
<th>MgO (%)</th>
<th>MnO (%)</th>
<th>P2O5 (%)</th>
<th>K2O (%)</th>
<th>SiO2 (%)</th>
<th>Na2O (%)</th>
<th>SO3 (%)</th>
<th>TiO2 (%)</th>
<th>NiO (%)</th>
<th>V2O5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) RH</td>
<td>1.73</td>
<td>1.31</td>
<td>1.1</td>
<td>0.84</td>
<td>0.83</td>
<td>1.81</td>
<td>2.66</td>
<td>88.51</td>
<td>0.31</td>
<td>0.32</td>
<td>0.18</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>b) Sufco#1</td>
<td>8.34</td>
<td>18.21</td>
<td>5.25</td>
<td>2.84</td>
<td>0.05</td>
<td>0.01</td>
<td>0.33</td>
<td>48.85</td>
<td>3.09</td>
<td>5.96</td>
<td>0.64</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>c) Sufco#2</td>
<td>12.09</td>
<td>11.9</td>
<td>3.62</td>
<td>3.94</td>
<td>0.03</td>
<td>0.25</td>
<td>1.13</td>
<td>62.48</td>
<td>0.81</td>
<td>1.83</td>
<td>0.68</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>d) 20RH-80Sufco#1</td>
<td>5.03</td>
<td>9.76</td>
<td>3.18</td>
<td>1.84</td>
<td>0.44</td>
<td>0.91</td>
<td>1.5</td>
<td>68.68</td>
<td>1.7</td>
<td>3.14</td>
<td>0.41</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>e) PRB</td>
<td>14.78</td>
<td>22.19</td>
<td>5.2</td>
<td>5.17</td>
<td>0.01</td>
<td>1.07</td>
<td>0.35</td>
<td>30.46</td>
<td>1.94</td>
<td>8.83</td>
<td>1.3</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>f) 13RH-87PRB</td>
<td>8.26</td>
<td>11.75</td>
<td>3.15</td>
<td>3.01</td>
<td>0.42</td>
<td>1.44</td>
<td>1.51</td>
<td>59.49</td>
<td>1.13</td>
<td>4.58</td>
<td>0.74</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>g) Petcoke</td>
<td>19.4</td>
<td>4.22</td>
<td>7.02</td>
<td>0.66</td>
<td>0.06</td>
<td>0.18</td>
<td>1.17</td>
<td>46.7</td>
<td>0.72</td>
<td>3.77</td>
<td>0.63</td>
<td>1.26</td>
<td>8.24</td>
</tr>
<tr>
<td>h) Illinois</td>
<td>20.18</td>
<td>3.22</td>
<td>16.46</td>
<td>0.89</td>
<td>0.03</td>
<td>0.1</td>
<td>2.1</td>
<td>51.22</td>
<td>1.06</td>
<td>2.79</td>
<td>0.98</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>i) 60Illinois-40PRB</td>
<td>18.02</td>
<td>10.81</td>
<td>11.96</td>
<td>2.60</td>
<td>0.02</td>
<td>0.49</td>
<td>1.40</td>
<td>42.92</td>
<td>1.41</td>
<td>5.21</td>
<td>1.11</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>j) Torrefied</td>
<td>2.67</td>
<td>51.72</td>
<td>8.28</td>
<td>10.39</td>
<td>4.73</td>
<td>4.16</td>
<td>4.61</td>
<td>6.82</td>
<td>1.6</td>
<td>5.03</td>
<td>0</td>
<td>*---</td>
<td>*---</td>
</tr>
<tr>
<td>k) 50Torrefied-50Sufco#2</td>
<td>11.95</td>
<td>12.33</td>
<td>3.7</td>
<td>4.04</td>
<td>0.1</td>
<td>0.29</td>
<td>1.32</td>
<td>61.71</td>
<td>0.83</td>
<td>1.84</td>
<td>0.67</td>
<td>*---</td>
<td>*---</td>
</tr>
</tbody>
</table>

* Not measured.
**Ash Deposition Probe Description**: A temperature-controlled ash deposit probe was used to determine deposition rate and profile for various fuels and operating conditions. The ash deposit collected from the horizontal surface of the probe, which is perpendicular to the flow direction of flue gas, includes inside (initial) deposit, outside deposit, and side deposit. The inside layer comprises the initial ash deposit, which is sticky, and should be scraped to be collected. The outside deposit, which is formed by large fly ash particles, can be collected by vigorous shaking; and it is different in composition than the inside layer [122]. The length and the diameter of the coupon are 7.37 and 6.03 (cm), respectively. More detailed information about the ash deposit probe can be found elsewhere [67, 122]. The probe was installed in the OFC for durations of 30, 60, and 120 minutes for each fuel condition. The calculated ash deposit mass is the sum of the collected ash deposit on the outside, inside, and sides of the coupon surface (see Figure 8-2) and the rate is the mass divided by the duration of installation.

**Figure 8-2**: Locations of ash deposit on the surface of heat exchanger tube depicted.

8.2.2. **Model description**

As the materials were described, we will now detail the model used to accurately predict the ash deposit formation in the combustor. We will first begin with a description of the
assumptions used and then move to a mathematical description of the various ash deposit mechanisms on the probe surface. Lastly, we will detail several other important inputs into our model including the ash deposit thickness, impaction efficiency, heat energy balance, ash deposit thermal conductivity, and the sticking efficiency. In the latter section we will present a comparison of this model to an existing sticking efficiency model and their ability to estimate the ash deposit rate.

*Model assumptions:* The main assumptions of this model are presented as follows:

1. The temperature of the particle before impacting the probe surface is equal to the temperature of the surrounding flue gas [156].
2. The temperature gradient inside the ash particles is neglected.
3. The thermal boundary layer thickness is obtained from the average Nusselt number, which accounts for the average convective heat transfer coefficient.
4. The eroding effect of the high momentum solid large particles on deposit surface, which leads to either natural or artificial shedding, is neglected. This is a reasonable assumption at short ash deposition times.
5. The geometry of the ash deposit surface may influence the deposition of ash particles on the probe surface. However, since this level of detail requires CFD modeling, and is, therefore, neglected in this study. We also believe that this is a reasonable assumption at short deposition times.
6. The temperature of the π side of the probe (see Figure 8-2) is assumed equal the probe surface temperature. This assumption is used for calculating thermophoresis and condensation at the π side.
7. The release fraction of Na and K from the fuel to the flue gas phase are the same.
8. The melt fractions of ash particles and the ash deposit are identical at the same temperatures; and obtained from thermodynamic modeling using FactSage 7.3 software by the University of Utah.

9. The side ash deposits of some tests were not measured. However, since the inside and side ash deposits are almost identical, it is assumed that the mass of side ash deposit is equal to the inside ash deposit.

**Ash Deposit Mechanisms:** In the present model, four essential mechanisms of ash deposition are considered including inertial impaction, thermophoresis, condensation, and eddy impaction. Particles above 10 µm, which do not follow the gas streamline, have enough force to contact with the heat exchanger surface and make inertial impaction ash deposit [85, 86]. Thermophoresis causes ash deposit due to the induced force on the particles because of the temperature gradient. Furthermore, condensation of vapor species on the colder surfaces whose temperature is lower than the dew point of species in the flue gas also leads to ash deposit on the probe [85]. Lastly, eddy impaction caused ash deposits occur when turbulent flows provide enough momentum to the fine ash particles to impact π side of the heat exchanger surfaces [67, 100]. A schematic diagram of ash deposit mechanisms for fouling formation is presented in Figure 8-3. The definition and a mathematical representation of each of these mechanisms is presented in detail in this paper.

**Inertial Impaction:** Inertial impaction is experienced predominately from large ash particles (typically > 10 µm) and is a strong function of velocity [132]. Particle velocity is governed by the momentum equation of a particle, which is determined by the balance of drag force, gravitational force, and the other forces as shown in the following equation [157]:

\[
\frac{dp}{dt} = \frac{1}{\rho} \left[ F_d - \nabla \cdot \left( \rho \vec{v} \vec{v} \right) + \rho \vec{g} + \mathbf{F}_{\text{other}} \right]
\]
Figure 8-3-Schematic diagram of fouling formation mechanisms.

\[
\frac{du_p}{dt} = \frac{18\mu_g C_D Re_p}{\rho_p d_p^2} \left( u_g - u_p \right) + \frac{g(\rho_p - \rho_g)}{\rho_p} + \text{Force}_{\text{balance}},
\]  

where \( u, \mu, \rho \) and \( d \) are the velocity, viscosity, density, and diameter, respectively. The subscripts of \( p \) and \( g \) mean particle and flue gas, respectively. The viscosity of flue gas is calculated using a simple equation by S.B. Hansen et al. [132]. \( C_D \) is the drag coefficient elsewhere [158], \( \text{Force}_{\text{balance}} \) is the balance of the other body forces, such as the thermophoretic force [83], and \( g \) is the gravitational constant (9.8 m·s\(^{-2}\)). Furthermore, \( Re_p \) refers to the particle Reynolds number \( (Re_p = \rho_g d_p v_t / \mu_g) \), where \( v_t \) is the terminal velocity of particle and is assumed to be identical to the flue gas velocity as both the particle and the fluid flow in the same direction and approximately at the same velocities. For other equations, it will be necessary to define the inertial impaction velocity of a particle \( (u_{in}) \), in which its time variation \( \frac{du_{in}}{dt} \) is assumed to be the balance of the first two terms of the right side of Eq. (8-1).
**Thermophoresis:** The thermophoresis mechanism is a phenomenon driven by the temperature gradient between the high-temperature small particles and the low-temperature probe surface. The flue gas molecules at the hot side of an ash aerosol particle, which are typically in the range of \(0.1 \mu m < d_p < 1.0 \mu m\), impinge and carry the particle towards the cold side [152, 159]. A thermophoretic force in the negative direction to such temperature gradient is a result [160]. Thermophoretic force \(F_{th}\), which is governed by the geometric relationship between the particle and the cold surface, is expressed by the following formula [161]:

\[
F_{th} = \Phi \frac{d_p \nabla T_{g-s} \mu_g^2}{2T_g \rho_g}, \quad (8-2)
\]

where \(T_g\) is the flue gas temperature, and \(\nabla T_{g-s}\) is temperature gradient between flue gas and the surface temperature. \(\Phi(\Lambda, Kn)\) is a function of the ratio of thermal conductivity of the particle to that of the flue gas \((\Lambda = k_p/k_g)\) and Knudsen number \((Kn)\) [96, 160-163]. The particle thermal conductivity, \(k_p\), is assumed constant at 2.0 W/m/K; and the flue gas thermal conductivity, \(k_g\), can be estimated using the references [132, 161]. For later equations, we will refer to the thermophoretic velocity \(u_{th}\) which is the particle velocity due to thermophoresis and can be derived from the thermophoretic force elsewhere [162]:

\[
u_{th} = \frac{3 \mu_g C_c \nabla T_{g-s}}{18 \pi \rho_g T_g (1 + 0.15 Re_p^{0.687})}, \quad (8-3)
\]

where \(C_c\) is the Stokes-Cunningham factor:

\[
C_c = 1 + Kn(1.257 + 0.4e^{-1.1Kn}) \quad (8-4)
\]

The effect of thermal conductivity ratio and Knudsen number on the thermophoretic force is presented in Figure 8-4. As displayed in Figure 8-4, the graphs of the expression \(-\Phi/2\pi\) are a function of thermal conductivity ratio \((\Lambda)\) and Knudsen number \((Kn)\). The negative sign represents
the direction of the thermophoretic force towards the cold surface. The expression $-\Phi/2\pi$ is smaller at the greater thermal conductivity ratios and the smaller $Kn$ numbers, which indicates that the thermophoretic force declines. $-\Phi/2\pi$ is independent on the values of the conductivity ratios at $Kn > 1.0$.

![Figure 8-4: Expression $-\Phi/2\pi$ as a function of Knudsen number ($Kn$) and thermal conductivity ratio ($\Lambda$) [162].](image)

**Condensation**: Ash deposition by condensation occurs when vapors can condense on the probe surface. The mass flux of vapor condensation, $I_{\text{cond}}$, is determined by the following expressions [113, 164]:

$$I_{\text{cond}} = Sh(T_g) \left( \frac{D_v(T_g)D_v(T_\text{s})}{D_h R_g} \right)^{1/2} \left[ \frac{p_v(T_g)}{T_g} - \frac{p_{v,\text{s}}(T_\text{s})}{T_\text{s}} \right], \quad (8-5)$$

$$Sh(T_g) = 0.023 \cdot Re^{0.8} \cdot Sc(T_g)^{0.4} \quad (8-6)$$

$$Sc(T_g) = \mu_g / (\rho_g D_v(T_g)), \quad (8-7)$$

where $D_h$ is the hydraulic diameter of probe diameter and $R_g$ is the universal gas constant. $D_v$, which was determined by S.B. Hansen et al. [132], is the diffusion coefficient of vapor at flue gas
temperature, $D_v(T_g)$, or the deposit surface temperature, $D_v(T_s)$. Other variables include $p_v(T_g)$, which is the partial vapor pressure of the condensing components, and $p_{v,s}(T_s)$, which is the saturation vapor pressure at the deposit surface temperature. The Reynolds number for Eq. (8-6) is defined as $Re = \rho_g \bar{U} D_h / \mu_g$ where $\bar{U}$ is the magnitude of the flue gas velocity in this study.

**Eddy impaction:** Eddy impaction is a process by which the fine ash particles can impact the $\pi$ side of the probe surface (see Figure 8-3) once they obtain a high enough momentum to follow the eddy streams [101, 156]. Circular vortices that cause the eddy streams form at Reynolds numbers of about 4.0 [165]. Only the particles with a diameter smaller than $d_{p-eddy}$, given by the following expression by W. D. Bachalo [166], may respond to eddy streams:

$$d_{p-eddy} \leq \sqrt{D_h \left[ \frac{u_{rms} \rho_g \mu_g}{\rho_g + 1} \right]}^{\frac{1}{2}},$$  \hspace{1cm} (8-8)

where $u_{rms}$ is root-mean-square speed of the flue gas. Particles with larger diameters may cross eddy streamlines and impact and stick on the upstream side of the probe surface due to their inertia [156]. The flux of ash deposition by eddy impaction, $I_{eddy}$, can be expressed by the following equation [156]:

$$I_{eddy} = u_r C_{eddy} f_p,$$  \hspace{1cm} (8-9)

where $u_r$ is the maximum reverse flue gas velocity, assumed to be equal to the gas velocity [156], $\bar{U}$, in the present study, and $C_{eddy}$ is the mass concentration of eddy particles. There is a lack of information in the literature about the definition of $C_{eddy}$. The authors of this study define it as a function of $Re$ and the concentration of fly ash particles upstream from the probe ($C_p$) as follows:

$$C_{eddy} = 8 \times 10^{-6} \cdot Re \cdot C_p.$$  \hspace{1cm} (8-10)
Therefore, even at very high \( Re \) numbers \( (Re \approx 10^5) \), \( C_{eddy} \) is less than \( C_p \). It should be noted that \( Re < 1000 \) for this study. Lastly, the probability of impacting and sticking the \( \pi \) side of probe surface \( (f_p) \) is determined by the following equation [156]:

\[
f_p = k_c \cdot Sr \cdot u_g / D_h,
\]

where \( k_c \) is a lumped empirical parameter including impaction and sticking efficiencies, and it is assumed 1.1 in this study [156]. Strouhal number, \( Sr \), is assumed to be 0.2 for the Reynolds numbers of the tubes at the range of \( 250 < Re < 2 \times 10^5 \) [167].

**Ash deposit thickness:** The ash deposit thickness on a coupon as a function of time is determined by X. Yang et al. [83] as follows:

\[
\frac{dL_{deposit}}{dt} = \frac{\eta_{stick} A_{arrival} + I_{cond}}{\rho_p (1 - \varepsilon_{deposit})},
\]

where \( L_{deposit} \), is the ash deposit thickness and \( \eta_{stick} \) is the sticking efficiency, or the ratio of the number of the particles that stick to the surface to the number of particles that impact the surface [90]. X. Yang et al. [83] defined \( A_{arrival} \) as the flow flux of the arrival of ash particles due to the inertial impaction and thermophoresis in a CFD model. Since this study is not a CFD model, it is modified by replacing \( A_{arrival} \) with \( \eta_{imp} C_p (u_{in} + u_{th}) \), in which \( \eta_{imp} \) is the impaction efficiency of fly ash particles surrounding the probe, and \( C_p \) is the concentration of fly ash particles also used in Eq. (8-12). The impaction efficiency \( \eta_{imp} \) is the ratio of the number of the impacted particles to the probe surface to the total number of the particles directed to the surface by the flue gas [168]. The inertial impaction velocity of a particle \( (u_{in}) \) and the thermophoretic velocity \( (u_{th}) \) were introduced previously. \( C_p \), is obtained by the following equation:

\[
C_p = \frac{x_{ash} \cdot m_{fuel} \cdot \rho_g}{m_g}
\]

(8-13)
where \( x_{\text{ash}} \) is the ash mass fraction defined as the ratio of the mass of ash to the mass of fuel, \( \dot{m}_{\text{fuel}} \) is the fuel rate defined as the ratio of the mass of fuel to time, \( \rho_g \) is the flue gas density, and \( \dot{m}_g \) is the flue gas flow rate defined as the ratio of the mass of flue gas to time.

The ash deposit porosity is also an important factor in determining ash deposit thickness. Ash deposit porosity \( \varepsilon_{\text{deposit}} \) varies according to the changing volume of liquid \( (V_l) \) and solid \( (V_s) \) phases of ash deposit as follows [169]:

\[
\varepsilon_{\text{deposit}} = 1 - \left[ (1 - \varepsilon_0) + \frac{V_l}{V_s} (1 - \varepsilon_0) \right],
\]

(8-14)

where \( \varepsilon_0 \) is the initial ash deposit porosity, which is assumed at the range of 0.6-0.9 [132, 169, 170]. Volume of liquid phase \( (V_l) \) is calculated by using the melt fraction of ash deposit and estimating its density, which is obtained from the literature [171, 172]. K. C. Mills and B. J. Keene [173] presented a review about the density of liquid phase of ash as a function of temperature and chemistry which can be used in connection with the volume of the liquid phase \( (V_l) \) mentioned previously.

**Impaction efficiency:** Impaction efficiency is expressed as a function of the effective Stokes number, \( St_{\text{eff}} \), which is calculated by the correlation of R. Israel et al. [174], and is valid for \( St_{\text{eff}} > 0.14 \):

\[
\eta_{\text{imp}} = \left[ 1 + 1.25 (St_{\text{eff}} - 0.125)^{-1} - 0.014 (St_{\text{eff}} - 0.125)^{-2} + 0.000508 (St_{\text{eff}} - 0.125)^{-3} \right]^{-1},
\]

(8-15)

The effective Stokes number is defined as follows [175]:

\[
St_{\text{eff}} = \psi \cdot St,
\]

(8-16)

where the Stokes number is expressed by \( St = \frac{\rho_p d_p^2 g}{9 \mu_g D_h} \) [175]. The particle Reynolds (\( Re_p \)) number determines Stokes correction factor \( (\psi) \) [175]:

112
\[ \psi = \frac{18}{Re_p} \left( R_{ep}^\frac{1}{3} - \sqrt{6} \tan^{-1}\left( \frac{R_{ep}}{\sqrt{6}} \right) \right). \] (8-17)

The impaction efficiency increases with increasing particle size. Particles with \( St < 0.1 \) follow the flow streamlines and their impaction efficiencies cannot be predicted by available equations [152]. Therefore, a constant impaction efficiency is assumed for such particles in this study [161].

**Heat energy balance:** The heat energy balance expresses the relationship between conduction, convection, and radiation heat transfer and is greatly affected by the ash deposit thickness. Heat transfer in the reactor occurs first from the flue gas to the ash deposit surface by convection and radiation; then, it transfers to the coupon surface by conduction [169]. Increasing ash deposit thickness leads to declining heat transfer from the flue gas to the heat exchanger surface. As a result, the deposit surface temperature, \( T_s \), will increase due to the increasing thermal resistance. The deposit surface temperature, which is coupled with the ash deposit thickness calculation, is represented by the following energy balance that is organized from the components of references [156, 176]:

\[
\frac{k_{deposit}}{L_{deposit}} (T_s - T_{coupon,s}) = h_g(T_g - T_s) + \sigma_r(\alpha_r T_g^4 - \epsilon_r T_s^4),
\] (8-18)

where \( T_{coupon,s} \) is the temperature of coupon surface, which is set at 922 K. \( L_{deposit} \) and \( k_{deposit} \) are the thickness and the thermal conductivity of ash deposit layer, respectively. \( h_g \) is the forced crossflow convective heat transfer coefficient from the flue gas. \( \epsilon_r \) is the emissivity of the ash deposit, and it declines with increasing temperature; the value of \( \epsilon_r \) for ash is in the literature [177, 178]. \( \sigma_r \) is the Stefan-Boltzmann constant. Kirchhoff’s law, which is assumed for this model, states that the absorbance thermal radiation, \( \alpha_r \), is equal to the emittance thermal radiation, \( \epsilon_r \), when they are in radiative thermodynamic equilibrium [176]. It should be noted that the heat flux to the
probe surface is typically calculated by CFD calculations and can be affected by the fluid type [179]. Using an average Nusselt number ($\overline{Nu} = h_g D_h / k_g$), the convective heat coefficient is estimated. As expressed in A. Zbogar et al. [180] and F. P. Incropera et al. [181], the overall Nusselt number of the external crossflow around a cylinder is correlated by a power law relationship of Hilpert: $\overline{Nu} = C \cdot Re^m \cdot Pr^{1/3}$; where $C$ and $m$ are constants varying by Reynolds numbers as stated in Table 8-3. Please refer to the references [180-183] for more information about the forced convective heat transfer.

<table>
<thead>
<tr>
<th>Re</th>
<th>C</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4-4</td>
<td>0.989</td>
<td>0.330</td>
</tr>
<tr>
<td>4-40</td>
<td>0.911</td>
<td>0.385</td>
</tr>
<tr>
<td>40-4000</td>
<td>0.683</td>
<td>0.466</td>
</tr>
<tr>
<td>4000-40,000</td>
<td>0.193</td>
<td>0.618</td>
</tr>
<tr>
<td>40,000-400,000</td>
<td>0.027</td>
<td>0.805</td>
</tr>
</tbody>
</table>

**Table 8-3: The parameters of overall Nusselt number cited in [181].**

_Ash deposit thermal conductivity:_ The ash deposit thermal conductivity is another essential input of our developed model. A simple equation to estimate the ash deposit thermal conductivity, $k_{deposit}$, is presented by G. H. Richards [169]:

$$k_{deposit} = (1 - F) k_s + F k_{g - N_2}, \quad (8-19)$$

where $k_s$ and $k_{g - N_2}$ are the thermal conductivity of the solid phase deposit and nitrogen gas, respectively. The thermal conductivity of nitrogen gas phase is approximated using a dataset [184]. $k_s$ is assumed to be a constant value of 3 W/m/K by A. L. Robinson [185], which represents the thermal conductivity of silica-based materials at high temperatures. For the torrefied woody biomass ash, which is low in Si and high in Ca, $k_s$ is assumed to be 0.4 W/m/K [186]. For more information about the thermal conductivities of materials, please refer to the following paper by
Lastly, $F$ is the weighting factor, which is a function of ash porosity and obtained from elsewhere [169].

$$F = \frac{2^n}{2^{n-1}} \left( 1 - \frac{1}{(1+\varepsilon_{\text{deposit}})^n} \right),$$  \hspace{1cm} (8-20)

where, $n$ is an empirical parameter (6.5) [169]. It is worthwhile to note that the thermal conductivities and the ash deposit porosities are calculated at the deposit surface temperatures. It should also be noted that the influence of radiation on thermal conductivity is neglected in this research.

**Sticking efficiency:** The sticking efficiency is a complex phenomenon that can be accurately represented by many parameters including the viscosity, kinetic energy, and melt fraction of ash particles [83, 90, 132, 152, 187]. Viscosity-based stickiness models imply that the sticking probability equals unity if the particle viscosity is lower than the critical viscosity [115, 188]. While some sources disagree on the exact value for the critical viscosity of ash particles, J. R. Fan et al. [189] identified it as $1 \times 10^5$ (Pa·s). This value may be assumed or determined by experimentation [152]. Kinetic energy can also influence the sticking efficiency. If the kinetic energy of a particle is higher than the interfacial energy after the impaction, the particle bounces off the surface and does not stick [115, 190]. On the other hand, if the melt fraction of either impacting ash particle or ash deposit is in the range of 0.15-0.7, the stickiness of ash particle to the surface can be significant. M. U. Garba et al. [191] combined the sticking probabilities based on ash viscosity and melt fraction in a CFD model to predict the deposition rate of ash slag of coal/biomass cofiring. Similarly, M. Zhou et al. [192] integrated the three aforementioned stickiness models in a CFD model to predict the ash deposit rate in a pilot furnace. Traditionally, the complex stickiness models are solved by CFD packages; however, two stickiness models that do not require CFD are investigated independently in this study and their results are compared:
**Melt Fraction Stickiness Model (MFSM):** The MFSM derives the sticking efficiency from the particle temperature, the ash deposit surface temperature, and the melt fraction of ash [187, 189, 193]. Neglecting ash shedding and the other deposit removing mechanisms such as erosion, melting, and debonding of ash deposit, the following equation for the sticking efficiency, $\eta_{\text{stick}}$, is presented [187]:

$$\eta_{\text{stick}} = P(T_p) + [1 - P(T_p)]P(T_s),$$

where $P(T_p)$ is the sticking probability of the impacting ash particles at the particle temperature and $P(T_s)$ is the sticking probability of the ash deposit at the deposit surface temperature [83]. The probabilities are needed as only part of an ash particle may stick whereas the remaining splashes [152]. The splash of impacting ash particles is ignored here. Typically, the stickiness probability is assumed to linearly increase from 0.0 to 1.0 for the melt fractions between 0.15 (at $T_{15}$) and 0.7 (at $T_{70}$), and it remains constant at a constant value of 1.0 for higher melt fractions [194, 195]. The melt fraction is obtained through thermodynamic equilibrium calculations based on the minimization of the Gibbs free energy, which is performed using FactSage 7.3 thermochemical package [196-199]. The melt fractions that depend on the ash composition and designated temperature are tabulated in the next section (Results and Discussion 8.3). The MFSM considers the temperature and composition of ash; and the velocity of ash particle influences the model through Reynolds numbers. It is generally assumed that the sticking probability of the particles and ash deposit are identical if they have the same temperature and composition. However, the sticking probability is also affected by the initial sticky condensation layer on the clean coupon surface (or the initial deposit surface). This sticky layer is dominant in building up the initial layer of ash deposit and it is a novel approach in estimating the stickiness in the MFSM. If the condensation of vapor species results in molten salt, the sticking efficiency will equal to unity and
ash deposit rate goes up significantly. This issue is clarified in the Results and Discussion section of this study.

**Kinetic Energy Stickiness Model (KESM):** The KESM is based on several other factors distinct from the MFSM. The first of which is the velocity of impacting ash particles. The impacting particles will stick to the surface if their velocity is lower than the critical velocity [70]. The critical particle velocity \( u_{p,\text{crit}} \) is defined according to the references [200, 201] as a function of Young’s moduli and Poisson’s ratios of particle and surface:

\[
u_{p,\text{crit}} = \left( \frac{2K}{d_p R^2} \right)^{10/7},
\]

where \( R \) is kinematic restitution coefficient with the constant value of 0.9 [201]. \( K \), which is an effective stiffness parameter, is calculated by [201]:

\[
K = 0.51 \left( \frac{5\pi(\frac{1-\nu_s^2}{E_s} + \frac{1-\nu_p^2}{E_p})}{4\rho_p^{3/2}} \right)^{2/5},
\]

where \( \nu_s \) and \( \nu_p \) are the Poisson’s ratios of surface and particle, respectively, with the constant value of 0.27 [201]. \( E_s \) and \( E_p \) are Young’s moduli of the surface and particle, respectively. The following equation can estimate Young’s moduli, which is valid for \( d_p < 110\mu m \) [201]:

\[
E = 3 \times 10^{81} \exp (-0.1302T)
\]

In addition to velocity, the viscosity of ash deposit on the surface also plays a role in the KESM as viscosity may decrease with increasing ash deposit surface temperature. The decrease in viscosity leads to the adhesion of the impacting ash particles even if they are non-sticky [132]. Likewise, the of kinetic energy also plays a substantial role in this model. This is because the ash particles may lose their kinetic energy upon impaction due to their penetration into the surface,
which also increases their overall stickiness [202]. The critical velocity of surface capture \(u_{s,\text{crit}}\), which allows one to estimate the ability of the ash deposit surface to capture the kinetic energy of coming particles, is described by E. Raask [202]:

\[
u_{s,\text{crit}} = \frac{\rho_p g^2 (d_p/2)^3 \mu_s}{\gamma^2}
\]  

(8-25)

where surface tension \(\gamma\) is calculated by a model of K.C. Mills [203], and the ash deposit surface viscosity, \(\mu_s\), is assumed as high as \(10^{10}\) Pa·s [132].

In the KESM, the sticking probability of the impacting particle \((P_p)\) at the stagnation point and the sticking probability caused by the surface \((P_s)\) are defined using the following equations [132, 152, 201]:

\[
P_p = \begin{cases} 0 & \text{for } u_p > u_{p,\text{crit}} \\ 1 & \text{for } u_p < u_{p,\text{crit}} \end{cases}
\]  

(8-26)

\[
P_s = \begin{cases} 1 & \text{for } u_p > u_{s,\text{crit}} \\ 0 & \text{for } u_p < u_{s,\text{crit}} \end{cases}
\]  

(8-27)

Similar to MFSM, neglecting ash shedding and the other deposit removing mechanisms, the stickiness of kinetic-based model is defined as the following [132]:

\[
\eta_{\text{stick}} = P_p + [1 - P_p] P_s.
\]  

(8-28)

Figure 8-5 shows the algorithm of our developed model using MATLAB R2016a. A total ash deposit thickness of the stagnation point is guessed at the first step of the model. The dynamic ash deposit thickness is then calculated at the upstream stagnation point of the coupon surface for three bins of fly ash particle sizes. At this point, the ash deposit at the upstream stagnation point includes the inertial impaction, thermophoresis, and condensation, as previously discussed. Furthermore, the three bins are defined by 0-20%, 20-80%, and 80-100% of cumulative concentration of fly ash [70] by making an analogy with the data of Q. Gao et al. [149]. For as
long as the model runs, the ash deposit thickness is calculated and the parameters of ash deposit such as porosity, thermal conductivity and deposit surface temperature are likewise calculated and updated. Eddy impaction, which builds up the eddy ash deposit, is also calculated at the stagnation point of the $\pi$ side of the coupon, and its impaction efficiency is assumed to be unity.

The inertial ash deposit thickness at the angular position ($\theta$) is estimated by multiplying absolute $\cos(\theta)$ with the inertial ash deposit thickness of the stagnation point (see Figure 8-5). More complex formula than the simple cosine function that is presented here can be found elsewhere [204, 205]. In a similar approach, the temperature gradient at the angle of $\theta$ is estimated by assuming that the ash deposit surface temperature of the $\pi$ side of the coupon and $\theta = \pi/2$ and $\theta = 3\pi/2$ are equal to the coupon surface temperature (922 K). Therefore, the thermophoresis ash deposit and condensed ash deposit can be calculated at the circumference of the coupon at each angular position ($\theta$) by estimating the temperature gradient between the ash deposit surface and the flue gas. It is assumed that the impaction efficiencies of the thermophoresis particles on the front and $\pi$ sides are the same. The eddy impaction ash deposit, which can be maximum deposit at the stagnation point of the $\pi$ side, is calculated at the angle of $\theta$ of the $\pi$ side using the similar procedure that explained for calculating inertial ash deposit at the angle of $\theta$ of the upstream side. The initial guess for the model uses the results of the first step and is updated after each iteration until the new guess and the calculated total ash deposit thickness of the stagnation point converged (The error is less than 0.01%). Afterwards, the total mass of the ash deposit on the coupon surface is calculated and compared with the experimental data. The logic of this algorithm is proven reasonable due to its success in predicting the ash deposit rate of a wide range of solid fuels. The numerical code of our developed model of ash deposit rate is presented in Appendix D: Numerical code of ash deposit rate. I developed this model myself.
8.3 Results and discussion

The following paragraphs compare the results of the MFSM and KESM approaches to the experimental data to determine their effectiveness in predicting the sticking efficiency and, subsequently, the accuracy of our model is discussed. The effect of fuel type and time is then discussed for several important factors included the deposit surface temperature, deposit thickness, and thermal resistance. Lastly, the dominant mechanisms of ash deposition on all surfaces of the coupon are presented.

The melt fraction temperature, which is distinct for each fuel type, is estimated by the simulation package FactSage 7.3. The values for this temperature depend heavily on the ash
composition, presented in Table 8-4 for each of the eleven fuels. The common parameters in this
table are shared for all the fuels in this study. FactSage 7.3 predicts the following four characteristic
temperatures of the melt fractions of ash particles and ash deposits:

- $T_0$ (the temperature at which the melt fraction of ash particle (or deposit) is 0)
- $T_{15}$ (the temperature at which the melt fraction of ash particle (or deposit) is 0.15)
- $T_{70}$ (the temperature at which the melt fraction of ash particle (or deposit) is 0.7)
- $T_{100}$ (the temperature at which the melt fraction of ash particle (or deposit) is 1.0)

The characteristic temperatures were determined via two steps. First, using the FToxid-
SLAGA database with the ash components listed in Table 8-2 as the inputs, the melt fractions at
various temperatures were calculated and tabulated. Second, the temperatures at which the melt
fractions reach 0.15, 0.7, and 1.0 were determined using interpolation.

In order to calculate the characteristic temperatures, the mole fractions of the alkali vapor
species is required. This was also accomplished using FactSage, which showed that KCl and NaCl
are the main alkali vapor species in the flue gas. S. B. Hansen et al. [132] provided a formula to
obtain the amount of the K released from the fuel to the gas phase, which was used in this study.
Due to their common properties, it was assumed that the release of Na and K to the gas phase is
the same. Additionally, it is assumed that K and Na of ash analysis represent K and Na of the fuel.
Furthermore, it was assumed that both Cl and S are completely released [68, 206]. These
assumptions provide a good estimation of the amount of alkali vapor species and their
condensation temperature ranges. The amount of Cl released is especially important as Cl
facilitates the transfer of the alkali components from the fuel to the surface, forming sulfates and
the sticky layer of ash deposit [14]. With NaCl and KCl identified as the major alkali species, the
amount of the other mineral vapor species can be neglected when calculating the condensation ash deposit on the probe surface. It should be noted that the contribution of these alkali vapor species (forming sub-micron particles) to the ash deposit mass is small compared to that of the coarse ash particles [68]; however, their role in making sticky layer on the surfaces to build up ash deposit is significant because they might increase the sticking efficiency to unity as discussed in the model description.

The analysis of alkali vapor condensation was performed for the fuels whose Cl contents are available (see Table 8-1); however, Cl concentrations were not available for all of the fuels of this study. The mole fractions of NaCl and KCl in the flue gas vary from one fuel to another, as does their flue gas temperatures. Since the accurate amounts of the released Na and K to the gas phase are not known and Cl content data is not complete; the calculation of the temperature ranges of alkali vapor condensation and the amount of alkali vapor species by Factsage are estimated. Therefore, the constant mole fraction of NaCl and KCl, and a common range of temperatures of their condensation, which are calculated by FactSage, are assumed as displayed in Table 8-4. This assumption is reasonable and does not change the results significantly as the presence of a condensation deposition mechanism is more indicative of behavior than the amount of condensation deposition happening. Moreover, specifying the common parameters that are valid for a range of fuels would facilitate the understanding and application of this model by the boiler operators. Careful analysis of the alkali vapor species and their condensation might be needed for longer operation times in which the ash deposit surface temperature may approach the flue gas temperature.

Alkali vapor species discussed above are important as they have been shown to condense on cooler surfaces, creating a sticky molten layer that acts like a glue, capturing further ash
particles [77]. Our thermodynamic modeling results confirm this and suggest that NaCl and KCl can condense partly on the colder external surfaces such as the clean coupon surface and the initial ash deposit surface. The resulting condensed phase reacts with S to make new compounds, which predominately consist of solid sulfates and molten salts. The solid sulfates are formed when S condenses at the lower surface temperatures of around 800 to 850 K. The molten salts are formed by the condensation of alkali vapor species at surface temperatures of around 851 to 1266 K and are sticky, efficiently capturing impacting particles. The upper limit of this temperature window (1266 K) is defined as the temperature at which 90% of the released K is in the vapor phase. Due to the stickiness of the molten salts, we assumed the sticking efficiency in this temperature range should be unity for the MFSM when there is an appropriate concentration of alkali and Cl in the gas phase. For this study, the molten salt deposits were predicted by the specific FTsalt database of FactSage, with the organic part of the fuel (C, H, O, N, and S), air, and the released K, Na, and Cl used as inputs. Meanwhile, as indicated in Table 8-4, it is reasonable to tabulate a common temperature range of alkali vapor condensation and their condensation temperature range for all the fuels of this study.

The ash aerosol concentration of the fuels, another important parameter for our model, was presented by Y. Wang [122]. The three bins of 0-20%, 20-80%, and 80-100% of cumulative fly ash PSD are defined by making an analogy with the data of Q. Gao et al. [149], which is a good estimation to be used in the model. For simplicity, it is assumed that the presented bin sizes in Table 8-4 are representative for all the fuels in this study.
Table 8-4: Distinct and Common Parameters used in the model. The fuels include: a) RH, b) Sufco#1, c) Sufco#2, d) 20RH-80Sufco#1, e) PRB, f) 13RH-87PRB, g) Petcoke, h) Illinois, i) 60Illinois-40PRB, j) Torrefied, k) 50Torrefied-50Sufco#2.

<table>
<thead>
<tr>
<th>Distinct Parameters</th>
<th>Unit</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
<th>Fuels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid fuel rate</td>
<td>(kg/s) x 10^3</td>
<td>0.239</td>
<td>0.961</td>
<td>0.961</td>
<td>1.09</td>
<td>1.26</td>
<td>1.32</td>
<td>0.378</td>
<td>0.989</td>
<td>1.08</td>
<td>1.26</td>
</tr>
<tr>
<td>Flue gas density [132]</td>
<td>(kg/m^3)</td>
<td>0.28</td>
<td>0.286</td>
<td>0.304</td>
<td>0.288</td>
<td>0.298</td>
<td>0.289</td>
<td>0.291</td>
<td>0.29</td>
<td>0.287</td>
<td>0.306</td>
</tr>
<tr>
<td>Flue gas velocity</td>
<td>(m/s)</td>
<td>0.92</td>
<td>0.73</td>
<td>0.60</td>
<td>0.76</td>
<td>0.67</td>
<td>0.72</td>
<td>0.75</td>
<td>0.67</td>
<td>0.68</td>
<td>0.64</td>
</tr>
<tr>
<td>Flue gas temperature at port 6 (T_g)</td>
<td>(K)</td>
<td>1277</td>
<td>1247</td>
<td>1173</td>
<td>1241</td>
<td>1198</td>
<td>1235</td>
<td>1228</td>
<td>1232</td>
<td>1245</td>
<td>1166</td>
</tr>
<tr>
<td>Viscosity of flue gas (μ_g) [132]</td>
<td>(Pa·s)x10^5</td>
<td>4.87</td>
<td>4.80</td>
<td>4.61</td>
<td>4.78</td>
<td>4.67</td>
<td>4.76</td>
<td>4.75</td>
<td>4.76</td>
<td>4.79</td>
<td>4.59</td>
</tr>
<tr>
<td>Fly ash concentration (C_p)</td>
<td>(kg/m^3) x 10^3</td>
<td>1.56</td>
<td>1.96</td>
<td>3.99</td>
<td>3.45</td>
<td>1.65</td>
<td>2.84</td>
<td>0.269</td>
<td>2.50</td>
<td>2.16</td>
<td>0.0665</td>
</tr>
<tr>
<td>Melt fraction temperatures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_0</td>
<td>(K)</td>
<td>1140</td>
<td>1160</td>
<td>1165</td>
<td>1175</td>
<td>1185</td>
<td>1180</td>
<td>1170</td>
<td>1180</td>
<td>1285</td>
<td>1180</td>
</tr>
<tr>
<td>T_15</td>
<td>(K)</td>
<td>1165</td>
<td>1180</td>
<td>1185</td>
<td>1190</td>
<td>1285</td>
<td>1190</td>
<td>1180</td>
<td>1190</td>
<td>1320</td>
<td>1190</td>
</tr>
<tr>
<td>T_70</td>
<td>(K)</td>
<td>1810</td>
<td>1495</td>
<td>1520</td>
<td>1530</td>
<td>1395</td>
<td>1380</td>
<td>1495</td>
<td>1415</td>
<td>1410</td>
<td>1410</td>
</tr>
<tr>
<td>T_100</td>
<td>(K)</td>
<td>1850</td>
<td>1585</td>
<td>1575</td>
<td>1790</td>
<td>1465</td>
<td>1580</td>
<td>1575</td>
<td>1465</td>
<td>1460</td>
<td>1560</td>
</tr>
</tbody>
</table>

| Common Parameters                     |            |       |       |       |       |       |       |       |       |       |       |
| Coupon surface temperature            | (K)        | 922   |       |       |       |       |       |       |       |       |       |
| Initial porosity of ash deposit       | ---        | 0.6   |       |       |       |       |       |       |       |       |       |
| Fly ash particle density (ρ_p)*       | (kg/m^3)   | 1300  |       |       |       |       |       |       |       |       |       |
| Bulk density of ash deposit*          | (kg/m^3)   | 550   |       |       |       |       |       |       |       |       |       |
| Fly ash PSD (three bins of beginning 20%, middle 60%, and end 20%) | (μm) | (1.6, 7.77, 95) |       |       |       |       |       |       |       |       |       |
| Temperature range of alkali vapor condensation | Solid Sulfide | 800 < T ≤ 850 | Sticking efficiency of MFSM is calculated from Eq. (8-21) |       |       |       |       |       |       |       |
|                                      | Molten Salt | 851 ≤ T ≤ 1266 | Sticking efficiency of MFSM is unity. |       |       |       |       |       |       |       |
|                                      | Alkali Vapor | 1266 < T | Sticking efficiency of MFSM is calculated from Eq. (8-21) |       |       |       |       |       |       |       |
| Alkali vapor species                 |            |       |       |       |       |       |       |       |       |       |       |
| NaCl                                 | (Mole Fraction)x10^5 | 5.59  |       |       |       |       |       |       |       |       |       |
| KCl                                  |            | 5.4   |       |       |       |       |       |       |       |       |       |
* Fly ash particle density and bulk density of ash deposit are assumed constant regarding the reported data in [207-209]. Due to the heavier minerals of Petcoke (g), its fly ash particle density and bulk density of ash deposit are assumed 2240 and 915 kg/m³, respectively [210].

** The melt fraction temperatures of the Torrefied woody biomass are uncommon because of the fuel’s unusual ash composition: low Si and Al content and very high Ca content (>50 wt. %). FactSage predicts that there would be 10 wt. % of molten salt but the melt fraction is not sensitive to the temperature. In addition, the ash content of this fuel is quite low (0.19 wt. %). Therefore, the same melt fraction of fuel (k) (50Torrefied-50Sufco#2) is assumed here. This assumption does not overestimate the stickiness model since T₀ > Tg, meaning that the melt fraction is zero.

**Table 8-5: The experimental ash deposition data for three operation times of 30, 60, and 120 minutes. The fuels include a) RH, b) Sufco#1, c) Sufco#2, d) 20RH-80Sufco#1, e) PRB, f) 13RH-87PRB, g) Petcoke, h) Illinois, i) 60Illinois-40PRB, j) Torrefied, k) Torrefied-50Sufco#2. Only one test was performed (60-minute test) for fuels (e), (h), and (i).**

<table>
<thead>
<tr>
<th>Fuels</th>
<th>a (g/m²)</th>
<th>b (g/m²)</th>
<th>c (g/m²)</th>
<th>d (g/m²)</th>
<th>e (g/m²)</th>
<th>f (g/m²)</th>
<th>g (g/m²)</th>
<th>h (g/m²)</th>
<th>i (g/m²)</th>
<th>j (g/m²)</th>
<th>k (g/m²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t₁=30 (minutes)</td>
<td>28.83</td>
<td>33.56</td>
<td>52.53</td>
<td>79.68</td>
<td>NA</td>
<td>31.65</td>
<td>24.14</td>
<td>NA</td>
<td>NA</td>
<td>0.85</td>
<td>7.89</td>
</tr>
<tr>
<td>t₂=60 (minutes)</td>
<td>60.99</td>
<td>77.13</td>
<td>116.76</td>
<td>112.12</td>
<td>36.51</td>
<td>76.96</td>
<td>69.50</td>
<td>53.26</td>
<td>36.17</td>
<td>1.52</td>
<td>23.00</td>
</tr>
<tr>
<td>t₃=120 (minutes)</td>
<td>130.22</td>
<td>148.33</td>
<td>237.53</td>
<td>294.97</td>
<td>NA</td>
<td>173.43</td>
<td>110.61</td>
<td>NA</td>
<td>NA</td>
<td>8.19</td>
<td>13.26</td>
</tr>
</tbody>
</table>
The experimental ash deposit data are presented in Table 8-5. There is only one test data (60-minute test) for the fuels of PRB (e), Illinois (h), and 60Illinois-40PRB (i).

Comparing the results of stickiness models: The MFSM and the KESM provided different results due to the unique parameters each model uses to estimate the stickiness. As previously shown, the MFSM is more complex than the KESM due to the estimation of the melt fractions and the initial sticky layer of condensation. The more complicated nature of the MFSM allows it to estimate the stickiness more effectively, as shown in Figure 8-6, compared to the ash deposit rates that are presented in Table 8-5. Figure 8-6 also shows that in general, the MFSM overestimates and the KESM underestimates the ash deposit rate experimental data; notwithstanding, the MFSM supports the experimental data better than the KESM. As displayed in Figure 8-6, the root mean square error of MFSM and KESM are 48.1 and 84.6, respectively. One of the reasons for this is that the model based on the KESM inaccurately predicts that many fly ash particles bounce off and do not stick to the surface, especially for Petcoke (g). Both stickiness models overestimate the ash deposit of 50Torrefied-50Sufco#2 (k), which might be because of the ash shedding during the experimental test of the fuel, particularly for the 120-minute test. The phenomenon of ash shedding is outside of the scope of this research and may be considered for future studies. In the absence of ash shedding, our model accurately predicts the ash deposition rate. The predictions may be further improved by specifically defining some of the input data of the model—such as the initial porosity and the densities of ash. This approach could also be considered in future studies.

As discussed, the MFSM provides more reasonable results than the KESM. The remainder of this section will provide predictions from the model based on the MFSM.
Dynamic changes of ash deposit properties: Many of the ash deposit properties, including the ash deposit surface temperature, ash deposit thickness, and the ash deposit’s thermal resistance dynamically change as a function of time (or deposit thickness). Plots of the aforementioned properties versus time for various fuels are shown in Figure 8-7. The ash deposit surface temperatures (top plots) are predicted to increase with time relative to the coupon surface temperature (922 K). Given more time, the maximum value for these plots would eventually
approach the flue gas temperature. The ash deposit thickness (middle plots) is predicted to increase over time, exhibiting a relatively linear trend. The condensed sticky layer, which forms in the temperature range of 851 to 1266 K, leads to the enhanced stickiness probability of the surface. Changes of ash deposit thermal resistance, which is the ratio of ash deposit thickness over its thermal conductivity, versus time are shown in Figure 8-7 (bottom plots). When observing the plots, it is important to understand that because the ash analysis of Torrefied woody biomass is predominantly Ca as opposed to Si, therefore its thermal conductivity is much less than the other fuels of this study. It is therefore preferable to present the plot of ash deposit thermal resistances instead of ash deposit thermal conductivities. In general, it is observed that thicker ash deposits contribute to a higher surface temperature and lower heat transfer due to the increased thermal resistance [77].

Our model shows that the ash deposit rate, surface temperature, and thermal conductivity can vary depending upon the fuel type. The ash deposit surface temperature and the ash deposit thickness of 20RH-80Sufco#1 (d) are high as shown by 20RH-80Sufco#1’s high thermal resistance relative to the other fuels. In contrast, the ash deposit surface temperature and the thickness of Petcoke (g) and Torrefied (j) are relatively low as shown by their low thermal resistances.

Our model was also able to determine the effect on the porosity of the ash deposit. As discussed in model description, the thermal conductivity of an ash deposit, which determines the ash deposit’s ability to transfer heat [130], is a function of ash porosity (see Eq. (8-19), Eq. (8-20), and reference [169]). As mentioned above, the deposit surface temperature rises over time due to the increasing thickness of ash deposit. Increasing the deposit surface temperature may lead to a decrease in the porosity of the ash deposit [152]. Neglecting the liquid volume of the condensed
sticky layer, the model results show that the deposit porosity does not change since the liquid volume of ash deposit is not significant.

It is worthwhile to note that the effect of sintering—or the flow of the liquid phase on the surface of the bonded ash particles to the low potential energy surface—on the thermal conductivity during heating ash has been studied by other researchers [130, 180, 211-213]. For our study, however, we assume that the ash deposit never reaches the sintering point, and it is therefore not addressed. It is suggested that further research be done on the relationship of this topic to our ash deposit models.

*Predicted shape of ash deposit in cross view of the probe:* For visualization of the data of our model, Figure 8-8 presents an approximate simulated shape of ash deposit on the coupon surface after 120 minutes of 20RH-80Sufco#1. As expected, the largest ash deposit thickness is formed at the stagnation point of upstream ($\theta = 0$) with inertial impaction being the dominant mechanism of ash deposit formation. The contribution of thermophoresis and condensation at the upstream are significant as long as the temperature gradient between the surface and the flue gas is high enough; however, their general contributions in ash deposit growth are much lower than the inertial impaction. Eddy impaction does not contribute to the ash deposit on the $\pi$ side of the coupon due to the high density of fly ash particle, meaning that the fly ash particles do not follow the eddy stream to impact the $\pi$ side of the coupon. At the angles $\theta = \pi/2$ and $3\pi/2$, inertial impaction contributes the least to ash deposit formation. Whereas thermophoresis is the dominant mechanism of ash deposit at such angles. Condensation in general contributes the least to ash deposit on the coupon surface; however, its contribution in making sticky layer is significant. Similar trends were observed for all other 10 fuel types.
Figure 8-7: Changes of ash deposit properties versus time. (top) Ash deposit surface temperature, (middle) Ash deposit thickness, and (bottom) ash deposit thermal resistance.
Figure 8-8: Predicted ash deposit shape after 120-minute operation of the blend of 20RH-80Sufco#1.

8.4 Conclusion

A dynamic model has been developed and validated to effectively predict the ash deposit rate of the combustion of eleven solid fuels in a 100 kW down-fired combustor. The fuels include coal, biomass, and their blends and petroleum coke. The four mechanisms of ash deposition: inertial impaction, thermophoresis, condensation, and eddy impaction for three size bins representing different PSDs of fly ash particles are calculated in the model. Thermodynamic
modeling results using FactSage indicate that NaCl and KCl are the main alkali vapor species in the flue gas. Two ash deposit stickiness models were investigated: the melt fraction stickiness model (MFSM) and the kinetic energy stickiness model (KESM), which differ in the way that they predict sticking efficiency of particles that arrive at the deposit surface. In the MFSM, the sticking efficiency is calculated by accounting for the condensation of alkali vapor species as molten salts on the cooler surfaces (clean coupon surface and ash deposit surface). This condensation creates a sticky layer that enhances the sticking efficiency of the impacting particles and accelerates the ash deposit growth. In the KESM, the sticking efficiency is determined by analyzing the sticking or rebounding of impacting particles and depends on the critical velocities of the particle and surface and not the condensation of alkali vapor species. There is a better agreement between the results of the MFSM and the experimental data due to the significant effect of the condensed sticky layer on ash deposit growth, and its novel approach to estimate the stickiness. The significant smaller root-mean-square-error of MFSM compared to KESM is a proof for such a claim. The results of the MFSM demonstrate that the inertial impaction and thermophoresis are the dominant mechanisms in ash deposit formation on the upstream and the π side surface, respectively. It was also shown that thermophoresis is significant at the angles of \( \theta = \pi/2 \) and \( 3\pi/2 \); and that thermophoretic forces have smaller values at larger thermal conductivity ratios and smaller \( Kn \) numbers. Likewise, the ash deposit surface temperature was shown to increase with the build up of ash deposit due to increasing thermal resistance. The developed model can be used to help the operators of the full-scale boilers readily predict ash deposits on the heat exchanger surfaces. It is suggested that more accurate stickiness models continue to be researched to predict the ash deposit rate for a wider variety of fuels.
9 Modelling of ash deposition rate of coal combustion and its blend with woody biomass in a 1.5 MW combustor.

9.1 Introduction

The objective of this chapter is to help boiler operators predict ash deposit rate during the combustion of the blend of prepared woody biomass (steam exploded and torrefied pellets) and coal compared to the pure coal on the heat exchanger surfaces. To fulfill such an objective, a model, which includes inertial impaction, thermophoresis and condensation as well as eddy impaction, was developed in Chapter 8 to predict ash deposition rate. Here the predicted ash deposition rates and the measured data of 1.5 MW combustor are compared. Furthermore, the dominant deposition mechanisms on the probe surface are discussed [214].

The experimental data of the combustor is presented in Chapter 6. The ash deposit samples were collected using the probe that was described in Chapter 5. I used the developed model in Chapter 8 to predict ash deposit rate of 1.5 MW combustor. However, the traverse of flue gas is horizontal in the 1.5 MW combustor. For this furnace and operating configuration, the deposit mechanisms will be in accordance with Figure 8-3, but rotated 90 degrees.

9.2 Results and discussion

The melt fraction stickiness model (MFSM), which was described in Chapter 8, utilized thermodynamic calculations by FactSage 7.3, performed by University of Utah. The following sections discuss the thermodynamic analysis and ash deposit rate model.
9.2.1 Thermodynamic analysis

Mineral equilibrium calculated by FactSage assumed that the release of Na and K to the gas phase are similar. Hansen [132] provided a formula to obtain the estimated amount of the released K from the fuel to the vapor phase, which is used at this study. In addition, it is hypothesized that both Cl and S are released fully [68, 206]. Chlorine provides a pathway for the alkali components vaporized from the fuel to condense on the surface and subsequently form sulfates and the sticky layer of ash deposit [14].

The equilibrium of alkali vapor species of the three fuels used in this study are presented in Figure 9-1. KCl and NaCl are the main alkali vapor species for the flue gas temperature range between ports 7 and 10, which are presented in Table 6-5. Condensation of alkali chlorine vapor species on a surface (heat exchanger surface or ash particle surface) typically act like a glue by making a sticky molten layer on the surface [77]. The Factsage thermodynamic package shows that NaCl and KCl convert to solid or molten sulfate compounds at the coupon surface temperature (or ash deposit) and therefore HCl is generated. However, the contribution to mass deposited of such vapor species compared to the coarse ash particles in ash deposit formation is small [68].

Theoretically, when the chlorine vapors (KCl and NaCl) approach the cold surface, the temperature drop in the gaseous film would cause the condensation of KCl and NaCl vapors, but at the same time, the sulfation reactions of alkali chlorines are also thermodynamically favored at the same temperatures, producing vapor HCl. The condensed phase includes molten salts and solids. For more clarification, the following reaction may happen at the coupon surface temperature (811 K):

$$4\text{KCl} + 2\text{SO}_2 + \text{O}_2 + 2\text{H}_2\text{O} \rightarrow 2\text{K}_2\text{SO}_4 + 4\text{HCl}.$$
This suggests that the alkali compound is not molten on the coupon surface, and therefore does not contribute significantly to the sticking efficiency and ash deposition rate. However, the condensation of alkali vapor species at the surface temperatures above 850 K (up to a surface temperature in which 90 wt.% of K is in vapor phase) form molten salts, which are sticky and cause retention the impacting particles. Therefore, we have assumed that the sticking efficiency of the melt fraction model becomes unity in this temperature window, when the appropriate concentration of alkali and chlorine are present. The temperature ranges of the condensation of alkali vapor species are presented in Table 9-1. These temperature ranges have been calculated by Factsage using the estimated concentrations of K and Na which were not measured in this experiment.

The SLAG feature in FactSage is useful to determine the expected ash compositions, which are presented at Figure 9-2. The mass percentage of quartz (SiO₂) is the dominant slag ash compound for all the three fuel analyses. The graphs change dramatically in the temperature range of 1345-1400 K. At temperatures below 1345 K, the melt is basically composed of Si, Al, K, Na, and O. When the temperature reaches 1345 K, some high-melting species, e.g. Ca and Fe, start to enter the molten phase, leading to drastic changes in composition between 1345 K and 1400 K. This indicates the melt fractions of ash, calculated by FactSage, depends upon the temperature as presented in Table 9-2. Similarly, as described in Chapter 9, the melt fraction of ash deposit and fly ash particles are identical at the same temperatures.
Figure 9-1 FactSage equilibrium mole fractions of alkali vapor species at the flue gas temperature range of ports 7 and 10.

Table 9-1: Temperature ranges of alkali vapor condensation.

<table>
<thead>
<tr>
<th></th>
<th>Sufco Coal</th>
<th>Torrefied blend</th>
<th>Steam Exploded blend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid Sulfide</td>
<td>800 &lt; T ≤ 850</td>
<td>800 &lt; T ≤ 850</td>
<td>800 &lt; T ≤ 850</td>
</tr>
<tr>
<td>Molten Salt</td>
<td>851 ≤ T ≤ 1290</td>
<td>851 ≤ T ≤ 1296</td>
<td>851 ≤ T ≤ 1256</td>
</tr>
<tr>
<td>Alkali Vapor</td>
<td>1290 &lt; T</td>
<td>1296 &lt; T</td>
<td>1256 &lt; T</td>
</tr>
</tbody>
</table>
Figure 9-2: Mass percentage of slag ash compounds, which is in liquid phase.
9.2.2 Ash deposition model

The model detailed in Chapter 8 was used to predict the experimental ash deposit rate of 1.5 MW combustor during coal combustion and its blend with woody biomass, which was described in Chapter 6. The input operating parameters for this model are presented in Table 6-5, Table 9-1, Table 9-2, and the following Table 9-3.

There is a good agreement between the model and experiments of ash deposition as shown in Figure 9-3. The model predicts a reasonably linear increase of ash deposit thickness versus time. Ash deposit in port 7 was about twice than port 10, which was likely due to the higher fly ash concentration and flue gas temperature at port 7, which is represented well by the model.

### Table 9-2: Temperature ranges of ash melt fractions.

<table>
<thead>
<tr>
<th>Melt fraction temperatures</th>
<th>Sufco Coal</th>
<th>Torrefied blend</th>
<th>Steam Exploded blend</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₀</td>
<td>1186</td>
<td>1187</td>
<td>1186</td>
</tr>
<tr>
<td>T₁₅</td>
<td>1187</td>
<td>1188</td>
<td>1187</td>
</tr>
<tr>
<td>T₇₀</td>
<td>1375</td>
<td>1372</td>
<td>1411</td>
</tr>
<tr>
<td>T₁₀₀</td>
<td>1425</td>
<td>1450</td>
<td>1470</td>
</tr>
</tbody>
</table>

### Table 9-3: Operating parameters of the 1.5 MW combustor.

<table>
<thead>
<tr>
<th>Port No.</th>
<th>Torrefied blend</th>
<th>Steam exploded blend</th>
<th>Sufco coal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fly ash concentration (kg/m³)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.23E-03</td>
<td>2.12E-03</td>
<td>1.72E-03</td>
</tr>
<tr>
<td>10</td>
<td>2.18E-03</td>
<td>2.04E-03</td>
<td>1.66E-03</td>
</tr>
<tr>
<td>Fly Ash PSD (three bins of 20%-60%-20%) μm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(1.24-5.43-90)</td>
<td>(1.72-5.83-95)</td>
<td>(1.6-7.77-95)</td>
</tr>
<tr>
<td>10</td>
<td>(1.84-9.65-95)</td>
<td>(1.84-7.77-95)</td>
<td>(1.6-4.7-90)</td>
</tr>
<tr>
<td>Viscosity of flue gas [132] (Pa.s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.97E-05</td>
<td>5.03E-05</td>
<td>5.08E-05</td>
</tr>
<tr>
<td>10</td>
<td>4.66E-05</td>
<td>4.81E-05</td>
<td>4.87E-05</td>
</tr>
<tr>
<td>Initial Porosity</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Figure 9-3: Comparing ash deposition at both ports of 7 and 10 for experiment and modeling.

The changes of ash deposit thickness versus time are shown in Figure 9-4. The changes of ash deposit growth are relatively linear. The condensed molten sticky layer, which forms at temperatures above 850 K on the surface, enhances the sticking efficiency of the surface and therefore, the ash deposit thickness increases after the break points in the plots. These break points are also displayed at the ash deposit surface temperatures (Figure 9-5) and ash deposit thermal conductivity (Figure 9-6), which will be discussed. It should be noted that since there are many parameters that involved in modelling of ash deposit rate, and some of them are distinct (e.g. Table 9-3), the model can resolve the differences between behavior of each fuel.

Changes of ash deposit surface temperature versus time are presented in Figure 9-5. The ash deposit surface temperatures increase with time and approaches the flue gas temperature at long time periods. The ash deposit surface temperature measured in port 7 while firing pure coal increases more quickly, which is likely due to the its higher flue gas temperature compared to the others. Generally, thicker ash deposits contribute to a higher surface temperature and lower thermal conductivity due to increasing the resistance [77]. The beginning point of all the trends is the
coupon surface temperature (811 K), which represents the heat exchanger surface temperature of clean reheater tubes. The largest bin size of PSD of fly ash particles is the only bin that was governed by the MFSM in this project. Therefore, the ash deposit surface temperatures of this project, which displayed in Figure 9-5, are based on such a bin. This approach provides a reasonable estimation of dynamic changes of ash deposit surface temperatures. However, further analysis is required in the future research.

**Figure 9-4: Model of ash deposit thickness versus time for ports 7 and 10.**

**Figure 9-5: Changes of ash deposit surface temperature versus time for ports 7 and 10.**
Ash deposit thermal conductivity is another important parameter, which changes with time as shown in Figure 9-6. As discussed in the model description of the previous chapter, the ash deposit thermal conductivity is a function of several parameters such as porosity, solid-phase conductivity, and gas-phase conductivity. The solid phase conductivity, $k_s$ (see Eq. 8-19), is assumed to be a constant value of 3 W/m/K [185]. Our model shows that the porosity does not change, and moreover, the solid phase thermal conductivity is assumed constant. However, the ash deposit thermal conductivity increases due to gas phase thermal conductivity of ash deposit. In general, the denser deposit structure leads to higher thermal conductivity [152, 185]. Therefore, the thermal conductivity of coal ash is typically higher than that of biomass ash because of its structure and porosity [118]. It should be noted that since both ash deposit thickness and thermal conductivity influence the ash deposit thermal resistance, higher thermal conductivity does not mean lower thermal resistance.

![Figure 9-6: Changes of ash deposit thermal conductivity versus time for ports 7 and 10.](image)

*Figure 9-6: Changes of ash deposit thermal conductivity versus time for ports 7 and 10.*
Figure 9-7: Predicted ash deposit shape after 90 minutes of Sufco coal operation at port 7.

Figure 9-7 presents the estimated shape of ash deposit on the coupon surface after 90 minutes of Sufco coal combustion at port#7. As expected, the largest ash deposit thickness is formed at the stagnation point upstream; and inertial impaction is the dominant mechanism in ash deposit formation. The contribution of thermophoresis is significant only at the beginning of ash deposit formation; but its total contribution in ash deposit formation is much lower than inertial impaction. Eddy impaction does not contribute to ash deposit at the π side of the coupon surface because the fly ash particles do not follow the eddy stream. At the angles \( \theta = \pi/2 \) and \( \theta = 3\pi/2 \), inertial impaction has the least contribution in ash deposit formation. However, thermophoresis
is the dominant mechanism of ash deposit for these angles and the $\pi$ side of the coupon surface. Condensation contributes least to the mass of deposit on the coupon surface. However, condensation of alkali vapor species forms a sticky layer on the coupon or deposit surface which facilitates ash deposit growth by other mechanisms.

9.3 Conclusion

Our developed model predicts the ash deposit rate of coal combustion and its blends with torrefied biomass and steam exploded biomass. The details of the experimental data and the model description are presented in Chapters 6 and 8, respectively. The four mechanisms of ash deposition: inertial impaction, thermophoresis, condensation, and eddy impaction for the three bins of PSD of fly ash particles are calculated in the model. The results of the model demonstrate that the inertial impaction and thermophoresis are the dominant mechanisms in ash deposit formation at the stagnation point and the $\pi$ side surface, respectively. Thermophoresis and condensation are significant at the angles of $\theta = \pi/2$ and $\theta = 3\pi/2$. The model demonstrates that ash deposit surface temperature increases by building up ash deposit overtime and it changes the thermal conductivity. The thermodynamic simulation package FactSage 7.3 was used and it shows that NaCl and KCl are the main alkali vapor species at the flue gas. The condensation of such alkali vapor species on the colder surfaces contributes to sticky layer that lead to increase in ash deposit growth. The results of this model match well with experimental data. As a summary for the operators of the full-scale boilers, it can be stated that during Sufco coal combustion and also its blend with woody biomass (15% biomass, 85% coal by weight mixture), the ash deposit rate of convection area increases significantly if the temperature surface of heat exchanger greater than 850 K.
10 Summary and conclusion

The impacts of partial replacement of coal with woody biomass in existing coal-fired power plants was investigated in this PhD thesis. The technologies that we investigated in this PhD thesis have the potential to reduce the net emission of CO$_2$ to the atmosphere and mitigate wildland fire hazard. These technologies include modelling co-milling of woody biomass with coal in an unmodified pulverizer, measuring and analysis of ash deposit during cofiring; and developing a model to predict ash deposit rate of a variety of fuels.

10.1 Analysis of the trajectories of the pulverized particles of woody biomass and coal during co-milling

We used Barracuda CPFD simulation package in order to simulate the differences in behavior between coal and biomass during co-milling in a pulverizer; and evaluate the impact of geometry and operational changes on the mill performance. Since there was a void of information in the literature and little was known by experts about the internal conditions in the mill, reasonable assumptions were made and validated for the location, PSD, and the rate of the particles generated by the mechanical action of the bowl and rolls in the model, resulting in a novel approach for CFD modeling of coal mill behavior. Post-processing analysis of the simulation by our generated C# code shows that the large woody biomass particles are entrained by air, generally bypass the classifiers and exit from the mill along with the fine particles as the product stream due to the lower density of biomass compared to coal particles. Parametric simulations indicate that small
changes to classifier vane angle and lengthening of the vortex finder of the cyclone can both decrease the number of large particles exiting along with the fine product stream. These two modifications provide a synergistic contribution to classifier performance. In addition, it was determined that changes in air velocity do not lead to a significant impact on the PSD of the product stream. Generally, a combination of simple modifications to hardware configuration and modifications to operating conditions may optimize mill performance by improving PSD while maintaining the product flow rate. Our novel contribution to the state of the art generated by this study includes the following:

- Reasonable assumptions were developed and validated for representing moving mechanical components of the pulverizer in order to converge CPFD simulations that match experimental measurements;
- A C# code was developed to post-process output files of Barracuda and track trajectory of individual particles in the pulverizer.
- The impact of simple manipulations to pulverizer geometry and operating parameters on pulverizer performance was determined and optimized for co-milling.

10.2 Cofiring the blend of woody biomass and coal and compare the samples of ash deposit and ash aerosol to those of coal combustion

In order to develop data set of ash behavior during cofiring of woody biomass and coal, experiments and demonstrations were performed in a 1500 kW\textsubscript{TH} (in University of Utah) and a 500 MW\textsubscript{e} furnace (Hunter power plant, Unit-3). Two types of woody biomass were tested: torrefied and steam exploded. The blends of pulverized woody biomass were prepared as a 15% biomass, 85% coal by weight mixture. We designed and constructed an ash deposit probe to
provide a representative heat exchanger surface where ash deposits could be collected and subsequently analyzed. The collected ash deposit samples were weighed carefully to determine ash deposit rate. Additionally, we designed and constructed an ash aerosol probe to collect and analyze the aerosol particles, which was operated by the University of Utah. The entrained ash aerosol and ash deposit samples were analyzed to determine PSD, composition and morphology.

PSD and composition of the entrained ash aerosol particles do not change significantly by cofiring woody biomass with coal at the blending ratios we tested. The combustion of fuels in 1500 kW\textsubscript{TH} (pilot) scale combustor were observed to produce a three modal PSD; however, the distribution was bimodal in the full-scale furnace. The extra mode of the PSD in pilot combustor is in the range of 0.01 to 0.1 µm, which is likely due to the nucleation of vapor species. In addition, the concentration of ash aerosol particles in the size range of 0.1-1.0 µm while cofiring biomass in the pilot combustor was slightly higher than that while firing coal, which is likely due to coagulation of nuclei. The concentration of alkali metals in the submicron entrained ash particles while firing the blended fuel was slightly higher than that of coal, which is likely due to the increased vaporization of organically bound metals. In both the pilot and full-scale combustor experiments, Al and Si content of the ash aerosols were slightly higher for the blend combustion than those for coal, which contribute to aluminosilicate compositions.

Similar to the results of the entrained ash particles, the analysis of ash deposit samples did not deviate significantly between the pure coal and the blended fuels. The measured ash deposit rates during pilot-scale testing depended upon the port location and the collection-time period; and the results were very similar for both pure coal and biomass co-firing. Deposition measurements showed that the rates were higher at the location closer to the burner of the pilot-scale combustor, which is likely due to the higher concentration of fly ash particles at that location because of ash
deposition on the floor of the unit. The ash deposit rates of Hunter-3 are almost identical at short collection times. However, at long collection times they are likely impacted by the shedding ash deposit from an upstream heat exchanger, or growth on the hot probe tip.

In addition to the above results, cofiring woody biomass with coal in Hunter#3 showed that the mill performance and flue gas species were impacted by co-milling. The torrefied blend increased the power requirement for the mills. In addition, NO$_x$ and SO$_2$ emissions were reduced during cofiring.

Our unique contribution to the state of the art resulting from this study includes the following:

- Design and construct a novel controlled-temperature ash deposit probe and an extractive entrained ash sampling probe in order to operate in both pilot and full-scale boilers.
- The behavior of co-firing biomass with coal was investigated at multiple scales, indicating the efficacy of performing pilot-scale testing.
- This study includes the analysis of coupled ash aerosol and ash deposit providing a more robust set of inputs for model development and validation.

10.3 Ash deposit model of the combustion of a range of solid fuels in a 100-kW combustor.

We developed a model in this study in order to help operators of utility boilers predict relative ash deposit rates for blends of solid fuels incorporating a broad range of fuel chemistries. The model accounts for four mechanisms of ash deposition: inertial impaction, thermophoresis, condensation, and eddy impaction. Thermodynamic modeling of mineral matter using FactSage indicated that NaCl and KCl are the main alkali vapor species in the flue gas. Two ash deposit
stickiness models were investigated: the melt fraction stickiness model (MFSM), which is based on the condensed sticky layer and melt fraction of the ash, and the kinetic energy stickiness model (KESM), which is a function of the critical velocity of fly ash particles. The condensation in the MFSM model creates a sticky layer that enhances the sticking efficiency of the impacting particles and accelerates the ash deposit growth. We evaluated the developed model with the experimental data of the combustion of eleven solid fuels in a 100 kW down-fired combustor. It was shown that the performance of the MFSM model was much better than the KESM model by calculating root-mean-squared-error when predicting the deposition rate while firing 11 different fuels. The enhanced performance of the MFSM model was likely due to accounting for the significant effect of the condensed alkali materials and their effect on sticking efficiency. In addition, the results of the MFSM demonstrated that the inertial impaction and thermophoresis are the dominant mechanisms in ash deposit formation on the upstream and the downstream side surfaces. The developed model that was evaluated against 100 kW experimental data was also able to predict the ash deposit rates occurring in the pilot-scale combustor (experimental data presented in Chapter 6).

Our contribution to this study includes the followings:

- We developed an ash deposition model by combining mechanistic models from the literature including descriptions of inertial impaction, thermophoresis, condensation and eddy impaction.

- A novel method of calculating sticking efficiency based on fuel alkali and chlorine composition calculated condensation temperatures was utilized.
The model was validated by predicting the ash deposit rate of a variety of solid fuels in a 100-kW combustor and was then used to predict the results of the pilot-scale experiments.

A model was developed to help the operators of utility boilers evaluate the expected relative deposition rate in the convective section (fouling) while firing blends of biomass and coal.
11 Recommended future work

A list of recommendations for future work in the area is presented here:

- Careful analysis of the effect of operating parameters on PSD of the product stream of co-milling woody biomass with coal.

- Our ash deposit probe was modified by adding cooling water and load cells and an extra thermocouple. However, due to dissimilar thermal expansion in the air- and water-cooled section of the probe, the weld broke rendering the water cooling jacket ineffective. For future studies a redesign of the cooling jacket is recommended.

- Design a more advanced ash deposit probe in order to record continuously the dynamic shape of ash deposit while maintaining the specifications of the designed probe of this research. This could be accomplished through mass or video measurements of the deposit.

- Determine the impact of the following parameters on the ash deposit rate of cofiring of biomass with coal:
  - higher percentages of woody biomass;
  - additives to reduce ash formation and ash deposition [215];
  - optimization of the operating parameters of boiler conditions to obtain maximum boiler performance during cofiring.
• Investigate the following parameters in improving the model of ash deposit rate:
  
  o More accurate estimation of melt fraction of bulk ash particles and ash deposits;
  
  o Develop a more robust methodology for representing realistic shape of ash deposit;
  
  o Inclusion of chemical reaction mechanisms on ash deposit rate;
  
  o Determination of the eroding effect of impacting ash particles and prediction of ash deposit rate over long periods;
  
  o Measurement the PSD of the super-micron fly ash particles (1 to 100 µm) and investigate its effect on the model;
References

1. Carbon emissions from burning biomass for energy. Partnership for Policy Integrity.


54. Library, W.E. Fuels and their ash content. Available from: [http://www.nzdl.org/gsdlmod?e=d-00000-00---off-0envl--00-0-----0-10-0---0---0direct-10---4-------0-1l--11-en-50---20-help---00-0-1-00-0-4----0-0-11-10-0utfZz-8-00&cl=CL1.1&d=HASH0128ef689bc19ae2d6124941.10.8&g=1](http://www.nzdl.org/gsdlmod?e=d-00000-00---off-0envl--00-0-----0-10-0---0---0direct-10---4-------0-1l--11-en-50---20-help---00-0-1-00-0-4----0-0-11-10-0utfZz-8-00&cl=CL1.1&d=HASH0128ef689bc19ae2d6124941.10.8&g=1).


61. Valmari, T., Potassium behaviour during combustion of wood in circulating fluidised bed power plants, in VTT Chemical Technology. 2000, Helsinki University of Technology.


120. O. P. Mahajan, P.L.W., Jr., *Porosity of coals and coal products*, in *Analytical methods for coal and coal products*.


159. John H. Cameron, K.G.-W., *Role of thermophoresis in the deposition of fume particles resulting from the combustion of high inorganic containing fuels with reference to kraft black liquor.* Fuel Processing Technology, 1999. **60:** p. 49-68.


175. Dr. Bradley R. Adams, D.A.R.F., Dr. Constance L. Senior, Dr. Hong Shig Shim, Dr. Huafeng Wang, Prof. Jost Wendt (University of Utah), Dr. Christopher Shaddix (Sandia), *Characterization of Oxy-combustion Impacts in Existing Coal-fired Boilers*. 2009.


207. *Fly Ash, Slag, Silica Fume, and Natural Pozzolans*, in *Design and Control of Concrete Mixtures*.


Appendix A: The developed code for analysis of particle trajectories

Introduction:
This code was developed by Tanner Jasperson, tanner.jasperson@gmail.com
It is provided as-is with no liability for any application or use of the code.

Any future public or personal use of the program, with or without modifications, may be done without permission of the author, provided that the original author is credited and listed as a contributor to any project or purpose for which the code is used.

Regardless of the modifications, this program must always be provided as an open-source, free resource. Future applications of this code must be developed and disturbed under the same license and restrictions.

Any modifications of this code must include this header along with notices stating when and how it was modified.

PURPOSE:
This code was developed to be applied to a specific use of the Barracuda CPFD. A certain 'classifier' was designed and used in the program to simulate the behavior of particulate matter in the system. The output files from that program are the inputs to this one.

This program relies solely on the data output (.GMV text files) from Barracuda. It analyzes the data and presents relevant information on the positions and properties of the particles in the system.

To do so, the program makes use of the ZedGraph utility (published under the GNU Lesser General Public License v2.1). It works with the UI to provide graphs and other visuals. There are 2 files associated with this utility (two drivers). They are dependencies of the program and must be present in the same directory as the compiled program for it to run.

This code was written and compiled in Microsoft Visual Studio, and it is suggested that any future work on the code be done in the same environment. The compiled program is provided under the conditions detailed above.

USE:
To begin with the program, select File > Import Data. A file dialog box will open and allow you to select the .GMV file you wish to analyze. Once it has been selected, the program will read the file and pick out the particles
that are inside of the defined classifier bounds. The particles are saved as Particle objects in a List. The bounds of the classifier are defined by the WallBoundary object. The shape of the classifier defined by this program is a section of an inverted cone.

Once the particles have been read, the program will display a 2-D plot of the particles inside of the classifier. Initially, this is an x-vs-y plot where all of the particles are projected onto the xy plane. The two drop-down menus by the graph’s axis can be used to modify what is being plotted. Mass, speed, and Particle Radius are properties of the individual particles as modeled by Barricuda. X, Y, Z, radius, and dWall (distance from the wall) come from the positional coordinates of the particle. PID is the unique ID given to each particle by Barricuda. The particles are of both wood and coal type.

Note that by default a parameter is applied on the speed of the particles, only showing the ones whose speed is less than 10% of the average speed.

The plot can also be modified visually by the options in the left-hand pane. Further details about the particles can be obtained by setting parameters on the particles in the ‘Parameters’ drop-down menu. There, you can set bounds on the mass, speed, and position of the particle. In that same menu, you can select to ‘View 3D explorer’ which will open a new window in which you can explore the positions of the particles in three dimensions. The same filters applied in the main page of the program are passed to the 3D explorer.

Any subset of selected particles can be exported to a .CSV for further analysis. This is done through the File Menu- File > Export… The program will open a file dialog box to allow you to choose where to save the file.

Another use of the program is to explore path of particles leaving the top of the classifier. This is referred to as ‘Reverse Engineering’ the classifier because it is used to determine how the particles that exit the classifier approach the exit. To do this, you will need a folder with all of the GMV files you wish to analyze. Barricuda outputs the files with names that reflect the time they correspond to, and the program relies on these file names to determine the order of the files in time.

To analyze particles over time, select File > Reverse Engineering. A new window will appear prompting you to select the particles and to set a minimum particle radius for the analysis. You will be given a file dialog box to use to select the particle file you wish to begin the analysis with, and the program will scan the directory for the other files. Reading all of the files can take a while, but once all of the data is loaded, all of the tools discussed above can be used to explore the particles in time.

Two new buttons will be available below the graph to step either forward or backward in time. Pressing the button will display the properties of the particles at the next time step. In addition, more of the ‘Classifier stats’ will be printed in the left-hand column for this time-inclusive analysis. Here, you will be able to see the number of exiting particles, their masses, and residence times. Note that one of the options in the ‘Parameters’ menu
is the ‘Swirl Y’, which is the minimum depth a particle must reach inside
the classifier in order to be considered to have ‘swirled’ inside of it.

The 3D explorer is especially useful with the time-inclusive data. Opening
it will display the same 3D navigator, but with lines tracing the path taken
by the particles with time. In addition, a listbox will allow you to select
specific particles to see their paths individually or as a group. Here you
will be able to adjust the ‘Swirl Y’, and select all swirling particles.
This allows you to view the paths of the particles that swirl before exiting.

FUTURE WORK:
This program was developed with hard-coded parameters for the classifier
size and shape. Future work on the WallBoundary object could allow for a
more robust classifier size definition. Furthermore, user-definable ‘exit
zones’ for use in the Reverse Engineering analysis could be added to replace
the hard-coded one.

*/

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using ZedGraph;

namespace ClassiParse
{
    public partial class ClassiParseForm : Form
    {
        //These are our global variables
        //particles is a list of all the particles that were loaded from
        //the file.
        //selectedParticles is a list of the particles that meet the
        //filter parameters.
        private List<Particle> particles, selectedParticles;

        //curve1 is wood, curve2 is coal. They are used in the plots as
        LineItems and BarItems
        CurveItem curve1, curve2;

        List<CurveItem> curvesClassifier;

        //pointSymbol is the object that defines the type of point used on
        //the scatter plot
        GraphicsPath pointSymbol;
    }
}
// classifier is a WallBoundary object that defines the bounds of
// the classifier itself. Modify the bounds in that .cs file
WallBoundary classifier;

// this will hold the units for the axis labels
Dictionary<string, string> units;

// this bool will keep track of when the graph is being updated
bool makingGraph;

// and this one will note when a file is being read. If it is
// true, any attempt to regraph will be canceled.
bool readingFile;

// this string contains the labels for the x axis
double[] xAxisLabels;

// This bool keeps track of whether or not we are doing reverse
// engineering
bool time = false;

// This is the total number of timesteps
int totAge = 1;

// This int keeps track of which timestep we are on
int timeStep = 0;

// This list keeps track of the file names corresponding to
timesteps
List<string> fileNames;

// these constants are for the types
const string COAL_TYPE = "2";
const string WOOD_TYPE = "1";

public ClassiParseForm()
{
    // initialize all the form components
    InitializeComponent();

    // initialize the lists of particles
    particles = new List<Particle>();
    selectedParticles = new List<Particle>();

    // initialize the classifier WallBoundary.
    // The parameter provided details which fitting method will be
    // used for the wall estimation.
    classifier = new WallBoundary(12);

    // calls the function that makes the pointSymbol (based on the
    // selected point size)
    makePointSymbol();
}
// set graph to not scale fonts (so if you resize it, the graph gets bigger, but the font doesn't)
zedGraph.GraphPane.IsFontsScaled = false;

// set the graph to show point values with they are highlighted with the mouse
zedGraph.IsShowPointValues = true;

// setup the graph options pane
updateGraphOptions();

// setup the units
units = new Dictionary<string, string>();
units["mass"] = "kg";
units["speed"] = "m/s";
units["Particle Radius"] = "µm";
units["radius"] = "m";
units["x"] = "m";
units["y"] = "m";
units["z"] = "m";
units["dWall"] = "m";
units["count"] = "";
units["% count"] = "%";
units["% mass"] = "%";
units["PID"] = "ID #";
units["type"] = "1 = wood, 2 = coal";

// set the selected items for the two axis
comboBoxXAxis.SelectedIndex = comboBoxXAxis.FindStringExact("Particle Radius");
comboBoxYAxis.SelectedIndex = comboBoxYAxis.FindStringExact("% mass");

// initiate the makingGraph bool
makingGraph = false;

// initiate the readingFile bool
readingFile = false;

// initiate the x axis labels string
xAxisLabels = new double[100];

fileNames = new List<string>();
fileNames.Add("no file loaded");

curvesClassifier = new List<CurveItem>();

}

// this function 'builds' the point used in the scatter plots. It is based off of selected value for point size
public void makePointSymbol()
{
    // set r to equal .05 times the selected value in the combobox.
float r = .05f * float.Parse(comboBoxPointSize.SelectedItem.ToString());

//create the pointSymbol. The code here contains two arrays, one with the coordinates of the lines we are going to draw (it makes a diamond) and the other dictates where the starting point is and where the lines are.
pointSymbol = new GraphicsPath{
    new PointF(-r, 0.0f), new PointF(0.0f, -r), new PointF(r, 0.0f), new PointF(0.0f, r), new PointF(-r, 0.0f)
},
    new[]{
        (byte)PathPointType.Start,
        (byte)PathPointType.Line, (byte)PathPointType.Line,
        (byte)PathPointType.Line, (byte)PathPointType.Line
    });

//adjusts the value of a parameter based on the selected units
public double ParameterUnits(double x, string unit)
{
    switch (unit)
    {
        case "%":
            x /= 100;
            break;
        case "µm":
            x /= 1000000;
            break;
        case "mm":
            x /= 1000;
            break;
        case "cm":
            x /= 100;
            break;
        case "in":
            x /= (100 / 2.54);
            break;
        case "g":
            x /= 1000;
            break;
        case "mg":
            x /= 1000000;
            break;
        case "m":
            break;
        case "m/s":
            break;
    }
    return x;
}
return x;

// checks the configured parameters and creates an arraylist (selectedParticles) of particles within them.
public void filterParticles()
{
    // remove all particles currently in the list
    selectedParticles.Clear();

    if (time)
    {
        selectedParticles = new List<Particle>(particles);
        return;
    }

    // set the cursor to the wait cursor while the graph is made
    Cursor.Current = Cursors.WaitCursor;
    // and set up the progress bar
    progressBar.Visible = true;
    progressBar.Maximum = particles.Count; // this is the total number of loop iterations that happen here
    int loops = 0; // this int will keep track of each loop

    progressStatus.Text = "Applying bounds"; // set the progress status
    Application.DoEvents();

    // dWall Parameters
    string[] dWallString =
        toolStripComboBoxDWallParameter.SelectedItem.ToString().Split(' ');
    bool dWallPercent = (dWallString[1] == "%");
    double dWallParameter = 10000;
    if (dWallString[0] != "")
        dWallParameter = ParameterUnits(double.Parse(dWallString[0]), dWallString[1]);

    // speed Parameters
    string[] speedString =
        toolStripComboBoxSpeedParameter.SelectedItem.ToString().Split(' ');
    string speedUnit = speedString[1];
    bool speedPercentMax = (speedUnit == "%-max");
    bool speedPercentMean = (speedUnit == "%-mean");
    if (speedPercentMax || speedPercentMean)
        speedUnit = "%";
    double speedParameter = 10000;
    if (speedString[0] != "")
        speedParameter = ParameterUnits(double.Parse(speedString[0]), speedUnit);

    // y bounds parameters
    bool ys1, ys2, ys3, ys4 = true;
ys1 = section1ToolStripMenuItem.Checked ||
top2ToolStripMenuItem.Checked ||
section2ToolStripMenuItem.Checked;
ys2 = section2ToolStripMenuItem.Checked ||
top2ToolStripMenuItem.Checked ||
middle2ToolStripMenuItem.Checked ||
threeSectionsToolStripMenuItem.Checked;
ys3 = section3ToolStripMenuItem.Checked ||
middle2ToolStripMenuItem.Checked ||
bottom2ToolStripMenuItem.Checked ||
threeSectionsToolStripMenuItem.Checked;
ys4 = section4ToolStripMenuItem.Checked ||
bottom2ToolStripMenuItem.Checked ||
bottom3ToolStripMenuItem.Checked;
if (fullClassifierToolStripMenuItem.Checked)
    ys1 = ys2 = ys3 = ys4 = true;
double[] yParameters = classifier.YBounds(ys1,ys2,ys3,ys4);

// particle radius parameters
double pRadLowParameter = 0;
double pRadHighParameter = 10000;
try
{
    pRadLowParameter =
double.Parse(toolStripTextBoxPRadiusLower.Text);
pRadHighParameter =
double.Parse(toolStripTextBoxPRadiusUpper.Text);
}
catch { }
bool pRadPercent =
ToolStripMenuItemPRadiusLowerPercent.Checked;
string pRadUnits = "µm";
if (pRadPercent)
pRadUnits = "%";
pRadLowParameter = ParameterUnits(pRadLowParameter,
pRadUnits);
pRadHighParameter = ParameterUnits(pRadHighParameter,
pRadUnits);

// particle mass parameters
double massLowParameter = 0;
double massHighParameter = 10000;
try
{
    pRadLowParameter =
double.Parse(toolStripTextBoxMassLower.Text);
pRadHighParameter =
double.Parse(toolStripTextBoxMassUpper.Text);
}
catch { }
bubble massPercent =
ToolStripMenuItemMassLowerPercent.Checked;
string massUnits = "kg";
if (massPercent)
massUnits = "%";
else if (ToolStripMenuItemMassLowerMG.Checked)
massUnits = "mg";
else if (ToolStripMenuItemMassLowerG.Checked)
massUnits = "g";
massLowParameter = ParameterUnits(massLowParameter, massUnits);
massHighParameter = ParameterUnits(massHighParameter, massUnits);

bool withinParamemters = true;
foreach (Particle p in particles)
{
    // check dWall parameters
    if (dWallPercent)
        withinParamemters = (p.dWall <= p.wallAt * dWallParameter);
    else
        withinParamemters = (p.dWall <= dWallParameter);

    // check speed parameters
    if (speedPercentMax)
        withinParamemters = (p.speed <= classifier.speedH * speedParameter) && withinParamemters;
    else if (speedPercentMean)
        withinParamemters = (p.speed <= classifier.speedA * speedParameter) && withinParamemters;
    else
        withinParamemters = (p.speed <= speedParameter) && withinParamemters;

    // check y low and y high parameter
    withinParamemters = (p.y <= yParameters[1]) && withinParamemters;
    withinParamemters = (p.y >= yParameters[0]) && withinParamemters;

    // check radius high and low parameter
    if (pRadPercent)
    {
        withinParamemters = (p.pRadius <= (classifier.radiusL + (classifier.radiusH - classifier.radiusL) * pRadHighParameter)) && withinParamemters;
        withinParamemters = (p.pRadius >= (classifier.radiusL + (classifier.radiusH - classifier.radiusL) * pRadLowParameter)) && withinParamemters;
    }
    else
    {
        withinParamemters = (p.pRadius <= pRadHighParameter*1000000) && withinParamemters;
        withinParamemters = (p.pRadius >= pRadLowParameter*1000000) && withinParamemters;
    }

    // check mass high and low parameter
    if (massPercent)
    {
    
}
withinParamemters = (p.mass <= (classifier.massL + (classifier.massH - classifier.massL) * massHighParameter)) &&
withinParamemters;
withinParamemters = (p.mass >= (classifier.massL + (classifier.massH - classifier.massL) * massLowParameter)) &&
withinParamemters;
else
{
    withinParamemters = (p.mass <= massHighParameter * 1000000) && withinParamemters;
    withinParamemters = (p.mass >= massLowParameter * 1000000) && withinParamemters;
}

//check particle type parameter
withinParamemters = (!(!coalParameterToolStripMenuItem.Checked && p.type == COAL_TYPE) &&
withinParamemters;
withinParamemters = (!(!woodParameterToolStripMenuItem.Checked && p.type == WOOD_TYPE) &&
withinParamemters;

//adds the particle to the 'selectedParticles' list if it has met all the parameters
if (withinParamemters)
    selectedParticles.Add(p);
loops++;
//if loops reaches 20, reset it and trigger the progress bar.
if (loops == 20)
{
    loops = 0;
    if (progressBar.Value + 20 > progressBar.Maximum)
        progressBar.Value = progressBar.Maximum - 1;
    else
        progressBar.Value += 20;
    Application.DoEvents();
}

//set the cursor back to normal
Cursor.Current = Cursors.Default;
//and hide and reset the progress bar and status
progressBar.Visible = false;
progressBar.Value = 0;
progressStatus.Text = "";
}

//this function updates the available graph options when the selected graph type changes
public void updateGraphOptions()
{
    //based on what is selected, make the needed changes to the GUI
switch (comboBoxGraphType.SelectedItem.ToString())
{
    case "Scatter Plot":
        checkBoxConnectPoints.Visible = true;
        checkBoxShowCoal.Visible = true;
        checkBoxShowWood.Visible = true;
        labelPointSize.Visible = true;
        comboBoxPointSize.Visible = true;
        labelDivisions.Visible = false;
        comboBoxDivisions.Visible = false;
        comboBoxYAxis.Items.Clear();
        comboBoxYAxis.Items.AddRange(new string[] { "mass", "Particle Radius", "speed", "x", "y", "z", "radius", "dWall", "PID", "type" });
        break;
    case "Histogram":
        checkBoxConnectPoints.Visible = false;
        checkBoxShowCoal.Visible = true;
        checkBoxShowWood.Visible = true;
        labelPointSize.Visible = false;
        comboBoxPointSize.Visible = false;
        labelDivisions.Visible = true;
        comboBoxDivisions.Visible = true;
        comboBoxYAxis.Items.Clear();
        comboBoxYAxis.Items.AddRange(new string[] { "% count", "% mass", "count", "mass" });
        break;
}
//this function calls the make() method for the selected type of graph
public void makeGraph()
{
    // if a file is being read, cancel this action.
    if (readingFile) return;

    // if a graph is currently being made, remove everything from selected particles and begin to remake it
    if (makingGraph)
    {
        selectedParticles.Clear();
        zedGraph.GraphPane.CurveList.Clear();
    }

    //set makingGraph to true for while we are running processes
    makingGraph = true;
    filterParticles();
    //based on what is selected, call the appropriate make() method
    switch (comboBoxGraphType.SelectedItem.ToString())
    {
        case "Scatter Plot":
            makeScatter(); break;
        case "Histogram":
            makeHyogram(); break;
    }
}
makeHistogram(); break;

// set the angles a, b, and c, based off of the axis
string xAxis, yAxis;
try
{
    xAxis = comboBoxXAxis.SelectedItem.ToString();
    yAxis = comboBoxYAxis.SelectedItem.ToString();
}
catch
{
    comboBoxXAxis.SelectedIndex = comboBoxXAxis.FindStringExact("Particle Radius");
    comboBoxYAxis.SelectedIndex = comboBoxYAxis.FindStringExact("% mass");
    xAxis = "Particle Radius";
    yAxis = "% mass";
}

string graphType = xAxis + yAxis;
double a, b, c;
double originLift = 0;
bool showClassifier = checkBoxShowClassifier.Checked;

// Set the angles depending on the type of graph desired
double halfPI = Math.PI / 2;
double fourthPI = Math.PI / 4;
switch (graphType)
{
    // the horizontal cases are -pi/4, -pi/2, pi/4
    case "yx":
        case "yz":
            a = -fourthPI;
            b = -halfPI;
            c = fourthPI;
            break;

    // The vertical cases are -pi/2, pi/4, pi/2
    case "zy":
        case "xy":
            a = -halfPI;
            b = fourthPI;
            c = halfPI;
            break;

    // top view cases are 0, 0, 0
    case "xz":
        case "zx":
            a = 0;
            b = 0;
            c = 0;
            break;
}
// if there is not a type, don't make the curve
default:
    a = 0;
b = 0;
c = 0;
    showClassifier = false;
    break;
}

// make the classifier curve
if (showClassifier)
{
    classifier.setAngles(a, b, c, originLift);
    LineItem classiOutlineTop =
    zedGraph.GraphPane.AddCurve("", classifier.outlineTopXs,
    classifier.outlineTopYs, Color.Brown);
    classiOutlineTop.Line.IsVisible = true;
    classiOutlineTop.Symbol.IsVisible = false;

    LineItem classiOutlineBot =
    zedGraph.GraphPane.AddCurve("", classifier.outlineBotXs,
    classifier.outlineBotYs, Color.Brown);
    classiOutlineBot.Line.IsVisible = true;
    classiOutlineBot.Symbol.IsVisible = false;

    LineItem extremePosXs = zedGraph.GraphPane.AddCurve("",
    classifier.extremePosXs()[0], classifier.extremePosXs()[1], Color.Brown);
    extremePosXs.Line.IsVisible = true;
    extremePosXs.Symbol.IsVisible = false;
    LineItem extremeNegXs = zedGraph.GraphPane.AddCurve("",
    classifier.extremeNegXs()[0], classifier.extremeNegXs()[1], Color.Brown);
    extremeNegXs.Line.IsVisible = true;
    extremeNegXs.Symbol.IsVisible = false;
    LineItem extremePosYs = zedGraph.GraphPane.AddCurve("",
    classifier.extremePosYs()[0], classifier.extremePosYs()[1], Color.Brown);
    extremePosYs.Line.IsVisible = true;
    extremePosYs.Symbol.IsVisible = false;
    LineItem extremeNegYs = zedGraph.GraphPane.AddCurve("",
    classifier.extremeNegYs()[0], classifier.extremeNegYs()[1], Color.Brown);
    extremeNegYs.Line.IsVisible = true;
    extremeNegYs.Symbol.IsVisible = false;
}

// Set the zoom up so that it shows the classifier, it that's what it should do
// The different graph types handle the scaling differently
switch (comboBoxGraphType.SelectedItem.ToString())
{
    case "Scatter Plot":
        zedGraph.AxisChange();
        zedGraph.Refresh();
        break;
    case "Histogram":
        zedGraph.RestoreScale(zedGraph.GraphPane);
break;
}

//set makingGraph to false now that we're done.
makingGraph = false;

}

//this function makes a histogram based on the selected x axis
public void makeHistogram()
{
    //determine the two axis types
    string xAxis;
    string yAxis;
    try
    {
        xAxis = comboBoxXAxis.SelectedItem.ToString();
        yAxis = comboBoxYAxis.SelectedItem.ToString();
    }
    catch
    {
        comboBoxXAxis.SelectedIndex = comboBoxXAxis.FindStringExact("Particle Radius");
        comboBoxYAxis.SelectedIndex = comboBoxYAxis.FindStringExact("% mass");
        xAxis = "Particle Radius";
        yAxis = "% mass";
    }

    //get the total number of particles
    double numParticles = selectedParticles.Count;

    //get the number of 'divisions' for the histogram's x axis
    int categories = int.Parse(comboBoxDivisions.SelectedItem.ToString());
    //in the case that 'type' is on the x-axis, categories will automatically be 2
    if (xAxis == "type")
    {
        categories = 2;
    }

    //set the cursor to the wait cursor while the graph is made
    Cursor.Current = Cursors.WaitCursor;
    //and set up the progress bar
    progressBar.Visible = true;
    progressBar.Maximum = (int)(numParticles * categories); //this is a guess at the total number of loops that will happen here
    int loops = 0; //this int will keep track of each loop iteration
    progressStatus.Text = "Building Graph";
    Application.DoEvents();
}
//remove the current curves from the graph pane.
zedGraph.GraphPane.CurveList.RemoveRange(0,
zedGraph.GraphPane.CurveList.Count);

//determine the range of the x axis item and determine the
histogram ranges. Also determine the total mass.
double xAxisMax = -100;
double xAxisMin = 100;
double totalMass = 0.0;
foreach (Particle p in selectedParticles)
{
    if (p.properties() == null) continue;
    foreach (Property prop in p.properties())
    {
        if (prop.name == xAxis)
        {
            if (prop.value < xAxisMin)
            {
                xAxisMin = prop.value;
            }
            if (prop.value > xAxisMax)
            {
                xAxisMax = prop.value;
            }
            totalMass += prop.value;
        }
    }
}
//set the 'step' of the histogram's x axis based on the
selected number of categories
double histogramSpacingStep = (xAxisMax - xAxisMin) /
categories;

//make two arrays: one to hold the axis titles and one to hold
the count
xAxisLabels = new double[categories];
double[] histogramWoodCount = new double[categories];
double[] histogramCoalCount = new double[categories];

double addAmount = 0.0;
for (int j = 0; j < categories; j++)
{
    //calculate 'i', the histogram range value at j
    double i = xAxisMin + j * histogramSpacingStep;
    //set the xAxis label number j with i
    xAxisLabels[j] = i;//.ToString("0.00E0");

    //initiate the count of particles in range j
histogramWoodCount[j] = 0.0;
histogramCoalCount[j] = 0.0;
through now

//increase the max by the number of particles we'll loop
//progressBar.Maximum += selectedParticles.Count;
//Application.DoEvents();

//add to the count of particles in range j for each
particle in the range.
for (int k = 0; k < selectedParticles.Count(); k++)
{
if (selectedParticles[k].properties() == null)
continue;
if (selectedParticles[k].properties()[xAxis] <= i)
{
//determine what to add to the count based on what
is selected in the y axis combo box
switch (yAxis)
{
case "% count": addAmount = 100 /

numParticles; break;

case "% mass": addAmount =
selectedParticles[k].properties()["mass"] / totalMass; break;
case "count": addAmount = 1; break;
case "mass": addAmount =
selectedParticles[k].properties()["mass"]; break;
}
//add one to the histogram value for this range.
if (selectedParticles[k].type == COAL_TYPE)
histogramCoalCount[j] += addAmount;
else
histogramWoodCount[j] += addAmount;
//remove particles so they aren't double-counted
selectedParticles.RemoveAt(k--);
// add the particles to 'loops' so that it doesn't

track them anymore
}

progress bar.

progressBar.Maximum)

loops += categories - j - 1;

//toggle the 'loops' counter
loops += 1;
//if loops reaches 50, reset it and trigger the
if (loops >= 1000)
{
loops -= 1000;
if (progressBar.Value + 1000 >

181


progressBar.Value = progressBar.Maximum - 1;
else
    progressBar.Value += 1000;
Application.DoEvents();
}
}

// increment the progress bar for this loop
if (progressBar.Value + 1 < progressBar.Maximum)
    progressBar.Value += 1;

if ((yAxis == "% mass" || yAxis == "% count") && j != 0)
{
    histogramCoalCount[j] += histogramCoalCount[j - 1];
    histogramWoodCount[j] += histogramWoodCount[j - 1];
}

/// remove stuff that doesn't matter with '.Take(stoppedAt).ToArray()'?

/// add any remaining values to the final category, then clear the list.
foreach (Particle p in selectedParticles)
{
    if (p.properties() == null) continue;
    switch (yAxis)
    {
    case "% count": addAmount = 100 / numParticles; break;
    case "% mass": addAmount = p.properties()["mass"] / totalMass; break;
    case "count": addAmount = 1; break;
    case "mass": addAmount = p.properties()["mass"]; break;
    }

    if (p.type == COAL_TYPE)
        histogramCoalCount[categories - 1] += addAmount;
    else
        histogramWoodCount[categories - 1] += addAmount;
}
selectedParticles.Clear();

// update the progress status
progressStatus.Text = "Drawing Graph";
Application.DoEvents();

// create and define the bars
curve1 = zedGraph.GraphPane.AddCurve("Wood", xAxisLabels, histogramWoodCount, Color.Red);
((LineItem)curve1).Symbol.IsVisible = false;
// bar1.Bar.Fill = new Fill(Color.Red);
// bar1.Bar.Border = new Border(false, Color.Red, 1);
curve2 = zedGraph.GraphPane.AddCurve("Coal", xAxisLabels,
histogramCoalCount, Color.Blue);
((LineItem)curve2).Symbol.IsVisible = false;
// bar2.Bar.Fill = new Fill(Color.Blue);
// bar2.Bar.Border = new Border(false, Color.Blue, 1);

if (checkBoxShowWood.Checked)
    zedGraph.GraphPane.CurveList.Remove(curve1);
if (!checkBoxShowCoal.Checked)
    zedGraph.GraphPane.CurveList.Remove(curve2);

zedGraph.GraphPane.Title.Text = xAxis + " vs " + yAxis;
zedGraph.GraphPane.XAxis.Title.Text = xAxis + " (" + units[xAxis] + ")";
zedGraph.GraphPane.YAxis.Title.Text = yAxis + " (" + units[yAxis] + ")";

filterParticles();

Cursor.Current = Cursors.Default; //and hide and reset the progress bar and status
progressBar.Visible = false;
progressBar.Value = 0;
progressStatus.Text = "";

}

//this function makes a scatter plot based on the selected x and y axis
public void makeScatter()
{
    //set the cursor to the waiting cursor while we process this
    Cursor.Current = Cursors.WaitCursor;
    //and set up the progress bar
    progressBar.Visible = true;
    progressBar.Maximum = selectedParticles.Count; // this is the total number of loop iterations that happen here
    int loops = 0; //this int will keep track of each loop iteration
    progressStatus.Text = "Building Graph";
    Application.DoEvents();

    //detect what was selected for the x and y axis. If an error occurs, use the default, x and y.
string xAxis;
string yAxis;
try
{
    xAxis = comboBoxXAxis.SelectedItem.ToString();
    yAxis = comboBoxYAxis.SelectedItem.ToString();
}
catch
{
    comboBoxXAxis.SelectedIndex = comboBoxXAxis.FindStringExact("x");
    comboBoxYAxis.SelectedIndex = comboBoxYAxis.FindStringExact("Y");

    xAxis = "x";
    yAxis = "y";
}

//removes the current curves from the graph pane.
zedGraph.GraphPane.CurveList.RemoveRange(0, zedGraph.GraphPane.CurveList.Count);

//create the two lists of point pairs that will be put onto the pane. Series 1 is for wood and series 2 is for coal
PointPairList series1 = new PointPairList();
PointPairList series2 = new PointPairList();

//loop through the particles and add them to either series 1 (wood) or series 2 (coal)
double x = 0;
double y = 0;
foreach (Particle p in selectedParticles)
{
    if (p.properties() == null) continue;

    x = p.properties()[xAxis];
    y = p.properties()[yAxis];

    if (p.type == COAL_TYPE)
        series1.Add(x, y);
    else
        series2.Add(x, y);

    //toggle the 'loops' counter
    loops++;
    //if loops reaches 50, reset it and trigger the progress bar.
    if (loops == 50)
    {
        loops = 0;
        if (progressBar.Value + 50 > progressBar.Maximum)
            progressBar.Value = progressBar.Maximum - 1;
        else
            progressBar.Value += 50;
    }
// sort the series
series1.Sort();
series2.Sort();

// set the name of the graph based off of the selected x and y
axis options
zedGraph.GraphPane.Title.Text = xAxis + " vs " + yAxis;

// set the axis labels as well
zedGraph.GraphPane.XAxis.Title.Text = xAxis;
zedGraph.GraphPane.YAxis.Title.Text = yAxis;

// add the curves to the graph pane. Name them and give them
colors
curve1 = zedGraph.GraphPane.AddCurve("wood", series1, Color.Red, SymbolType.UserDefined);
curve2 = zedGraph.GraphPane.AddCurve("coal", series2, Color.Blue, SymbolType.UserDefined);

// if the graph settings for which curves to show are set, remove the curves that aren't checked
if (!checkBoxShowWood.Checked)
    zedGraph.GraphPane.CurveList.Remove(curve1);
if (!checkBoxShowCoal.Checked)
    zedGraph.GraphPane.CurveList.Remove(curve2);

// set the symbol (marker) for the curve to be the custom
sybmol (so it can be resized)
((LineItem)curve1).Symbol.UserSymbol = pointSymbol;
((LineItem)curve2).Symbol.UserSymbol = pointSymbol;

// set the color for the symbol fill
((LineItem)curve1).Symbol.Fill = new Fill(Color.Red);
((LineItem)curve2).Symbol.Fill = new Fill(Color.Blue);

// if the checkbox for connecting the points is checked, connect them.
((LineItem)curve1).Line.IsVisible = checkboxConnectPoints.Checked;
((LineItem)curve2).Line.IsVisible = checkboxConnectPoints.Checked;

// set the cursor back to normal
Cursor.Current = Cursors.Default;
// and hide and reset the progress bar and status
progressBar.Visible = false;
progressBar.Value = 0;
progressStatus.Text = "";

} //Function for when the 'connect points' checkbox is checked/unchecked
private void checkboxConnectPoints_CheckedChanged(object sender, EventArgs e) {
    //set the cursor to the wait cursor.
    Cursor.Current = Cursors.WaitCursor;

    //the try is to catch the error that occurs when the checkbox state is changed and no data is loaded.
    try {
        //connect the points or not depending on if the checkbox was checked or unchecked
        ((LineItem)curve1).Line.IsVisible = checkboxConnectPoints.Checked;
        ((LineItem)curve2).Line.IsVisible = checkboxConnectPoints.Checked;

        //refresh the graph
        zedGraph.Refresh();
    } catch { } 

    //set the cursor back to default
    Cursor.Current = Cursors.Default;
}

//Function for when the 'show wood' checkbox is checked/unchecked
private void checkBoxShowWood_CheckedChanged(object sender, EventArgs e) {
    //set the cursor to the wait cursor
    Cursor.Current = Cursors.WaitCursor;

    //the try is for the case when the checkbox is changed and no data is loaded.
    try {
        //show or hide the wood data (data1) depending on if the checkbox was checked or unchecked.
        if (checkBoxShowWood.Checked)
            zedGraph.GraphPane.CurveList.Add(curve1);
        else
            zedGraph.GraphPane.CurveList.Remove(curve1);
    } catch { } 

    //update the graph by indicating an axis change.
    zedGraph.AxisChange();
    zedGraph.Refresh();

    //set the cursor back to default
    Cursor.Current = Cursors.Default;
}
private void checkBoxShowCoal_CheckedChanged(object sender, EventArgs e)
{
    //set the cursor to the wait cursor
    Cursor.Current = Cursors.WaitCursor;

    //the try is for the case when the checkbox is changed and no data is loaded.
    try
    {
        //show or hide the coal data (data2) depending on if the checkbox was checked or unchecked.
        if (checkBoxShowCoal.Checked)
        {
            zedGraph.GraphPane.CurveList.Add(curve2);
        }
        else
        {
            zedGraph.GraphPane.CurveList.Remove(curve2);
        }
    }
    catch {}

    //update the graph by indicating an axis change.
    zedGraph.AxisChange();
    zedGraph.Refresh();

    //set the cursor back to default
    Cursor.Current = Cursors.Default;
}

private void comboBoxYAxis_SelectedIndexChanged(object sender, EventArgs e)
{
    //remake the graph with the new axis and refresh it
    makeGraph();
}

private void comboBoxXAxis_SelectedIndexChanged(object sender, EventArgs e)
{
    //remake the graph with the new axis and refresh it.
    makeGraph();
}

private void comboBoxGraphType_SelectedIndexChanged(object sender, EventArgs e)
{
    //call the update graph options method to change the available options
    updateGraphOptions();

    //call the makegraph() method to remake the graph with the new type
makeGraph();
}

private void setClassifierStats()
{
    double maxMass = 0.0, maxRadius = 0.0, minMass = 1000.0, minRadius = 1000.0, maxSpeed = 0.0;
    double cumMass = 0.0, cumRadius = 0.0, cumSpeed = 0.0;

    foreach (Particle p in particles)
    {
        p.setTimeStep(timeStep);
        if (p.mass < minMass)
            minMass = p.mass;
        if (p.mass > maxMass)
            maxMass = p.mass;
        if (p.pRadius < minRadius)
            minRadius = p.pRadius;
        if (p.pRadius > maxRadius)
            maxRadius = p.pRadius;
        if (p.speed > maxSpeed)
            maxSpeed = p.speed;
        cumMass += p.mass;
        cumRadius += p.pRadius;
        cumSpeed += p.speed;
    }
    classifier.count = particles.Count();
    classifier.massH = maxMass;
    classifier.massL = minMass;
    classifier.massA = cumMass / classifier.count;
    classifier.radiusH = maxRadius;
    classifier.radiusL = minRadius;
    classifier.radiusA = cumRadius / classifier.count;
    classifier.speedH = maxSpeed;
    classifier.speedA = cumSpeed / classifier.count;

    // let's output the classifier stats now
    labelClassifierStats.Text = "Particle Radius (µm):
        Max:  " + alignText(classifier.radiusH) + "
        Min:  " + alignText(classifier.radiusL) + "
        Mean: " + alignText(classifier.radiusA) + " \n\n        Speed (m/s):
        Max:  " + alignText(classifier.speedH) + "
        Mean: " + alignText(classifier.speedA) + " \n\n        Count: " + classifier.count;
}

// when import data is pressed in the menu, open a file dialogue to
get the file to import, then import it.
private void importDataToolStripMenuItem_Click(object sender, EventArgs e)
{
    setTime(false);
}
//create an 'OpenFileDialog' to let the user select the particle data file
OpenFileDialog openDialog = new OpenFileDialog();
openDialog.Title = "Open Text File";
openDialog.Filter = "VARS files|*.vars";

//remove all particles in the current list
particles.Clear();

//if the user selects a file (and doesn't press cancel)
if (openDialog.ShowDialog() == DialogResult.OK)
{
    //set readingFile to true
    readingFile = true;
    
    //set the cursor to the wait cursor while the file is read
    Cursor.Current = Cursors.WaitCursor;
    
    //we're going to keep the progress bar updated. Its max length is the number of bytes in the file we're reading, and we'll update it as the file is read.
    FileInfo fi = new FileInfo(openDialog.FileName);
    progressBar.Maximum = (int)fi.Length/1024 + 1024;
    //converted to KB, plus an extra one so we don't exceed the max
    progressBar.Visible = true; //show the progress bar during loading
    progressStatus.Text = "Reading File"; //set the progress status
    Application.DoEvents();
    
    //set the title of the application window to include the uploaded file name
    string filename = openDialog.SafeFileName;
    Text = "ClassiParse " + filename;
    fileNames.Clear();
    fileNames.Add(filename);
    
    //and this double will keep track of how many kilobytes have been read so far
    double kilobytes = 0.0;
    
    //inside a try to catch any errors with reading the file or with the format of the file
    try
    {
        //we're going to do this with a buffered stream to handle really big files.
        //That way the file doesn't have to be loaded all at once.
        
    
}
using (BufferedStream bs = new BufferedStream(fs))
using (StreamReader sr = new StreamReader(bs))
{
    //loop through each line of the file
    string line;
    while ((line = sr.ReadLine()) != null)
    {
        //add to the total number of read kilobytes
        kilobytes += (double)line.Length / 1024;
        //when we've reached 20 KB, trigger the
        progressBar and start the count over
        if (kilobytes >= 20)
        {
            kilobytes -= 20;
            try
            {
                progressBar.Value += 20;
            }
            catch
            {
                progressBar.Value =
                progressBar.Maximum - 1;
            }
            Application.DoEvents();
        }
        //don't read the lines that start with '#' -
        these are the comments at the top
        if (!(line[0] == '#'))
        {
            //split the string on whitespace and
            string[] l = line.Trim().Split(new[] { " " }, StringSplitOptions.RemoveEmptyEntries);
            //if it didn't split right, it must be
            seperated by tabs instead of spaces. Split it again.
            if (l.Length == 1)
                l = line.Trim().Split('\t');
            //if the line successfully split, parse
            the data from it.
            if (l.Length >= 1)
            {
                //define the x, y, z, radius, and
                wallAtY values for the particle on this line
                double pX = double.Parse(l[l.Length - 3]) - classifier.offsetX;
                double pY = double.Parse(l[l.Length - 2]);
                double pZ = double.Parse(l[l.Length - 1]) - classifier.offsetZ;
            }
        }
    }
}
double pR = Math.Sqrt(pX * pX + pZ * pZ);
double way = classifier.wallAtY(pY);

//define the speed of the particle
double pS = double.Parse(l[7]);

//if the particle is within the bounds
defined by the WallBoundary 'classifier', add it to the list
if (pY <= classifier.topY && pY >= classifier.bottomY && (way - pR) > 0)
{
    Particle p = new Particle(l[2],
l[3], l[4], pX, pY, pZ, way, pR, pS);
    particles.Add(p);
}
}

//set the cursor back to normal
Cursor.Current = Cursors.Default;
//and hide the progress bar again, and reset the
progress status
progressBar.Visible = false;
progressBar.Value = 0;
progressStatus.Text = "";

//set the classifier stats
setClassifierStats();

//we're done with readingFile, set it to false
readingFile = false;

//show the graph and refresh it to ensure that it
focuses on the data
makeGraph();
zedGraph.Refresh();

} catch (Exception ex)
{
    //if there is an error in reading the file, display an
error message and put the cursor back to normal.
    Cursor.Current = Cursors.Default;
    progressBar.Visible = false; //also hide the progress
    bar
    progressBar.Value = 0; //reset the value and the
    progress status
    progressStatus.Text = "";
    MessageBox.Show("Error: Could not read file from disk.
    Original error: " + ex.Message);
}
}
private string alignText(double d)
{
    d = Math.Round(d, 6);
    string[] ds;
    if (d.ToString().Contains('.'))
        ds = d.ToString().Split('.');
    else
    if (ds[1].Length > 6) ds[1] = ds[1].Substring(0, 6);
    return string.Format("{0,4}.{1,-6}", ds[0], ds[1]);
}

private void exportChartDataToolStripMenuItem_Click(object sender, EventArgs e)
{
    PointPairList woodPoints = (PointPairList) curve1.Points;
    PointPairList coalPoints = (PointPairList) curve2.Points;
    //create a 'SaveFileDialog' so the user can select where to
    save the .csv
    SaveFileDialog saveFile = new SaveFileDialog();
    //set the base filename
    saveFile.FileName = "Graph Data";
    //sets the directory back to what it was before closing
    saveFile.RestoreDirectory = true;
    //by default this dialog will only see .csv files, but there
    is the option to show all files
    saveFile.Filter = "|*.csv|All files (*)|.*|";
    saveFile.DefaultExt = ".csv";
    saveFile.FilterIndex = 1;
    //show the saveFile dialog
    saveFile.ShowDialog();
    //using a streamwriter, write the contents of
    'selectedParticles' to a .csv in the specified location
    using (StreamWriter sw = new StreamWriter(saveFile.FileName))
    {
        //this is the 'title line' for the columns in the .csv
        sw.WriteLine("Wood Data, , ,Coal Data");
        //for each particle, write its properties. The columns are
        seperated by commas, and the \n ends the line.
        for (int i = 0; i < xAxisLabels.Length; i++)
        {
            if (i < woodPoints.Count)
193

```csharp
{    sw.Write(xAxisLabels[i] + "","");
    sw.Write(woodPoints[i].Y + ",");
} else
{
    sw.Write(" , ");
} sw.Write(" , ");
if (i < coalPoints.Count)
{
    sw.Write(xAxisLabels[i] + "","");
    sw.Write(coalPoints[i].Y + ",");
} else
{
    sw.Write(" , ");
} sw.Write("\n");
}

// Show a message box indicating that the data was saved. Ask the user if they wish to open it. If they select 'Yes,' open the file.
if (MessageBox.Show("Data saved to " + saveFile.FileName + ". Open file now?", "Data Exported!", MessageBoxButtons.YesNo) == DialogResult.Yes)
    System.Diagnostics.Process.Start(saveFile.FileName);
}

// when export data is pressed in the menu, open a file dialogue to prompt where to save the .csv to, then save it.
private void exportDataToCSVToolStripMenuItem_Click(object sender, EventArgs e)
{
    // if a file is being read, cancel this action.
    if (readingFile) return;

    // get a complete list of particles by running 'filterparticles' again
    filterParticles();

    // create a 'SaveFileDialog' so the user can select where to save the .csv
    SaveFileDialog saveFile = new SaveFileDialog();

    // set the default filename
    saveFile.FileName = "Particle Data";

    // sets the directory back to what it was before closing
    saveFile.RestoreDirectory = true;

    // by default this dialog will only see .csv files, but there is the option to show all files
```
saveFile.Filter = "|*.csv|All files (*.*)|*.*";
saveFile.DefaultExt = "*.csv";
saveFile.FilterIndex = 1;

//show the saveFile dialog
saveFile.ShowDialog();

//using a streamwriter, write the contents of
'selectedParticles' to a .csv in the specified location
using (StreamWriter sw = new StreamWriter(saveFile.FileName))
{
    //this is the 'title line' for the columns in the .csv
    sw.WriteLine("PID, Type, Particle Radius, Mass, x, y, z,
    Speed, Distance from Wall");

    //for each particle, write its properties. The columns are
    seperated by commas, and the \n ends the line.
    foreach (Particle p in selectedParticles)
    {
        sw.Write(p.properties()["PID"] + ",");
        sw.Write(p.type + ",");
        sw.Write(p.pRadius + ",");
        sw.Write(p.mass + ",");
        sw.Write(p.x + ",");
        sw.Write(p.y + ",");
        sw.Write(p.z + ",");
        sw.Write(p.speed + ",");
        sw.Write(p.dWall + \\
    }
}

    //Show a message box indicating that the data was saved. Ask
    the user if they wish to open it. If they select 'Yes,' open the file.
    if (MessageBox.Show("Data saved to " + saveFile.FileName + ".
    Open file now?", "Data Exported!", MessageBoxButtons.YesNo) ==
    DialogResult.Yes)
    System.Diagnostics.Process.Start(saveFile.FileName);
}

    //when exit is pressed in the menu
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Close();
    Environment.Exit(Environment.ExitCode);
}

    //used for the wood/coal composition selection parameters that can
be checked on/off. Changes the check, then updates the graph.
private void compositionParameterFlipCheck(object sender,
EventArgs e)
{
    //flip the checked state of the sender
((ToolStripMenuItem)sender).Checked = !((ToolStripMenuItem)sender).Checked;
filterAndGraph(sender, e);

//method for all of the y bound parameters menu items. Makes sure that the clicked one is checked, then updates the graph.
private void yBoundsCheckChanged(object s, EventArgs e)
{
//uncheck all of the y bounds parameters
ToolStripMenuItem sender = (ToolStripMenuItem)s;
foreach (ToolStripMenuItem tsmi in yBoundsParameterToolStripMenuItem.DropDownItems)
{
    tsmi.Checked = false;
    foreach (ToolStripMenuItem tsmi1 in tsmi.DropDownItems)
    tsmi1.Checked = false;
}
//check the sender
sender.Checked = true;
//if sender is a subitem of the 2 or 3 sections groups, check the parent as well.
    if (sender == top3ToolStripMenuItem || sender == bottom3ToolStripMenuItem)
        threeSectionsToolStripMenuItem.Checked = true;
    if (sender == top2ToolStripMenuItem || sender == middle2ToolStripMenuItem || sender == bottom2ToolStripMenuItem)
        twoSectionsToolStripMenuItem.Checked = true;
    filterAndGraph(sender, e);
}
//handle the units for the pRadius parameter - only one can be checked.
private void ToolStripMenuItemPRadiusUM_Click(object sender, EventArgs e)
{
    ToolStripMenuPRadiusLowerUM.Checked = true;
    ToolStripMenuPRadiusUpperUM.Checked = true;
    ToolStripMenuPRadiusLowerPercent.Checked = false;
    ToolStripMenuPRadiusUpperPercent.Checked = false;
    filterAndGraph(sender, e);
}
//handle the units for the pRadius parameter - only one can be checked.
private void ToolStripMenuPRadiusPercent_Click(object sender, EventArgs e)
{
    ToolStripMenuPRadiusLowerUM.Checked = false;
ToolStripMenuItemPRadiusUpperUM.Checked = false;
ToolStripMenuItemPRadiusLowerPercent.Checked = true;
ToolStripMenuItemPRadiusUpperPercent.Checked = true;

filterAndGraph(sender, e);
}

// when the 'reset to default' option is selected in the parameters menu
private void resetToDefaultToolStripMenuItem_Click(object sender, EventArgs e)
{
    // set everything to its defaults
    woodParameterToolStripMenuItem.Checked = true;
    coalParameterToolStripMenuItem.Checked = true;
    toolStripComboBoxDWallParameter.SelectedIndex =
        toolStripComboBoxDWallParameter.FindElement("30 mm");
    toolStripComboBoxSpeedParameter.SelectedIndex =
        toolStripComboBoxSpeedParameter.FindElement("10 %-mean");
    toolStripTextBoxMassLower.Text = "--";
    toolStripTextBoxMassUpper.Text = "--";
    toolStripTextBoxPRadiusLower.Text = "--";
    toolStripTextBoxPRadiusUpper.Text = "--";

    ToolStripMenuPRadiusLowerUM.Checked = true;
    ToolStripMenuPRadiusUpperUM.Checked = true;
    ToolStripMenuPRadiusLowerPercent.Checked = false;
    ToolStripMenuPRadiusUpperPercent.Checked = false;

    ToolStripMenuMassLowerG.Checked = true;
    ToolStripMenuMassLowerKG.Checked = false;
    ToolStripMenuMassLowerMG.Checked = false;
    ToolStripMenuMassLowerPercent.Checked = false;
    ToolStripMenuMassUpperG.Checked = true;
    ToolStripMenuMassUpperKG.Checked = false;
    ToolStripMenuMassUpperMG.Checked = false;
    ToolStripMenuMassUpperPercent.Checked = false;

    foreach (ToolStripMenuItem tsmi in
        yBoundsParameterToolStripMenuItem.DropDownItems)
    {
        tsmi.Checked = false;
        foreach (ToolStripMenuItem tsmi1 in tsmi.DropDownItems)
            tsmi1.Checked = false;
    }  
    fullClassifierToolStripMenuItem.Checked = true;

    // if this is a time plot, the default is to show everything
    if (time)
    {
        setFiltersToMax();
    }

    filterAndGraph(sender, e);
}
private void comboBoxDivisions_SelectedIndexChanged(object sender, EventArgs e)
{
    // remake the graph with the new divisions and refresh it.
    makeGraph();
}

// When the close menu button is pressed, exit
private void ClassiParseForm_FormClosed(object sender, FormClosedEventArgs e)
{
    Environment.Exit(Environment.ExitCode);
}

// When the 3D explorer button is pressed, launch a 3D space
plotter form
private void ThreeDExplorerToolStripMenuItem_Click(object sender, EventArgs e)
{
    _3DSpacePlotter threeDForm = new _3DSpacePlotter(selectedParticles, time, fileNames[timeStep]);
    threeDForm.Show();
}

public void setTime(bool t)
{
    time = t;
    if (time)
    {
        // set totAge to the number of time intervals
        totAge = particles[0].age();

        Text = "ClassiParse at time " + (totAge - timeStep);

        // Show the time stats group box
        groupBoxTimeStats.Visible = true;
    }
    else
    {
        // set totAge to show that there is no time data
        totAge = 1;

        // Hide the time stats group box
        groupBoxTimeStats.Visible = false;
    }

    // set timeStep to 0
    timeStep = 0;

    // show or hide the timestep buttons, depending on time
    buttonBackTimeStep.Visible = time;
buttonForwardTimeStep.Visible = time;

if (timeStep <= 1)
{
    buttonBackTimeStep.Enabled = false;
}
else
{
    buttonBackTimeStep.Enabled = true;
}

if (timeStep >= totAge - 1)
{
    buttonForwardTimeStep.Enabled = false;
}
else
{
    buttonForwardTimeStep.Enabled = true;
}
}

// when the 'reverse engineering' option is selected in the menu.
private void reverseEngineeringToolStripMenuItem_Click(object sender, EventArgs e)
{
    readingFile = true;

    // Launch the Multiple File Picker.
    ReverseEngineeringFilePicker multiFileForm = new ReverseEngineeringFilePicker(classifier);
    multiFileForm.ShowDialog();

    // get the particles from the Multiple File Picker
    particles = multiFileForm.particles;

    if (particles.Count() > 0)
    {
        // get the file names as well
        fileNames = multiFileForm.fileNames;

        // set graph options to show all particles
        setFiltersToMax();

        setTime(true);

        readingFile = false;

        setClassifierStats();

        // show graph
        filterAndGraph(null, null);
    }
}
// When the point size is changed
private void comboBoxPointSize_SelectedIndexChanged(object sender, EventArgs e)
{
    // remake the point symbol based on the selected size (see function), then remake the graph.
    makePointSymbol();
    makeGraph();
}

private void setToShowAllToolStripMenuItem_Click(object sender, EventArgs e)
{
    setFiltersToMax();
}

private void buttonBackTimeStep_Click(object sender, EventArgs e)
{
    timeStep--;
    if (timeStep <= 0)
    {
        timeStep = 0;
        buttonBackTimeStep.Enabled = false;
    }
    else
    {
        buttonBackTimeStep.Enabled = true;
    }
    if (timeStep < totAge - 1)
    {
        buttonForwardTimeStep.Enabled = true;
    }
    else
    {
        buttonForwardTimeStep.Enabled = false;
    }
    Text = "ClassiParse at time " + (totAge - timeStep);
    updateParticleTime();
    filterAndGraph(sender, e);
}

private void buttonForwardTimeStep_Click(object sender, EventArgs e)
{
    timeStep++;
    if (timeStep <= 0)
    {
        timeStep = 0;
        buttonBackTimeStep.Enabled = false;
else
{
    buttonBackTimeStep.Enabled = true;
}
if (timeStep >= totAge - 1)
{
    timeStep = totAge - 1;
    buttonForwardTimeStep.Enabled = false;
} else
{
    buttonForwardTimeStep.Enabled = true;
}

Text = "ClassiParse at time " + (totAge - timeStep);
updateParticleTime();
filterAndGraph(sender, e);

private void updateParticleTime()
{
    foreach (Particle p in particles)
    {
        p.setTimeStep(timeStep);
    }
}

private void checkBoxShowClassifier_CheckedChanged(object sender, EventArgs e)
{
    // update the graph, adding or removing the classifier.
    makeGraph();
}

private void setFiltersToMax()
{
    // set everything to the max state
    woodParameterToolStripMenuItem.Checked = true;
coilParameterToolStripMenuItem.Checked = true;
toolStripComboBoxDWallParameter.SelectedIndex =
toolStripComboBoxDWallParameter.FindStringExact("100 %");
toolStripComboBoxSpeedParameter.SelectedIndex =
toolStripComboBoxSpeedParameter.FindStringExact("100 %-max");
toolStripTextBoxMassLower.Text = "---";
toolStripTextBoxMassUpper.Text = "---";
toolStripTextBoxPRadiusLower.Text = "---";
toolStripTextBoxPRadiusUpper.Text = "---";

    ToolStripMenuItemPRadiusLowerUM.Checked = true;
    ToolStripMenuItemPRadiusUpperUM.Checked = true;
ToolstripMenuItemPRadiusLowerPercent.Checked = false;
ToolstripMenuItemPRadiusUpperPercent.Checked = false;

ToolstripMenuItemMassLowerG.Checked = true;
ToolstripMenuItemMassLowerKG.Checked = false;
ToolstripMenuItemMassLowerMG.Checked = false;
ToolstripMenuItemMassLowerPercent.Checked = false;
ToolstripMenuItemMassUpperG.Checked = true;
ToolstripMenuItemMassUpperKG.Checked = false;
ToolstripMenuItemMassUpperMG.Checked = false;
ToolstripMenuItemMassUpperPercent.Checked = false;

doi (ToolstripMenuItem tsmi in yBoundsParameterToolstripMenuItem.DropDownItems)
{
    tsmi.Checked = false;
    doi (ToolstripMenuItem tsmi1 in tsmi.DropDownItems)
        tsmi1.Checked = false;
}
fullClassifierToolstripMenuItem.Item.Checked = true;

filterAndGraph(null, null);

private void toolstripTextBoxSwirlDepth_TextChanged(object sender, EventArgs e)
{
    classifier.topY = double.Parse(toolstripTextBoxSwirlDepth.Text);
calculateAndShowTimeStats();
}

// when parameters are changed, this is called.
private void filterAndGraph(object Sender, EventArgs e)
{
    // if a file is being read, cancel this action.
    if (readingFile) return;

    // apply the filter to the particles, remaking 'selectedParticles'
    filterParticles();

    // remake the graph
    makeGraph();

    // if this is a time plot, show the time stats.
    if (time)
    {
        calculateAndShowTimeStats();
    }
}

private void calculateAndShowTimeStats()
{  
    Console.WriteLine("Calculating and showing time stats.");
    
    // declare the variables  
    double averageWoodResTime = 0.0;  
    double averageCoalResTime = 0.0;  
    double averageWoodDeepResTime = 0.0;  
    double averageCoalDeepResTime = 0.0;  
    double totalCoalMass = 0.0;  
    double totalWoodMass = 0.0;  
    double woodSwirlMass = 0.0;  
    double coalSwirlMass = 0.0;  
    int woodSwirlCount = 0;  
    int coalSwirlCount = 0;  
    int totalCoalCount = 0;  
    
    // Loop through all of the selected particles  
    foreach (Particle p in selectedParticles)  
    {  
        // recalculate the boundary properties based on the  
        // classifier's parameters  
        p.calculateTimeProperties(classifier);  
        
        // determine its type  
        if (p.type == COAL_TYPE)  
        {  
            averageCoalResTime += p.getResidenceTime();  
            if (p.GetMaxPen() > 0.0)  
            {  
                coalSwirlCount++;  
                averageCoalDeepResTime +=  
                p.getDeepResidenceTime();  
                coalSwirlMass += p.mass;  
            }  
            totalCoalCount++;  
            totalCoalMass += p.mass;  
        }  
        else  
        {  
            averageWoodResTime += p.getResidenceTime();  
            if (p.GetMaxPen() > 0.0)  
            {  
                woodSwirlCount++;  
                averageWoodDeepResTime +=  
                p.getDeepResidenceTime();  
                woodSwirlMass += p.mass;  
            }  
            totalWoodMass += p.mass;  
        }  
    }  
    
    // finish finding the average res time  
    averageCoalResTime /= totalCoalCount;
averageWoodResTime /= (selectedParticles.Count -
totalCoalCount);
averageCoalDeepResTime /= coalSwirlCount;
averageWoodDeepResTime /= woodSwirlCount;

// Convert the masses from kg to g
totalCoalMass *= 1000;
totalWoodMass *= 1000;
coalSwirlMass *= 1000;
woodSwirlMass *= 1000;

// show the label
Math.Round(averageCoalResTime, 2) + "\n  Wood: " +
Math.Round(averageWoodResTime, 2) +
"\nNumber (Mass) of Particles\n  Coal: " + totalCoalCount + " (" +
Math.Round(totalCoalMass,2) + " g)" +
"\n  Wood: " + (selectedParticles.Count - totalCoalCount) + " (" +
Math.Round(totalWoodMass,2) + " g)" +
"\nNumber (Mass) that Swirl\n  Coal: " + coalSwirlCount + " (" +
Math.Round(coalSwirlMass,2) + " g)" +
"\n  Wood: " + woodSwirlCount + " (" +
Math.Round(woodSwirlMass,2) + " g)"");

partial class ClassiParseForm
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// inside the method.
    /// </summary>
    /// <param name="controls">Components that make up this form</param>
    private void InitializeComponent(System.Drawing.Container controls)
    {
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    this.zedGraph = new ZedGraph.ZedGraphControl();
    this.checkboxConnectPoints = new System.Windows.Forms.CheckBox();
    this.checkBoxShowWood = new System.Windows.Forms.CheckBox();
    this.checkBoxShowCoal = new System.Windows.Forms.CheckBox();
    this.comboBoxXAxis = new System.Windows.Forms.ComboBox();
    this.comboBoxYAxis = new System.Windows.Forms.ComboBox();
    this.comboBoxPointSize = new System.Windows.Forms.ComboBox();
    this.labelPointSize = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.groupBoxGraphOptions = new System.Windows.Forms.GroupBox();
    this.checkBoxShowClassifier = new System.Windows.Forms.CheckBox();
    this.comboBoxDivisions = new System.Windows.Forms.ComboBox();
    this.comboBoxGraphType = new System.Windows.Forms.ComboBox();
    this.labelDivisions = new System.Windows.Forms.Label();
    this.menuStrip1 = new System.Windows.Forms.MenuStrip();
    this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.importDataToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.exportChartDataToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.exportAllParticlesToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.toolStripSeparator3 = new System.Windows.Forms.ToolStripSeparator();
    this.reverseEngineeringToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.toolStripSeparator2 = new System.Windows.Forms.ToolStripSeparator();
    this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.parametersToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.dWallParameterToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.toolStripComboBoxDWallParameter = new System.Windows.Forms.ToolStripComboBox();
    this.speedToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.toolStripComboBoxSpeedParameter = new System.Windows.Forms.ToolStripComboBox();
    this.yBoundsParameterToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.toolStripComboBoxYBoundsParameter = new System.Windows.Forms.ToolStripComboBox();
}
this.fullClassifierToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.section1ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.section2ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.section3ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.section4ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.twoSectionsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.top2ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.middle2ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.bottom2ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.threeSectionsToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.top3ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.bottom3ToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.particleRadiusParameterToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemPRadiusLower = new System.Windows.Forms.ToolStripMenuItem();
this.toolStripTextBoxPRadiusLower = new System.Windows.Forms.ToolStripTextBox();
this.ToolStripMenuItemPRadiusLowerUM = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemPRadiusLowerPercent = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemPRadiusUpper = new System.Windows.Forms.ToolStripMenuItem();
this.toolStripTextBoxPRadiusUpper = new System.Windows.Forms.ToolStripTextBox();
this.ToolStripMenuItemPRadiusUpperUM = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemPRadiusUpperPercent = new System.Windows.Forms.ToolStripMenuItem();
this.massParameterToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassLower = new System.Windows.Forms.ToolStripMenuItem();
this.toolStripTextBoxMassLower = new System.Windows.Forms.ToolStripTextBox();
this.ToolStripMenuItemMassLowerKG = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassLowerMG = new System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassLowerPercent = new 
System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassUpper = new 
System.Windows.Forms.ToolStripMenuItem();
this.toolStripTextBoxMassUpper = new 
System.Windows.Forms.ToolStripTextBox();
this.ToolStripMenuItemMassUpperKG = new 
System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassUpperG = new 
System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassUpperMG = new 
System.Windows.Forms.ToolStripMenuItem();
this.ToolStripMenuItemMassUpperPercent = new 
System.Windows.Forms.ToolStripMenuItem();
this.adjustSwirlParameterToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.toolStripTextBoxSwirlDepth = new 
System.Windows.Forms.ToolStripTextBox();
this.mToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.woodParameterToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.coalParameterToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.toolStripSeparator1 = new 
System.Windows.Forms.ToolStripSeparator();
this.resetToDefaultToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.setToShowAllToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.toolStripSeparator4 = new 
System.Windows.Forms.ToolStripSeparator();
this.ThreeDExplorerToolStripMenuItem = new 
System.Windows.Forms.ToolStripMenuItem();
this.progressBar = new System.Windows.Forms.ProgressBar();
this.progressStatus = new System.Windows.Forms.Label();
this.groupBoxDataBox = new System.Windows.Forms.GroupBox();
this.labelClassifierStats = new System.Windows.Forms.Label();
this.buttonBackTimeStep = new System.Windows.Forms.Button();
this.buttonForwardTimeStep = new 
this.groupBoxTimeStats.SuspendLayout();
this.menuStrip1.SuspendLayout();
this.groupBoxDataBox.SuspendLayout();
this.groupBoxTimeStats.SuspendLayout();
this.SuspendLayout();

//
// zedGraph
//
this.zedGraph.Anchor =
this.zedGraph.AutoSize = true;
this.zedGraph.EditButtons =
System.Windows.FormsMouseButton.None;
this.zedGraph.EditModifierKeys =
this.zedGraph.Location = new System.Drawing.Point(367, 27);
this.zedGraph.Name = "zedGraph";
this.zedGraph.ScrollGrace = 0D;
this.zedGraph.ScrollMaxX = 0D;
this.zedGraph.ScrollMaxY = 0D;
this.zedGraph.ScrollMaxY2 = 0D;
this.zedGraph.ScrollMinX = 0D;
this.zedGraph.ScrollMinY = 0D;
this.zedGraph.ScrollMinY2 = 0D;
this.zedGraph.Size = new System.Drawing.Size(771, 563);
this.zedGraph.TabIndex = 1;
//
// checkboxConnectPoints
//
this.checkboxConnectPoints.AutoSize = true;
this.checkboxConnectPoints.Location = new System.Drawing.Point(8, 55);
this.checkboxConnectPoints.Name = "checkboxConnectPoints";
this.checkboxConnectPoints.Size = new System.Drawing.Size(98, 17);
this.checkboxConnectPoints.TabIndex = 4;
this.checkboxConnectPoints.Text = "Connect Points";
this.checkboxConnectPoints.UseVisualStyleBackColor = true;
this.checkboxConnectPoints.CheckedChanged += new System.EventHandler(this.checkboxConnectPoints_CheckedChanged);
//
// checkBoxShowWood
//
this.checkBoxShowWood.AutoSize = true;
this.checkBoxShowWood.Checked = true;
this.checkBoxShowWood.CheckState =
this.checkBoxShowWood.Location = new System.Drawing.Point(8, 78);
this.checkBoxShowWood.Name = "checkBoxShowWood";
this.checkBoxShowWood.Size = new System.Drawing.Size(114, 17);
this.checkBoxShowWood.TabIndex = 5;
this.checkBoxShowWood.Text = "Show Wood (Red)"
this.checkBoxShowWood.UseVisualStyleBackColor = true;
this.checkBoxShowWood.CheckedChanged += new System.EventHandler(this.checkBoxShowWood_CheckedChanged);
//
// checkBoxShowCoal
//
this.checkBoxShowCoal.AutoSize = true;
this.checkBoxShowCoal.Checked = true;
this.checkBoxShowCoal.CheckState = System.Windows.Forms.CheckState.Checked;
this.checkBoxShowCoal.Location = new System.Drawing.Point(8, 101);
this.checkBoxShowCoal.Name = "checkBoxShowCoal";
this.checkBoxShowCoal.Size = new System.Drawing.Size(107, 17);
this.checkBoxShowCoal.TabIndex = 6;
this.checkBoxShowCoal.Text = "Show Coal (Blue)";
this.checkBoxShowCoal.UseVisualStyleBackColor = true;
this.checkBoxShowCoal.CheckedChanged += new System.EventHandler(this.checkBoxShowCoal_CheckedChanged);

// comboBoxXAxis
//
this.comboBoxXAxis.FormattingEnabled = true;
this.comboBoxXAxis.Items.AddRange(new object[] {
    "mass",
    "speed",
    "Particle Radius",
    "x",
    "y",
    "z",
    "radius",
    "dWall",
    "PID",
    "type"});
this.comboBoxXAxis.Location = new System.Drawing.Point(693, 596);
this.comboBoxXAxis.Name = "comboBoxXAxis";
this.comboBoxXAxis.Size = new System.Drawing.Size(112, 21);
this.comboBoxXAxis.TabIndex = 7;
this.comboBoxXAxis.Text = "Particle Radius";
this.comboBoxXAxis.SelectedIndexChanged += new System.EventHandler(this.comboBoxXAxis_SelectedIndexChanged);

// comboBoxYAxis
//
this.comboBoxYAxis.Anchor = System.Windows.Forms.AnchorStyles.Left;
this.comboBoxYAxis.FormattingEnabled = true;
this.comboBoxYAxis.Items.AddRange(new object[] {
    "% mass",
    "% count",
    "count",
    "mass"});
this.comboBoxYAxis.Location = new System.Drawing.Point(264, 288);
this.comboBoxYAxis.Name = "comboBoxYAxis";
this.comboBoxYAxis.Size = new System.Drawing.Size(97, 21);
this.comboBoxYAxis.TabIndex = 8;
this.comboBoxYAxis.Text = "% mass";
thiscomboBoxYAxis.SelectedIndexChanged += new System.EventHandler(this.comboBoxYAxis_SelectedIndexChanged);

// comboBoxPointSize
//
this.comboBoxPointSize.FormattingEnabled = true;
this.comboBoxPointSize.Items.AddRange(new object[] {
    "1",
    "2",
    "5",
    "9"});
this.comboBoxPointSize.Location = new System.Drawing.Point(68, 145);
this.comboBoxPointSize.Name = "comboBoxPointSize";
this.comboBoxPointSize.Size = new System.Drawing.Size(39, 21);
this.comboBoxPointSize.TabIndex = 9;
this.comboBoxPointSize.Text = "5";
this.comboBoxPointSize.SelectedIndexChanged += new System.EventHandler(this.comboBoxPointSize_SelectedIndexChanged);

// labelPointSize
//
this.labelPointSize.AutoSize = true;
this.labelPointSize.Location = new System.Drawing.Point(7, 149);
this.labelPointSize.Name = "labelPointSize";
this.labelPointSize.Size = new System.Drawing.Size(57, 13);
this.labelPointSize.TabIndex = 10;
this.labelPointSize.Text = "Point Size:";

// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(4, 13);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(107, 16);
this.label2.TabIndex = 11;
this.label2.Text = "Graph Options";

this.groupBoxGraphOptions.Controls.Add(this.checkBoxShowClassifier);
this.groupBoxGraphOptions.Controls.Add(this.comboBoxDivisions);
this.groupBoxGraphOptions.Controls.Add(this.comboBoxGraphType);
this.groupBoxGraphOptions.Controls.Add(this.labelDivisions);
this.groupBoxGraphOptions.Controls.Add(this.label2);
this.groupBoxGraphOptions.Controls.Add(this.checkboxConnectPoints);

this.groupBoxGraphOptions.Controls.Add(this.checkBoxShowClassifier);
this.groupBoxGraphOptions.Controls.Add(this.comboBoxDivisions);
this.groupBoxGraphOptions.Controls.Add(this.comboBoxGraphType);
this.groupBoxGraphOptions.Controls.Add(this.labelDivisions);
this.groupBoxGraphOptions.Controls.Add(this.label2);
this.groupBoxGraphOptions.Controls.Add(this.checkboxConnectPoints);
this.groupBoxGraphOptions.Controls.Add(this.labelPointSize);
this.groupBoxGraphOptions.Controls.Add(this.comboBoxPointSize);
this.groupBoxGraphOptions.Controls.Add(this.checkBoxShowWood);
this.groupBoxGraphOptions.Controls.Add(this.checkBoxShowCoal);
this.groupBoxGraphOptions.ForeColor = System.Drawing.SystemColors.ControlText;
this.groupBoxGraphOptions.Location = new System.Drawing.Point(12, 27);
this.groupBoxGraphOptions.Name = "groupBoxGraphOptions";
this.groupBoxGraphOptions.Size = new System.Drawing.Size(220, 176);
this.groupBoxGraphOptions.TabIndex = 17;
this.groupBoxGraphOptions.TabStop = false;
/
// checkBoxShowClassifier
//
this.checkBoxShowClassifier.AutoSize = true;
this.checkBoxShowClassifier.Checked = true;
this.checkBoxShowClassifier.CheckState = System.Windows.Forms.CheckState.Checked;
this.checkBoxShowClassifier.Location = new System.Drawing.Point(7, 124);
this.checkBoxShowClassifier.Name = "checkBoxShowClassifier";
this.checkBoxShowClassifier.Size = new System.Drawing.Size(133, 17);
this.checkBoxShowClassifier.TabIndex = 20;
this.checkBoxShowClassifier.Text = "Show Classifier Outline";
this.checkBoxShowClassifier.TextAlign = System.Drawing.ContentAlignment.TopRight;
this.checkBoxShowClassifier.UseVisualStyleBackColor = true;
this.checkBoxShowClassifier.CheckedChanged += new System.EventHandler(this.checkBoxShowClassifier_CheckedChanged);
/
// comboBoxDivisions
/
this.comboBoxDivisions.FormattingEnabled = true;
this.comboBoxDivisions.Items.AddRange(new object[] { "2", "3", "4", "5", "10", "15", "20", "25", "30", "35", "40", "45", "50", "60", "70", "80",...
"90",
"100",
"125",
"150",
"175",
"200",
"300",
"400",
"500",
"750",
"1000"});
this.comboBoxDivisions.Location = new System.Drawing.Point(60, 55);
this.comboBoxDivisions.Name = "comboBoxDivisions";
this.comboBoxDivisions.Size = new System.Drawing.Size(46, 21);
this.comboBoxDivisions.TabIndex = 19;
this.comboBoxDivisions.Text = "100";
this.comboBoxDivisions.SelectedIndexChanged += new System.EventHandler(this.comboBoxDivisions_SelectedIndexChanged);
//
// comboBoxGraphType
//
this.comboBoxGraphType.FormattingEnabled = true;
this.comboBoxGraphType.Items.AddRange(new object[] {
    "Scatter Plot",
    "Histogram"});
this.comboBoxGraphType.Location = new System.Drawing.Point(8, 32);
this.comboBoxGraphType.Name = "comboBoxGraphType";
this.comboBoxGraphType.Size = new System.Drawing.Size(121, 21);
this.comboBoxGraphType.TabIndex = 12;
this.comboBoxGraphType.Text = "Scatter Plot";
this.comboBoxGraphType.SelectedIndexChanged += new System.EventHandler(this.comboBoxGraphType_SelectedIndexChanged);
//
// labelDivisions
//
this.labelDivisions.AutoSize = true;
this.labelDivisions.Location = new System.Drawing.Point(6, 59);
this.labelDivisions.Name = "labelDivisions";
this.labelDivisions.Size = new System.Drawing.Size(52, 13);
this.labelDivisions.TabIndex = 13;
this.labelDivisions.Text = "Divisions:";
//
// menuStrip1
//
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
    this.fileToolStripMenuItem,
    this.parametersToolStripMenuItem});
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Size = new System.Drawing.Size(1150, 24);
this.menuStrip1.TabIndex = 19;
this.menuStrip1.Text = "menuStrip1";

//
// fileToolStripMenuItem
//
this.fileToolStripMenuItem.DropDownItems.AddRange(new 
System.Windows.Forms.ToolStripItem[] {
    this.importDataToolStripMenuItem,
    this.exportChartDataToolStripMenuItem,
    this.exportAllParticlesToolStripMenuItem,
    this.toolStripSeparator3,
    this.reverseEngineeringToolStripMenuItem,
    this.toolStripSeparator2,
    this.exitToolStripMenuItem});
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
this.fileToolStripMenuItem.Text = "File";

//
// importDataToolStripMenuItem
//
this.importDataToolStripMenuItem.Name = "importDataToolStripMenuItem";
this.importDataToolStripMenuItem.Size = new System.Drawing.Size(218, 22);
this.importDataToolStripMenuItem.Text = "Import Data";
this.importDataToolStripMenuItem.Click += new System.EventHandler(this.importDataToolStripMenuItem_Click);

//
// exportChartDataToolStripMenuItem
//
this.exportChartDataToolStripMenuItem.Name = "exportChartDataToolStripMenuItem";
this.exportChartDataToolStripMenuItem.Size = new System.Drawing.Size(218, 22);
this.exportChartDataToolStripMenuItem.Text = "Export Chart Data only";
this.exportChartDataToolStripMenuItem.Click += new System.EventHandler(this.exportDataToCSVToolStripMenuItem_Click);

//
// toolStripSeparator3
//
this.toolStripSeparator3.Name = "toolStripSeparator3";
this.toolStripSeparator3.Size = new System.Drawing.Size(215, 6);

// reverseEngineeringToolStripMenuItem
//
this.reverseEngineeringToolStripMenuItem.Name = "reverseEngineeringToolStripMenuItem";
this.reverseEngineeringToolStripMenuItem.Size = new System.Drawing.Size(218, 22);
this.reverseEngineeringToolStripMenuItem.Text = "Reverse Engineering";
this.reverseEngineeringToolStripMenuItem.Click += new System.EventHandler(this.reverseEngineeringToolStripMenuItem_Click);

// toolStripSeparator2
//
this.toolStripSeparator2.Name = "toolStripSeparator2";
this.toolStripSeparator2.Size = new System.Drawing.Size(215, 6);

// exitToolStripMenuItem
//
this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";
this.exitToolStripMenuItem.Size = new System.Drawing.Size(218, 22);
this.exitToolStripMenuItem.Text = "Exit";
this.exitToolStripMenuItem.Click += new System.EventHandler(this.exitToolStripMenuItem_Click);

// parametersToolStripMenuItem
//
this.parametersToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
    this.dWallParameterToolStripMenuItem,
    this.speedToolStripMenuItem,
    this.yBoundsParameterToolStripMenuItem,
    this.particleRadiusParameterToolStripMenuItem,
    this.massParameterToolStripMenuItem,
    this.adjustSwirlParameterToolStripMenuItem,
    this.woodParameterToolStripMenuItem,
    this.coalParameterToolStripMenuItem,
    this.toolStripSeparator1,
    this.resetToDefaultToolStripMenuItem,
    this.setToShowAllToolStripMenuItem,
    this.toolStripSeparator4,
    this.ThreeDExplorerToolStripMenuItem});

this.parametersToolStripMenuItem.Name = "parametersToolStripMenuItem";
this.parametersToolStripMenuItem.Size = new System.Drawing.Size(78, 20);
this.parametersToolStripMenuItem.Text = "Parameters";

//
// dWallParameterToolStripMenuItem

213
this.dWallParameterToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.toolStripComboBoxDWallParameter});
this.dWallParameterToolStripMenuItem.Name = "dWallParameterToolStripMenuItem";
this.dWallParameterToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
this.dWallParameterToolStripMenuItem.Text = "Wall Proximity";
//
// this.toolStripComboBoxDWallParameter
//
this.toolStripComboBoxDWallParameter.FlatStyle = System.Windows.Forms.FlatStyle.Standard;
this.toolStripComboBoxDWallParameter.Items.AddRange(new object[] {
  "5 mm",
  "10 mm",
  "30 mm",
  "50 mm",
  "100 mm",
  "1 %",
  "10 %",
  "100 %"});
this.toolStripComboBoxDWallParameter.Name = "toolStripComboBoxDWallParameter";
this.toolStripComboBoxDWallParameter.Size = new System.Drawing.Size(121, 23);
this.toolStripComboBoxDWallParameter.Text = "30 mm";
this.toolStripComboBoxDWallParameter.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.toolStripComboBoxDWallParameter.SelectedItemChanged += new System.EventHandler(this.filterAndGraph);
//
// speedToolStripMenuItem
//
this.speedToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.toolStripComboBoxSpeedParameter});
this.speedToolStripMenuItem.Name = "speedToolStripMenuItem";
this.speedToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
this.speedToolStripMenuItem.Text = "Speed Limit";
//
// this.toolStripComboBoxSpeedParameter
//
this.toolStripComboBoxSpeedParameter.FlatStyle = System.Windows.Forms.FlatStyle.Standard;
this.toolStripComboBoxSpeedParameter.Items.AddRange(new object[] {
  "0.001 m/s",
  "0.01 m/s",
  "0.1 m/s",
  "1 %-mean",
  "10 %-mean"});
"100 %-mean",
"1 %-max",
"10 %-max",
"100 %-max"));
this.toolStripComboBoxSpeedParameter.Name =
"toolStripComboBoxSpeedParameter";
this.toolStripComboBoxSpeedParameter.Size = new
System.Drawing.Size(121, 23);
this.toolStripComboBoxSpeedParameter.Text = "10 %-mean";
this.toolStripComboBoxSpeedParameter.SelectedIndexChanged +=
new System.EventHandler(this.filterAndGraph);
//
// yBoundsParameterToolStripMenuItem
//
this.yBoundsParameterToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
    this.fullClassifierToolStripMenuItem,
    this.section1ToolStripMenuItem,
    this.section2ToolStripMenuItem,
    this.section3ToolStripMenuItem,
    this.section4ToolStripMenuItem,
    this.twoSectionsToolStripMenuItem,
    this.threeSectionsToolStripMenuItem});
this.yBoundsParameterToolStripMenuItem.Name =
"yBoundsParameterToolStripMenuItem";
this.yBoundsParameterToolStripMenuItem.Size = new
System.Drawing.Size(157, 22);
this.yBoundsParameterToolStripMenuItem.Text = "Y Bounds";
//
// fullClassifierToolStripMenuItem
//
this.fullClassifierToolStripMenuItem.Checked = true;
this.fullClassifierToolStripMenuItem.CheckState =
this.fullClassifierToolStripMenuItem.Image =
(System.Drawing.Image)(resources.GetObject("fullClassifierToolStripMenuItem.Image"));
this.fullClassifierToolStripMenuItem.Name =
"fullClassifierToolStripMenuItem";
this.fullClassifierToolStripMenuItem.Size = new
System.Drawing.Size(143, 22);
this.fullClassifierToolStripMenuItem.Text = "Full Classifier";
this.fullClassifierToolStripMenuItem.Click += new
System.EventHandler(this.yBoundsCheckChanged);
//
// section1ToolStripMenuItem
//
this.section1ToolStripMenuItem.Image =
(System.Drawing.Image)(resources.GetObject("section1ToolStripMenuItem.Image"));
this.section1ToolStripMenuItem.Name =
"section1ToolStripMenuItem";
this.section1ToolStripMenuItem.Size = new
System.Drawing.Size(143, 22);
this.section1ToolStripMenuItem.Text = "Section 1";
this.section1ToolStripMenuItem.Click += new
System.EventHandler(this.yBoundsCheckChanged);
this.section1ToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.section1ToolStripMenuItem.Text = "Section 1";
this.section1ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// section2ToolStripMenuItem
//
this.section2ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("section2ToolStripMenuItem.Image")));
this.section2ToolStripMenuItem.Name = "section2ToolStripMenuItem";
this.section2ToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.section2ToolStripMenuItem.Text = "Section 2";
this.section2ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// section3ToolStripMenuItem
//
this.section3ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("section3ToolStripMenuItem.Image")));
this.section3ToolStripMenuItem.Name = "section3ToolStripMenuItem";
this.section3ToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.section3ToolStripMenuItem.Text = "Section 3";
this.section3ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// section4ToolStripMenuItem
//
this.section4ToolStripMenuItem.Name = "section4ToolStripMenuItem";
this.section4ToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.section4ToolStripMenuItem.Text = "Section 4";
this.section4ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// twoSectionsToolStripMenuItem
//
this.twoSectionsToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.top2ToolStripMenuItem, this.middle2ToolStripMenuItem, this.bottom2ToolStripMenuItem });
this.twoSectionsToolStripMenuItem.Name = "twoSectionsToolStripMenuItem";
this.twoSectionsToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.twoSectionsToolStripMenuItem.Text = "2 Sections ...";
//
// top2ToolStripMenuItem
//
this.top2ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("top2ToolStripMenuItem.Image")));
this.top2ToolStripMenuItem.Name = "top2ToolStripMenuItem";
this.top2ToolStripMenuItem.Size = new System.Drawing.Size(123, 22);
this.top2ToolStripMenuItem.Text = "Top 2";
this.top2ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);
//
// middle2ToolStripMenuItem
//
this.middle2ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("middle2ToolStripMenuItem.Image")));
this.middle2ToolStripMenuItem.Name = "middle2ToolStripMenuItem";
this.middle2ToolStripMenuItem.Size = new System.Drawing.Size(123, 22);
this.middle2ToolStripMenuItem.Text = "Middle 2";
this.middle2ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);
//
// bottom2ToolStripMenuItem
//
this.bottom2ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("bottom2ToolStripMenuItem.Image")));
this.bottom2ToolStripMenuItem.Name = "bottom2ToolStripMenuItem";
this.bottom2ToolStripMenuItem.Size = new System.Drawing.Size(123, 22);
this.bottom2ToolStripMenuItem.Text = "Bottom 2";
this.bottom2ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);
//
// threeSectionsToolStripMenuItem
//
this.threeSectionsToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.top3ToolStripMenuItem, this.bottom3ToolStripMenuItem });
this.threeSectionsToolStripMenuItem.Name = "threeSectionsToolStripMenuItem";
this.threeSectionsToolStripMenuItem.Size = new System.Drawing.Size(143, 22);
this.threeSectionsToolStripMenuItem.Text = "3 Sections ...";
// top3ToolStripMenuItem
//
this.top3ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("top3ToolStripMenuItem.Image")));
this.top3ToolStripMenuItem.Name = "top3ToolStripMenuItem";
this.top3ToolStripMenuItem.Size = new System.Drawing.Size(123, 22);
this.top3ToolStripMenuItem.Text = "Top 3";
this.top3ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// bottom3ToolStripMenuItem
//
this.bottom3ToolStripMenuItem.Image = ((System.Drawing.Image)(resources.GetObject("bottom3ToolStripMenuItem.Image")));
this.bottom3ToolStripMenuItem.Name = "bottom3ToolStripMenuItem";
this.bottom3ToolStripMenuItem.Size = new System.Drawing.Size(123, 22);
this.bottom3ToolStripMenuItem.Text = "Bottom 3";
this.bottom3ToolStripMenuItem.Click += new System.EventHandler(this.yBoundsCheckChanged);

// particleRadiusParameterToolStripMenuItem
//
this.particleRadiusParameterToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.ToolStripMenuItemPRadiusLower, this.ToolStripMenuItemPRadiusUpper });
this.particleRadiusParameterToolStripMenuItem.Name = "particleRadiusParameterToolStripMenuItem";
this.particleRadiusParameterToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
this.particleRadiusParameterToolStripMenuItem.Text = "Particle Radius";

//
// ToolStripMenuItemPRadiusLower
//
this.ToolStripMenuItemPRadiusLower.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] { this.toolStripTextBoxPRadiusLower, this.ToolStripMenuItemPRadiusLowerUM, this.ToolStripMenuItemPRadiusLowerPercent });
this.ToolStripMenuItemPRadiusLower.Name = "ToolStripMenuItemPRadiusLower";
this.ToolStripMenuItemPRadiusLower.Size = new System.Drawing.Size(144, 22);
this.ToolStripMenuItemPRadiusLower.Text = "Lower Bound";

//
// toolStripTextBoxPRadiusLower
//
this.toolStripTextBoxPRadiusLower.Name = "toolStripTextBoxPRadiusLower";
this.toolStripTextBoxPRadiusLower.Size = new System.Drawing.Size(100, 23);
this.toolStripTextBoxPRadiusLower.Text = "--";
this.toolStripTextBoxPRadiusLower.TextChanged += new System.EventHandler(this.filterAndGraph);
//
// ToolStripMenuItemPRadiusLowerUM
//
this.ToolStripMenuItemPRadiusLowerUM.Checked = true;
this.ToolStripMenuItemPRadiusLowerUM.CheckState = System.Windows.Forms.CheckState.Checked;
this.ToolStripMenuItemPRadiusLowerUM.Name = "ToolStripMenuItemPRadiusLowerUM";
this.ToolStripMenuItemPRadiusLowerUM.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemPRadiusLowerUM.Text = "µm";
this.ToolStripMenuItemPRadiusLowerUM.Click += new System.EventHandler(this.ToolStripMenuItemPRadiusUM_Click);
//
// ToolStripMenuItemPRadiusLowerPercent
//
this.ToolStripMenuItemPRadiusLowerPercent.Name = "ToolStripMenuItemPRadiusLowerPercent";
this.ToolStripMenuItemPRadiusLowerPercent.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemPRadiusLowerPercent.Text = "%";
this.ToolStripMenuItemPRadiusLowerPercent.Click += new System.EventHandler(this.ToolStripMenuItemPRadiusPercent_Click);
//
// ToolStripMenuItemPRadiusUpper
//
    this.toolStripTextBoxPRadiusUpper,
    this.ToolStripMenuItemPRadiusUpperUM,
    this.ToolStripMenuItemPRadiusUpperPercent});
this.ToolStripMenuItemPRadiusUpper.Name = "ToolStripMenuItemPRadiusUpper";
this.ToolStripMenuItemPRadiusUpper.Size = new System.Drawing.Size(144, 22);
this.ToolStripMenuItemPRadiusUpper.Text = "Upper Bound";
//
// toolStripTextBoxPRadiusUpper
//
this.toolStripTextBoxPRadiusUpper.Name = "toolStripTextBoxPRadiusUpper";
this.toolStripTextBoxPRadiusUpper.Size = new System.Drawing.Size(100, 23);
this.toolStripTextBoxPRadiusUpper.Text = "--";
this.toolStripTextBoxPRadiusUpper.TextChanged += new System.EventHandler(this.filterAndGraph);

// ToolStripMenuItemPRadiusUpperUM
//
this.ToolStripMenuItemPRadiusUpperUM.Checked = true;
this.ToolStripMenuItemPRadiusUpperUM.CheckState = System.Windows.Forms.CheckState.Checked;
this.ToolStripMenuItemPRadiusUpperUM.Name = "ToolStripMenuItemPRadiusUpperUM";
this.ToolStripMenuItemPRadiusUpperUM.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemPRadiusUpperUM.Text = "µm";
this.ToolStripMenuItemPRadiusUpperUM.Click += new System.EventHandler(this.ToolStripMenuItemPRadiusUpperUM_Click);

// ToolStripMenuItemPRadiusUpperPercent
//
this.ToolStripMenuItemPRadiusUpperPercent.Name = "ToolStripMenuItemPRadiusUpperPercent";
this.ToolStripMenuItemPRadiusUpperPercent.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemPRadiusUpperPercent.Text = "%";
this.ToolStripMenuItemPRadiusUpperPercent.Click += new System.EventHandler(this.ToolStripMenuItemPRadiusUpperPercent_Click);

// massParameterToolStripMenuItem
//
this.massParameterToolStripMenuItem.Name = "massParameterToolStripMenuItem";
this.massParameterToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
this.massParameterToolStripMenuItem.Text = "Mass";

// ToolStripMenuItemMassLower
//
this.ToolStripMenuItemMassLower.Name = "ToolStripMenuItemMassLower";
this.ToolStripMenuItemMassLower.Size = new System.Drawing.Size(144, 22);
this.ToolStripMenuItemMassLower.Text = "Lower Bound";
// toolStripTextBoxMassLower
//
this.toolStripTextBoxMassLower.Name = "toolStripTextBoxMassLower";
this.toolStripTextBoxMassLower.Size = new System.Drawing.Size(100, 23);
this.toolStripTextBoxMassLower.Text = "--";
this.toolStripTextBoxMassLower.TextChanged += new System.EventHandler(this.filterAndGraph);

//
// ToolStripMenuItemMassLowerKG
//
this.ToolStripMenuItemMassLowerKG.Name = "ToolStripMenuItemMassLowerKG";
this.ToolStripMenuItemMassLowerKG.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemMassLowerKG.Text = "kg";

//
// ToolStripMenuItemMassLowerG
//
this.ToolStripMenuItemMassLowerG.Checked = true;
this.ToolStripMenuItemMassLowerG.Name = "ToolStripMenuItemMassLowerG";
this.ToolStripMenuItemMassLowerG.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemMassLowerG.Text = "g";

//
// ToolStripMenuItemMassLowerMG
//
this.ToolStripMenuItemMassLowerMG.Name = "ToolStripMenuItemMassLowerMG";
this.ToolStripMenuItemMassLowerMG.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemMassLowerMG.Text = "mg";

//
// ToolStripMenuItemMassLowerPercent
//
this.ToolStripMenuItemMassLowerPercent.Name = "ToolStripMenuItemMassLowerPercent";
this.ToolStripMenuItemMassLowerPercent.Size = new System.Drawing.Size(160, 22);
this.ToolStripMenuItemMassLowerPercent.Text = "%";

//
// ToolStripMenuItemMassUpper
//
this.ToolStripMenuItemMassUpperMG,
    this.ToolStripMenuItemMassUpperPercent});
    this.ToolStripMenuItemMassUpper.Name = "ToolStripMenuItemMassUpper";
    this.ToolStripMenuItemMassUpper.Size = new System.Drawing.Size(144, 22);
    this.ToolStripMenuItemMassUpper.Text = "Upper Bound";
    //
    // toolstripTextBoxMassUpper
    //
    this.toolStripTextBoxMassUpper.Name = "toolStripTextBoxMassUpper";
    this.toolStripTextBoxMassUpper.Size = new System.Drawing.Size(100, 23);
    this.toolStripTextBoxMassUpper.Text = "--";
    this.toolStripTextBoxMassUpper.TextChanged += new System.EventHandler(this.filterAndGraph);
    //
    // ToolStripMenuItemMassUpperKG
    //
    this.ToolStripMenuItemMassUpperKG.Name = "ToolStripMenuItemMassUpperKG";
    this.ToolStripMenuItemMassUpperKG.Size = new System.Drawing.Size(160, 22);
    this.ToolStripMenuItemMassUpperKG.Text = "kg";
    //
    // ToolStripMenuItemMassUpperG
    //
    this.ToolStripMenuItemMassUpperG.Checked = true;
    this.ToolStripMenuItemMassUpperG.Name = "ToolStripMenuItemMassUpperG";
    this.ToolStripMenuItemMassUpperG.Size = new System.Drawing.Size(160, 22);
    this.ToolStripMenuItemMassUpperG.Text = "g";
    //
    // ToolStripMenuItemMassUpperMG
    //
    this.ToolStripMenuItemMassUpperMG.Name = "ToolStripMenuItemMassUpperMG";
    this.ToolStripMenuItemMassUpperMG.Size = new System.Drawing.Size(160, 22);
    this.ToolStripMenuItemMassUpperMG.Text = "mg";
    //
    // ToolStripMenuItemMassUpperPercent
    //
    this.ToolStripMenuItemMassUpperPercent.Name = "ToolStripMenuItemMassUpperPercent";
    this.ToolStripMenuItemMassUpperPercent.Size = new System.Drawing.Size(160, 22);
    this.ToolStripMenuItemMassUpperPercent.Text = "%";
this.coalParameterToolStripMenuItem.Name = "coalParameterToolStripMenuItem";
this.coalParameterToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
this.coalParameterToolStripMenuItem.Text = "Coal";
this.coalParameterToolStripMenuItem.Click += new System.EventHandler(this.compositionParameterFlipCheck);

// toolStripSeparator1
// this.toolStripSeparator1.Name = "toolStripSeparator1";
// this.toolStripSeparator1.Size = new System.Drawing.Size(154, 6);

// resetToDefaultToolStripMenuItem
// this.resetToDefaultToolStripMenuItem.Name = "resetToDefaultToolStripMenuItem";
// this.resetToDefaultToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
// this.resetToDefaultToolStripMenuItem.Text = "Reset to Default";
this.resetToDefaultToolStripMenuItem.Click += new System.EventHandler(this.resetToDefaultToolStripMenuItem_Click);

// setToShowAllToolStripMenuItem
// this.setToShowAllToolStripMenuItem.Name = "setToShowAllToolStripMenuItem";
// this.setToShowAllToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
// this.setToShowAllToolStripMenuItem.Text = "Set to Show All";
this.setToShowAllToolStripMenuItem.Click += new System.EventHandler(this.setToShowAllToolStripMenuItem_Click);

// toolStripSeparator4
// this(toolStripSeparator4.Name = "toolStripSeparator4";
// this(toolStripSeparator4.Size = new System.Drawing.Size(154, 6);

// ThreeDExplorerToolStripMenuItem
// this.ThreeDExplorerToolStripMenuItem.Name = "ThreeDExplorerToolStripMenuItem";
// this.ThreeDExplorerToolStripMenuItem.Size = new System.Drawing.Size(157, 22);
// this.ThreeDExplorerToolStripMenuItem.Text = "3-D Explorer";
this.ThreeDExplorerToolStripMenuItem.Click += new System.EventHandler(this.ThreeDExplorerToolStripMenuItem_Click);

// progressBar
//
this.progressBar.Location = new System.Drawing.Point(5, 653);
this.progressBar.Name = "progressBar";
this.progressBar.Size = new System.Drawing.Size(129, 20);
this.progressBar.TabIndex = 20;
this.progressBar.Visible = false;
//
// progressStatus
//
this.progressStatus.AutoSize = true;
this.progressStatus.Location = new System.Drawing.Point(9, 637);
this.progressStatus.Name = "progressStatus";
this.progressStatus.Size = new System.Drawing.Size(0, 13);
this.progressStatus.TabIndex = 21;
//
// groupBoxDataBox
//
this.groupBoxDataBox.Controls.Add(this.labelClassifierStats);
this.groupBoxDataBox.Location = new System.Drawing.Point(13, 209);
this.groupBoxDataBox.Name = "groupBoxDataBox";
this.groupBoxDataBox.Size = new System.Drawing.Size(219, 161);
this.groupBoxDataBox.TabIndex = 22;
this.groupBoxDataBox.TabStop = false;
this.groupBoxDataBox.Text = "Classifier Stats";
//
// labelClassifierStats
//
this.labelClassifierStats.AutoSize = true;
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.labelClassifierStats.Location = new System.Drawing.Point(6, 16);
this.labelClassifierStats.Name = "labelClassifierStats";
this.labelClassifierStats.Size = new System.Drawing.Size(154, 140);
this.labelClassifierStats.TabIndex = 0;
this.labelClassifierStats.Text = "Particle Radius (µm):
Max:
Min:
Mean:
Speed (m/s):
Max:
Mean:
Count:";
//
// buttonBackTimeStep
//

// buttonForwardTimeStep
//

// groupBoxTimeStats
//
this.groupBoxTimeStats.Controls.Add(this.labelTimeStats); this.groupBoxTimeStats.Location = new System.Drawing.Point(13, 376); this.groupBoxTimeStats.Name = "groupBoxTimeStats"; this.groupBoxTimeStats.Size = new System.Drawing.Size(219, 241); this.groupBoxTimeStats.TabIndex = 23; this.groupBoxTimeStats.TabStop = false; this.groupBoxTimeStats.Text = "Time Stats"; this.groupBoxTimeStats.Visible = false;

// labelTimeStats
//
this.labelTimeStats.Size = new System.Drawing.Size(189, 210);
this.labelTimeStats.TabIndex = 0;
this.labelTimeStats.Text = "Tot. Avg. Res. Time:\nCoal:\nWood:\nSwirl Avg. Res. Time\nCoal:\nWood:\nNumber (Mass) of Particles\nCoal:\nWood:\n" +
"Number (Mass) that Swirl\nCoal:\nWood:";

//
// ClassiParseForm
//
this.AllowDrop = true;
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.ClientSize = new System.Drawing.Size(1150, 676);
this.Controls.Add(this.groupBoxTimeStats);
this.Controls.Add(this.buttonForwardTimeStep);
this.Controls.Add(this.buttonBackTimeStep);
this.Controls.Add(this.groupBoxDataBox);
this.Controls.Add(this.progressStatus);
this.Controls.Add(this.progressBar);
this.Controls.Add(this.groupBoxGraphOptions);
this.Controls.Add(this.comboBoxYAxis);
this.Controls.Add(this.comboBoxXAxis);
this.Controls.Add(this.zedGraph);
this.MainMenuStrip = this.menuStrip1;
this.Name = "ClassiParseForm";
this.Text = "ClassiParse";
this.FormClosed += new System.Windows.Forms.FormClosedEventHandler(this.ClassiParseForm_FormClosed);

System.Windows.Forms.FormClosedEventHandler(this.ClassiParseForm_FormClosed)

this.groupBoxGraphOptions.ResumeLayout(false);
this.groupBoxGraphOptions.PerformLayout();
this.menuStrip1.ResumeLayout(false);
this.menuStrip1.PerformLayout();
this.groupBoxDataBox.ResumeLayout(false);
this.groupBoxDataBox.PerformLayout();
this.groupBoxTimeStats.ResumeLayout(false);
this.groupBoxTimeStats.PerformLayout();
this.ResumeLayout(false);
this.PerformLayout();
this.PerformLayout();

#endregion
private ZedGraph.ZedGraphControl zedGraph;
private System.Windows.Forms.CheckBox checkboxConnectPoints;
private System.Windows.Forms.CheckBox checkBoxShowWood;
private System.Windows.Forms.CheckBox checkBoxShowCoal;
private System.Windows.Forms.ComboBox comboBoxXAxis;
private System.Windows.Forms.ComboBox comboBoxYAxis;
private System.Windows.Forms.ComboBox comboBoxPointSize;
private System.Windows.Forms.Label labelPointSize;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.GroupBox groupBoxGraphOptions;
private System.Windows.Forms.ComboBox comboBoxGraphType;
private System.Windows.Forms.Label labelDivisions;
private System.Windows.Forms.ComboBox comboBoxDivisions;
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.ToolStripComboBox toolStripComboBoxDWallParameter;
private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem importDataToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem exportAllParticlesToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem parametersToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem dWallParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem(toolStripComboBoxDWallParameter);
private System.Windows.Forms.ToolStripMenuItem yBoundsParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem fullClassifierToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem section1ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem section2ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem section3ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem section4ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem twoSectionsToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem top2ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem middle2ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem bottom2ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem threeSectionsToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem top3ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem bottom3ToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem particleRadiusParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem massParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem woodParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem coalParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuItemPRadiusLower;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxPRadiusLower;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuitemPRadiusLowerUM;
private System.Windows.Forms.ToolStripMenuItem ToolStripMenuitemPRadiusLowerPercent;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxPRadiusUpper;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxMassLower;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxMassUpper;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
private System.Windows.Forms.ToolStripMenuItem resetToDefaultToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem exportChartDataToolStripMenuItem;
private System.Windows.Forms.Label progressStatus;
private System.Windows.Forms.GroupBox groupBoxDataBox;
private System.Windows.Forms.ToolStripMenuItem speedToolStripMenuItem;
private System.Windows.Forms.ToolStripComboBox toolStripComboBoxSpeedParameter;
private System.Windows.Forms.Label labelClassifierStats;
private System.Windows.Forms.ToolStripMenuItem ThreeDExplorerToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem reverseEngineeringToolStripMenuItem;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator3;
private System.Windows.Forms.ToolStripMenuItem setToShowAllToolStripMenuItem;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator2;
private System.Windows.Forms.ToolStripMenuItem adjustSwirlParameterToolStripMenuItem;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxSwirlDepth;
private System.Windows.Forms.GroupBox groupBoxTimeStats;
private System.Windows.Forms.CheckBox checkBoxShowClassifier;
private System.Windows.Forms.ToolStripTextBox mToolStripTextBox;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator4;
private System.Windows.Forms.ToolStripTextBox toolStripTextBoxSwirlDepth;
private System.Windows.Forms.ToolStripMenuItem mToolStripMenuItem;

class _3DAxis
{
    public double size;
    private double[] angles;
    private double[] xs;
    private double[] ys;

    public _3DAxis(double s, double l)
    { size = s;
      xs = new double[5];
      ys = new double[5];
      angles = new double[3];
    }

    public double threeDX(double x, double y, double z)
    {
        double xx = Math.Cos(angles[2]) * Math.Cos(angles[0]) -
                    Math.Sin(angles[2]) * Math.Sin(angles[0]) * Math.Sin(angles[1]);
        double yx = -Math.Cos(angles[2]) * Math.Sin(angles[0]);
        double zx = -Math.Sin(angles[2]) * Math.Cos(angles[1]);
        return xx * x + yx * y + zx * z;
    }

    public double threeDY(double x, double y, double z)
    {
        double xy = Math.Cos(angles[2]) * Math.Sin(angles[0]);
        double yz = Math.Sin(angles[2]) * Math.Cos(angles[1]);
        return xy * x + yz * y;
    }
}
double yy = Math.Cos(angles[2]) * Math.Cos(angles[0]) * 
Math.Sin(angles[1]) - Math.Sin(angles[2]) * Math.Sin(angles[0]);
double zy = Math.Cos(angles[2]) * Math.Sin(angles[1]);
return xy * x + yy * y + zy * z;
}

public double[] axisXs()
{
    return xs;
}

public double[] axisYs()
{
    return ys;
}

public void setAngles(double a, double b, double c)
{
    angles = new double[] { a, b, c };
    xs[0] = threeDX(size, 0, 0);
    xs[1] = threeDX(0, 0, 0);
    xs[2] = threeDX(0, size, 0);
    xs[3] = threeDX(0, 0, 0);
    xs[4] = threeDX(0, 0, size);
    ys[0] = threeDY(size, 0, 0);
    ys[1] = threeDY(0, 0, 0);
    ys[2] = threeDY(0, size, 0);
    ys[3] = threeDY(0, 0, 0);
    ys[4] = threeDY(0, 0, size);
}

public partial class _3DSpacePlotter : Form
{
    public const string COAL = "2";
    public const string WOOD = "1";

    private List<Particle> selectedParticles;

    private _3DAxis axis;
    private const double originLift = 0.618595;

    private bool time = false;
    private bool firstRound = true;
    private WallBoundary classi;

    private string fileName;
    private Random rnd;
    Color cor;
private bool addingToList = false;
private bool start = true;

public _3DSpacePlotter(List<Particle> sps, bool t, string fn) {
    // Build the form
    InitializeComponent();

    selectedParticles = sps;
    axis = new _3DAxis(0.4, 0.6);
    time = t;
    fileName = fn;
    classi = new WallBoundary(22);

    // Let's start the angles in a place so that the classifier is upright
    trackBarAAngle.Value = (int)(-Math.PI / 2 * 100);
    trackBarBAngle.Value = 0;
    trackBarCAngle.Value = (int)(Math.PI / 2 * 100);

    // Set up the zed graph stuff
    zedGraph.GraphPane.Title.IsVisible = false;
    zedGraph.GraphPane.XAxis.Title.IsVisible = false;
    zedGraph.GraphPane.YAxis.Title.IsVisible = false;
    zedGraph.GraphPane.XAxis.IsVisible = false;
    zedGraph.GraphPane.YAxis.IsVisible = false;
    zedGraph.GraphPane.Legend.IsVisible = false;

    // set graph to not scale fonts (so if you resize it, the graph gets bigger, but the font doesn't)
    zedGraph.GraphPane.IsFontsScaled = false;

    Text = "3D Explorer: " + fileName;
    start = true;
}

private void makePlot() {
    double a = trackBarAAngle.Value / 100.0;
    double b = trackBarBAngle.Value / 100.0;
    double c = trackBarCAngle.Value / 100.0;

    if (!time) {
        if (selectedParticles.Count > 200)
            checkBoxShowLegend.Checked = false;
        makeStdPlot(a, b, c);
    }
}
else
{
    makeTimePlot(a, b, c);
}

// draw the classifier outline on top of the graph.
outlineClassifier(a, b, c);

// set the bounds
zedGraph.Refresh();
zedGraph.AxisChange();
updateParticleStats();
}

private void makeTimePlot(double a, double b, double c)
{
    zedGraph.GraphPane.CurveList.Clear();
    zedGraph.GraphPane.Legend.IsVisible =
    checkBoxShowLegend.Checked;

    int age = selectedParticles[0].age();
    int pos = 0;
    int loops = 0;

    // set the cursor to the wait cursor while the graph is made
    Cursor.Current = Cursors.WaitCursor;
    Application.DoEvents();

    foreach (Particle p in selectedParticles)
    {
        // if this item is not selected, skip to the next particle
        // Since we are going through in order, each loop will be for
        // the next particle in the list.
        // Here we check if that position (index) is one of the
        // selected indicies.
        if
        (!listBoxParticlesToShow.SelectedIndices.Contains(pos++) && !firstRound)
        {
            continue;
        }
    }

    firstRound = false;
    List<double> xl = new List<double>();
    List<double> yl = new List<double>();

    for (int i = 0; i < age; i++)
    {
        p.setTimeStep(i);
        if (p.properties() == null) continue;
        p.setProperties();
xl.Add(p.ThreeDX(a, b, c, originLift));
yl.Add(p.ThreeDY(a, b, c, originLift));
}

int trueAge = xl.Count;
double[] xs = new double[trueAge];
double[] ys = new double[trueAge];

for (int i = 0; i < trueAge; i++)
{
    xs[i] = xl[i];
    ys[i] = yl[i];
}

rnd = new Random(int.Parse(p.PID));
cor = Color.FromArgb(rnd.Next(256), rnd.Next(256), rnd.Next(256));

LineItem particle_curve =
zedGraph.GraphPane.AddCurve("particle: " + p.PID, xs, ys, cor);
    particle_curve.Line.IsVisible = true;
    particle_curve.Symbol.Type = SymbolType.Circle;
    particle_curve.Symbol.Fill.Color = cor;
    particle_curve.Symbol.Fill.Type = FillType.Solid;
    particle_curve.Line.IsSmooth = true;
    particle_curve.Line.SmoothTension = 0.1F;

    // increment loops
    loops++;
    if (loops == 100)
    {
        loops = 0;
        Application.DoEvents();
    }
}

// set the cursor back to normal
Cursor.Current = Cursors.Default;

private void makeStdPlot(double a, double b, double c)
{
    zedGraph.GraphPane.CurveList.Clear();
    int numWood = 0;
    int numCoal = 0;

    foreach (Particle p in selectedParticles)
    {
        if (p.type == COAL)
        {
            numCoal++;
        }
    }
else {
    numWood++;
}
}

double[] coal_xs = new double[numCoal];
double[] coal_ys = new double[numCoal];

double[] wood_xs = new double[numWood];
double[] wood_ys = new double[numWood];

int i = 0;
int j = 0;

foreach (Particle p in selectedParticles) {
    if (p.type == COAL) {
        coal_xs[i] = p.ThreeDX(a, b, c, originLift);
        coal_ys[i] = p.ThreeDY(a, b, c, originLift);
        i++;
    } else {
        wood_xs[j] = p.ThreeDX(a, b, c, originLift);
        wood_ys[j] = p.ThreeDY(a, b, c, originLift);
        j++;
    }
}

LineItem coal_curve =
zedGraph.GraphPane.AddCurve("coalparticles", coal_xs, coal_ys,
Color.Blue);
coal_curve.Line.IsVisible = false;
coal_curve.Symbol.Type = SymbolType.Circle;

LineItem wood_curve =
zedGraph.GraphPane.AddCurve("woodparticles", wood_xs, wood_ys, Color.Red);
wood_curve.Line.IsVisible = false;
wood_curve.Symbol.Type = SymbolType.Circle;

private void outlineClassifier(double a, double b, double c) {
    // show a cartesian axis
    /*axis.setAngles(a, b, c);
    LineItem axisCurve =
    zedGraph.GraphPane.AddCurve("axis",axis.axisXs(),axis.axisYs(),Color.Black");
    axisCurve.Line.IsVisible = true;
    axisCurve.Symbol.IsVisible = false;/*
    classi.setAngles(a, b, c, originLift);
    LineItem classiOutlineTop = zedGraph.GraphPane.AddCurve("",
    classi.outlineTopXs, classi.outlineTopYs, Color.Brown);
classiOutlineTop.Line.IsVisible = true;
classiOutlineTop.Symbol.IsVisible = false;

LineItem classiOutlineBot = zedGraph.GraphPane.AddCurve("",
cassi.outlineBotXs, classi.outlineBotYs, Color.Brown);
classiOutlineBot.Line.IsVisible = true;
classiOutlineBot.Symbol.IsVisible = false;

LineItem extremePosXs = zedGraph.GraphPane.AddCurve("",
cassi.extremePosXs()[0], cassi.extremePosXs()[1], Color.Brown);
extremePosXs.Line.IsVisible = true;
extremePosXs.Symbol.IsVisible = false;
LineItem extremeNegXs = zedGraph.GraphPane.AddCurve("",
cassi.extremeNegXs()[0], cassi.extremeNegXs()[1], Color.Brown);
extremeNegXs.Line.IsVisible = true;
extremeNegXs.Symbol.IsVisible = false;
LineItem extremePosYs = zedGraph.GraphPane.AddCurve("",
cassi.extremePosYs()[0], cassi.extremePosYs()[1], Color.Brown);
extremePosYs.Line.IsVisible = true;
extremePosYs.Symbol.IsVisible = false;
LineItem extremeNegYs = zedGraph.GraphPane.AddCurve("",
cassi.extremeNegYs()[0], cassi.extremeNegYs()[1], Color.Brown);
extremeNegYs.Line.IsVisible = true;
extremeNegYs.Symbol.IsVisible = false;
}

private void trackBarAAngle_Scroll(object sender, EventArgs e)
{
    makePlot();
}

private void trackBarBAngle_Scroll(object sender, EventArgs e)
{
    makePlot();
}

private void trackBarCAngle_Scroll(object sender, EventArgs e)
{
    makePlot();
}

private void listBoxParticlesToShow_SelectedIndexChanged(object sender, EventArgs e)
{
    // If several items are being added to the list, don't update
    // the plot until they're all added.
    if (!addingToList)
        makePlot();
}

private void fillListBox()
{
    int count = 0;
    foreach (Particle p in selectedParticles)
private void buttonAllWood_Click(object sender, EventArgs e)
{
    // clear the selected items
    listBoxParticlesToShow.SelectedItems.Clear();

    // Set addingToList to true so that the form doesn't try to
    // redraw every time we add a particle
    addingToList = true;

    // select only the wood ones
    for (int i = 0; i < listBoxParticlesToShow.Items.Count; i++)
    {
        if (listBoxParticlesToShow.Items[i].ToString().Contains("Wood"))
        {
            listBoxParticlesToShow.SelectedItems.Add(i);
        }
    }

    // set adding to list back to false, then make the plot
    addingToList = false;
    makePlot();
}

private void buttonAllCoal_Click(object sender, EventArgs e)
{
    // clear the selected items
    listBoxParticlesToShow.SelectedItems.Clear();

    // Set addingToList to true so that the form doesn't try to
    // redraw every time we add a particle
    addingToList = true;

    // select only the coal ones
    for (int i = 0; i < listBoxParticlesToShow.Items.Count; i++)
    {
        if (listBoxParticlesToShow.Items[i].ToString().Contains("Coal"))
        {
            listBoxParticlesToShow.SelectedItems.Add(i);
        }
    }

    // set adding to list back to false, then make the plot
    addingToList = false;
    makePlot();
}
// Update the values in the Particle Stats box
private void updateParticleStats()
{
    double avgResTime = 0.0;
    double avgDepth = 0.0;

    int pos = 0;

    // loop through all particles
    foreach (Particle p in selectedParticles)
    {
        // if this item is not selected, skip to the next particle
        // Since are going through in order, each loop will be for
        // the next particle in the list.
        // Here we check if that position (index) is one of the
        // selected indicies.
        if (!listBoxParticlesToShow.SelectedIndices.Contains(pos++) && !firstRound)
        {
            continue;
        }

        // Add to the res time and depth
        avgResTime += p.getResidenceTime();
        avgDepth += p.GetMaxPen();
    }

    // divide by the count to get the average
    avgResTime /= listBoxParticlesToShow.SelectedIndices.Count;
    avgDepth /= listBoxParticlesToShow.SelectedIndices.Count;

    // Update the properties on the window
    labelResidenceTime.Text = Math.Round(avgResTime, 2) + " GMVs";
    labelMaxPen.Text = Math.Round(avgDepth, 2) + " m";
}

private void checkBoxShowLegend_CheckedChanged(object sender, EventArgs e)
{
    // update the plot with our without the legend.
    makePlot();
}

private void buttonClose_Click(object sender, EventArgs e)
{
    Close();
}

private void _3DSpacePlotter_Shown(object sender, EventArgs e)
{
    if (start)
    {
        // this will keep the graph from drawing until it is ready
addingToList = true;

makePlot();
// If this is a 'time' plot
if (time)
{
    // show the particle line chooser
    listBoxParticlesToShow.Visible = true;

    // add the items to the list
    fillListBox();
}
else
{
    // hide all of the 'time' features
    listBoxParticlesToShow.Visible = false;
    buttonAllCoal.Visible = false;
    buttonAllWood.Visible = false;
    groupBoxParticleStats.Visible = false;
    checkBoxShowLegend.Visible = false;
}
zedGraph.Refresh();
zedGraph.AxisChange();

// no longer adding to the list
addingToList = false;
start = false;
}

private void buttonSwirlPars_Click(object sender, EventArgs e)
{
    // FIXME (add filter to find particles that swirl)
    classi.topY = double.Parse(textBoxSwirlDepth.Text);

    // clear the selected items
    listBoxParticlesToShow.SelectedIndices.Clear();

    // Set addingToList to true so that the form doesn't try to
    // redraw every time we add a particle
    addingToList = true;

    // select only the coal ones
    for (int i = 0; i < selectedParticles.Count; i++)
    {
        selectedParticles[i].calculateTimeProperties(classi);

        if (selectedParticles[i].GetMaxPen() > 0)
        {
            listBoxParticlesToShow.SelectedIndices.Add(i);
        }
    }
}
partial class _3DSpacePlotter
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        this.zedGraph = new ZedGraph.ZedGraphControl();
        this.trackBarAAngle = new System.Windows.Forms.TrackBar();
        this.trackBarBAngle = new System.Windows.Forms.TrackBar();
        this.trackBarCAngle = new System.Windows.Forms.TrackBar();
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.listBoxParticlesToShow = new System.Windows.Forms.ListBox();
        this.buttonAllWood = new System.Windows.Forms.Button();
        this.buttonAllCoal = new System.Windows.Forms.Button();
        this.groupBoxParticleStats = new System.Windows.Forms.GroupBox();
        this.labelMaxPen = new System.Windows.Forms.Label();
        this.labelResidenceTime = new System.Windows.Forms.Label();
        this.label5 = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
        this.listBoxParticlesToShow.SuspendLayout();
    
    
    #endregion
}
this.checkBoxShowLegend = new System.Windows.Forms.CheckBox();
this.buttonSwirlPars = new System.Windows.Forms.Button();
this.textBoxSwirlDepth = new System.Windows.Forms.TextBox();
this.label6 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();

((System.ComponentModel.ISupportInitialize)(this.trackBarAAngle)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.trackBarBAngle)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.trackBarCAngle)).BeginInit();

this.groupBoxParticleStats.SuspendLayout();
this.SuspendLayout();

// zedGraph
this.zedGraph.IsShowPointValues = true;
this.zedGraph.Location = new System.Drawing.Point(462, 12);
this.zedGraph.Name = "zedGraph";
this.zedGraph.ScrollGrace = 0D;
this.zedGraph.ScrollMaxX = 0D;
this.zedGraph.ScrollMaxY = 0D;
this.zedGraph.ScrollMaxY2 = 0D;
this.zedGraph.ScrollMinX = 0D;
this.zedGraph.ScrollMinY = 0D;
this.zedGraph.ScrollMinY2 = 0D;
this.zedGraph.Size = new System.Drawing.Size(600, 596);
this.zedGraph.TabIndex = 0;

// trackBarAAngle
this.trackBarAAngle.Location = new System.Drawing.Point(12, 28);
this.trackBarAAngle.Maximum = 314;
this.trackBarAAngle.Minimum = -314;
this.trackBarAAngle.Name = "trackBarAAngle";
this.trackBarAAngle.Size = new System.Drawing.Size(317, 45);
this.trackBarAAngle.TabIndex = 1;
this.trackBarAAngle.TickFrequency = 5;
this.trackBarAAngle.Scroll += new System.EventHandler(this.trackBarAAngle_Scroll);

System.EventHandler(this.trackBarBAngle_Scroll);
this.trackBarBAngle.Location = new System.Drawing.Point(12, 97);
this.trackBarBAngle.Maximum = 314;
this.trackBarBAngle.Name = "trackBarBAngle";
this.trackBarBAngle.Size = new System.Drawing.Size(317, 45);
this.trackBarBAngle.TabIndex = 2;
this.trackBarBAngle.TickFrequency = 5;
this.trackBarBAngle.Scroll += new System.EventHandler(this.trackBarBAngle_Scroll);
this.trackBarBAngle.Maximum = 314;
this.trackBarBAngle.Minimum = -314;
this.trackBarBAngle.Name = "trackBarBAngle";
this.trackBarBAngle.Size = new System.Drawing.Size(317, 45);
this.trackBarBAngle.TabIndex = 2;
this.trackBarBAngle.TickFrequency = 5;
this.trackBarBAngle.Scroll += new System.EventHandler(this.trackBarBAngle_Scroll);

//
// trackBarCAngle
//
this.trackBarCAngle.Location = new System.Drawing.Point(12, 163);
this.trackBarCAngle.Maximum = 314;
this.trackBarCAngle.Minimum = -314;
this.trackBarCAngle.Name = "trackBarCAngle";
this.trackBarCAngle.Size = new System.Drawing.Size(317, 45);
this.trackBarCAngle.TabIndex = 3;
this.trackBarCAngle.TickFrequency = 5;
this.trackBarCAngle.Scroll += new System.EventHandler(this.trackBarCAngle_Scroll);

//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(12, 12);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(79, 13);
this.label1.TabIndex = 4;
this.label1.Text = "Y-Axis Rotation";

//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(12, 81);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(79, 13);
this.label2.TabIndex = 5;
this.label2.Text = "Z-Axis Rotation";

//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(12, 147);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(79, 13);
this.label3.TabIndex = 6;
this.label3.Text = "X-Axis Rotation";

//
// listBoxParticlesToShow
//
System.Drawing.Point(12, 214);
this.listBoxParticlesToShow.Name = "listBoxParticlesToShow";
this.listBoxParticlesToShow.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended;
this.listBoxParticlesToShow.Size = new System.Drawing.Size(193, 394);
this.listBoxParticlesToShow.TabIndex = 7;
this.listBoxParticlesToShow.SelectedIndexChanged += new System.EventHandler(this.listBoxParticlesToShow_SelectedIndexChanged);

this.buttonAllWood.Location = new System.Drawing.Point(12, 619);
this.buttonAllWood.Name = "buttonAllWood";
this.buttonAllWood.Size = new System.Drawing.Size(88, 37);
this.buttonAllWood.TabIndex = 8;
this.buttonAllWood.Text = "Select All Wood";
this.buttonAllWood.UseVisualStyleBackColor = true;
this.buttonAllWood.Click += new System.EventHandler(this.buttonAllWood_Click);

this.buttonAllCoal.Location = new System.Drawing.Point(117, 619);
this.buttonAllCoal.Name = "buttonAllCoal";
this.buttonAllCoal.Size = new System.Drawing.Size(88, 37);
this.buttonAllCoal.TabIndex = 9;
this.buttonAllCoal.Text = "Select All Coal";
this.buttonAllCoal.UseVisualStyleBackColor = true;
this.buttonAllCoal.Click += new System.EventHandler(this.buttonAllCoal_Click);

this.groupBoxParticleStats.Controls.Add(this.labelMaxPen);
this.groupBoxParticleStats.Controls.Add(this.labelResidenceTime);
this.groupBoxParticleStats.Controls.Add(this.label5);
this.groupBoxParticleStats.Controls.Add(this.label4);
this.groupBoxParticleStats.Location = new System.Drawing.Point(240, 214);
this.groupBoxParticleStats.Name = "groupBoxParticleStats";

this.groupBoxParticleStats.Controls.Add(this.labelMaxPen);
this.groupBoxParticleStats.Controls.Add(this.labelResidenceTime);
this.groupBoxParticleStats.Controls.Add(this.label5);
this.groupBoxParticleStats.Controls.Add(this.label4);
this.groupBoxParticleStats.Location = new System.Drawing.Point(240, 214);
this.groupBoxParticleStats.Name = "groupBoxParticleStats";

this.groupBoxParticleStats.Controls.Add(this.labelMaxPen);
this.groupBoxParticleStats.Controls.Add(this.labelResidenceTime);
this.groupBoxParticleStats.Controls.Add(this.label5);
this.groupBoxParticleStats.Controls.Add(this.label4);
this.groupBoxParticleStats.Location = new System.Drawing.Point(240, 214);
this.groupBoxParticleStats.Name = "groupBoxParticleStats";
this.groupBoxParticleStats.Size = new System.Drawing.Size(165, 99);
this.groupBoxParticleStats.TabIndex = 11;
this.groupBoxParticleStats.TabStop = false;
this.groupBoxParticleStats.Text = "Particle Stats";
    //
    // labelMaxPen
    //
this.labelMaxPen.AutoSize = true;
this.labelMaxPen.Location = new System.Drawing.Point(99, 70);
this.labelMaxPen.Name = "labelMaxPen";
this.labelMaxPen.Size = new System.Drawing.Size(15, 13);
this.labelMaxPen.TabIndex = 3;
this.labelMaxPen.Text = "m";
    //
    // labelResidenceTime
    //
this.labelResidenceTime.AutoSize = true;
this.labelResidenceTime.Location = new System.Drawing.Point(99, 33);
this.labelResidenceTime.Name = "labelResidenceTime";
this.labelResidenceTime.Size = new System.Drawing.Size(36, 13);
this.labelResidenceTime.TabIndex = 2;
this.labelResidenceTime.Text = "GMVs";
    //
    // label5
    //
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(9, 70);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(80, 13);
this.label5.TabIndex = 1;
this.label5.Text = "Classifier Depth";
    //
    // label4
    //
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(6, 33);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(87, 13);
this.label4.TabIndex = 0;
this.label4.Text = "Residence Time:";
    //
    // checkBoxShowLegend
    //
this.checkBoxShowLegend.AutoSize = true;
this.checkBoxShowLegend.Location = new System.Drawing.Point(240, 344);
this.checkBoxShowLegend.Name = "checkBoxShowLegend";
this.checkBoxShowLegend.Size = new System.Drawing.Size(135, 17);
this.checkBoxShowLegend.TabIndex = 12;
this.checkBoxShowLegend.Text = "Show Legend on Chart";
this.checkBoxShowLegend.UseVisualStyleBackColor = true;
this.checkBoxShowLegend.CheckedChanged += new
System.EventHandler(this.checkBoxShowLegend_CheckedChanged);
//
// buttonSwirlPars
//
this.buttonSwirlPars.Location = new System.Drawing.Point(223,
619);
this.buttonSwirlPars.Name = "buttonSwirlPars";
this.buttonSwirlPars.Size = new System.Drawing.Size(90, 36);
this.buttonSwirlPars.TabIndex = 13;
this.buttonSwirlPars.Text = "Select Swirling Particles";
this.buttonSwirlPars.UseVisualStyleBackColor = true;
this.buttonSwirlPars.Click += new
System.EventHandler(this.buttonSwirlPars_Click);
//
// textBoxSwirlDepth
//
this.textBoxSwirlDepth.Location = new System.Drawing.Point(328,
629);
this.textBoxSwirlDepth.Name = "textBoxSwirlDepth";
this.textBoxSwirlDepth.Size = new System.Drawing.Size(47, 20);
this.textBoxSwirlDepth.TabIndex = 14;
this.textBoxSwirlDepth.Text = "1.1";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(381, 632);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(15, 13);
this.label6.TabIndex = 15;
this.label6.Text = "m";
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(325, 613);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(59, 13);
this.label7.TabIndex = 16;
this.label7.Text = "Swirl Y limit";
//
// _3DSpacePlotter
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.ClientSize = new System.Drawing.Size(1074, 668);
this.Controls.Add(this.label7);
this.Controls.Add(this.label6);
this.Controls.Add(this.textBoxSwirlDepth);
this.Controls.Add(this.buttonSwirlPars);
this.Controls.Add(this.checkBoxShowLegend);
this.Controls.Add(this.groupBoxParticleStats);
this.Controls.Add(this.buttonAllCoal);
this.Controls.Add(this.buttonAllWood);
this.Controls.Add(this.listBoxParticlesToShow);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.label1);
this.Controls.Add(this.trackBarCAngle);
this.Controls.Add(this.trackBarBAngle);
this.Controls.Add(this.trackBarAAngle);
this.Controls.Add(this.zedGraph);
this.Name = "3DSpacePlotter";
this.Text = "3DSpacePlotter";
thisShown += new
System.EventHandler(this._3DSpacePlotter_Shown);
((System.ComponentModel.ISupportInitialize)(this.trackBarAAngle)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.trackBarBAngle)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.trackBarCAngle)).EndInit();
this.groupBoxParticleStats.ResumeLayout(false);
this.groupBoxParticleStats.PerformLayout();
this.ResumeLayout(false);
this.PerformLayout();
}
#endregion

private ZedGraph.ZedGraphControl zedGraph;
private System.Windows.Forms.TrackBar trackBarAAngle;
private System.Windows.Forms.TrackBar trackBarBAngle;
private System.Windows.Forms.TrackBar trackBarCAngle;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.ListBox listBoxParticlesToShow;
private System.Windows.Forms.Button buttonAllWood;
private System.Windows.Forms.Button buttonAllCoal;
private System.Windows.Forms.GroupBox groupBoxParticleStats;
private System.Windows.Forms.Label labelMaxPen;
private System.Windows.Forms.Label labelResidenceTime;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.CheckBox checkBoxShowLegend;
private System.Windows.Forms.TextBox textBoxSwirlDepth;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
}
//the particle class represents each 'particle' in the classifier.
public class Particle
{
    public const string COAL = "2";
    public const string WOOD = "1";

    //the particle has 10 members:
    //PID is the particle ID.
    //pRadius is the radius of the particle
    //type represents if the particle is wood or coal
    //x is the x-coordinate of the particle
    //y is the y-coordinate of the particle
    //z is the z-coordinate of the particle
    //wallAt is the 'radius' of the wall in the classifier at the
    particle's y (height)
    //radius is the distance the particle is from the center of the
classifier (r in cylindrical coords)
     //properties is a dictionary pairing strings with these values to
make accessing them easier in the graphing function

    // Consider large particles to be > 300 micron

    //The density of woody biomass and coal is 985 and 1250 kg/m3
respectively
    private double rhoWood = 985;
    private double rhoCoal = 1250;

    public double dWall { get; set; }
    public string PID { get; set; }

    //2 for coal, 1 for wood
    public string type { get; set; }

    public double wallAt { get; set; }
    private List <Dictionary<string, double>> dataStore { get; }

    public double x { get; set; }
    public double y { get; set; }
    public double z { get; set; }

    public double speed { get; set; }
    public double pRadius{ get; set; }
    public double mass { get; set; }
    public int timeStep = 0;
    private int residenceTime = 0;
private int deepResidenceTime = 0;
private double maxPenetration = 0.0;

// Default constructor for making a particle object without
// updating its history
public Particle(string p)
{
    dataStore = new List<Dictionary<string, double>>();
    PID = p;
}

// normal constructor
public Particle(string p, string mr, string t, double xx, double
yy, double zz, double way, double r, double s)
{
    dataStore = new List<Dictionary<string, double>>();
    PID = p;
    addData(mr, t, xx, yy, zz, way, r, s);
}

public void addData(string mr, string t, double xx, double yy,
double zz, double way, double r, double s)
{
    Dictionary<string, double> properties = new Dictionary<string,
double>();
    type = t;
    wallAt = way;
    dWall = way - r;
    x = xx;
    y = yy;
    z = zz;

    pRadius = double.Parse(mr);

    mass = 4 / 3 * Math.PI * Math.Pow(pRadius / 1000000, 2);
    if (type == "2")
        mass *= rhoCoal;
    else
        mass *= rhoWood;

    speed = s;

    properties.Add("Particle Radius", pRadius);
    properties.Add("mass", mass);
    properties.Add("x", xx);
    properties.Add("y", yy);
    properties.Add("z", zz);
    properties.Add("radius", r);
properties.Add("dWall", dWall);
properties.Add("type", double.Parse(type));
properties.Add("speed", speed);
properties.Add("wallAt", wallAt);
properties.Add("PID", double.Parse(PID));

    // Add the data at position '0' in the list because the data will be passed from last to first.
    dataStore.Insert(0, properties);
}

    // This function is called once a particle's time data has been fully set up. It calculates the residence time and max penetration
public void calculateTimeProperties(WallBoundary classifier)
{
    int inTime = 0;
    int inDeepTime = 0;
    int outTime = dataStore.Count;
    double maxPen = 0.0;
    bool inClassifier = false;
    bool inVanes = false;
    bool entered = false;
    bool enteredDeep = false;

    // loop through the data in order (from start to finish)
    for (int i = 0; i < dataStore.Count; i++)
    {
        Dictionary<string, double> par = dataStore[i];

        // if there isn't any data for this timestep, continue
        if (par == null)
        {
            // First check if it ever entered. If it did, note the exit time.
            if (entered)
            {
                outTime = i;
            }
            continue;
        }

        inVanes = classifier.isInVanes(par["x"], par["y"],
par["z"]);
        inClassifier = classifier.isInClassifier(par["x"],
par["y"], par["z"]);

        if (inVanes || inClassifier && !entered)
        {
            entered = true;
            inTime = i;
        }

        if (inClassifier && !enteredDeep)
        {
            enteredDeep = true;
inDeepTime = i;
}

// If we are in the classifier, check the maxPen stat
if (inClassifier)
{
    // Get the current penetration
double currPen =
    classifier.penetrationDepth(par["y"]);

    // if we are deeper than ever, update the max depth
    if (currPen > maxPen)
    {
        maxPen = currPen;
    }
}

// Now we can record the stats
maxPenetration = maxPen;
residenceTime = outTime - inTime;
if (enteredDeep)
{
    deepResidenceTime = outTime - inDeepTime;
}
else
{
    deepResidenceTime = 0;
}

public int getDeepResidenceTime()
{
    return deepResidenceTime;
}

public int getResidenceTime()
{
    return residenceTime;
}

public double GetMaxPen()
{
    return maxPenetration;
}

public void addData()
{
    dataStore.Insert(0, null);
}

public void setTimeStep(int i)
{
    timeStep = i;
}
if (timeStep > age() - 1)
{
    addData();
}
if (dataStore[timeStep] != null)
{
    type = dataStore[timeStep]["type"] + ";
    dWall = dataStore[timeStep]["dWall"];
    x = dataStore[timeStep]["x"];
    y = dataStore[timeStep]["y"];
    z = dataStore[timeStep]["z"];
    pRadius = dataStore[timeStep]["Particle Radius"];
    mass = dataStore[timeStep]["mass"];
    speed = dataStore[timeStep]["speed"];
    wallAt = dataStore[timeStep]["wallAt"];
}

public double ThreeDX(double a, double b, double c, double ol)
{
    double xx = Math.Cos(c) * Math.Cos(a) - Math.Sin(c) * Math.Sin(a) *
    Math.Sin(ol);
    double yx = -Math.Cos(c) * Math.Sin(a) - Math.Sin(c) * Math.Cos(a) *
    Math.Sin(ol);
    double zx = Math.Sin(c) * Math.Cos(ol);
    return xx * x + yx * (y - ol) + zx * z;
}

public double ThreeDY(double a, double b, double c, double ol)
{
    double xy = Math.Cos(c) * Math.Sin(a) * Math.Sin(b) +
    Math.Sin(c) * Math.Cos(ol);
    double yy = Math.Cos(c) * Math.Cos(a) * Math.Sin(b) -
    Math.Sin(c) * Math.Sin(ol);
    double zy = Math.Cos(c) * Math.Cos(ol);
    return xy * x + yy * (y - ol) + zy * z;
}

public Dictionary<string, double> properties()
{
    return dataStore[timeStep];
}

public Dictionary<string, double> properties(int i)
{
    setTimeStep(i);
    return dataStore[timeStep];
}

public int age()
{
    return dataStore.Count;
}
static class Program
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new ClassiParseForm());
    }
}

public partial class ReverseEngineeringFilePicker : Form
{
    public const int COAL = 2;
    public const int WOOD = 1;

    private List<string> gmvs;
    private bool error = true;
    private WallBoundary classifier;

    public List<Particle> particles { get; }
    private List<Label> fileLabels;
    public List<string> fileNames { get; }

    public ReverseEngineeringFilePicker(WallBoundary wb)
    {
        classifier = wb;
        InitializeComponent();
        gmvs = new List<string>();
        fileLabels = new List<Label>();
        fileNames = new List<string>();
        particles = new List<Particle>();
    }

    private void resetFilePicker()
    {
        gmvs.Clear();
        fileLabels.Clear();
        particles.Clear();
    }

    private void multiFilePickerButton_Click(object sender, EventArgs e)
{ // Scan first file, make a List<Particle> exitParticles.
  // Scan subsequent files, finding Particle objects with a matching PID.
  // When a Particle in exitParticles is identified, use Particle.addData to add the data to that particle.
  // For any particle that isn't in the file, add a null data object to the particle data
  // so that the history of all the Particle objects will be the same length

  // Remove any currently found files.
  groupBoxFileLabels.Controls.Clear();
  gmvs.Clear();
  fileLabels.Clear();

  //create an 'OpenFileDialog' to let the user select the particle data file
  OpenFileDialog openDialog = new OpenFileDialog();
  openDialog.Title = "Open Text File";
  openDialog.Filter = "VARS files|*.vars";

  //if the user selects a file (and doesn't press cancel)
  if (openDialog.ShowDialog() == DialogResult.OK)
    {
      try
      {
        //Get the file path from the selected file.
        string fileName = openDialog.FileName;
        string[] splitFile = fileName.Split('\\');

        //create the filepath and filename from the split array
        string filePath = "";
        for (int i = 0; i < splitFile.Length; i++)
          {
            if (i == splitFile.Length - 1)
            {
              fileName = splitFile[i];
            }
            else
            {
              filePath += splitFile[i] + "\\";
            }
          }

        //Get the rest of the files in the folder. Create a List of the gmv file names. Create a label for the file.
        string[] files = Directory.GetFiles(filePath);
        int label_y = 21;
        const int LABEL_SPACING = 18;
        const int LABEL_X = 9;
        foreach (string file in files)
```csharp
{
    string[] s = file.Split('\\');
    string name = s[s.Length - 1];

    if (Regex.IsMatch(name, "particles_Gmv.*.vars"))
    {
        gmvs.Add(file);
        fileNames.Add(name);

        Label l = new Label();
        l.Text = name;
        l.AutoSize = true;
        l.Location = new Point(LABEL_X, label_y);
        label_y += LABEL_SPACING;
        l.Size = new Size(120, 13);

        fileLabels.Add(l);
        groupBoxFileLabels.Controls.Add(l);
    }
}

//update the GUI with the number of files found. Show their names.

int fileCount = gmvs.Count();

labelFilesFound.Visible = true;
labelFilesFound.Text = fileCount + " Files Found. Begin Upload?";

buttonBeginUpload.Visible = true;

} catch (Exception ex)
{
    MessageBox.Show("Error: Could not read file from disk. Original error: " + ex.Message);
    error = true;
}
}

public bool wasSuccessful()
{
    return !error;
}

private void buttonBeginUpload_Click(object sender, EventArgs e)
{
    buttonBeginUpload.Enabled = false;
    multiFilePickerButton.Enabled = false;

    // this will keep track of the file that is being scanned, for error output.
    string currentFile = "";
```
// remove all particles in the current list
particles.Clear();

// Determine the 'oldest' GMV file. Select that one and remove it from 'gmvs'
string lastFile = gmvs[gmvs.Count() - 1];
gmvs.Remove(lastFile);

// set particles equal to particlesInExitArea
particles.AddRange(particlesInExitArea(lastFile));

List<string> exitPIDs = new List<string>();

// make a list of the PIDs of the exit particles
foreach (Particle p in particles)
{
    exitPIDs.Add(p.PID + "");
}

// loop through the remaining gmvs from last to first, updating each particle's history.
for (int i = gmvs.Count() - 1; i >= 0; i --)
{
    currentFile = gmvs[i];
    Console.WriteLine("Scanning " + currentFile);
    bool added = updateParticleHistory(currentFile, exitPIDs, gmvs.Count() - i + 1);
    if (!added) break;
}

// Now that all particles have been updated with their time data, calculate the time properties
foreach (Particle p in particles)
{
    p.calculateTimeProperties(classifier);
}

// particles were updated successfully. Return to main window for processing.
Close();
}

private bool updateParticleHistory(string file, List<string> exitPs, int age)
{
    Console.WriteLine("Age is: " + age);
    string[] s = file.Split('\\');
    string currentFile = s[s.Length - 1];
    //set the cursor to the wait cursor while the file is read
    Cursor.Current = Cursors.WaitCursor;
    //
/we're going to keep the progress bar updated. Its max length is the number of bytes in the file we're reading, and we'll update it as the file is read.

FileInfo fi = new FileInfo(file);
progressBar.Maximum = (int)fi.Length / 1024 + 1024;
//converted to KB, plus an extra one so we don't exceed the max
progressBar.Visible = true; //show the progress bar during loading
progressStatus.Text = "Reading: " + currentFile; //set the progress status
Application.DoEvents();

//and this double will keep track of how many kilobytes have been read so far
double kilobytes = 0.0;

try
{
  //we're going to do this with a buffered stream to handle really big files.
  //That way the file doesn't have to be loaded all at once.
  using (FileStream fs = File.Open(file, FileMode.Open,
 FileAccess.Read, FileShare.ReadWrite))
  using (BufferedStream bs = new BufferedStream(fs))
  using (StreamReader sr = new StreamReader(bs))
  {
    //loop through each line of the file
    string line;
    while ((line = sr.ReadLine()) != null)
    {
      //add to the total number of read kilobytes
      kilobytes += (double)line.Length / 1024;

      //when we've reached 20 KB, trigger the progress bar and start the count over
      if (kilobytes >= 20)
      {
        kilobytes -= 20;
        try
        {
          progressBar.Value += 20;
        }
        catch
        {
          progressBar.Value = progressBar.Maximum - 1;
        }
        Application.DoEvents();
      }
      //don't read the lines that start with '#' - these are the comments at the top
      if (!line[0] == '#')
      {
        //add to the total number of read kilobytes
        kilobytes += (double)line.Length / 1024;

        //when we've reached 20 KB, trigger the progress bar and start the count over
        if (kilobytes >= 20)
        {
          kilobytes -= 20;
          try
          {
            progressBar.Value += 20;
          }
          catch
          {
            progressBar.Value = progressBar_MAXIMUM - 1;
          }
          Application.DoEvents();
        }
        //don't read the lines that start with '#' - these are the comments at the top
        if (!line[0] == '#')
        {
          //add to the total number of read kilobytes
          kilobytes += (double)line.Length / 1024;

          //when we've reached 20 KB, trigger the progress bar and start the count over
          if (kilobytes >= 20)
          {
            kilobytes -= 20;
            try
            {
              progressBar.Value += 20;
            }
            catch
            {
              progressBar.Value = progressBar.Maximum - 1;
            }
            Application.DoEvents();
          }
        }
      }
    }
  }
}

256
//split the string on whitespace and remove any empty entries
string[] l = line.Trim().Split(new[] { " " }, StringSplitOptions.RemoveEmptyEntries);

//if it didn't split right, it must be separated by tabs instead of spaces. Split it again.
if (l.Length == 1)
    l = line.Trim().Split('t');

//if the line successfully split, parse the data from it.
if (l.Length >= 1)
{
    // Find the coordinates of the particle
    double pX = double.Parse(l[l.Length - 3]) - classifier.offsetX;
    double pY = double.Parse(l[l.Length - 2]);
    double pZ = double.Parse(l[l.Length - 1]) - classifier.offsetZ;
    double pR = Math.Sqrt(pX * pX + pZ * pZ);

    //if the particle is one of the exit particles, update its history
    if (exitPs.Contains(l[2]))
    {
        //define wallAtY and speed values for the particle on this line
        double way = classifier.wallAtY(pY);
        double pS = double.Parse(l[7]);

        particles[exitPs.IndexOf(l[2])].addData(l[3], l[4], pX, pY, pZ, way, pR, pS);

        Console.WriteLine(l[2] + " found!");
    }

    // Otherwise, if 'Scan All GMVs for exit particles' is selected, check if the particle is in the exit zone.
    if (checkBoxScanAll.Checked)
    {
        //if the particle is within the bounds defined by the WallBoundary 'classifier', and is a large particle, add it to the list
        if (classifier.isInTopExitZone(pY, pR) && double.Parse(l[3]) >= classifier.largeRad)
        {
            //define wallAtY and speed values for the particle on this line
            double way = classifier.wallAtY(pY);
            double pS = double.Parse(l[7]);
        }
    }
}
constructor with only the PID.  

Particle p = new Particle(l[2]);  

// Since this isn't the 'first' point, add blank time steps to the particle up to this point in time  
for (int i = 0; i < age; i++)  
{
    p.addData();
}

// Now add the rest of the data to the right time  
way, pR, pS);

p.addData(l[3], l[4], pX, pY, pZ,  

// Add this particle's PID to the list of exit particles for future scans  
exitPs.Add(p.PID);  

// Add this particle to the list of particles.  
particles.Add(p);

}

}

}

}

}

// now that the entire file has been read, update any particles that didn't appear in this file with 'null'  
foreach (Particle p in particles)  
{
    if (p.age() < age)  
    {
        p.addData();
    }
}

//set the cursor back to normal  
Cursor.Current = Cursors.Default;

//reset the progress status  
progressBar.Visible = false;  
progressBar.Value = 0;  
progressStatus.Text = "";
}
}

catch (Exception ex)
{
    //if there is an error in reading the file, display an error message and put the cursor back to normal.  
    Cursor.Current = Cursors.Default;
    progressBar.Visible = false; //also hide the progress bar
progressBar.Value = 0; //reset the value and the progress status
progressStatus.Text = "";
MessageBox.Show("Error: Could not read file: " + currentFile + " from disk. Original error: " + ex.Message);
resetFilePicker();
return false;
}
return true;
}

private List<Particle> particlesInExitArea(string file)
{
List<Particle> fileParticles = new List<Particle>();

string[] s = file.Split('\');
string currentFile = s[s.Length - 1];

//set the cursor to the wait cursor while the file is read
Cursor.Current = Cursors.WaitCursor;

//we're going to keep the progress bar updated. Its max length is the number of bytes in the file we're reading, and we'll update it as the file is read.
FileInfo fi = new FileInfo(file);
progressBar.Maximum = (int)fi.Length / 1024 + 1024;
//converted to KB, plus an extra one so we don't exceed the max
progressBar.Visible = true; //show the progress bar during loading
progressStatus.Text = "Reading: " + currentFile; //set the progress status
Application.DoEvents();

//and this double will keep track of how many kilobytes have been read so far
double kilobytes = 0.0;

try
{
    //we're going to do this with a buffered stream to handle really big files.
    //That way the file doesn't have to be loaded all at once.
    using (BufferedStream bs = new BufferedStream(fs))
    using (StreamReader sr = new StreamReader(bs))
    {
        //loop through each line of the file
        string line;
        while ((line = sr.ReadLine()) != null)
        {
        
259
add to the total number of read kilobytes
kilobytes += (double)line.Length / 1024;

// when we've reached 20 KB, trigger the progress bar and start the count over
if (kilobytes >= 20)
{
    kilobytes -= 20;
    try
    {
        progressBar.Value += 20;
    }
    catch
    {
        progressBar.Value = progressBar.Maximum - 1;
    }
    Application.DoEvents();
}

// don't read the lines that start with '#' - these are the comments at the top
if (!(line[0] == '#'))
{
    // split the string on whitespace and remove any empty entries
    string[] l = line.Trim().Split(new[] { " " }, StringSplitOptions.RemoveEmptyEntries);

    // if it didn't split right, it must be separated by tabs instead of spaces. Split it again.
    if (l.Length == 1)
        l = line.Trim().Split('\\t');

    // if the line successfully split, parse the data from it.
    if (l.Length >= 1)
    {
        // define the x, y, z, radius, and wallAtY values for the particle on this line
        double pX = double.Parse(l[l.Length - 3]) - classifier.offsetX;
        double pY = double.Parse(l[l.Length - 2]);
        double pZ = double.Parse(l[l.Length - 1]) - classifier.offsetZ;
        double pR = Math.Sqrt(pX * pX + pZ * pZ);
        double way = classifier.wallAtY(pY);
        double pS = double.Parse(l[7]);

        // if the particle is within the bounds defined by the WallBoundary 'classifier', and is a large particle, add it to the list
if (pY <= classifier.exitYHi && pY >= classifier.exitYLo && (classifier.exitR - pR) > 0 && double.Parse(l[3]) >= classifier.largeRad)
{
    Particle p = new Particle(l[2], l[3], l[4], pX, pY, pZ, way, pR, pS);
    fileParticles.Add(p);
}

//set the cursor back to normal
Cursor.Current = Cursors.Default;
//reset the progress status
progressBar.Visible = false;
progressBar.Value = 0;
progressStatus.Text = "";

} catch (Exception ex)
{
    //if there is an error in reading the file, display an error message and put the cursor back to normal.
    Cursor.Current = Cursors.Default;
    progressBar.Visible = false; //also hide the progress bar
    progressBar.Value = 0; //reset the value and the progress status
    progressStatus.Text = "";
    MessageBox.Show("Error: Could not read file: " + currentFile + " from disk. Original error: " + ex.Message);
    resetFilePicker();
    return null;
}
return fileParticles;

private void textBoxLargeRad_TextChanged(object sender, EventArgs e)
{
    //try to set the the classifier 'large rad' parameter to the contents of the text box.
    try
    {
        double largeRadVal = double.Parse(textBoxLargeRad.Text);
        if (largeRadVal < 0)
        {
            throw new Exception();
        }
        classifier.largeRad = largeRadVal;
    }
    catch
    {
}
// if there is an error, set it to 200
textBoxLargeRad.Text = 200 + "";
classifier.largeRad = 200;
}
}

partial class ReverseEngineeringFilePicker
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
based.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.label1 = new System.Windows.Forms.Label();
        this.groupBoxFileLabels = new System.Windows.Forms.GroupBox();
        this.labelFilesFound = new System.Windows.Forms.Label();
        this.buttonBeginUpload = new System.Windows.Forms.Button();
        this.progressBar = new System.Windows.Forms.ProgressBar();
        this.progressStatus = new System.Windows.Forms.Label();
        this.textBoxLargeRad = new System.Windows.Forms.TextBox();
        this.label2 = new System.Windows.Forms.Label();
        this.label3 = new System.Windows.Forms.Label();
        this.checkBoxScanAll = new System.Windows.Forms.CheckBox();
        this.SuspendLayout();
        
        // label1
        // this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(12, 54);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(120, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Select the first GMV file:";

//
// multiFilePickerButton
//
this.multiFilePickerButton.Anchor =
this.multiFilePickerButton.Location = new System.Drawing.Point(174, 50);
this.multiFilePickerButton.Name = "multiFilePickerButton";
this.multiFilePickerButton.Size = new System.Drawing.Size(82, 20);
this.multiFilePickerButton.TabIndex = 1;
this.multiFilePickerButton.Text = "Choose File";
this.multiFilePickerButton.UseVisualStyleBackColor = true;
this.multiFilePickerButton.Click += new System.EventHandler(this.multiFilePickerButton_Click);

//
// groupBoxFileLabels
//
this.groupBoxFileLabels.Anchor =
this.groupBoxFileLabels.Location = new System.Drawing.Point(13, 126);
this.groupBoxFileLabels.Name = "groupBoxFileLabels";
this.groupBoxFileLabels.Size = new System.Drawing.Size(243, 230);
this.groupBoxFileLabels.TabStop = false;
this.groupBoxFileLabels.Text = "Files Found";

//
// labelFilesFound
//
this.labelFilesFound.AutoSize = true;
this.labelFilesFound.Location = new System.Drawing.Point(12, 80);
this.labelFilesFound.Name = "labelFilesFound";
this.labelFilesFound.Size = new System.Drawing.Size(61, 13);
this.labelFilesFound.TabIndex = 3;
this.labelFilesFound.Text = "Files Found";
this.labelFilesFound.Visible = false;

//
// buttonBeginUpload
//
this.buttonBeginUpload.Anchor =
this.buttonBeginUpload.Location = new System.Drawing.Point(174, 76);
this.buttonBeginUpload.Name = "buttonBeginUpload";
this.buttonBeginUpload.Size = new System.Drawing.Size(82, 21);
this.buttonBeginUpload.TabIndex = 4;
this.buttonBeginUpload.Text = "Begin Upload";
this.buttonBeginUpload.UseVisualStyleBackColor = true;
this.buttonBeginUpload.Visible = false;
this.buttonBeginUpload.Click += new System.EventHandler(this.buttonBeginUpload_Click);
//
// progressBar
//
this.progressBar.Location = new System.Drawing.Point(15, 382);
this.progressBar.Name = "progressBar";
this.progressBar.Size = new System.Drawing.Size(239, 20);
this.progressBar.TabIndex = 21;
this.progressBar.Visible = false;
//
// progressStatus
//
this.progressStatus.AutoSize = true;
this.progressStatus.Location = new System.Drawing.Point(12, 359);
this.progressStatus.Name = "progressStatus";
this.progressStatus.Size = new System.Drawing.Size(0, 13);
this.progressStatus.TabIndex = 22;
//
// textBoxLargeRad
//
this.textBoxLargeRad.Location = new System.Drawing.Point(15, 25);
this.textBoxLargeRad.Name = "textBoxLargeRad";
this.textBoxLargeRad.Size = new System.Drawing.Size(46, 20);
this.textBoxLargeRad.TabIndex = 23;
this.textBoxLargeRad.Text = "200";
this.textBoxLargeRad.TextChanged += new System.EventHandler(this.textBoxLargeRad_TextChanged);
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(67, 28);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(43, 13);
this.label2.TabIndex = 24;
this.label2.Text = "microns";

// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(12, 9);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(231, 13);
this.label3.TabIndex = 25;
this.label3.Text = "Select Exit Zone Particles with a Radius
above:";

// checkBoxScanAll
//
this.checkBoxScanAll.AutoSize = true;
this.checkBoxScanAll.Location = new System.Drawing.Point(15,
103);
this.checkBoxScanAll.Name = "checkBoxScanAll";
this.checkBoxScanAll.Size = new System.Drawing.Size(250, 17);
this.checkBoxScanAll.TabStop = false;
this.checkBoxScanAll.Text = "Scan ALL GMVs for Exit Particles
(takes longer)";
this.checkBoxScanAll.UseVisualStyle = true;

// ReverseEngineeringFilePicker
//
this.AcceptButton = this.multiFilePickerButton;
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.ClientSize = new System.Drawing.Size(266, 408);
this.Controls.Add(this.checkBoxScanAll);
this.Controls.Add(this.label3);
this.Controls.Add(this.label2);
this.Controls.Add(this.textBoxLargeRad);
this.Controls.Add(this.progressStatus);
this.Controls.Add(this.progressBar);
this.Controls.Add(this.buttonBeginUpload);
this.Controls.Add(this.labelFilesFound);
this.Controls.Add(this.groupBoxFileLabels);
this.Controls.Add(this.multiFilePickerButton);
this.Controls.Add(this.label1);
this.MaximizeBox = false;
this.MinimizeBox = false;
this.Name = "ReverseEngineeringFilePicker";
this.ShowInTaskbar = false;
this.Text = "Multiple File Picker";
this.TopMost = true;
this.ResumeLayout(false);
this.PerformLayout();
public class WallBoundary
{
    //WallBoundary has 5 members and 1 function.
    //wallPrecision represents which order of polynomial is used in
    //estimating the wall radius for a given height
    //offsetX is the distance that the center of the classifier is
    //from 0 in the x-direction
    //offsetZ is the distance that the center of the classifier is
    //from 0 in the z-direction
    //topY is the y-value of the top of the classifier (the top of the
    //volume to be considered)
    //bottomY is the y-value of the bottom of the classifier (the
    //bottom of the volume to be considered)
    //wallAtY() returns the 'radius' of the wall for a given Y. This
    //is estimated based off of functions that fit the
    //    wall 'radius' to the height. Different degrees of accuracy
    //can be acheived (at the cost of processing time)
    //    by using first through sixth order polynomials. The default
    //    (set in ClassiParseForm.cs) is 22.

    public int wallPrecision;

    public double offsetX { get; }
    public double offsetZ { get; }

    public double topY { get; set; }
    public double bottomY { get; }

    public double massH { get; set; }
    public double massL { get; set; }
    public double massA { get; set; }
    public double radiusH { get; set; }
    public double radiusL { get; set; }
    public double radiusA { get; set; }
    public double speedH { get; set; }
    public double speedA { get; set; }
    public double count { get; set; }
}
public double exitX { get; }
public double exitYHi { get; }
public double exitYLo { get; }
public double exitZ { get; }
public double exitR { get; }

public double largeRad { get; set; }

private double y1, y2, y3, y4, y5;

// this are for drawing the classifier in 3d
private double[] angles;

// These are for drawing the classifier in 2d. Top and bottom x and z coordinates.
private const int NUM_DRAWING_POINTS = 100;
private double[] txs;
private double[] tzs;
private double[] bxs;
private double[] bzs;

public double[] outlineTopXs { get; set; }
public double[] outlineTopYs { get; set; }
public double[] outlineBotXs { get; set; }
public double[] outlineBotYs { get; set; }

public WallBoundary (int w)
{
    wallPrecision = w;
    offsetX = 0.53525;
    offsetZ = 0.53525;

    topY = 1.1;
    bottomY = 0.618595;

    y1 = topY;
    y2 = (3 * topY + bottomY) / 4;
    y3 = (topY + bottomY) / 2;
    y4 = (topY + 3 * bottomY) / 4;
    y5 = bottomY;

    exitX = 0.0;
    exitZ = 0.0;
    exitR = 0.127;
    exitYHi = 1.45166;
    exitYLo = 1.28789;

    //FIXME: Add option in reverse engineering file picker for filtering out by size to adjust this
    largeRad = 200; // microns

txs = new double[NUM_DRAWING_POINTS];
tzs = new double[NUM_DRAWING_POINTS];
bxs = new double[NUM_DRAWING_POINTS];
bzs = new double[NUM_DRAWING_POINTS];

outlineTopXs = new double[NUM_DRAWING_POINTS];
outlineTopYs = new double[NUM_DRAWING_POINTS];
outlineBotXs = new double[NUM_DRAWING_POINTS];
outlineBotYs = new double[NUM_DRAWING_POINTS];

angles = new double[3];

setupOutline();
setAngles(0, 0, 0, 0);
}

public double threeDX(double x, double y, double z, double ol)
{
    double xx = Math.Cos(angles[2]) * Math.Cos(angles[0]) -
                Math.Sin(angles[2]) * Math.Sin(angles[0]) * Math.Sin(angles[1]);
    double yx = -Math.Cos(angles[2]) * Math.Sin(angles[0]) -
                Math.Sin(angles[2]) * Math.Cos(angles[0]) * Math.Sin(angles[1]);
    double zx = -Math.Sin(angles[2]) * Math.Cos(angles[1]);
    return xx * x + yx * (y - ol) + zx * z;
}

public double threeDY(double x, double y, double z, double ol)
{
    double xy = Math.Cos(angles[2]) * Math.Sin(angles[0]) *
                Math.Sin(angles[1]) + Math.Sin(angles[2]) * Math.Cos(angles[0]);
    double yy = Math.Cos(angles[2]) * Math.Cos(angles[0]) * Math.Sin(angles[1]) -
                Math.Sin(angles[2]) * Math.Sin(angles[0]);
    double zy = Math.Cos(angles[2]) * Math.Cos(angles[1]);
    return xy * x + yy * (y - ol) + zy * z;
}

public double[][] extremePosXs()
{
    double x1 = outlineTopXs.Max();
    int x1i = outlineTopXs.ToList().IndexOf(x1);
    double y1 = outlineTopYs[x1i];

    double x2 = outlineBotXs.Max();
    int x2i = outlineBotXs.ToList().IndexOf(x2);
    double y2 = outlineBotYs[x2i];

    return new double[][] { new double[] { x1, x2 }, new double[] { y1, y2 } };
}

public double[][] extremeNegXs()
{
    double x1 = outlineTopXs.Min();
    int x1i = outlineTopXs.ToList().IndexOf(x1);
    double y1 = outlineTopYs[x1i];

    double x2 = outlineBotXs.Min();
    int x2i = outlineBotXs.ToList().IndexOf(x2);
    double y2 = outlineBotYs[x2i];

    return new double[][] { new double[] { x1, x2 }, new double[] { y1, y2 } };
}
double x2 = outlineBotXs.Min();
int x2i = outlineBotXs.ToList().IndexOf(x2);
double y2 = outlineBotYs[x2i];

return new double[][] { new double[] { x1, x2 }, new double[] { y1, y2 } };
}

public double[][] extremePosYs()
{
    double y1 = outlineTopYs.Max();
    int y1i = outlineTopYs.ToList().IndexOf(y1);
double x1 = outlineTopXs[y1i];

double y2 = outlineBotYs.Max();
int y2i = outlineBotYs.ToList().IndexOf(y2);
double x2 = outlineBotXs[y2i];

return new double[][] { new double[] { x1, x2 }, new double[] { y1, y2 } };
}

public double[][] extremeNegYs()
{
    double y1 = outlineTopYs.Min();
    int y1i = outlineTopYs.ToList().IndexOf(y1);
double x1 = outlineTopXs[y1i];

double y2 = outlineBotYs.Min();
int y2i = outlineBotYs.ToList().IndexOf(y2);
double x2 = outlineBotXs[y2i];

return new double[][] { new double[] { x1, x2 }, new double[] { y1, y2 } };
}

public void setAngles(double a, double b, double c, double ol)
{
    angles[0] = a;
    angles[1] = b;
    angles[2] = c;

    for (int i = 0; i < NUM_DRAWING_POINTS; i++)
    {
        outlineTopXs[i] = threeDX(txs[i], topY, tzs[i], ol);
        outlineTopYs[i] = threeDY(txs[i], topY, tzs[i], ol);
        outlineBotXs[i] = threeDX(bxs[i], bottomY, bzs[i], ol);
        outlineBotYs[i] = threeDY(bxs[i], bottomY, bzs[i], ol);
    }
}

private void setupOutline()
{
    double topr = wallAtY(topY);
double botr = wallAtY(bottomY);

int nps = NUM_DRAWING_POINTS / 2;

for (int i = 0; i <= nps; i++)
{
    txs[i] = -topr + (2 * topr) / (nps - 1) * i;
    bxs[i] = -botr + (2 * botr) / (nps - 1) * i;
    tzs[i] = Math.Sqrt(Math.Abs(Math.Pow(topr, 2) -
            Math.Pow(txs[i], 2)));
    bzs[i] = Math.Sqrt(Math.Abs(Math.Pow(botr, 2) -
            Math.Pow(bxs[i], 2)));
}

for (int i = 0; i < nps; i++)
{
    txs[nps + i] = topr - (2 * topr) / (nps - 1) * i;
    bxs[nps + i] = botr - (2 * botr) / (nps - 1) * i;
    tzs[nps + i] = -Math.Sqrt(Math.Abs(Math.Pow(topr, 2) -
            Math.Pow(txs[i], 2)));
    bzs[nps + i] = -Math.Sqrt(Math.Abs(Math.Pow(botr, 2) -
            Math.Pow(bxs[i], 2)));
}

// returns the boundary conditions (top and bottom) for the
selected sections
public double[] YBounds(bool sec1, bool sec2, bool sec3, bool sec4)
{
    double t = y5;
    double b = y1;
    if (sec1)
    {
        t = y1;
        b = y2;
    }
    if (sec2)
    {
        if (t < y2)
            t = y2;
        if (b > y3)
            b = y3;
    }
    if (sec3)
    {
        if (t < y3)
            t = y3;
        if (b > y4)
            b = y4;
    }
    if (sec4)
if (t < y4)
    t = y4;
b = y5;
return new double[] {b, t};

public double wallAtY(double y)
{
    switch (wallPrecision)
    {
    case 11:
        return wallAtYLinear(y);
    case 21:
        return wallAtYPoly(y);
    case 12:
        return wallAtYLinear2(y);
    default:
        return wallAtYPoly2(y);
    }
}

private double wallAtYPoly2(double y) //polynomial fit to the wall
data v2
{
    return r;
}

private double wallAtYLinear2(double y) //linear fit to the wall
data v2
{
    double r = 0.35250681 * y - 0.11252744;
    return r;
}

private double wallAtYPoly(double y) //polynomial fit to the original wall data
{
    return r;
}

private double wallAtYLinear(double y) //linear fit to the original wall data
{
    double r = 0.366474 * y - 0.125207;
    return r;
}
// Determines if the x, y, z coordinates are inside of the classifier
public bool isInClassifier(double x, double y, double z)
{
    // First check the y
    if (y > topY || y < bottomY)
    return false;

    // Next find the radius and compare it to wallAtY
    double r = Math.Sqrt(Math.Pow(x, 2) + Math.Pow(z, 2));
    if (r > wallAtY(y))
    return false;

    // If the coordinates pass both tests, they are inside the particle
    return true;
}

public bool isInVanes(double x, double y, double z)
{
    // In this case, the particle needs to be above the top y and within the top radius
    // First check the y
    if (y < topY)
    return false;

    // Next find the radius and compare it to wallAtY for the top y
    double r = Math.Sqrt(Math.Pow(x, 2) + Math.Pow(z, 2));
    if (r > wallAtY(topY))
    return false;

    // If the coordinates pass both tests, they are inside the vanes
    return true;
}

public bool isInTopExitZone(double pY, double pR)
{
    return pY <= exitYHi && pY >= exitYLo && (exitR - pR) > 0;
}

// Assume that this function will only be called for particles inside the classifier.
public double penetrationDepth(double y)
{
    return topY - y;
}
Appendix B: Fabrication drawings of ash deposit probe

The drawings of the designed ash deposit probe as well as the steel frame are presented in this section. It should be noted that there might be some minor changes during construction, which are not reflected in the drawings. Meanwhile, the drawings are still valid.
<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>DESCRIPTION</th>
<th>MATERIAL</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pipe</td>
<td>2&quot; SCH 10, 316SS</td>
<td>1</td>
</tr>
</tbody>
</table>

DETAIL A

SCALE 1 : 12

Fabricate 1
DETIAL K
SCALE 1:8

5 15/16"

Number of threads per inch: 11 1/2

SECTION N-N

DETIAL L
SCALE 1:8

22 1/4"

Number of threads per inch: 11 1/2

9 holes in 6 rows

DETAIL P
SCALE 1:8

1 1/2"

Ø 3/8"

SECTION N-N

Fabricate 1

ITEM NO. | DESCRIPTION | MATERIAL | QTY.
--- | --- | --- | ---
2 | Pipe | 1" SCH 40, 316 SS | 1
<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>DESCRIPTION</th>
<th>MATERIAL</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Cap</td>
<td>SS</td>
<td>1</td>
</tr>
</tbody>
</table>

Number of threads per inch: 11 1/2

Fabricate 1
We may use a metal sheet instead of flange.
The least diameter of the spring is 1.315".

Fabricate 1

SOLIDWORKS Educational Product. For Instructional Use Only.
ITEM NO. | DESCRIPTION   | MATERIAL  | QTY.
----------|---------------|-----------|-----
8         | T Connection  | Aluminum  | 1   

NPT tapered threads used at this section.

NPT tapered threads used at this half section.

Number of threads per inch: 11 1/2

SECTION AL-AL

Fabricate 1
<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>DESCRIPTION</th>
<th>MATERIAL</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Connection-0B</td>
<td>SS</td>
<td>1</td>
</tr>
</tbody>
</table>

Fabricate 1

1-5/16-12 Thread

SECTION AM-AM

Ø 2 3/8"

Ø 1 5/16"

1 1/2"

1/2"

2"
ITEM NO. | DESCRIPTION | MATERIAL | QTY.
--- | --- | --- | ---
11 | Compression Hanger Assembly | SS | 1

Fabricate 1

直径 1.5 英寸

高 22 英寸
Appendix C: Summary of the PI data of Hunter#3

A summary of the plant instrumentations (PI) data of Hunter#3 furnace is presented at this section.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Time period (minute)</th>
<th>T (F)</th>
<th>P (psi)</th>
<th>Mass Flow (klb/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Air</td>
<td>30</td>
<td>377.19</td>
<td>--</td>
<td>869.80</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>383.11</td>
<td>--</td>
<td>874.35</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>406.08</td>
<td>--</td>
<td>883.55</td>
</tr>
<tr>
<td>Secondary Air</td>
<td>30</td>
<td>521.01</td>
<td>--</td>
<td>1529.30</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>521.68</td>
<td>--</td>
<td>1538.05</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>517.44</td>
<td>--</td>
<td>1512.52</td>
</tr>
<tr>
<td>OFA</td>
<td>30</td>
<td>521.01</td>
<td>--</td>
<td>1162.71</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>521.68</td>
<td>--</td>
<td>1165.64</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>517.44</td>
<td>--</td>
<td>1182.29</td>
</tr>
<tr>
<td>Pulv. Solid Fuel</td>
<td>30</td>
<td>Ambient T &amp; P</td>
<td>388.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td>393.81</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td></td>
<td></td>
<td>406.65</td>
</tr>
<tr>
<td>FW</td>
<td>30</td>
<td>483.86</td>
<td>2683.32</td>
<td>2703.86</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>484.09</td>
<td>2682.15</td>
<td>2704.84</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>482.45</td>
<td>2686.60</td>
<td>2688.91</td>
</tr>
<tr>
<td>Attmp-1</td>
<td>30</td>
<td>331.35</td>
<td>--</td>
<td>63.68</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>330.87</td>
<td>--</td>
<td>69.40</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>330.02</td>
<td>--</td>
<td>63.08</td>
</tr>
<tr>
<td>Attemp-2</td>
<td>30</td>
<td>331.35</td>
<td>--</td>
<td>88.69</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>330.87</td>
<td>--</td>
<td>75.64</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>330.02</td>
<td>--</td>
<td>92.39</td>
</tr>
<tr>
<td>Attemp-RH</td>
<td>30</td>
<td>330.48</td>
<td>--</td>
<td>145.55</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>330.53</td>
<td>--</td>
<td>179.38</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>329.27</td>
<td>--</td>
<td>82.00</td>
</tr>
<tr>
<td>RH-1</td>
<td>30</td>
<td>637.92</td>
<td>549.71</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>637.97</td>
<td>550.68</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>635.97</td>
<td>541.72</td>
<td>--</td>
</tr>
</tbody>
</table>
## Sufco Coal Combustion

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Time period (minute)</th>
<th>T (°F)</th>
<th>P (psi)</th>
<th>Mass Flow (klb/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Air</td>
<td>30</td>
<td>366.00</td>
<td>--</td>
<td>880.34</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>363.69</td>
<td>--</td>
<td>877.10</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>351.76</td>
<td>--</td>
<td>870.56</td>
</tr>
<tr>
<td>Secondary Air</td>
<td>30</td>
<td>527.64</td>
<td>--</td>
<td>1608.32</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>523.21</td>
<td>--</td>
<td>1567.06</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>517.08</td>
<td>--</td>
<td>1558.13</td>
</tr>
<tr>
<td>OFA</td>
<td>30</td>
<td>527.64</td>
<td>--</td>
<td>1158.21</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>523.21</td>
<td>--</td>
<td>1160.56</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>517.08</td>
<td>--</td>
<td>1164.00</td>
</tr>
<tr>
<td>Pulv. Coal</td>
<td>30</td>
<td></td>
<td></td>
<td>385.33</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td>381.79</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td></td>
<td></td>
<td>370.47</td>
</tr>
<tr>
<td>FW</td>
<td>30</td>
<td>484.19</td>
<td>2694.54</td>
<td>2773.15</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>483.21</td>
<td>2700.90</td>
<td>2770.97</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>481.78</td>
<td>2687.66</td>
<td>2716.80</td>
</tr>
<tr>
<td>Attemp-1</td>
<td>30</td>
<td>332.16</td>
<td>--</td>
<td>74.05</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>331.23</td>
<td>--</td>
<td>47.31</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>331.15</td>
<td>--</td>
<td>54.98</td>
</tr>
<tr>
<td>Attemp-2</td>
<td>30</td>
<td>332.16</td>
<td>--</td>
<td>43.31</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>331.23</td>
<td>--</td>
<td>99.09</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>331.15</td>
<td>--</td>
<td>78.37</td>
</tr>
<tr>
<td>Attemp-RH</td>
<td>30</td>
<td>331.80</td>
<td>--</td>
<td>209.31</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>331.43</td>
<td>--</td>
<td>60.45</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>330.40</td>
<td>--</td>
<td>94.19</td>
</tr>
<tr>
<td>RH-1</td>
<td>30</td>
<td>638.71</td>
<td>551.52</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>643.29</td>
<td>545.59</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>633.50</td>
<td>539.76</td>
<td>--</td>
</tr>
</tbody>
</table>
The numerical code of ash deposit rate is presented here. The data and parameters are described (commented) after % signs.

```
%% This is a code to calculate the ash deposition rate (gr/mm2) by Matlab
%% This code was developed by Seyedhassan Fakourian (sh.fakourian@gmail.com), with no liability
%% of use or application.
%% Ash deposition consists mostly of four mechanisms: Inertial Impaction, Thermophoresis,
%% Condensation, and Eddy Impaction
%% This code is able to calculate ash deposit thickness for three bins of size, which are
%% representative of PSD range.

close all; clc; clear;

% Introduction of the fuels

answer = questdlg('Fuel?', 'Fuel',
    'RH', 'Sufco1', 'Sufco2', 'RH');
    % 'RH', 'Sufco1', 'Sufco2', '20RH-80Suf1', 'PRB', '13RH-87PRB', 'Petcoke', 'Illinois',
    '60Illinois-40PRB', 'TorrWood', '50Torre-50Suf2');

switch answer
    case 'RH'
        Fuel = 1.0;
    case 'Sufco1'
        Fuel = 2.0;
    case 'Sufco2'
        Fuel = 3.0;
end

answer = questdlg('How long?', 'Time',...
    '30 min', '60 min', '120 min','120 min');

switch answer
    case '30 min'
        t_end = 1800;
    case '60 min'
        t_end = 3600;
    case '120 min'
        t_end = 7200;
end

T_set_point = 922;                                  % set point of coupon surface temperature (K)
Length = 2.9*2.54*0.01;                             % Length of the coupons (m)
d_c = 2.375*0.00254;                                % Probe Outer Diameter (m)
k = round(t_end/5);                                 % defining how many times in Max. this code
Thick_deposit = linspace(0,0.002,k);                % Thickness of ash deposit, Guess (m)
Resistance = zeros(1,k);                            % Ash deposit thermal resistance (m2K/W)
k_deposit = zeros(1,k);                             % Thermal conductivity of ash (W/m.K)
t = linspace(0,t_end,k);                            % Time period (seconds)
PSD1_w = 0.2;   PSD2_w = 0.60;   PSD3_w = 0.2;      % Proportion of each PSD bin

%% Defining three bins of fly ash particles, representative of PSD of fly ash particles
```
for PSD = 1:1:3;
    if Fuel==1.0
        \( D_p1 = 1.6e-6; \quad D_p2 = 7.77e-6; \quad D_p3 = 9.5e-5; \)
        if PSD==1
            \( \text{Diam}_p = D_p1; \)
            Flow = linspace(1.09e-04, (1.56e-03), 3); % kg/m3 fly ash
        \%
        \end{align*}
    \else\% Suco1
        \( D_p1 = 1.6e-6; \quad D_p2 = 7.77e-6; \quad D_p3 = 9.5e-5; \)
        if PSD==1
            \( \text{Diam}_p = D_p1; \)
            Flow = linspace(1.09e-04, (1.96e-03), 3); % kg/m3 fly ash
        \end{align*}
    \%
    \end{align*}
end
else\% Suco2
    \( D_p1 = 1.6e-6; \quad D_p2 = 7.77e-6; \quad D_p3 = 9.5e-5; \)
    if PSD==1
        \( \text{Diam}_p = D_p1; \)
        Flow = linspace(1.82e-04, (3.99e-03), 3); % kg/m3 fly ash
    \%
end
\%
T_ds = zeros(1,k); % It resets Deposit Surface Temp.
if Fuel==1.0
    T_gas = 1277; % Flue gas Temp. (K)
density_g = 357.09/T_gas; % Flue gas density (kg/m3)
viscosity_g = ((6.5592e-07)*T_gas^0.6081)/(1+54.714/T_gas); % Flue gas viscosity
V_gas = 0.92; % Flue gas velocity m/s
Mol_gas = 0.029; % Flue gas MW (kg/gmol) Calculated by Mathcad
k_deposit = 2.0; % Ash deposit thermal conductivity W/(m.K)
Porosity_deposit0 = 0.6; % the ratio of the area of the pores to the total area of the oxide
291
X_NaCl = 5.59e-05; \% Mole fraction of NaCl at the flue gas
X_KCl = 5.4e-05; \% Mole fraction of KCl at the flue gas

surface_tension = 0.26;
density_p = 1300; \% Fly ash particle density (kg/m^3)
T100 = 1850; T70 = 1810; T15 = 1165; T0 = 1140; \% Ash deposit melt fraction

Temperatures, which generated by FactSage
T_solid = 851; Tcut = 1266; \% Temperature window of molten salt of condensed vapor species, meaning that from T_solid to Tcut the condensation is sticky

```matlab
elseif Fuel==2.0
  \% Similar TO FUEL 1
  T_gas = 1247;
density_g = 357.09/T_gas;
viscosity_g = ((6.5592e-07)*T_gas^0.6081)/(1+54.714/T_gas);
V_gas = 0.73;
Mol_gas = 0.029;
k_deposit = 2.0;
Porosity_deposit0 = 0.6;
X_NaCl = 5.59e-05;
X_KCl = 5.4e-05;
surface_tension = 0.31;
density_p = 1300;
T100 = 1585; T70 = 1455; T15 = 1180; T0 = 1160;
T_solid = 851; Tcut = 1266;
end
```
```matlab
elseif Fuel==3
  \% Similar TO FUEL 1
  T_gas = 1173;
density_g = 357.09/T_gas;
viscosity_g = ((6.5592e-07)*T_gas^0.6081)/(1+54.714/T_gas);
V_gas = 0.6;
Mol_gas = 0.029;
k_deposit = 2.0;
Porosity_deposit0 = 0.6;
X_NaCl = 5.59e-05;
X_KCl = 5.4e-05;
surface_tension = 0.34;
density_p = 1300;
T100 = 1575; T70 = 1520; T15 = 1185; T0 = 1165;
T_solid = 851; Tcut = 1266;
end
```

```matlab
density_deposit_bulk = 550; \% Bulk density of ash deposit
density_true = density_deposit_bulk/(1-Porosity_deposit0); \% density of liquid phase of ash deposit
T_p = 1.0 * T_gas; \% we assume that the temperature of particle equals to flue gas temperature.

R_g = 8.314/Mol_gas; \% Gas Constant J/(kg.K)
k_p = 2.0; \% Particle and gas thermal conductivity
_k_gas = 15*viscosity_g*R_g/4;
W/(m.K)
k_solid = 3; \% Thermal conductivity of solid phase of ash W/m.K
Cp_gas = 1340; \% Cp of Flue Gas J/(kg.K) -- From Hansen
Pr = viscosity_g*Cp_gas/k_gas; \% Pr number of flue gas from Hansen's Thesis

Re = density_g * V_gas * d_c / viscosity_g; \% Re number of flue gas
if Re < 4 && Re > 0.4
  Nu = 0.989*(Re^0.33)*(Pr^(-1/3)); h = Nu * k_gas/d_c; dx = k_gas/h;
elseif Re < 40 && Re > 4
  Nu = 0.914*(Re^0.385)*(Pr^(-1/3)); h = Nu * k_gas/d_c; dx = k_gas/h;
elseif Re < 4000 && Re > 40
  Nu = 0.683*(Re^0.466)*(Pr^(-1/3)); h = Nu * k_gas/d_c; dx = k_gas/h;
elseif Re < 40000 && Re > 4000
  Nu = 0.193*(Re^0.618)*(Pr^(-1/3)); h = Nu * k_gas/d_c; dx = k_gas/h;
end
T_ds = zeros(1,k); \% Dynamic Ash deposit surface temperature (K)
T_ds(1) = T_set_point; \% Initial ash deposit surface temperature is equal to coupon surface set point temperature
```

292
emissivity(1) = (-0.3/700)*(T_ds(1)-673) + 0.6; % Radiation Emissivity

% Code runs less than 20 cycles to converge with the Error less than 0.01%. Meanwhile, I defined 1000 cycles
for m = 1:1:1e3

    for i = 2:1:length(Thick_deposit)
        Resistance = (Thick_deposit./k_deposit);   % Ash deposit thermal resistance
        if T_ds(i)==0
            xx0 = T_ds(i-1);
        else
            xx0 = T_ds(i);
        end
        emissivity(i) = (-0.3/700)*(xx0-673) + 0.6;
        fun = @(xx) Resistance(i)*h*(T_gas-xx)+(emissivity(i)*(5.67e-8)*Resistance(i)*((T_gas^4)-(xx.^4)))+T_set_point-xx;
        T_ds(i) = fzero(fun,xx0);
    end

%% Thermophoresis Impaction
V_th = zeros(1,k);                      % Thermophoretic velocity
Re_p = density_g * V_gas / Diam_p / viscosity_g;   % Re number of fly ash particle
conductivity_ratio = k_p / k_gas;
Pressure = 88000*1.05;                             % (pa) pressure of flue gas
mean_length = viscosity_g*(pi/(2*Pressure*density_g)^0.5);
Kn = 2*mean_length/Diam_p;              % Kn number
Ktc = 1.10;     alpha_e = 1; alpha_m = 1.0;  Ce = 2.17*(2-alpha_e)/alpha_e;     Cm = 1.13*(2-alpha_m)/alpha_m;          % From Henrik Lutro MSc Thesis
coeff = (-12*pi*Ktc*(1+(Ce*conductivity_ratio*Kn)))/(2+conductivity_ratio+2*Ce*conductivity_ratio*Kn);  % It is valid for all the ranges of Kn numbers
Cc = 1+Kn*(1.257+0.4*exp(-1.1*Kn));
for j = 1:1:k
    V_th(j) = coeff*3*viscosity_g*Cc*((T_ds(j)-T_gas)/dx)/(18*pi*density_g*T_gas*(1+0.15*(Re_p^0.687)));
end
for theta = 1:1:181
    delta(theta) = (T_ds(j)-T_set_point)*abs(cosd(theta-1))+(T_gas-T_ds(j)); % It is the temperature of the surface as a function of theta at time t(j). [(Theta=91 is the upstream stagnation point)]
end

%% Inertial Impaction
if Re_p < 0.01
    C_D = 24 / Re_p;                                   % Drag Coefficient for Stokes number
elseif Re_p>0.01 && Re_p<20
    C_D = (24 / Re_p)*[1+0.1315*(Re_p*(0.82-0.05*log(Re_p))));
elseif Re_p>20 && Re_p<260
    C_D = (24 / Re_p)*[1+0.1935*(Re_p^0.6305));
elseif Re_p>260 && Re_p<1500
    C_D = 10^((1.6425-1.1242*log(Re_p)+0.1558*(log(Re_p))^2));
end
a = 18*viscosity_g*C_D*Re_p/(density_p *24 *Diam_p^2);
b = V_gas;
c = 9.8*(density_p-density_g)/ density_p;
V_inertial(1) = V_gas;
for j = 2:1:k
    V_inertial(j) = ((t(j)-t(j-1))*(c+a*b)+V_inertial(j-1))/(1+(a*(t(j)-t(j-1))));
end

%% Condensation
\[ I_{\text{condens}} = Sh \times (\text{Diff}_v \times \text{Diff}_d)^{0.5} \times ((\text{Pressure}_v / T_{\text{gas}}) - \text{Pressure}_{\text{sat}} / T_{\text{ds}}) / (d \times R_g) \]

\[ \text{Diff}_v = (3.8001 \times 10^{-10}) \times (T_{\text{gas}}^{1.7780}); \]

Estimated diffusion coefficient by Hansen

\[ \text{Sc} = \text{viscosity}_g / (\text{density}_g \times \text{Diff}_v); \]
\[ Sh = 0.023 \times (\text{Re}^{0.8}) \times (\text{Sc}^{0.4}); \]

% Saturate Vapor Pressure for NaCl (From DIPPR)
\[ A_1 = 85.587; \quad B_1 = -31057; \quad C_1 = -7.6371; \quad D_1 = 2.6231 \times 10^{-7}; \quad E_1 = 2; \]
\[ \text{Pressure}_{\text{sat}} = \text{zeros}(1,k); \]
\[ \text{for } j = 1:1:k \]
\[ \text{Pressure}_{\text{sat}}(j) = \exp(A_1 + B_1 / T_{\text{ds}}(j) + C_1 \times \log(T_{\text{ds}}(j)) + D_1 \times T_{\text{ds}}(j)^E_1); \]
\[ \text{end} \]

% Partial Vapor Pressure
\[ \text{I}_{\text{condens}}(j) = X_{\text{NaCl}} \times Sh \times (\text{Diff}_v \times \text{Diff}_d(j))^{0.5} \times ((\text{Pressure}_v / T_{\text{gas}}) - \text{Pressure}_{\text{sat}} / T_{\text{ds}}(j)) / (d \times R_g); \]

% Saturate Vapor Pressure for KCl (From DIPPR)
\[ A_2 = 67.041; \quad B_2 = -27778; \quad C_2 = -5.2577; \quad D_2 = 3.0381 \times 10^{-22}; \quad E_2 = 6; \]
\[ \text{Pressure}_{\text{sat}} = \text{zeros}(1,k); \]
\[ \text{for } j = 1:1:k \]
\[ \text{Pressure}_{\text{sat}}(j) = \exp(A_2 + B_2 / T_{\text{ds}}(j) + C_2 \times \log(T_{\text{ds}}(j)) + D_2 \times T_{\text{ds}}(j)^E_2); \]
\[ \text{end} \]

% Partial Vapor Pressure
\[ \text{I}_{\text{condens}}(j) = \text{I}_{\text{condens}}_{\text{NaCl}}(j) + \text{I}_{\text{condens}}_{\text{KCl}}(j); \]

% Impaction Efficiency & Stickiness Efficiency
\[ \text{stoke} = \text{density}_p \times \text{Diam}_p^2 \times \text{V}_{\text{gas}} / (9 \times \text{viscosity}_g \times d_c); \]
\[ \text{aa} = 0.125; \quad \text{bb} = 1.25; \quad \text{cc} = 0.014; \quad \text{dd} = 0.00000508; \]
\[ \text{stoke} = \text{stoke} \times ((18 / \text{Re}_p)^{(1/3)} - (\sqrt{6} \times \text{atan}((\text{Re}_p^{1/3}) / \sqrt{6}))) / (\text{stoke} - \text{stoke} \times \text{stoke}); \]
\[ \text{if } \text{stoke} < 0.14 \]
\[ \text{stoke} = 0.13; \]
\[ \text{end} \]
\[ \phi = \text{stoke} - \text{aa}; \]
\[ \text{Impaction} = 1 / (1 + (\text{bb} / \phi) \times (\text{cc} / \phi^2) + (\text{dd} / \phi^3)); \]

if \( T_p > T_{15} \)
\[ \text{melt}_p = ((0.7 - 0.15) / (T_{70} - T_{15})) \times (T_p - T_{15}) + 0.15; \]
\[ \text{if } \text{melt}_p > 1.0 \]
\[ \text{melt}_p = 1.0; \]
\[ \text{end} \]
\[ \text{if } \text{melt}_p > 0.15 \]
\[ \text{Stickiness} = (\text{melt}_p - 0.15) / (0.7 - 0.15); \]
\[ \text{else } \text{Stickiness} = 0; \]
\[ \text{end} \]
\[ \text{if } \text{Stickiness} > 1.0 \]
\[ \text{Stickiness} = 1.0; \]
\[ \text{end} \]
\[ \text{else } \text{Stickiness} = 0; \]
\[ \text{end} \]
melt_fraction_s = zeros(1,k); Stickiness_eff_s = zeros(1,k); Stickiness_eff = zeros(1,k);

for j = 1:1:k
    if T_ds(j) > T15 && T_ds(j)>T_set_point
        melt_fraction_s(j) = ((0.7-0.15)/(T70-T15))*(T_ds(j)-T15)+0.15;
    end
    if melt_fraction_s(j) > 1.0
        melt_fraction_s(j) = 1.0;
    end

    if Stickiness_eff_s(j)>0.15
        Stickiness_eff_s(j) = (melt_fraction_s(j) - 0.15)/(0.7-0.15);
    else Stickiness_eff_s(j) = 0;
    end

    if Stickiness_eff_s(j) > 1.0
        Stickiness_eff_s(j) = 1.0;
    end
    else Stickiness_eff_s(j) = 0;
end

if   T_ds(j) < Tcut && T_ds(j) > T_solid
    Stickiness_eff_s(j) = 1.0;
end

Stickiness_eff(j) = Stickiness_eff_p + (1-Stickiness_eff_p)*Stickiness_eff_s(j);
end

%%%%%%%%%%%%%%%%%%%%% Kinetic Energy Stickiness Model (KESM)%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% To activate the KESM, the following section shall be uncommented %%%%%%%%%%%%%%
% vp = 0.27;  vs = 0.27;
% Ep = (3e81)*exp(-0.1302*T_gas);   % It is valid for particles <110 Micron
% Es = (3e81)*exp(-0.1302*T_ds(j));
% kp = (1-vp^2)/(pi*Ep);
% ks = (1-vs^2)/(pi*Es);
% RR = 0.9;
% K = 0.51 * (5*pi*vp*(kp+ks)/(4*density_p^1.5))^0.4;
% Vcr = (2*K/(Diam_p * RR^2))^(10/7)
% V_p = V_gas;
% if V_p > Vcr
%    Stickiness_eff_p = 0;
% else
%    Stickiness_eff_p = 1;
% end
% viscosity_d = 10^10;
% Vscr = density_p*(9.8^2)*((Diam_p*0.5)^3)*(viscosity_d)/(surface_tension^2)
% melt_fraction_s = zeros(1,k); Stickiness_eff_s = zeros(1,k); Stickiness_eff = zeros(1,k);
% for j = 1:1:k
%    if V_p > Vscr
%        Stickiness_eff_s(j) = 1;
%    else
%        Stickiness_eff_s(j) = 0;
%    end
%    Stickiness_eff(j) = Stickiness_eff_p + (1-Stickiness_eff_p)*Stickiness_eff_s(j);
% end

%%% Deposit Porosity and Thermal Conductivity
Porosity_deposit = zeros(1,k); F = zeros(1,k); k_N2 = zeros(1,k);
Por = 1-Porosity_deposit0;

for j = 1:1:k
    if T_ds(j) > T0 && T_ds(j)<=T15
        melt_fraction_deposit(j) = ((0.15-0)/(T15-T0))*(T_ds(j)-T0);
    elseif T_ds(j)>T15 && T_ds(j)<=T70
        melt_fraction_deposit(j) = ((0.70-0.15)/(T70-T15))*(T_ds(j)-T15)+0.15;
    elseif T_ds(j)>T70 && T_ds(j)<=T100
        melt_fraction_deposit(j) = ((1.0-0.70)/(T100-T70))*(T_ds(j)-T70)+0.70;
    elseif T_ds(j)>T100
        melt_fraction_deposit(j) = 1.0;
    else melt_fraction_deposit(j) = 0;
end
% Vl_Vs(j) = melt_fraction_deposit(j)*density_true./(density_liq*(1-
melt_fraction_deposit(j)));  
Porosity_deposit(j) = 1-(Poro+(Vl_Vs(j))*Poro);  
n = 6.5;            % it is an empirical parameter  
F(j) = ((2^n)/((2^n-1))^1-(1/((1+Porosity_deposit(j)).^n))));  
A_N2 = 0.00033143;  B_N2 = 0.7722;  C_N2 = 16.32;  D_N2 = 373.72;  
k_N2(j) = (A_N2 * T_ds(j))/(1+B_N2);% k_solid = 4.0;        % W/mK  
k_deposit(j) = (1-F(j)) * k_solid + (F(j) * k_N2(j));
% Eddy Impaction % I_eddy= V_gas * concentration_ash * f_p & f_p = k_c * f_v & f_v = Sr * V_gas / d_c  
R_g_mole = 8.314;  
v_rms = sqrt(3 * R_g_mole * T_gas / Mol_gas);  
d_eddy = sqrt(d_c) / sqrt(v_rms * density_g * ((density_p/density_g)+1) / viscosity_g);  
% Calculate particle diameter of eddy impaction  
Sr = 0.2;        % Strouhal number  
f_v = Sr * V_gas / d_c;        % frequency of vortex shedding  
k_c = 1.1;        % an empirical parameter  
f_p = k_c * f_v;        % It is linearly related to frequency of vortex shedding. It is the probability of impacting and sticking the rear side.  
concentration_ash = 8e-06*Re*Flow_p;        % kg/m3 Concentration of particle that participate in eddy impaction. (I assumed it)  
if Diam_p < d_eddy  
i_eddy= V_gas * concentration_ash * f_p;        % (kg/(m2.s)) flux of eddy impaction  
else i_eddy = 0;  
end  

%% Deposit Thickness Calculation  
Thick_deposit_check = Thick_deposit;  
Thick_deposit = zeros(1,k);  
for j = 2:1:k  
Thick_deposit(j) = ((Stickiness_eff(j)*Impaction_eff*Flow_p*(V_inertial(j)+V_th(j))+I_condens(j))/(density_p*(1-
Porosity_deposit(j))))*(t(j)-t(j-1))+Thick_deposit(j-1);  
end  
if abs((Thick_deposit_check(j) -Thick_deposit(j))*100/Thick_deposit(j)) < 0.01  
break  
end  

end  

%% The above calculations are for one bin of PSD. The results are multiplied by weight percentage for the bin: PSD1_w = 0.2;   PSD2_w = 0.60;   PSD3_w = 0.2;  
Thick_deposit_1 = zeros(1,k);  
Thick_deposit_2 = zeros(1,k);  
Thick_deposit_3 = zeros(1,k);  
if PSD==1  
Thick_deposit_1 = Thick_deposit*PSD1_w;   Impaction_eff_1 = Impaction_eff*PSD1_w;   Stickiness_eff_1 = Stickiness_eff*PSD1_w;   V_inertial_1 = V_inertial*PSD1_w;   V_th_1 = V_th*PSD1_w;   I_condens_1 = I_condens*PSD1_w;   I_eddy_1 = I_eddy*PSD1_w;   Flow_p1 = Flow_p*PSD1_w;   V_th_s_1 = V_th_s*PSD1_w;   T_ds_1 = T_ds*PSD1_w;   Resistance_1 = Resistance*PSD1_w;   Porosity_deposit_1 = Porosity_deposit*PSD1_w;   k_deposit_1 = k_deposit*PSD1_w;  
elseif PSD==2  
Thick_deposit_2 = Thick_deposit*PSD2_w;   Impaction_eff_2 = Impaction_eff*PSD2_w;   Stickiness_eff_2 = Stickiness_eff*PSD2_w;   V_inertial_2 = V_inertial*PSD2_w;   V_th_2 = V_th*PSD2_w;   I_condens_2 = I_condens*PSD2_w;   I_eddy_2 = I_eddy*PSD2_w;   Flow_p2 = Flow_p*PSD2_w;   V_th_s_2 = V_th_s*PSD2_w;   T_ds_2 = T_ds*PSD2_w;   Resistance_2 = Resistance*PSD2_w;   Porosity_deposit_2 = Porosity_deposit*PSD2_w;   k_deposit_2 = k_deposit*PSD2_w;  
elseif PSD==3  
Thick_deposit_3 = Thick_deposit*PSD3_w;   Impaction_eff_3 = Impaction_eff*PSD3_w;   Stickiness_eff_3 = Stickiness_eff*PSD3_w;
\[ V_{\text{inertial}}_3 = V_{\text{inertial}} \times \text{PSD}3_w; \quad V_{\text{th}}_3 = V_{\text{th}} \times \text{PSD}3_w; \quad I_{\text{condens}}_3 = I_{\text{condens}} \times \text{PSD}3_w; \quad I_{\text{eddy}}_3 = I_{\text{eddy}} \times \text{PSD}3_w; \quad \text{Flow}_p3 = \text{Flow}_p; \quad V_{\text{th}}_s_3 = V_{\text{th}}_s \times \text{PSD}3_w; \quad T_{ds}_3 = T_{ds} \times \text{PSD}3_w; \quad \text{Resistance}_3 = \text{Resistance} \times \text{PSD}3_w; \quad \text{Porosity}_\text{deposit}_3 = \text{Porosity}_\text{deposit} \times \text{PSD}3_w; \quad k_{\text{deposit}}_3 = k_{\text{deposit}} \times \text{PSD}3_w; \]

end

end

%% Put together all the results of above calculations
\[ \text{Impaction}_\text{eff} = \text{Impaction}_\text{eff}1 + \text{Impaction}_\text{eff}2 + \text{Impaction}_\text{eff}3; \]
\[ \text{Stickiness}_\text{eff} = \text{Stickiness}_\text{eff}_1 + \text{ Stickiness}_\text{eff}_2 + \text{ Stickiness}_\text{eff}_3; \]
\[ V_{\text{inertial}} = V_{\text{inertial}}_1 + V_{\text{inertial}}_2 + V_{\text{inertial}}_3; \]
\[ V_{\text{th}} = V_{\text{th}}_1 + V_{\text{th}}_2 + V_{\text{th}}_3; \]
\[ V_{\text{th}}_s = V_{\text{th}}_s_1 + V_{\text{th}}_s_2 + V_{\text{th}}_s_3; \]
\[ I_{\text{condens}} = I_{\text{condens}}_1 + I_{\text{condens}}_2 + I_{\text{condens}}_3; \]
\[ I_{\text{eddy}} = I_{\text{eddy}}_1 + I_{\text{eddy}}_2 + I_{\text{eddy}}_3; \]
\[ T_{ds} = T_{ds}_1 + T_{ds}_2 + T_{ds}_3; \]
\[ \text{Resistance} = \text{Resistance}_1 + \text{Resistance}_2 + \text{Resistance}_3; \]
\[ \text{Porosity}_\text{deposit} = \text{Porosity}_\text{deposit}_1 + \text{Porosity}_\text{deposit}_2 + \text{Porosity}_\text{deposit}_3; \]
\[ k_{\text{deposit}} = k_{\text{deposit}}_1 + k_{\text{deposit}}_2 + k_{\text{deposit}}_3; \]
\[ \text{Flow}_p = \text{Flow}_p1 + \text{Flow}_p2 + \text{Flow}_p3; \]
\[ \text{Thick}_\text{deposit} = \text{zeros}(1,k); \]

for \( j = 2:1:k \)
\[ \text{Thick}_\text{deposit}(j) = \]
\[ ((\text{Stickiness}_\text{eff}(j) \times \text{Impaction}_\text{eff} \times \text{Flow}_p \times (V_{\text{inertial}}(j) + V_{\text{th}}(j)) + I_{\text{condens}}(j))/(\text{density}_p \times (1 - \text{Porosity}_\text{deposit}(j)))) \times (t(j) - t(j-1)) + \text{Thick}_\text{deposit}(j-1); \]

end

%% Distinguish the role of EACH mechanism in ash deposit growth on the stagnation point of the front side

% effect of Thermophoresis

for \( j = 2:1:k \)
\[ \text{Thick}_\text{deposit} \_\text{th}(1) = 0; \]
\[ \text{V}_{\text{inertial}} \_\text{reserve}(j) = \text{V}_{\text{inertial}}(j); \]
\[ \text{V}_{\text{inertial}} \_\text{reserve}(j) = 0; \]
\[ \text{I}_{\text{condens}} \_\text{reserve}(j) = \text{I}_{\text{condens}}(j); \]
\[ \text{I}_{\text{condens}} \_\text{reserve}(j) = 0; \]
\[ \text{Thick}_\text{deposit} \_\text{th}(j) = \]
\[ ((\text{Stickiness}_\text{eff}(j) \times \text{Impaction}_\text{eff} \times \text{Flow}_p \times (\text{V}_{\text{inertial}} \_\text{reserve}(j) + \text{V}_{\text{th}}(j)) + \text{I}_{\text{condens}} \_\text{reserve}(j))/(\text{density}_p \times (1 - \text{Porosity}_\text{deposit}(j)))) \times (t(j) - t(j-1)) + \text{Thick}_\text{deposit} \_\text{th}(j-1); \]

end

% effect of Inertial

for \( j = 2:1:k \)
\[ \text{Thick}_\text{deposit} \_\text{inertial}(1) = 0; \]
\[ \text{V}_{\text{th}} \_\text{reserve}(j) = \text{V}_{\text{th}}(j); \]
\[ \text{V}_{\text{th}} \_\text{reserve}(j) = 0; \]
\[ \text{I}_{\text{condens}} \_\text{reserve}(j) = \text{I}_{\text{condens}}(j); \]
\[ \text{I}_{\text{condens}} \_\text{reserve}(j) = 0; \]
\[ \text{Thick}_\text{deposit} \_\text{inertial}(j) = \]
\[ ((\text{Stickiness}_\text{eff}(j) \times \text{Impaction}_\text{eff} \times \text{Flow}_p \times (\text{V}_{\text{inertial}}(j) + \text{V}_{\text{th}} \_\text{reserve}(j)) + \text{I}_{\text{condens}} \_\text{reserve}(j))/(\text{density}_p \times (1 - \text{Porosity}_\text{deposit}(j)))) \times (t(j) - t(j-1)) + \text{Thick}_\text{deposit} \_\text{inertial}(j-1); \]

end

% effect of Condensation

for \( j = 2:1:k \)
\[ \text{Thick}_\text{deposit} \_\text{Condensation}(1) = 0; \]
\[ \text{V}_{\text{th}} \_\text{reserve}(j) = \text{V}_{\text{th}}(j); \]
\[ \text{V}_{\text{th}} \_\text{reserve}(j) = 0; \]
\[ \text{V}_{\text{inertial}} \_\text{reserve}(j) = \text{V}_{\text{inertial}}(j); \]
\[ \text{V}_{\text{inertial}} \_\text{reserve}(j) = 0; \]
\[ \text{Thick}_\text{deposit} \_\text{Condensation}(j) = \]
\[ ((\text{Stickiness}_\text{eff}(j) \times \text{Impaction}_\text{eff} \times \text{Flow}_p \times (\text{V}_{\text{inertial}} \_\text{reserve}(j) + \text{V}_{\text{th}} \_\text{reserve}(j)) + \text{I}_{\text{condens}}(j))/(\text{density}_p \times (1 - \text{Porosity}_\text{deposit}(j)))) \times (t(j) - t(j-1)) + \text{Thick}_\text{deposit} \_\text{Condensation}(j-1); \]

end

%% Ash deposit calculation around the surface of pipe

for \( \text{theta} = 1:1:181 \)
\[ \text{delta}(\text{theta}) = (T_{ds}(j) - T_{\text{set point}}) \times \text{abs}(\cos(\text{theta}-1)) + (T_{\text{gas}} - T_{ds}(j)); \]
\[ T_{s}(\text{theta}) = T_{\text{gas}} - \text{delta}(\text{theta}); \]
\[ \text{Pressure}_{v \text{ sat s NaCl}}(\theta) = \exp(A1+B1/T_{s}(\theta)+C1\log(T_{s}(\theta))+D1\cdot T_{s}(\theta)^{E1}); \]

\[ \text{Pressure}_{v \text{ sat s KCL}}(\theta) = \exp(A2+B2/T_{s}(\theta)+C2\log(T_{s}(\theta))+D2\cdot T_{s}(\theta)^{E2}); \]

\[ \text{I}_{\text{condens NaCl}}(\theta) = X_{\text{NaCl}}\cdot S_{h}\cdot \left((\text{Diff}_{v g}\cdot \text{Diff}_{v ds}(j))^{0.5}\right)\cdot \left((\text{Pressure}_{v g \text{ NaCl}}/T_{\text{gas}})-\left(\text{Pressure}_{v \text{ sat s NaCl}}(\theta)/T_{s}(\theta)\right)\right)/\left(d_{c}\cdot R_{g}\right); \]

\[ \text{I}_{\text{condens KCL}}(\theta) = X_{\text{KCl}}\cdot S_{h}\cdot \left((\text{Diff}_{v g}\cdot \text{Diff}_{v ds}(j))^{0.5}\right)\cdot \left((\text{Pressure}_{v g \text{ KCL}}/T_{\text{gas}})-\left(\text{Pressure}_{v \text{ sat s KCL}}(\theta)/T_{s}(\theta)\right)\right)/\left(d_{c}\cdot R_{g}\right); \]

\[ \text{I}_{\text{condens}}(\theta) = \text{I}_{\text{condens NaCl}}(\theta) + \text{I}_{\text{condens KCL}}(\theta); \]

\[ \text{end} \]

for \( \theta = 1:1:181 \)

\[ \text{Thick deposit th pipe front}(\theta) = \left(\text{Stickiness eff}(j)\cdot \text{Impaction eff}\cdot \text{Flow p}\cdot \left(\text{V inertial reserve}(j)+\text{V th s}(\theta)\right)+\text{I}_{\text{condens reserve}}(j)\right)/\left(\text{density p}\cdot \left(1-\text{Porosity deposit}(j)\right)\right)\cdot t(j); \]

\[ \text{Thick deposit th pipe back}(\theta) = \left(\text{Stickiness eff}(1)\cdot \text{Impaction eff}\cdot \text{Flow p}\cdot \left(\text{V inertial reserve}(j)+\text{V th s}(1)\right)+\text{I}_{\text{condens reserve}}(j)\right)/\left(\text{density p}\cdot \left(1-\text{Porosity deposit}(j)\right)\right)\cdot t(j); \]

\[ \text{Thick deposit Condensation pipe front}(\theta) = \left(\text{Stickiness eff}(j)\cdot \text{Impaction eff}\cdot \text{Flow p}\cdot \left(\text{V inertial reserve}(j)+\text{V th s}(\theta)\right)+\text{I}_{\text{condens}}(j)\right)/\left(\text{density p}\cdot \left(1-\text{Porosity deposit}(j)\right)\right)\cdot t(j); \]

\[ \text{Thick deposit Condensation pipe back}(\theta) = \left(\text{Stickiness eff}(1)\cdot \text{Impaction eff}\cdot \text{Flow p}\cdot \left(\text{V inertial reserve}(j)+\text{V th s}(1)\right)+\text{I}_{\text{condens}}(j)\right)/\left(\text{density p}\cdot \left(1-\text{Porosity deposit}(j)\right)\right)\cdot t(j); \]

\[ \text{if Thick deposit th pipe front}(\theta) < \text{Thick deposit th}(j) \]

\[ \text{Thick deposit th pipe front}(\theta) = \text{Thick deposit th}(j); \]

\[ \text{end} \]

\[ \text{if Thick deposit Condensation pipe front}(\theta) < \text{Thick deposit Condensation}(j) \]

\[ \text{Thick deposit Condensation pipe front}(\theta) = \text{Thick deposit Condensation}(j); \]

\[ \text{end} \]

\[ \text{Thick deposit pipe front}(\theta) = \text{Thick deposit th pipe front}(\theta) + \text{Thick deposit Condensation pipe front}(\theta) + \text{Thick deposit inertial pipe}(\theta); \]

\[ \text{Thick deposit pipe Back}(\theta) = \text{Thick deposit th pipe back}(\theta) + \text{Thick deposit Condensation pipe back}(\theta) + \text{Thick deposit Eddy pipe}(\theta); \]

\[ \text{end} \]

\[ \theta = 0:1:180; \]

\[ \text{xp} = \left(d_{c}\cdot 1000/2\right)\cdot \cosd(\theta); \]

\[ \text{yp} = \left(d_{c}\cdot 1000/2\right)\cdot \sind(\theta); \]

\[ \text{p1} = \text{plot}(100+\text{xp},100+\text{yp},'-k'); \]

\[ \text{hold on}; \]

\[ \text{p2} = \text{plot}(100+\text{xp},100-\text{yp},'-k'); \]

\[ \text{hold on}; \]

\[ \text{p3} = \text{plot}(100+(\left(d_{c}\cdot \text{Thick deposit pipe Front}(1)\right)\cdot 1000/2)\cdot \cosd(\theta), \text{Thick deposit pipe front}(1000+100+\text{yp},'-k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p4} = \text{plot}(100+(\left(d_{c}\cdot \text{Thick deposit th pipe front}(1)\right)\cdot 1000/2)\cdot \cosd(\theta), \text{Thick deposit th pipe front}(1000+100+\text{yp},'-k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p5} = \text{plot}(100+(\left(d_{c}\cdot \text{Thick deposit th pipe front}(1)\right)\cdot 1000/2)\cdot \cosd(\theta), 100-\text{yp}-\text{Thick deposit th pipe back}(1000,-'k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p6} = \text{plot}(100+\text{xp}, \text{Thick deposit inertial pipe}(1000+100+\text{yp},':k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p7} = \text{plot}(100+(\left(d_{c}\cdot \text{Thick deposit Condensation pipe front}(1)\right)\cdot 1000/2)\cdot \cosd(\theta), \text{Thick deposit Condensation pipe front}(1000+100+\text{yp},'-k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p8} = \text{plot}(100+(\left(d_{c}\cdot \text{Thick deposit Condensation pipe front}(1)\right)\cdot 1000/2)\cdot \cosd(\theta), 100-\text{yp}-\text{Thick deposit Condensation pipe back}(1000,-'k', 'LineWidth', 2); \]

\[ \text{hold on}; \]

\[ \text{p9} = \text{plot}(100+\text{xp}, 100-\text{yp}\cdot \text{Thick deposit Eddy pipe}(1000,-'k', 'LineWidth', 1); \]

\[ \text{hold on}; \]

\[ \text{set(gca,'FontSize',14,'FontName','Times New Roman'); } \]

\[ \text{xlabel('Ash deposit thickness (mm)','FontSize',18,'FontName','Times New Roman'); } \]

\[ \text{ylabel('Ash deposit thickness (mm)','FontSize',18,'FontName','Times New Roman'); } \]

\[ \text{legend([p1(1), p3(1), p6(1), p4(1), p7(1), p9(1)], 'Surface of Probe','Total Ash deposition thickness','Inertial impaction','Thermophoresis deposit','Condensation deposit','Eddy impaction','Location','East'); } \]

\[ \text{xlim([50 200]); } \]

\[ \text{hold on}; \]

\[ 298 \]
%% Calculation of Ash Deposition Rate (g/m²); Note: for OFC 100 kW modeling, the ash deposit of
three quarters of the surface are in place

\[
\text{Radius} \_\text{Front}(1) = (\text{Thick} \_\text{deposit} \_\text{inertial} \_\text{pipe}(1) + \text{Thick} \_\text{deposit} \_\text{th} \_\text{pipe} \_\text{front}(1) + \text{Thick} \_\text{deposit} \_\text{Condensation} \_\text{pipe} \_\text{front}(1)) ;
\]

\[
\text{Radius} \_\text{Back}(1) = (\text{Thick} \_\text{deposit} \_\text{Condensation} \_\text{pipe} \_\text{back}(1) + \text{Thick} \_\text{deposit} \_\text{th} \_\text{pipe} \_\text{back}(1) + \text{Thick} \_\text{deposit} \_\text{Eddy} \_\text{pipe}(1)) ;
\]

\[
\text{for} \ \theta = 1:1:180 \\
\text{Radius} \_\text{Front}(\theta + 1) = ((\text{Thick} \_\text{deposit} \_\text{inertial} \_\text{pipe}(\theta + 1) + \text{Thick} \_\text{deposit} \_\text{th} \_\text{pipe} \_\text{front}(\theta + 1) + \text{Thick} \_\text{deposit} \_\text{Condensation} \_\text{pipe} \_\text{front}(\theta + 1)) + \text{Radius} \_\text{Front}(\theta)) \cdot 0.5 ;
\]

\[
\text{bin} \_\text{Front}(\theta + 1) = \pi \cdot ((\text{Radius} \_\text{Front}(\theta + 1) + (0.5 \cdot d_c)) \text{^} 2 - (d_c \cdot 0.5) \text{^} 2) / 360 ;
\]

\[
\text{Radius} \_\text{Back}(\theta + 1) = ((\text{Thick} \_\text{deposit} \_\text{Condensation} \_\text{pipe} \_\text{back}(\theta + 1) + \text{Thick} \_\text{deposit} \_\text{th} \_\text{pipe} \_\text{back}(\theta + 1) + \text{Thick} \_\text{deposit} \_\text{Eddy} \_\text{pipe}(\theta + 1)) + \text{Radius} \_\text{Back}(\theta + 1)) \cdot 0.5 ;
\]

\[
\text{bin} \_\text{Back}(\theta + 1) = \pi \cdot ((\text{Radius} \_\text{Back}(\theta + 1) + (0.5 \cdot d_c)) \text{^} 2 - (d_c \cdot 0.5) \text{^} 2) / 360 ;
\]

end

\[
\text{Area} = \pi \cdot d_c \cdot \text{Length} \cdot 3/4 ;
\]

% Surface area Coupons

\[
\text{Volume} \_\text{deposit} = \text{Length} \cdot (\text{sum(bin} \_\text{Front}) + \text{sum(bin} \_\text{Back}) - \text{sum(bin} \_\text{Back}(45:135))) ;
\]

% Volume of the ash deposits (m³)

\[
\text{density} \_\text{deposit} \_\text{bulk} = \text{density} \_\text{true} \cdot (1 - \text{Porosity} \_\text{deposit}(k)) ;
\]

\[
\text{Ash} \_\text{Deposit} \_\text{Area} = \text{Volume} \_\text{deposit} \cdot \text{density} \_\text{deposit} \_\text{bulk} \cdot 1000 / \text{Area} ;
\]

% Ash Deposit Rate (g/m²)

%% Plot the results

figure;
plot(t/60, Thick_deposit*1000, 'b' );
xlabel('Time (min)'); ylabel('Deposit Thickness (mm)');
hold on;
plot(t/60, Thick_deposit_inertial*1000, 'r' );
hold on;
plot(t/60, Thick_deposit_th*1000, 'c' );
hold on;
plot(t/60, Thick_deposit_Condensation*1000, 'g' );
hold on;
legend('Total', 'Inertial', 'Thermophoresis', 'Condensation', 'Location', 'NorthWest')
figure
plot(t/60, T_ds);
xlabel('Time (min)'); ylabel('Deposit Surface Temp (K)');

figure;
plot(t/60, V_th*1000);
xlabel('Time (min)'); ylabel('Velocity_t_h (mm/s)');
figure
\[ \text{t}_2 = \text{linspace}(0, \text{t}_{\text{end}}, \text{length(Resistance)}) ; \]
plot(t2/60, Resistance);
xlabel('Time (min)'); ylabel('Resistance (K*m^2/W)');
legend('The resistance of ash deposit', 'Location', 'NorthWest');
figure
plot(Porosity_deposit, T_ds);
xlabel('Deposit Porosity'); ylabel('Deposit Surface Temp (K)');
legend('The changes of ash deposit surface temperature vs. porosity', 'Location', 'SouthWest');
ylim([800 1400]);
figure
stoke = linspace(0.14, 100, 1000);
for \[ i = 1:1:\text{length(stoke)} \]
\[ \phi(i) = \text{stoke}(i) - \text{aa} ; \]
\[ \text{Impaction} \_\text{eff}(i) = 1 / ((\text{bb} / \phi(i)^{2}) - (\text{cc} / \phi(i)^{3}) + (\text{dd} / \phi(i)^{4})) ; \]
end
semilogx(stoke, Impaction_eff, 'b');
xlim([0.14 100.0]);
xlabel('Stokes Number'); ylabel('Impaction Efficiency');
figure
Kn = linspace(1e-04,10,1000);
conductivity_ratio = exp(linspace(log(1),log(1000),10));
for mm = 1:length(conductivity_ratio)
    for i = 1:length(Kn)
        coeff(mm,i) = (-12*pi*Ktc*(1+(Ce*conductivity_ratio(mm)*Kn(i))))/((2+conductivity_ratio(mm)*Kn(i))*(1+(3*Cm*Kn(i))));
        coeff_figure(mm,i) = -coeff(mm,i)/(2*pi);
    end
    semilogx(Kn, coeff_figure(mm, 1:end), 'Linewidth', 2);
    hold on;
    lgd(mm) = ['-\Lambda', sprintf(' = %0.0f', conductivity_ratio(mm))];
end
legend(lgd);
set(gca, 'FontSize', 14, 'FontName', 'Times New Roman')
xlabel('Kn Number', 'FontSize', 18, 'FontName', 'Times New Roman'); ylabel('-\Phi/2\pi', 'FontSize', 18, 'FontName', 'Times New Roman');

% figure
% theta = 1:1:181;
% plot(theta, Thick_deposit_Condensation_pipe_front)
% figure
% plot(theta, I_condens_s);
% figure
% plot(t, I_condens);
% figure
% plot(theta, Thick_deposit_th_pipe_front)

uiwait(msgbox(sprintf('Ash Deposit of fuel #%d for #%d (min) is #%G (g/m2). The ash deposit thickness at the upstream side of the coupon is #%G (mm)', Fuel, t_end/60, Ash_Deposit_Area, Thick_deposit(k)*1000)););