UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA

Modelling and Prediction of Aerosol Precursors in the CLOUD chamber

Pedro Miguel Ferreira Mendeiros

Mestrado em Engenharia Física

Dissertação orientada por:
António Joaquim Rosa Amorim Barbosa
António Rodrigues Tomé

2025

Acknowledgments

I am deeply grateful for the guidance and mentorship I received throughout this demanding process. My sincere thanks go to my supervisors, António Amorim and António Tomé, for their expertise, insightful direction and support, as their patience in this research proved absolutely invaluable to the completion of this dissertation.

I must also acknowledge the invaluable support of the entire CLOUD Collaboration. I am especially grateful to the team members, particularly Pedro Rato, for their help in explaining the ins and outs of the experiment, providing the tools and maintaining the infrastructure that was fundamental to the analysis presented here.

I am also truly grateful for the essential financial support that made this work possible. My thanks are extended to FCT (Fundação para a Ciência e Tecnologia) for a research grant under the "PP-CLOUD: Participlação na Experiência CLOUD" project (Reference: CERN/FIS-COM/0008/2021).

On a personal level, this thesis would not have been completed without the constant support from my loved ones. To my parents, thank you for your constant encouragement. A special note of thanks goes to my girlfriend Filipa, for being my rock and source of comfort, and for her immense patience and understanding throughout the many long hours dedicated to this work. Finally, thank you to my friends João, Rúben and Alexandre for the necessary distractions, the laughter, and the camaraderie that kept me grounded.

Resumo

A experiência CLOUD, localizada no CERN, é um aparto científico inovador dedicado à ciência atmosférica, sendo a primeira vez que um acelerador de física de alta energia é utilizado para estudar a ciência atmosférica e climática. A colaboração CLOUD é uma equipa interdisciplinar de cientistas de 17 institutos em nove países. A peça central do experimento é uma câmara de aço inoxidável de 26.1 metros cúbicos, onde são recriadas condições atmosféricas controladas, e cientistas podem controlar rigorosamente fatores ambientais como a temperatura, operando até cerca de -70ºC, e introduzir múltiplas fontes de luz para simular reações fotolíticas. Um feixe de piões do PS do CERN simula os raios cósmicos galácticos, e um campo elétrico de alta tensão permite remover todos os iões para estudar a sua influência. A câmara está equipada com até 40 instrumentos analíticos de última geração, incluindo espetrómetros de massa e Scanning Mobility Particle Sizers, que monitorizam continuamente as condições da câmara, a química da fase gasosa e a composição dos aerossóis.

O projeto CLOUD já avançou significativamente na compreensão do papel de aerossóis na formação de nuvens: pesquisas iniciais de 2011 demonstraram uma ligação entre os raios cósmicos e a nucleação de aerossóis, mostrando que os iões gerados pelos raios cósmicos aumentam significativamente a formação de partículas de aerossol na média troposfera, elevando as taxas de formação em até dez vezes ou mais. No entanto, concluiu-se que o ácido sulfúrico, a água e a amónia sozinhos, mesmo potenciados pelos raios cósmicos, eram insuficientes para explicar a formação de aerossóis observada na atmosfera, indicando o envolvimento de outros vapores. Um estudo de 2014 demonstrou o papel significativo de vapores biogénicos oxidados, como o alfa-pineno, juntamente com o ácido sulfúrico na formação de novas partículas de aerossol, um processo que é potenciado pelos iões dos raios cósmicos quando as concentrações de vapores são baixas. Uma publicação importante de 2016, baseada em medições do CLOUD, estabeleceu os principais processos para a FNP em toda a troposfera e mostrou que a ionização por raios cósmicos contribui para quase um terço de todas as partículas formadas. Mais recentemente, uma síntese de 2023 enfatizou os papéis cruciais de vapores condensáveis, como orgânicos altamente oxigenados e oxoácidos de iodo, juntamente com estabilizadores como amónia, aminas e iões, no estabelecimento de uma base mecanicista para modelos de qualidade do ar e do clima.

Este trabalho teve como foco melhorias metodológicas para o cálculo da Taxa de Formação de Novas Partículas, um cálculo complexo devido aos processos de perda inerentes à câmara: deposição na parede, coagulação e diluição. A taxa de formação é definida pela soma da taxa de aumento observada na concentração de partículas com as perdas. Os processos de perda incluem o Sumidouro de Diluição (perda devido ao fluxo contínuo de ar limpo), o Sumidouro de Parede (perda por difusão, crítica para partículas de diâmetros inferiores a 3 nanómetros) e o Sumidouro de Coagulação (perda devido à colisão e agregação de partículas).

Foi desenvolvido um novo script de cálculo em Python para abordar as limitações de implementações anteriores em R e Matlab, que apresentavam dificuldades de interpretação e uma inconsistência dimensional no cálculo do sumidouro de coagulação. O novo script em Python foi concebido para ser

transparente e robusto, cujo avanço fundamental foi a correção da fórmula do Sumidouro de Coagulação, garantindo que o cálculo fosse dimensionalmente consistente e refletisse com precisão a taxa de perda de partículas, um ponto crítico, já que a coagulação é uma via de perda significativa para partículas recém-formadas. Além disso, o script usa dados de fluxo em tempo real dos sensores da câmara, em vez de uma constante fixa, para calcular dinamicamente o sumidouro de diluição, fornecendo um resultado mais fisicamente representativo das condições dinâmicas da câmara. O filtro Savitzky-Golay foi selecionado para o cálculo da taxa de variação da concentração de partículas, o qual foi considerado o mais eficiente e fiável, pois evita o atraso temporal de outras técnicas e fornece uma diferenciação robusta de dados ruidosos. O script também utiliza os instrumentos CPC e PSM como referência primária para a concentração total de partículas nos seus respetivos limites de diâmetro (2.5 nm para CPC, 1.7 nm para PSM), aproveitando a sua eficiência superior de deteção nesses limiares.

A análise comparativa validou a nova metodologia. Para os Sumidouros de Diluição e Parede, os scripts em Python e em Matlab mostraram um alto grau de concordância, ao passo que a implementação em R exibiu picos anormais e artefactos. No entanto, no Sumidouro de Coagulação, as implementações anteriores apresentaram valores significativamente mais altos, enquanto o presente trabalho gerou valores mais baixos e estáveis, o que é uma consequência direta da correção da fórmula dimensionalmente inconsistente. A Taxa de Formação final calculada pela implementação em Python, embora mais sensível e mostrando valores positivos e negativos, é considerada a mais cientificamente sólida devido aos seus cálculos consistentes e ao método de derivada mais responsivo.

O projeto também incluiu a modelação e previsão da concentração de ácido sulfúrico, uma molécula essencial para a Formação de Novas Partículas, devido à sua baixíssima volatilidade e alta higroscopicidade. O ácido sulfúrico é produzido na troposfera pela oxidação do dióxido de enxofre pelo radical hidroxilo, um processo impulsionado pela intensidade de radiação ultravioleta (UV). Para esta modelação, um algoritmo foi desenvolvido para identificar e extrair "intervalos estáveis" de concentração de ácido sulfúrico, aplicando o filtro Savitzky-Golay e um processo de validação rigoroso, incluindo a verificação de que o declive do ajuste linear se encontrava abaixo de um limiar pré-definido.

A modelação explicativa focou-se em entender os fatores subjacentes que impulsionam o declive da concentração de ácido sulfúrico durante os períodos de aumento, para a qual foi utilizado um modelo linear devido à sua simplicidade e facilidade de interpretação do declive como taxa de aumento. O modelo base, utilizando apenas as características fundamentais (como ozono, dióxido de enxofre e intensidade UV), demonstrou uma limitada capacidade de explicação da variância, contudo, a introdução de transformações não lineares (polinómios, logaritmos) aumentou substancialmente a capacidade de explicação. A inclusão de Termos de Interação elevou o poder explicativo ainda mais, atingindo um valor final de R^2 de 0.720, o que validou a hipótese de que relações não lineares e sinérgicas são cruciais nos processos químicos da câmara CLOUD. Na modelação preditiva, que visa prever a concentração de ácido sulfúrico, foram desenvolvidos e otimizados dois modelos: o Elastic Net (um modelo linear com regularização) e o XGBoost (um método de aprendizagem automática não linear). O Elastic Net, após ajuste de parâmetros, alcançou um erro de previsão baixo (RMSE de 0.54326), servindo como uma linha de base robusta. O modelo XGBoost, após otimização intensiva, alcançou um RMSE de 0.5855. Apesar de o Elastic Net ter um RMSE marginalmente menor, o modelo XGBoost foi selecionado como o preditor final. O XGBoost demonstrou uma distribuição de erros mais uniforme e aleatória nos gráficos de diagnóstico, significando uma superior capacidade de ajuste à complexidade dos dados e uma maior robustez para prever em novas condições.

Em conclusão, o projeto resultou num framework computacional de cálculo da FNP mais robusto e transparente e num modelo preditivo fiável para o declive da concentração de ácido sulfúrico. Os

passos futuros incluem a integração dos scripts Python no Sistema DAQ da colaboração CLOUD para monitorização em tempo real, a expansão do framework preditivo para incluir outras espécies atmosféricas importantes (como moléculas orgânicas altamente oxigenadas e aminas) e a exploração de modelos de aprendizagem de máquina informados, que incorporariam leis fundamentais para garantir previsões estatisticamente precisas e fisicamente plausíveis.

Palavras chave: Formação de Nuvens, Formação de Novas Partículas, Aprendizagem Automática, Aerossóis

Abstract

The CLOUD (Cosmics Leaving Outdoor Droplets) experiment at CERN addresses the largest source of uncertainty in global climate models: the effect of aerosols in cloud formation. This dissertation focused on optimizing data analysis routines on the CLOUD collaboration, aiming to reduce these climate uncertainties. The primary objective was twofold: to enhance the accuracy of the New Particle Formation rate calculation and to develop a robust predictive model for the rate of change of the key aerosol precursor, sulfuric acid.

The methodological work involved the re-implementation and optimization of the Formation Rate calculation script in Python, which critically corrected a dimensional inconsistency in the coagulation sink term found in previous implementations, and incorporated advanced signal processing techniques for a more scientifically rigorous result. Furthermore, Machine Learning models were developed using CLOUD16 campaign data to predict the sulfuric acid concentration, incorporating key experimental parameters. After comparison, the non-linear XGBoost model was selected as the superior predictive tool, demonstrating a greater capacity to capture the complex, non-linear relationships driving sulfuric acid dynamics.

In conclusion, this work provides a new, reliable analytical and predictive framework for the CLOUD collaboration, contributing directly to more precise data interpretation and supporting a deeper understanding of aerosol formation mechanisms.

Keywords: Cloud Formation, New Particle Formation, Machine Learning, Aerosols

Index

List of Figures

List of Tables

Nomenclature

List of Abbreviations

APCI Atmospheric Pressure Chemical Ionization

CCN Cloud Condensation Nuclei

CI Chemical Ionization

CIMS Chemical Ionization Mass Spectrometer

CLOUD Cosmics Leaving OUtdoor Droplets

CPC Condensation Particle Counter

CSC CLOUD Slow Control

DAQ Data Acquisition

DBMS Database MAnagement System

DEG Diethylene Glycol

DIP Data Interchange Protocal

DMA Differential Mobility Analyzer

EI Electron Impact

GCR Galatic Cosmic Rays

GDE General Dynamic Equation

HOM Highly Oxygenated Organic Molecule

HPLC High-Performance Liquid Chromatography

MCPC Mixing-Type Condensation Particle Counter

NAIS Neutral Cluster and Air Ion Spectrometers

NCI Negative Chemical Ionization

nCNC Nano-Condensation Nucleus Counter

NIST National Institute of Standards and Technology

NPF New Particle Formation

nSMPS NanoScan Scanning Mobility Particle Sizer

NTP Network Time Protocol

PS Proton Synchroton

PSM Particle Size Magnifier

RMSE Root Mean Squared Error

SCADA Supervisory Control and Data Acquisition

Simple Network Time Protocol SNTP

SMPS Scanning Mobility Particle Sizer

UCPC Ultrafine Condensation Particle Counter

UTC Coordinated Universal Time

UV Ultraviolet

XGBoost eXtreme Gradient Boosting

List of Symbols

C_a, C_b	Convolution Kernel
C_C	Cunningham Correction Factor
c	Mean Thermal Speed
D	Diffusion Coefficient
d^*	Kelvin Diameter
d_p	Particle Diameter
F	Experimentally Determined Factor
$F_{\text{syntethic air}}$	Volumetric Flow Rate of Synthetic Air
g	Correction Factor
J	Formation Rate
k_B	Boltzmann Constant
k_{coag}	Coagulation Coefficient
k_{dil}	Dilution Coefficient
k_{wall}	Wall-loss Coefficient
l	Characteristic Length Parameter
m	Slope
m_p	Mass
N	Particle Number Concentration
p_d	Saturation Vapor Pressure on a Droplet's Surface
p_s	Saturation Vapor Pressure
Q_{flow}	Volumetric Flow Rate through a Tube
R	Gas Constant
R	Relative Difference
R^2	Coefficient of Determination
s	Relative Slope
S_{coag}	Coagulation Sink
S_{dil}	Dilution Sink
S_{wall}	Wall Sink
T	Temperature
V_{chamber}	Volume of the CLOUD chamber
Sh	Sherwood Number
SR	Saturation Ratio
γ	Surface Tension
Δt	Time Interval Length
η	Gas Viscosity
η_{eff}	Efficiency Factor
μ	Molar Volume
σ	Standard Deviation
ξ	Gormley-Kennedy Parameter

Chapter 1

Introduction to the CLOUD Experiment and Research Objectives

1.1 The CLOUD Experiment: An Overview

The Cosmics Leaving OUtdoor Droplets (CLOUD) experiment [1], located at the Proton Synchrotron (PS) [2] at CERN, is a groundbreaking scientific endeavor focused on atmospheric science and its influence on global climate. Its primary goal is to study the microphysics of how galactic cosmic rays (GCRs) interact with aerosols in controlled environments, and to understand the climate implications of these interactions, which is particularly significant because it marks the first time a high-energy physics accelerator has been used to study atmospheric and climate science, aiming to reduce the largest source of uncertainty in present climate models and radiative forcing related to aerosols and their effect on clouds.

Bringing together atmospheric physicists, chemists, and particle physicists, the CLOUD collaboration is an interdisciplinary team made up of scientists from 17 institutes across nine countries. At its core, the experiment utilizes a 26.1 m^3 electropolished stainless steel chamber, shown in Figure 1.1, often referred to as "the cleanest box in the world". Inside, true atmospheric conditions are recreated with exceptional purity and precision, as the chamber is filled with synthetic air derived from liquid nitrogen and oxygen, and humidified with ultra-pure water, where scientists can meticulously control various environmental factors, including temperature (from -65ºC to +100ºC for bakeout, operating down to -70ºC with 0.01ºC stability), and introduce multiple light sources at different wavelengths, simulating photolytic reactions caused by sunlight [3]. A 3.5 GeV/c pion beam from the CERN PS simulates galactic cosmic rays, allowing for adjustable "cosmic ray" intensity, and a high-voltage electric field cage can remove all ions to study their influence.

1. INTRODUCTION TO THE CLOUD EXPERIMENT AND RESEARCH OBJECTIVES

Figure 1.1: **The CLOUD chamber, the central component of the experiment at CERN.**The stainless-steel sphere, with a volume of 26 cubic meters, serves as a precisely controlled environment for studying atmospheric nucleation. The intricate network of pipes, cables, and external instruments connected to the chamber's many access ports highlights the complexity of the experimental setup required to control and measure a wide range of atmospheric parameters.

The experiment also allows for rapid adiabatic expansions to create liquid and ice clouds, enabling studies on cloud microphysics. To monitor the complex processes, the chamber is equipped with up to 40 state-of-the-art analytical instruments, including various mass spectrometers, scanning mobility particle sizers (SMPS), and a Flow Tube System (FLOTUS). These instruments continuously monitor chamber conditions, gas phase chemistry, the chemical composition of aerosols and clusters, small ion clusters, charged/neutral aerosol particles, cloud droplets, precursor gases, nucleating vapours, oxidising agents, humidity, and light spectrum/intensity. The CLOUD experiment has significantly advanced the understanding of New Particle Formation (NPF) and its link to cloud formation and global climate. Before CLOUD, sulfuric acid (H_2SO_4) was primarily thought to drive atmospheric nucleation, with organic compounds mainly responsible for particle growth, now however, CLOUD's extensive measurements have revealed a more complex picture.

In 2011, initial research demonstrated a link between cosmic rays and aerosol nucleation [4]. It showed that ions generated by cosmic rays significantly enhance the formation of aerosol particles in the mid-troposphere and above, boosting formation rates by up to ten-fold or more. This work, led by Jasper Kirkby, also found that in cooler temperatures, sulfuric acid and water vapor could quickly form clusters without additional vapors, but in the lowest layer of the atmosphere, additional vapors like ammonia were required, and critically, the study highlighted that sulfuric acid, water, and ammonia alone, even enhanced with cosmic ray, were insufficient to explain observed atmospheric aerosol formation, indicating other vapors must be involved.Further studies showed that while amines can cluster with sulfuric acid to produce new aerosol particles, cosmic rays have little influence on this specific formation process, except at very low overall rates.

A 2014 study demonstrated that oxidized biogenic vapors, such as alpha-pinene emitted by trees, play a significant role alongside sulfuric acid in the formation of new aerosol particles [5]. This interaction is enhanced by ions from cosmic rays, particularly when concentrations of sulfuric acid and organic vapors are low, and may account for seasonal variations in atmospheric aerosols.

In 2016, a major publication presented a global model of aerosol formation based entirely on CLOUD laboratory measurements [6]. This study established the main processes for new particle formation throughout the troposphere, which is responsible for about half of all cloud seed particles, and it conclusively showed that observed particle concentrations can only be explained if additional molecules, either organic compounds or ammonia, participate in nucleation. It also found that ionization by cosmic rays contributes to nearly one-third of all particles formed, though small changes in cosmic rays over the solar cycle do not significantly affect today's polluted climate. This work is expected to reduce the variation in predicted global temperatures by climate models, thereby sharpening climate predictions.

Another 2016 publication from CLOUD reported the discovery that aerosol particles can form purely from organic vapors produced naturally, a process known as pure biogenic nucleation [7]. This process was found to be the primary source of particles in the pristine pre-industrial atmosphere, which suggests that estimates of aerosol radiative forcing from anthropogenic activities, and consequently modeled climate sensitivities, might need to be revised downwards, and more recently, CLOUD experiments have shown that iodic acid can form aerosol particles even without sulfuric acid, with nucleation rates highly dependent on temperature. Other research has explored the role of sesquiterpenes in biogenic new particle formation [8], [9] .

A 2023 synthesis summarized the current understanding, emphasizing the crucial roles of condensable vapors like highly oxygenated organics and iodine oxoacids, along with stabilizers such as ammonia, amines, and ions from galactic cosmic rays [10]. This synthesis aims to create a mechanistic foundation for air quality and climate models.

Overall, the CLOUD experiment has provided unprecedented insights into cloud formation processes, highlighting the complex interplay of various atmospheric compounds, temperature, and cosmic rays, and its findings are crucial for developing more reliable and realistic climate models, ultimately aiming to reduce uncertainties in climate projections and clarify the extent of the anthropogenic contribution to present climate change.

1.2 The Data Acquisition System (DAQ)

In scientific research and industrial settings, a Data Acquisition System (DAQ) serves as a crucial link, with its central function being to combine data from diverse, autonomous, and often heterogeneous sources into a single, coherent, and understandable database. These systems are designed to offer high adaptability and flexibility, enabling rapid setup and agile responses to any changes in instruments or experimental parameters, and by doing so, they facilitate online monitoring of measurements in near real-time, with minimal delays of a few seconds, and ensure the secure archiving of raw data, monitoring databases, and of course, processed data for future analyses.

Within the specific context of the CLOUD experiment at CERN, the DAQ system fulfills an equally vital role, but with challenging requirements tailored to the complexity of the research [11]. As mentioned, the CLOUD experiment investigates the nucleation and growth of aerosol particles and their clustering into cloud droplets under controlled atmospheric conditions, and utilizes an ultraclean 26.1 m^3 chamber and approximately 50 advanced instruments connected via sampling probes. The composition of these instruments changes on each experimental campaign according to the stated goals, necessitating that the DAQ system be quickly configurable, often within a single installation week, and capable of adjusting to modifications throughout the campaign,that typically occur once or twice a year, lasting about three months, during which data are collected continuously.

Data collection is performed from individual instrument files that are updated by each instrument's

autonomous data collection system, and the main storage server acts as the central collection point for this raw data. Data synchronization is primarily based on the rsync tool and protocol, which optimizes transfer by identifying and moving only the changes in files, thereby reducing network traffic and speeding up the process, and rsync clients are configured to operate in short, repetitive cycles, ensuring permanent synchronization and data recoverability even in the event of central DAQ failures. While most instruments write to ASCII files, some, like time-of-flight mass spectrometers, produce binary files that require special handling, such as a web server solution from the manufacturer (TOFWERK AG [12]) to provide remote data access via JSON-based network communication for low-latency monitoring. Data from the slow control system are published through CERN's Data Interchange Protocol (DIP) [13] and collected by dedicated processes.

One of the greatest challenges faced by the DAQ system is managing the diversity of instruments, since these instruments operate on multiple platforms, including various Linux distributions and Microsoft Windows versions (ranging from Windows XP to 10), with some even using real-time operating systems. Furthermore, instruments employ a number of frameworks for readout, storage, and analysis, spanning open-source solutions (such as C/C++, Python, R, MySQL, Qt) and proprietary and closed-source software (such as MATLAB, LabView, Igor, and TOFWERK software), so the difficulty is compounded by the fact that many instrument DAQ systems were developed by third parties who may not be available to make changes to the software or data format.

For online monitoring, a centralized MySQL database management system (DBMS) is hosted on a dedicated server within the CLOUD network, exclusively for the CERN control room. The database schema is normalized, with data organized into tables primarily indexed by measurement time, recorded as a Unix timestamp in milliseconds, and C++/Qt-based software, developed by the collaboration, handles data ingestion, supporting various text-based file formats and enabling efficient processing through parallel threads. Real-time monitoring is displayed by a CLOUD-specific C++/Qt application that updates data from the database with a 30-second refresh rate, and to manage load from data-intensive instruments, a maximum ingestion time is set for each instrument, ensuring efficient resource allocation. Preprocessing and calculation of physics quantities, such as particle nucleation rates, are also performed on the main server to guide experiments, and Users can also interact with the system by updating run tables and logbooks to comment on data and track experimental conditions.

A critical challenge in data acquisition is the precise synchronization of the multiple instrument clocks. Due to infrequent NTP polling, lack of administrator privileges, or incorrect timezone configurations, computer clocks can drift by more than 100 seconds from Coordinated Universal Time (UTC) during a campaign, so the implemented solution involves using the open-source Simple Network Time Protocol (SNTP) tool, NetTime [14], continuously recording the difference between UTC and each computer's clock. These time corrections are centrally collected with the raw data and also applied to the processed data offline, allowing for synchronization to approximately 10-ms precision.

Data are stored and replicated in multiple locations, including CERN and the University of Lisbon, where the databases are rebuilt locally from the raw data and metadata, an approach that does not use SQL database mirroring and provides independent access to online monitoring. At CERN, data are transferred to the EOS (open source distributed disk storage system) storage system and accessible via CERNBox, offering access to raw and processed data through a web browser or specific clients, and the Lisbon server acts as a primary distribution point for monitoring data outside the CLOUD network, facilitating easier access without the need for port forwarding or individual logins. Remote monitoring via these sites typically has a little delay of up to 10 minutes compared to the CLOUD control room.

The CLOUD Slow Control (CSC) system, redesigned in 2018, is responsible for managing and

monitoring the CLOUD facility with an emphasis on standard control, the use of industrial components, and hardware protection. This system is made up of a few layers: the field layer (hardware devices and sensors), the control layer (PLCs and interlocks), and the Supervisory Control and Data Acquisition (SCADA) system, and CSC data are rapidly published (every 1-10 seconds) on the CERN network via the Data Interchange Protocol (DIP) for access by the DAQ system, and the full dataset is archived into an Oracle database.

For the future, significant upgrades are underway, including a redesigned fiber optics network infrastructure with optimized routing. A third replication source at CERN, physically close to the experimental area, will be implemented to leverage the new network and reduce setup times, also eliminating a single point of failure for external monitoring. Furthermore, updates to CLOUD applications will include optional automatic transfer of changes to the external database to streamline handling frequent configuration updates.

1.3 Key Instrumentation for Data Collection

The successful operation of the CLOUD experiment and the acquisition of high-quality scientific data depend on a sophisticated suite of measurement instruments. These instruments are designed to provide real-time, precise measurements of a wide range of physical and chemical parameters within the chamber, including particle size distributions, chemical composition, and concentrations of various gases. The data collected by these instruments are the raw material for all subsequent analysis and are fundamental to achieve the scientific objectives of the experiment. The following subsections provide a detailed description of the most important instruments used in the CLOUD campaigns.

1.3.1 Condensation Particle Counter (CPC)

Condensation Particle Counters (CPCs) are instruments designed to detect ultrafine particles, which are too small to be visible through optical techniques, generally having a detection limit of about 0.1 μm. These devices operate by growing small aerosol particles in a sample to sizes that can be optically detected. Their use dates back to John Aitken in 1888, who developed an instrument to measure the number concentration of particles above 0.02 μm, usually referred to as "Aitken nuclei". Over the last century, CPC designs have been improved, resulting in several types.

The operating principle of a CPC is based on three fundamental processes [15]: supersaturation the working fluids, growth of particles by condensation of vapors, and detection of the resulting particles.

To induce the growth of ultrafine particles by condensation, CPCs create a volume of vapor supersaturation. This process is essential due to the Kelvin effect, which describes how the saturation vapor pressure, p_d, on a droplet's surface is greater than that for a flat liquid surface p_s, and is a function of the particle diameter, d_p [16]. This relationship is expressed by Equation 1.1:

$$\frac{p_d}{p_s} = \exp\left(\frac{4\nu\gamma}{RTd_p}\right) \tag{1.1}$$

,

where ν is the molar volume of the liquid, γ is the surface tension, R is the gas constant, and T the temperature.

Practically, the smaller the particle diameter, the greater the saturation ratio, SR = p_0/p_s, required to initiate condensation. The minimum size needed to initiate condensation at a given SR level in a CPC (typically between 1.5 and 3) is called the "Kelvin diameter", $d*$, and particles larger than $d*$ will grow,

while those smaller will not. The presence of nonvolatile solute particles (such as hygroscopic particles) in a volatile solvent (like water vapor) can lower the saturation vapor pressure. This effect, described by Raoult's law for ideal solutions, competes with the Kelvin effect, which can lead to lower detection limits for hygroscopic particles, like sodium chlorides, when water is the working fluid. Additionally, if a particle carries an electrostatic charge, the vapor pressure at its surface and the saturation ratio required for condensation are also reduced.

Regarding Droplet Growth, the expansion of particles by condensation can be calculated by considering several factors, including the diffusion coefficient of the condensing vapor, the vapor pressure and temperature in the surrounding gas, and the molar volume of the liquid [17]. For particles with a diameter smaller than 0.1 μm, a Fuchs correction becomes important in these calculations. Additionally, the temperature on the droplet surface increases because of the latent heat of condensation [18], which describes the growth of a homogeneous liquid droplet or even the growth of insoluble particles with wettable surfaces.

For Droplet Detection, the goal is to grow particles to a near-uniform size, typically between 2 and 15 μm, which makes them large enough to be detected by optical means. It has been observed that the final droplet size is very uniform, and while it depends on the system configuration, operating conditions, and particle number density, it is practically independent of the initial particle size distribution [19]. Historically, early methods for detection included using a glass magnifier to count particles [20] or mounting a camera on the CPC to photograph them [21], [22], and later photoelectric CPCs used light-extinction methods [23], while more recent models employ light-scattering techniques, where either individual particle signals are counted or the intensity of scattered light indicates concentration.

Historically, these instruments have evolved from John Aitken's [20] original designs and are now categorized into several types based on how they induce supersaturation. One type is the Expansion-Type CPC, which operates cyclically, where an aerosol stream is first humidified with water vapor until saturation, then rapidly cooled by either volume expansion or pressure release within an expansion chamber, which creates supersaturation, causing water vapor to condense on particles [21], [22], [23]. A key disadvantage of this type is its cyclic flow, making it incompatible with applications requiring a steady-state flow.

Another type is the Conductive Cooling CPC, which provides a continuous flow with a saturator, a condenser, and a particle detector. The aerosol goes through an alcohol reservoir (saturator) maintained at high temperatures, becoming saturated with the working fluid. It then enters a condenser tube that is kept at a lower temperature, where gas cooling by conduction results in supersaturation in the aerosol stream, which grows particles to about 12 μm, that are then detected by a light-scattering system [24], [25], [26]. Ultrafine Condensation Particle Counters (UCPCs) are optimized versions of this type, capable of detecting particles as small as 2.5 nm [27].

The Differential Diffusion CPCs are continuous-flow instruments that use water as the working fluid, addressing concerns associated with alcohol fumes. Their principle relies on the differential rates of heat and mass transport, where a cold aerosol flow first passes through a cool, wet preconditioner, becoming saturated with water vapor, which then enters a heated, wet-walled growth tube, and because water vapor diffuses to the centerline faster than heat is transferred from the walls, maximum supersaturation is achieved along the flow's centerline, activating particle growth [28], [29], [30]. The enlarged droplets are then individually counted as they go through a focused laser beam.

Finally, Mixing-Type CPCs (MCPCs) achieve condensation by rapidly combining two streams at different temperatures, where both streams can be saturated aerosol flows, or one can be a saturated vapor with a high boiling point, such as propylene glycol or butanol. This fast mixing in a nozzle assembly

results in condensation within a continuous, steady-state flow. MCPCs offer faster response times and minimum diffusional loss of aerosol particles because the aerosol stream does not need to pass through a saturator, making them suitable for rapid-scanning measurements [17], [31], [32].

1.3.2 Nano Scanning Mobility Particle Sizer (nSMPS)

The NanoScan SMPS is a portable analytical instrument designed to measure nanoparticle size distributions and concentrations, and it can also monitor concentration at a single diameter with high time resolution [33]. This battery-operated sizer provides an affordable method for nanoparticle measurement in various research and industrial hygiene applications. The instrument operates using an electrical mobility based sizing technique, which is the benchmark method for sizing airborne nanoparticles, similar to a Scanning Mobility Particle Sizer Spectrometer and validated by the National Institute of Standards and Technology (NIST). Electrical mobility is a convenient method for measuring particle size since it can be directly measured and is a first-principle function of size.

The NanoScan SMPS incorporates five primary components into a compact chassis. First, an inlet conditioner is used to remove larger particles, typically with a 500 nm cutpoint, which is crucial to prevent clogging and to avoid issues where larger particles with multiple charges might have the same electrical mobility as smaller, singly charged particles which in turn would degrade the instrument's resolution and accuracy. Next, an aerosol charger prepares the particles for sizing.

The NanoScan SMPS employs a patented unipolar "Corona-Jet" charger, which charges nanoparticles [34]. This specific charger design significantly increases nanoparticle counting efficiency by charging a higher percentage of nanometer-sized particles, charging approximately ten times more 10 nm particles compared to traditional bipolar chargers, and it also eliminates the logistical challenges associated with radioactive materials often used in older charging methods. In order to the instrument accurately compute particle diameter from electrical mobility, the number of charges per particle must be known, and it is important to note, of course, that only charged particles are measured.

The unipolar charger ensures a well-characterized and repeatable induced charge distribution. Following the charger, a Radial Differential Mobility Analyzer (DMA) is used to classify particles by their electrical mobility [35], where a charged aerosol particle experiences an electric field, causing it to move through the gas, while simultaneously encountering an opposing drag force from the gas flow. The Radial DMA design is beneficial because it features shorter particle residence times, leading to greater nanoparticle transmission efficiencies [36]. There, particle-free air enters a circular channel from the bottom, achieving laminar flow after passing through a flow straightener, and the polydisperse sample aerosol is introduced tangentially from the top. An electric field is created between the grounded top plate and a negatively charged bottom plate, so the DMA operates similarly to a bandpass filter in electronics; at each voltage setting, it outputs a "slice" of aerosol corresponding to a specific electrical mobility diameter. The size-classified aerosol then exits through the bottom center port and proceeds to the particle counter. Excess flow exits through the top center port, and both the sheath and excess flows are part of a closed, recirculating loop designed to optimize the laminarity of the flow field.

Downstream of the DMA, a Condensation Particle Counter (CPC) acts as the particle counter/detector, measuring the number of particles in each size bin. The NanoScan SMPS uses an isopropanol-based CPC, which provides accurate measurements across both high and low concentrations, while using a working fluid that is acceptable in workplace environments.

Finally, control software manages the instrument's operation, data analysis, and data logging. This can be done directly through the instrument's color touch screen, which offers graphical data display and

onboard data storage, or through the NanoScan Manager software for expanded analysis capabilities.

The NanoScan SMPS is designed for ease of use and is a versatile analytical tool, as it is highly valuable for diverse applications because it can present size distribution data weighted as number, surface area, or mass. Its uses span general applied research, indoor/outdoor air quality, nanotechnology, combustion and emissions, mobile studies, health effects/inhalation toxicology, and industrial hygiene (including worker exposure and source tracking).

1.3.3 Particle Size Magnifier (PSM)

The Airmodus Particle Size Magnifier (PSM) is a versatile tool used with a Condensation Particle Counter (CPC) to detect and analyze aerosol particles and clusters as small as approximately 1 nm in mobility diameter, especially for measurements of sub-3 nm particles, and was originally developed at the University of Helsinki to study atmospheric new particle formation and related processes [37]. The combined system is often referred to as a nano-Condensation Nucleus Counter (nCNC), with the PSM specifically being the pre-conditioner part where particles are grown before counting.

The fundamental principle behind the PSM, like all Condensation Particle Counters (CPCs), involves creating a supersaturation of a working fluid to grow sampled aerosol particles by condensation until they are large enough to be detected optically. The Airmodus PSM specifically operates using diethylene glycol (DEG) as its working fluid, which is considered highly suitable for activating sub-3 nm particles [38].

Inside the PSM, the process begins by mixing two flows: the sample flow containing the aerosol particles and a separate flow saturated with DEG vapor. To minimize particle losses, the PSM's inlet flow rate is set at a relatively high 2.5 liters per minute (lpm). This sample flow is then mixed turbulently with a heated flow originating from the saturator, a heated sintered metal tube, that is kept at 70–85 °C, where DEG evaporates into a filtered and compressed air flow (the saturator flow). The rate of this saturator flow can be adjusted between 0.1 and 1.3 lpm, which directly influences the instrument's lowest detectable size, or "cut-off diameter," typically ranging from about 1 to 4 nm.

Following the mixing section, the particles enter a cylindrical growth tube, where the wall temperature is maintained at 1–5 °C. It is within this growth tube that the largest DEG supersaturation along the particle trajectories occurs, leading to particle activation and growth. Since these grown particles are still not large enough for optical counting, they are then sampled from the centerline of the growth tube (at 1.0 lpm) and sent to a downstream Condensation Particle Counter (CPC) for further growth and counting.

The "cut-off diameter" of the PSM, defined as the size at which the instrument detects half of the particles, is influenced by the previously mentioned Kelvin effect: smaller particles require higher supersaturation to activate [39]. Therefore, by changing the supersaturation, which is controlled by the saturator flow rate and temperature settings, the lowest detectable size can be adjusted. It's essential to recognize that the composition of the particles also affects their activation probability and, consequently, the cut-off size, introducing uncertainty since ambient aerosol composition is often unknown. Therefore, the reported particle diameters are considered "activation equivalent diameters," assuming the measured particles activate similarly to the calibration aerosols.

1.3.4 Chemical Ionization Mass Spectrometer (CIMS)

Chemical Ionization (CI) mass spectrometry, initially proposed by Burnaby Munson and Frank H. Field in 1966, is a soft ionization technique that is fundamental in mass spectrometry, particularly for

the identification and quantification of organic molecules [40]. Unlike electron impact (EI) ionization, which typically imparts a high energy of 70 eV to the analyte, CI applies significantly less energy, usually ranging from 1 to 4 eV. This gentler process results in less fragmentation of the molecular ion and thus produces a simpler mass spectrum. The reduced fragmentation means that the molecular ion peak, crucial for molecular weight determination, is often more prominent or even the base peak, especially for molecules that are fragile or fragment excessively under EI conditions.

The operation of a CI mass spectrometer involves several key steps within a specialized ion source chamber, which is maintained at a relatively higher pressure, typically around 10^{-3} mbar, to facilitate ion-molecule reactions. First, a reagent gas (such as methane, isobutane, or ammonia) is released into this chamber in a large excess compared to the analyte sample. Electrons, generated externally from a heated filament (often made of tungsten, rhenium, or iridium) and typically accelerated to energies between 200 and 1000 eV (or sometimes 70 eV), are then introduced into this high-pressure environment. Due to its overwhelming abundance, the electron beam preferentially ionizes the reagent gas molecules rather than the analyte molecules.

These initial reagent ions then experience a sequence of secondary ion-molecule reactions within the source chamber to form more stable reagent ions. For instance, when methane is used, the primary methane radical cation ($CH_4^+\bullet$) reacts with another methane molecule to generate a methenium ion ($CH_5^+\bullet$) and a methyl radical ($CH_3\bullet$). This stable methenium ion, or other similar stable ions from different reagent gases (e.g., NH_4^+ from ammonia or $C_4H_9^+$ from isobutane), acts as a superacid and a source of protons in the gas phase, and when the analyte molecules (M) are introduced into the same chamber, they react with these stable reagent ions, predominantly through a proton transfer reaction. This reaction leads to the formation of a protonated analyte, which is the conjugate acid of the analyte. For this proton transfer to be efficient and energetically favorable (exothermic), the proton affinity of the reagent ion must be less than that of the analyte molecule. The exothermicity of this reaction, which is the difference in proton affinity between the neutral reagent molecule and the neutral analyte molecule, largely determines the extent of fragmentation of the analyte ions.

The resulting protonated analyte is relatively stable, enduring less fragmentation compared to ions produced by high-energy electron impact ionization. This stability results in a prominent M^+ peak in the mass spectrum, providing critical information for determining the molecular mass of the compound, especially for delicate molecules where an M^+ peak might be weak or absent in EI spectra. The careful selection of the reagent gas allows for some control over the degree of fragmentation and, consequently, the amount of structural information obtained.

While CI offers significant advantages such as increased sensitivity (due to ion signal concentration in fewer peaks) and the ability to identify intact molecular ions, it does have limitations, as it is primarily limited to volatile samples that can be easily vaporized in the ion source. Additionally, the reduced fragmentation means that CI typically provides less structural information compared to EI, and the spectra can also be sensitive to source conditions like pressure, temperature, and impurities, which makes it challenging to generate universal libraries of CI spectra for compound identification [41].

Despite these limitations, CI is a powerful tool widely used for the identification, structural elucidation, and quantification of organic compounds across various fields, including biochemistry, environmental analysis, and forensic investigations. There are also variants like Negative Chemical Ionization (NCI) [42], which is selective for analytes that can stabilize a negative charge (such as acidic or electronegative compounds), and Atmospheric Pressure Chemical Ionization (APCI) [43], often coupled with HPLC and operating at atmospheric pressure for enhanced sensitivity and ionization efficiency.

1.4 Research Objectives and Thesis Structure

This thesis aims to address key challenges in understanding atmospheric new particle formation and the dynamics of sulfuric acid concentration within the controlled environment of the CLOUD experiment. The overarching research objectives are twofold: first, to enhance the accuracy and robustness of new particle formation rate calculations through significant methodological advancements, and second, to develop a reliable predictive framework for forecasting sulfuric acid concentration slopes.

Chapter 2 specifically focuses on the methodology and comparative analysis of new particle formation rates. It begins by establishing the fundamental principles of particle formation and growth and thoroughly describes the various processes that influence observed particle number concentrations, including the dilution, wall, and coagulation sinks. A core objective of this chapter is to detail the improvements implemented in a new Python-based computational script, which includes rectifying a dimensionally inconsistent coagulation sink, integrating dynamic flow data for a more accurate dilution sink, and selecting the Savitzky-Golay filter for derivative calculations. The chapter culminates in a comparative analysis of the results from this new script against previous R and Matlab implementations, underscoring the enhanced robustness and scientific consistency of the updated methodology, which is slated for adaptation into real-time visualization at the CLOUD chamber.

Building on this, Chapter 3 is dedicated to the modeling and prediction of sulfuric acid concentration. Its primary objective is to transcend purely descriptive analysis by establishing a robust, data-driven framework for forecasting the concentration slope of sulfuric acid within the CLOUD chamber experiments, utilizing data from the CLOUD16 campaign. The chapter outlines its approach by first providing a foundational theoretical background on sulfuric acid formation, followed by a detailed description of the necessary data preparation and cleaning procedures. The modeling strategy involves a two-tiered approach: an initial explanatory phase to understand key drivers and relationships, emphasizing interpretability and identifying physical and chemical influences, and a central predictive modeling phase focused on developing and optimizing both a regularized linear model (Elastic Net) and a powerful nonlinear ensemble method (XGBoost) to capture complex data patterns.

The chapter concludes with a comprehensive comparative analysis of the final, optimized models, using quantitative metrics like Root Mean Squared Error and qualitative assessments from diagnostic plots to select the most suitable model for practical predictive applications. Ultimately, this chapter aims to provide a valuable tool for understanding and anticipating atmospheric nucleation events.

Chapter 2

Methodology and Comparative Analysis of New Particle Formation Rates

This chapter details the theoretical background and methodological advancements related to the calculation of new particle formation rates within the CLOUD experiment. It first establishes the fundamental principles governing particle formation and growth, followed by a comprehensive description of the various processes influencing the observed particle number concentrations, including the dilution, wall, and coagulation sinks. Subsequently, the chapter outlines the improvements implemented in a new Python-based computational script, explaining the correction of a dimensionally inconsistent coagulation sink, the use of dynamic flow data for the dilution sink, and the selection of the Savitzky-Golay filter for a more accurate derivative calculation.

Finally, the chapter presents a comparative analysis of the results from the new script against previous R and Matlab implementations, highlighting the enhanced robustness and scientific consistency of the updated methodology, which will be adapted for real-time visualization at the CLOUD chamber.

2.1 Introduction to New Particle Formation (NPF) and Formation Rate Concepts

2.1.1 The Atmospheric Significance of New Particle Formation

As fundamental components of the atmosphere, aerosol particles influence both air quality and climate, as they affect the Earth's radiative balance directly by scattering and absorbing solar radiation (aerosol-radiation interactions), and indirectly by serving as cloud condensation nuclei (CCN), which controls cloud formation, albedo, lifetime, and precipitation (aerosol-cloud interactions). [44], [45]. A significant source of these particles, particularly in unpolluted environments and the free troposphere, is the process of new particle formation in which gaseous precursor molecules cluster together to form stable particles, typically in the sub-3 nm size range [46]. These newly formed particles can then grow by condensation of available vapors and coagulation with other particles, eventually reaching sizes large enough to act as CCN [47].

Beyond their climatic role, NPF events also significantly contribute to ambient particulate matter concentrations, particularly ultrafine particles, which, due to their small size, can penetrate deep into the respiratory system, posing potential adverse health effects [48], so understanding the processes that govern NPF, including the identification of key precursor gases, the mechanisms of initial cluster formation, and the subsequent growth pathways, is crucial for comprehensive air quality assessments and the

development of effective mitigation strategies.

However, the complexity and variability of atmospheric conditions make direct observations of NPF challenging. Laboratory and controlled chamber experiments provide a crucial environment to isolate and study specific NPF mechanisms under well-defined conditions [49], that allow for precise control over gas-phase concentrations, temperature, humidity, and ion levels, enabling the investigation of various NPF pathways (e.g., sulfuric acid-amine nucleation, highly oxygenated organic molecule (HOM) nucleation, ion-induced nucleation). This controlled approach is essential for reducing uncertainties in atmospheric models that seek to predict aerosol populations and their climate impacts.

2.1.2 Defining the Particle Formation Rate

A key parameter characterizing the intensity of an NPF event is the formation rate (J), which is defined as the number of new particles formed per unit volume per unit time that grow to exceed a specific detection threshold diameter ($d_{p,min}$). It is typically expressed in units of $\mathrm{cm}^{-3}\mathrm{s}^{-1}$. Mathematically, the formation rate can be conceptualized as follows:

$$J_{d_{p,\min}} = \frac{dN(d_p > d_{p,\min})}{dt} + \mathrm{Losses}(d_p > d_{p,\min}) \tag{2.1}$$

where $\frac{dN(d_p > d_{p,\min})}{dt}$ represents the observed increase rate in the concentration of particles greater than $d_{p,\min}$, and $\mathrm{Losses}(d_p > d_{p,\min})$ accounts for the removal of particles in this size range due to processes such as coagulation, deposition, and dilution. The formation rate is a critical parameter that quantifies new particle sources, and by determining aerosol concentrations and CCN, it influences cloud properties and Earth's radiative balance, making its accuracy essential for climate models. Furthermore, ultrafine particles formed during NPF can have adverse health effects, so understanding their formation rates is crucial for air quality assessments and mitigation strategies, so measuring J under varying atmospheric conditions and with different precursor molecules also provides insights into the chemical and physical processes driving NPF.

In atmospheric field measurements, determining J requires continuous monitoring of particle number size distributions using instruments like Scanning Mobility Particle Sizers (SMPS) or Neutral cluster and Air Ion Spectrometers (NAIS). The calculation involves analyzing the temporal evolution of the size distribution, taking into account the growth and losses of the particles.

The particle formation rate in the CLOUD experiment represents a primary output parameter, derived from the measured particle number size distributions obtained via the data acquisition (DAQ) system. However, the calculation of J from CLOUD data is complicated by several factors intrinsic to the chamber that result in significant particle loss processes: wall deposition, where particles adhere to the chamber surfaces; coagulation, involving particle-particle collisions that reduce total number concentration; and dilution, resulting from the continuous flow operation and sampling. Each of these loss mechanisms is size-dependent and necessitates precise characterization and correction to ensure an accurate determination of the true particle formation rate.

Therefore, calculating the formation rate in CLOUD involves sophisticated analysis techniques. To determine particle formation rates, an aerosol general dynamic equation (GDE) is solved by accounting for NPF, growth, coagulation, wall losses, and dilution. Following the standardized protocols of [50], J is calculated by combining the time derivative of the particle concentration with these loss terms, ensuring consistent and comparable results across chamber experiments. The present work on remaking a script to calculate the formation rate within the CLOUD DAQ team is a direct contribution to this essential

analysis step.

2.1.3 Fundamental Processes Influencing Particle Number Concentration

2.1.3.1 Dilution Sink

In the continuous flow mode of the CLOUD chamber, a constant flow of ultra-clean synthetic air is introduced into the chamber, which serves multiple purposes: maintaining a stable pressure within the chamber, compensating for the volume of air sampled by the various analytical instruments, and ensuring a controlled, low background environment.

This continuous influx of clean air and the simultaneous removal of chamber air by sampling instruments leads to a dilution of the particle population within the chamber, and particles are effectively removed from the chamber volume at a rate proportional to the rate at which the chamber air is exchanged. This dilution effect reduces the particle number concentration over time and must be accurately accounted for in the formation rate calculation.

The loss rate due to dilution, S_{dil}, for particles above a certain size threshold, d_p, is directly proportional to the total particle number concentration above that size, $N_{>d_p}$ and the dilution coefficient , k_{dil}, that represents the fractional rate at which the air in the chamber is replaced. It is determined by the total volumetric flow rate of synthetic clean air entering the chamber, $\text{Flow}_{\text{synthetic air}}$, and the known volume of the CLOUD chamber, V_{chamber}. The relationship is expressed by the following equations:

$$S_{\text{dil}} = N_{>d_p} \cdot k_{\text{dil}} [\text{cm}^{-3}\text{s}^{-1}], \tag{2.2}$$

$$\text{with } k_{\text{dil}} = \frac{\text{Flow}_{\text{synthetic air}}}{V_{\text{chamber}}} \ [\text{s}^{-1}] \tag{2.3}$$

The k_{dil} is a well-defined parameter for each experiment based on the controlled flow settings and chamber geometry. Accurate measurement and control of the synthetic air flow rate are therefore essential for precisely determining the dilution loss rate, which provides a baseline removal term that is independent of particle properties like size or composition, but dependent on the overall particle concent ration.

2.1.3.2 Wall Sink

Wall deposition is a critical particle loss mechanism in experimental chambers, particularly for sub-3-nm aerosol particles, and must be accurately accounted for when quantifying particle dynamics during NPF events.

Unlike atmospheric conditions where coagulation onto existing particle populations is the primary sink for newly formed particles, particle losses to chamber walls are typically much more significant in chamber experiments because the existing particle population is usually absent or minimal, so given that particle losses are high, considering these losses is essential for accurate measurements, and a unified method for correcting these losses has been developed to ensure comparability of results across different chamber experiments.

The main mechanism responsible for particle deposition to chamber walls is diffusional loss, [51], which is driven by the Brownian motion of particles, leading to their migration towards surfaces where the concentration is lower. The cleanliness and material of the chamber walls also play a role; rough

surfaces can enhance deposition, and non-conductive materials like Teflon can cause efficient ion scavenging. Conversely, robust and electrically conductive materials like stainless steel, which the CLOUD chamber is made of, are more suitable for the study of ionic processes.

The wall sink, S_{wall}, representing the diffusional loss rate of particles to the chamber walls, is a crucial term in calculating the total particle formation rate. The calculation for the wall loss rate is expressed as:

$$S_{\text{wall}}(T) = \sum_i N_{d_{p_i}-d_{p_{i+1}}} \cdot k_{\text{wall}}(d_p, T) \ [\text{cm}^{-3}\text{s}^{-1}] \tag{2.4}$$

where $N_{d_{p_i}-d_{p_{i+1}}}$ is the particle number concentration in a specific size bin, and k_{wall} is the wall-loss coefficient for particles of a given particle diameter, d_p, at a specific temperature, T.

The wall-loss coefficient, k_{wall}, is a factor that in previous chamber experiments [52], was determined experimentally and adjusted based on the theoretical temperature dependence of the diffusion coefficient ($D \propto (T/Tref)^{1.75}$)[53] and the wall loss dependence on the diffusion coefficient ($k_{\text{wall}} \propto D^{0.5}$)[54]. While the protocol for CLOUD chamber data analysis recommends an equation that utilizes an experimentally determined factor F, such as:

$$k_{\text{wall}}(d_p, T) = F \cdot \left(\frac{T}{T_{\text{ref}}}\right)^{0.875} \cdot \left(\frac{d_{p,\text{ref}}}{d_p}\right) \ [\text{s}^{-1}], \tag{2.5}$$

for this specific work, the experimentally determined factor F was not available, so k_{wall} was calculated directly based on its proportionality to the square root of the diffusion coefficient of the particles, D. This approach is founded on the principle that diffusional losses are the most critical mechanism for particles with diameters smaller than 100 nm, with the loss coefficient being inversely proportional to the mobility diameter in these size ranges. The diffusion coefficient is, in turn, calculated for each particle mobility diameter, d_p, and temperature, T, using the Stokes-Einstein equation [55], which describes the diffusion of a particle in a gas phase:

$$D(d_p) = \frac{k_B T C_C(d_p)}{3\pi\eta d_p} \ [\text{cm}^2\text{s}^{-1}] \tag{2.6}$$

where k_B is the Boltzmann constant, T is the absolute temperature, η is the gas viscosity, C_C is the Cunningham correction factor, which is essential for small (sub-micrometer) particles where "slip" at the particle surface is significant [56]. This factor depends on particle diameter and the gas mean free path, which in turn depends on temperature and pressure.

Through this methodology, the temperature and size dependencies are implicitly incorporated into the calculation of k_{wall} via D, allowing for a consistent estimation of the wall loss rate.

2.1.3.3 Coagulation Sink

Coagulation is a fundamental atmospheric process where aerosol particles collide and adhere, leading to the formation of larger particles and a decrease in the total particle number concentration. This process acts as a significant loss mechanism for newly formed particles, as they are scavenged by larger, more abundant particles. To accurately determine the new particle formation rate, it is essential to quantify and account for the rate at which particles of a given size d_p are lost due to coagulation.

The fundamental parameter governing coagulation is the coagulation coefficient (or coagulation kernel), $k_{\text{coag}}(d_p, d_p')$, which quantifies the collision frequency between two particles of diameter d_p and d_p', and depends on factors such as particle size, temperature, pressure, and the medium's viscosity. The

Fuchs interpolation formula [57] is commonly used to calculate $k_{\text{coag}}(d_p, d'_p)$ across different regimes (continuum, free molecular, and transition), and its full expression is given by:

$$k_{\text{coag}}(d_p, d'_p) = 2\pi(D(d_p) + D(d'_p))(d_p + d'_p) \times \left[\frac{d_p + d'_p}{d_p + d'_p + 2(g(d_p)^2 + g(d'_p)^2)^{1/2}} + \frac{8(D(d_p) + D(d'_p))}{(c(d_p)^2 + c(d'_p)^2)^{1/2}(d_p + d'_p)} \right]^{-1} \quad [\text{cm}^3\text{s}^{-1}] \tag{2.7}$$

where

$$c(d_p) = \left(\frac{8k_B T}{\pi m(d_p)} \right)^{1/2} \quad [\text{cm s}^{-1}], \tag{2.8}$$

$$l(d_p) = \frac{8D(d_p)}{\pi c(d_p)} \quad [\text{cm}], \tag{2.9}$$

$$g(d_p) = \frac{1}{3D(d_p)l(d_p)}[(d_p + l(d_p))^3 - (d_p^2 + l(d_p)^2)^{3/2}] - d_p \quad [\text{cm}], \tag{2.10}$$

where k_B is the Boltzmann constant, T is the absolute temperature , $m(d_p)$ is the mass of a particle with diameter d_p (kg), $D(d_p)$ is its diffusion coefficient, and $c(d_p)$ is its mean thermal speed. Additionally, $l(d_p)$ represents a characteristic length parameter, often related to the particle's mean free path in the gas, and $g(d_p)$ is a dimensionless correction factor that accounts for the particle's kinetic motion and interaction with the surrounding gas.

The coagulation sink, S_{coag}, for particles of a specific diameter d_p is defined as the rate at which particles of size d_p are removed from the atmosphere (or chamber volume) by coagulating with all other particles present in the system. Consistent with other loss terms in this work, this sink represents the change in number concentration per unit time for a given size bin. In a discrete particle size distribution, it is expressed as [16]:

$$S_{\text{coag}}(d_p) = N_{d_p} \sum_{d'_p = d_{p,\min}}^{d'_p = \max} K(d_p, d'_p) N_{d'_p} \quad [\text{cm}^{-3}\text{s}^{-1}] \tag{2.11}$$

where N_{d_p} is the number concentration of particles in the size bin of interest (diameter d_p), and $N_{d'_p}$ is the number concentration of particles in any other size bin centered at d'_p.

It is important to acknowledge that the term "coagulation sink" is sometimes used in atmospheric literature to refer specifically to a first-order loss rate constant for particles (with units of $[\text{s}^{-1}]$), where it would represent $S_{\text{coag}}(d_p)/N_{d_p}$ as defined above. However, for the purpose of this thesis, and to maintain consistency with the dimensional definitions of other loss terms (e.g., dilution loss S_{dil}), the coagulation sink, S_{coag} will strictly refer to the loss rate of particles in units of $[\text{cm}^{-3}s^{-1}]$. This precise definition is essential for the accurate and consistent implementation of the formation rate calculation, particularly when comparing or integrating with other models or experimental setups.

2.2 Methodological Advancements in Formation Rate Calculation

2.2.1 Background and Limitations of the Pre-existing Scripts

The accurate determination of new particle formation rates from CLOUD chamber experiments is a complex computational task, relying on robust tools to interpret intricate particle dynamics data, and over time, the methodologies employed have evolved to meet increasing demands for precision and transparency in atmospheric aerosol research. Prior to the present work, the CLOUD collaboration utilized two primary computational scripts for J derivation. While these earlier implementations provided foundational capabilities, their inherent characteristics presented opportunities for significant refinement and advancement in methodological clarity and computational rigor, which ultimately motivated the comprehensive re-implementation undertaken in this thesis.

An earlier script, primarily developed in R, demonstrated certain complexities in its structure and documentation. Its codebase was, at times, difficult to fully interpret, posing challenges for detailed auditing and maintenance by collaborators. Instances were noted where control structures, such as while loops, exhibited behavior that did not fully align with anticipated logical pathways, highlighting areas for enhanced robustness in computational flow. Furthermore, certain constants integrated into key calculations, most notably within the diameter-to-mass conversion function, lacked explicit documentation regarding their derivation or precise impact on the derived particle properties. This ambiguity presented an opportunity for increased transparency and a more rigorous justification of all included parameters. This R script also employed denoising functions for particle number size distribution data that, while serving their purpose, could benefit from optimization, and subsequent comparative analyses indicated that more advanced denoising techniques could potentially preserve genuine atmospheric signals with greater fidelity and reduce the risk of introducing unintended artifacts into the data representation. These aspects collectively indicated areas where the analytical framework could be strengthened to enhance reliability and facilitate future development.

Concurrently, a separate computational implementation in Matlab was also instrumental in formation rate calculations. This script, generally following established conceptual approaches, presented a specific area for refinement concerning the calculation of the coagulation sink. Its implementation of the coagulation sink utilized an expression that incorporated a single particle concentration term (i.e., $\sum K(d_p, d'_p)N_{d'_p}$), despite being intended to represent a loss rate in [$\mathrm{cm^{-3}s^{-1}}$]. As rigorously established in Section 2.1.3.3 a precise dimensional analysis requires that for the coagulation sink to accurately reflect a loss rate in [$\mathrm{cm^{-3}s^{-1}}$], it must account for the concentration of both the target particle population and the concentration of the particles with which it is coagulating ($N_{d_p}\sum k_{\mathrm{coag}}(d_p, d'_p)N_{d'_p}$). This subtle but critical distinction in dimensional consistency presented an opportunity to refine the calculation for a more physically accurate representation of particle loss due to coagulation. Considering that coagulation constitutes a significant loss pathway for newly formed particles, addressing this point was crucial for improving the precision of derived new particle formation rates and enhancing the comparability of experimental results.

These observations, encompassing aspects from computational structure and documentation to the precision of specific physical parameterizations, highlighted the compelling need for an evolved and rigorously validated computational framework, and the present work directly addresses these opportunities for refinement by providing a transparent, robust, and dimensionally consistent re-implementation of the formation rate calculation, which will be detailed in the subsequent sections of this chapter.

2.2.2 Development of the Remade Formation Rate Script

This section details the design and implementation of the new Python-based computational framework developed for determining new particle formation (NPF) rates from CLOUD chamber experiments. Building upon the theoretical foundations of aerosol dynamics and directly addressing the methodological limitations identified in previous computational approaches (as discussed in Section 2.2.1), this re-implementation aims to provide a transparent, robust, and dimensionally consistent tool for scientific analysis and broader applicability.

2.2.2.1 General Architecture and Workflow

The newly developed computational framework for determining new particle formation rates is implemented as a comprehensive and modular Python script, which prioritizes clarity, maintainability, and extensibility, facilitating future developments and collaborative contributions. The overall workflow of the script, as conceptualized in the design diagram in Figure 2.1 follows a logical sequence of operations, transforming raw experimental measurements into the final particle formation rates.

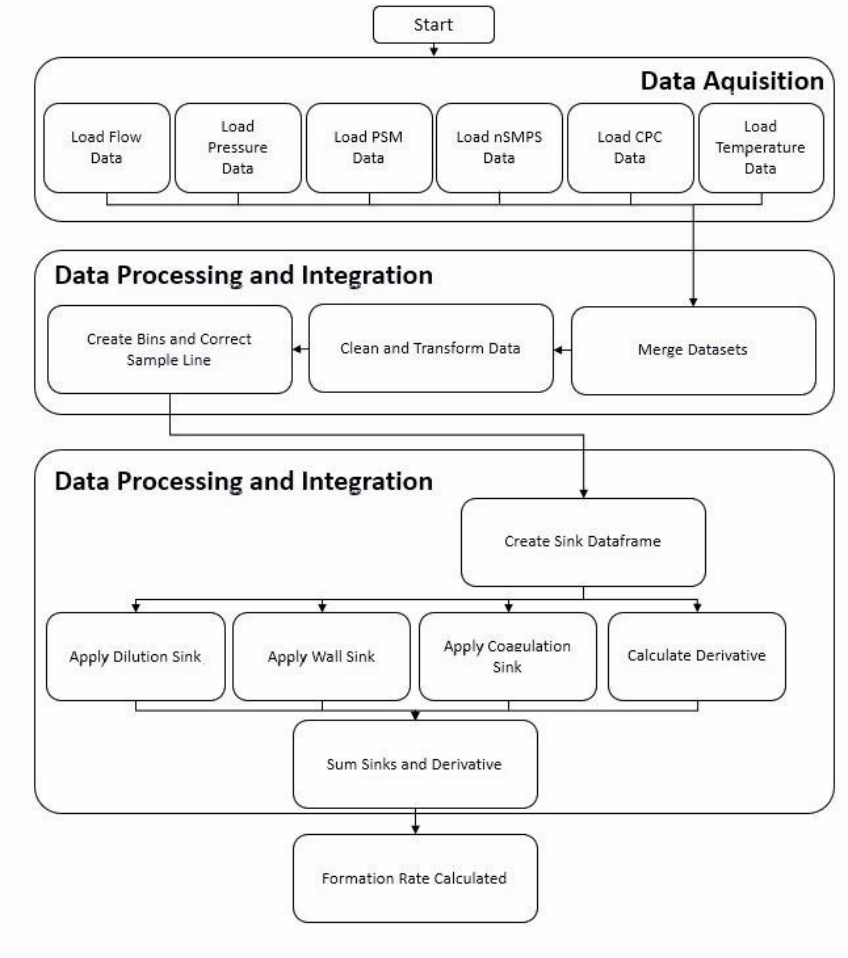


Figure 2.1: **High-Level Workflow of the Python Formation Rate Script.** This flowchart illustrates the systematic stages involved in the calculation of new particle formation rates (J) from CLOUD chamber experimental data. The process begins with Data Acquisition (loading various sensor and instrument data), proceeds through multiple phases of Data Processing and Integration (merging datasets, cleaning and transforming, creating particle size bins, and applying sampling line loss corrections), and culminates in the calculation and summation of individual loss/gain terms (dilution, wall, and coagulation sinks, as well as the particle concentration derivative) to determine the final formation rate.

The process begins with a series of dedicated data loading functions (*load_...*), each responsible for retrieving specific types of raw data (e.g., pressure, temperatures, chamber flows, particle concentrations from CPC, PSM, and nSMPS) directly from the experimental database. These functions perform initial pre-processing steps, such as time synchronization and handling of missing values and ensuring data consistency.

Once individual data streams are loaded, a central data aggregation function (*gatherData*) consolidates all relevant information into a single, unified DataFrame. This critical step also involves essential transformations and corrections, most notably the detailed sampling line loss correction, which accounts for particle losses before measurement. Additionally, *gatherData* is responsible for defining and calculating specific particle size bins (e.g., '2_50', '1_70') that are central to the formation rate calculation.

Following data preparation, the script proceeds to calculate the various terms of the formation rate equation through a set of specialized functions, which are the dilution sink (*DilutionSink*), wall sink (*WallSink*), and the coagulation sink (*CoagulationSink*), where each of these functions takes the aggregated data and the target particle diameter as inputs, returning the calculated loss rate for that specific process, and a dedicated derivative function (*derivative*) computes the time rate of change of particle concentration for the chosen size bin.

Finally, the results from these individual calculation functions are comprehensively summed to yield the total new particle formation rate for the specified particle diameters, so the script organizes the results of its calculations into a structured format, which will at a later stage be adapted to provide the final formation rate in real time at the CLOUD chamber experiment. This modular architecture ensures that each step of the calculation is transparent and individually verifiable, contributing to the overall reliability of the derived formation rates.

The code for the *GatherData*, *DilutionSink*, *WallSink*,*CoagulationSink* functions and the the calculation of J can be found in Figures A.1 and A.2 of the Appendix.

2.2.2.2 Key Design Principles and Advancements

The development of this remade formation rate script was meticulously guided by several key design principles, all aimed at comprehensively addressing the limitations and opportunities for improvement identified in the previous R and Matlab implementations. The primary objectives underpinning this re-implementation centered on establishing a more robust, transparent, and scientifically accurate computational framework.

Foremost among these principles was the commitment to enhanced transparency and modularity. The script is meticulously structured with a clear separation of concerns, utilizing distinct functions for each logical step of the data processing and calculation pipeline, for instance, specific functions are dedicated solely to loading particular data types (e.g., *load_pressure*, *load_temperatures*), while others manage data aggregation (*gatherData*) or individual physical calculations (e.g., *CoagulationSink*). This modular architecture significantly elevates code readability, making it considerably easier for researchers to comprehend the exact operations performed at each stage, as such clarity is vital for fostering greater confidence in the scientific analysis and facilitating collaborative development.

A critical advancement in this re-implementation is the rigorous adherence to improved accuracy and dimensional consistency which is most prominently demonstrated in the coagulation sink calculation, unlike previous approaches where inconsistencies in dimensional representation may have arisen, this script meticulously implements the coagulation sink equation, ensuring all terms and intermediate calculations strictly conform to their physical units. This rectification of a significant potential source of

error directly contributes to the derivation of more accurate and physically sound formation rates.

Furthermore, the script incorporates robust data processing and optimized denoising techniques: the deliberate application of the Savitzky-Golay filter provides a sophisticated method for smoothing inherently noisy experimental data while rigorously preserving underlying physical trends. This leads to more reliable numerical derivatives of particle concentrations, which are fundamental to the formation rate equation.

Finally, the transition to Python, a widely adopted and actively supported programming language in the scientific computing community, significantly enhances the script's maintainability and extensibility. The clean, function-based architecture of the script inherently facilitates easier debugging, streamlines future updates, and allows for the seamless integration of new features, analysis methodologies, or expanded experimental capabilities, which ensures that the re-engineered framework not only addresses past limitations but also stands as a reliable and adaptable tool for future aerosol dynamics research.

2.2.2.3 Data Acquisition and Initial Pre-processing

The initial stage of the formation rate calculation script is dedicated to the robust acquisition and preliminary processing of various raw experimental data streams originating from the CLOUD chamber. This foundational step is paramount for establishing a clean, consistent, and time-synchronized dataset, which is indispensable for ensuring the accuracy and reliability of all subsequent aerosol dynamics calculations.

A crucial pre-processing step, particularly for ensuring the integrity of time series data, involves time synchronization. Since some instrument's raw data often arrives with 'random' or irregular timestamps, timestamps are converted from milliseconds to seconds (*times = times // 1000*) and then precisely snapped to whole-second intervals by converting back to milliseconds (*times = times * 1000*). Subsequently, these dataframes are merged with a standardized array of times that are multiples of one second, ensuring a consistent 1-second time step, and missing data points introduced by this synchronization are systematically addressed, typically through forward-filling (*fillna(method='ffill')*), to ensure a complete and continuous record for downstream processing.

The script begins by loading essential environmental data critical for physical calculations within the chamber, where dedicated functions retrieve atmospheric pressure values, which are vital for deriving parameters such as the mean free path of air molecules, temperature data, that is collected from multiple sensors which introduces the need of calculating a time-resolved average across all specified sensors to provide a representative chamber temperature and chamber flow rates, with individual flow contributions summed and necessary unit conversions performed (e.g., from L min^{-1} to cm^3 s^{-1}) to obtain the total synthetic air flow into the chamber, a crucial parameter for calculating the dilution sink.

Data from particle counting instruments are also loaded and pre-processed to provide accurate time-series measurements of particle concentrations. A dedicated function retrieves total particle concentrations from the Condensation Particle Counter (CPC), representing particles larger than 2.5 nm ('N>2_50'), and a significant improvement in this re-implementation includes the precise loading and processing of data from the Particle Size Multiplier (PSM), which provides particle concentration data for particles larger than 1.7 nm ('N>1_70'). This inclusion of smaller size range data is crucial for accurately capturing the initial stages of new particle formation. For both CPC and PSM data, an initial calculation of the time derivative of particle concentration is performed directly within their respective loading functions, which is achieved using a Savitzky-Golay filter, a crucial preliminary step for subsequent formation rate calculations [58], [59].

Finally, the script is responsible for acquiring the high-resolution particle number size distribution data from the Nano-Scanning Mobility Particle Sizer (nSMPS), which provides particle counts across numerous defined size bins. This comprehensive data acquisition stage ensures that all necessary parameters are available and prepared in a consistent format for the subsequent processing and calculation steps.

2.2.2.4 Data Aggregation and Primary Conversions

Following the initial data acquisition and pre-processing, the gatherData function serves as a central hub within the computational framework, responsible for integrating all the prepared time series data into a single, comprehensive DataFrame. This function is critical for harmonizing diverse datasets and performing essential primary transformations and corrections that are prerequisites for the accurate calculation of particle formation rates and its primary roles include the precise merging of all acquired data streams, the creation of specific particle number concentration bins, and the crucial correction for sampling line losses.

A fundamental task of gatherData is to efficiently merge all individual pre-processed DataFrames (containing pressure, temperatures, flows, CPC, PSM, and nSMPS data) into a unified structure. This merging process is meticulous, prioritizing the maintenance of a consistent and reliable time base across all datasets. Crucially, the nSMPS timestamps are designated as the 'original' or reference time points. This strategy is employed not because the nSMPS possesses the highest temporal resolution (indeed, it typically does not measure every single second), but rather to avoid the complexities and potential inaccuracies of extrapolating its high-detail size distribution data, or incurring significant data loss due to its less frequent measurements so by aligning the higher-resolution data from other instruments (like CPC, PSM, and environmental sensors) to the nSMPS's timestamps, the integrity and completeness of the essential nSMPS size distribution dataset, critical for coagulation sink and bin creation, are robustly preserved, so the resulting DataFrame provides a synchronized view of all experimental parameters and particle measurements over the entire study period.

Within *gatherData*, precisely defined particle number concentration bins are derived from the available instrument data. These bins are essential for calculating formation rates specific to certain particle growth stages, and the script focuses on two primary bins: '2_50' (representing the total particle concentration with 2.50 nm diammeter) and '1_70' (for particles of 1.70 nm diameter).

A significant challenge in accurately determining these concentrations arises from the inherent differences and detection efficiencies among the instruments. The nSMPS provides a detailed particle size distribution, which is typically output in a log-scale distribution, so, for use in concentration calculations, this nSMPS log distribution is first converted into linear particle counts per size bin, allowing for direct summation of particle numbers [60]. While invaluable for understanding the shape of the particle spectrum and for calculating size-dependent processes like coagulation, the nSMPS tends to detect particles poorly at the lowest diameters (typically below $\sim$5 nm). In contrast, the Condensation Particle Counter (CPC) and the Particle Size Multiplier (PSM) are designed for highly efficient total number concentration measurements above their respective detection thresholds (2.5 nm for CPC and 1.7 nm for PSM), which results in the total particle counts for overlapping diameter ranges often do not agree perfectly between the nSMPS, CPC, and PSM.

To address this discrepancy and ensure the most reliable total particle number concentrations for these critical bins, the CPC and PSM are utilized as the primary reference instruments for their respective lower diameter cut-offs. Specifically, the total number concentration for the '2_50' bin is directly anchored to

the CPC measurement (>2.5 nm). Similarly, the total number concentration for the '1_70' bin is directly obtained from the PSM data (>1.7 nm). While the nSMPS data remains vital for understanding the distribution of particles within these and other size ranges (e.g., for coagulation sink calculations which require detailed size information), the total number concentrations used for calculating the derivative are thus corrected by leveraging the superior detection efficiency and reliability of the CPC and PSM for their specific lower limits. This approach ensures that the overall particle numbers used in the formation rate equation are as accurate as possible for the defined size ranges, providing a robust basis for analyzing new particle formation and growth across different size regimes.

The following step is a critical transformation: the correction for particle losses occurring within the sampling lines. Particles can deposit onto the walls of the tubing connecting the chamber to the measurement instruments, leading to an underestimation of their actual concentrations, so this correction is vital for obtaining accurate particle number concentrations inside the chamber, which are then used in subsequent formation rate calculations. The process involves several physics-based computations, many of which utilize parameters previously defined. The properties of air, such as viscosity and mean free path, and fundamental particle characteristics, including mean thermal speed, the Knudsen number, and the Cunningham slip correction factor, are calculated using the same established formulas as described in Section 2.1.3.2.

These intermediate parameters are used to calculate key dimensionless numbers that characterize the transport of particles within the sampling lines. The Gormley-Kennedy [61] parameter , ξ, quantifies the extent of particle diffusion within the tube and is calculated as:

$$\xi = \frac{\pi D L}{\mathrm{Q}_{\mathrm{flow}}} \tag{2.12}$$

where D is diffusion coefficient, L is the tube length, and $\mathrm{Q}_{\mathrm{flow}}$ is the volumetric flow rate through the tube. Following this, the Sherwood number (Sh), which describes the ratio of convective to diffusive transport [62], is determined using the following empirical relationship:

$$\mathrm{Sh} = 3.66 + \frac{0.2672}{\xi + 0.10079\xi^{1/3}} \tag{2.13}$$

All these parameters converge in the calculation of the size-dependent efficiency factor (η_{eff}) for each sampling line [63]. This factor specifically quantifies particle losses due to diffusion to the tube walls under laminar flow conditions. The efficiency is modeled as an exponential decay dependent on the product of ξ and Sh:

$$\eta_{\mathrm{eff}} = e^{-\xi \cdot \mathrm{Sh}} \tag{2.14}$$

Finally, to obtain the true particle concentrations within the CLOUD chamber, the measured concentrations from the instruments are divided by this calculated η_{eff} factor. This rigorous, physics-based correction ensures that the particle concentration values used for subsequent calculations of formation rates accurately reflect the conditions inside the chamber, providing a robust basis for analysis.

2.2.2.5 Calculation of Individual Particle Loss/Gain Terms

This section details the core functions that calculate the individual loss and gain terms contributing to the total particle formation rate. These functions operate on the comprehensive, corrected, and synchronized data prepared in the preceding steps, and each calculation is based on established physical

principles and is implemented to ensure accuracy and dimensional consistency, as a foundational step toward the final determination of the new particle formation rate.

The *DilutionSink* function calculates the term S_{dil}, which quantifies the loss of particles due to the continuous flow of clean air into and out of the CLOUD chamber which is a fundamental sink term in the analysis of chamber experiments, as it reduces the concentration of particles over time. A key advancement in this re-mplementation is the use of real-time flow data from the chamber's sensors, replacing the fixed-value dilution constant used in previous versions. This improvement ensures that the calculation accurately reflects the dynamic flow conditions throughout the experiment. The dilution sink is defined by the rate at which the chamber air is exchanged, based on the chamber volume (V_{chamber}) and the total volumetric flow rate ($\text{Flow}_{\text{synthetic air}}$) of the clean air, and its theoretical basis is detailed in Equations 2.2 and 2.3. The function takes the real-time flow data, chamber volume, and the current particle concentration as inputs and returns the calculated dilution sink term, which is subsequently added to the total sink.

The *WallSink* function is responsible for calculating the loss of particles due to their diffusion to the chamber's internal surfaces. This wall loss term, S_{wall}, is a significant factor, particularly for small particles that have high diffusivities. The function calculates this sink term by first determining a size-dependent wall loss rate constant, k_{wall}, for each individual diameter bin within the particle size distribution, which is derived from a detailed physical model that considers the chamber geometry and particle diffusion rates, utilizing the particle diffusion coefficient, D. Specifically, the wall loss rate constant is calculated as being proportional to the square root of the diffusion coefficient, using a constant of 0.0077 $\text{s}^{-0.5}\ \text{cm}^{-1}$, as established in [64] for the CLOUD chamber. The fundamental equations for air properties (viscosity, mean free path) and particle characteristics (mean thermal speed, Knudsen number, Cunningham correction), are the ones shown previously in Equation 2.6. The *WallSink* function then applies this unique rate constant to the corresponding particle concentration in each bin, and finally sums these individual loss rates across all relevant bins to obtain the total wall loss term for a given size range. This rigorous, size-resolved calculation ensures that the final wall sink term accurately reflects the complex diffusion dynamics within the chamber and is subsequently added to the total sink.

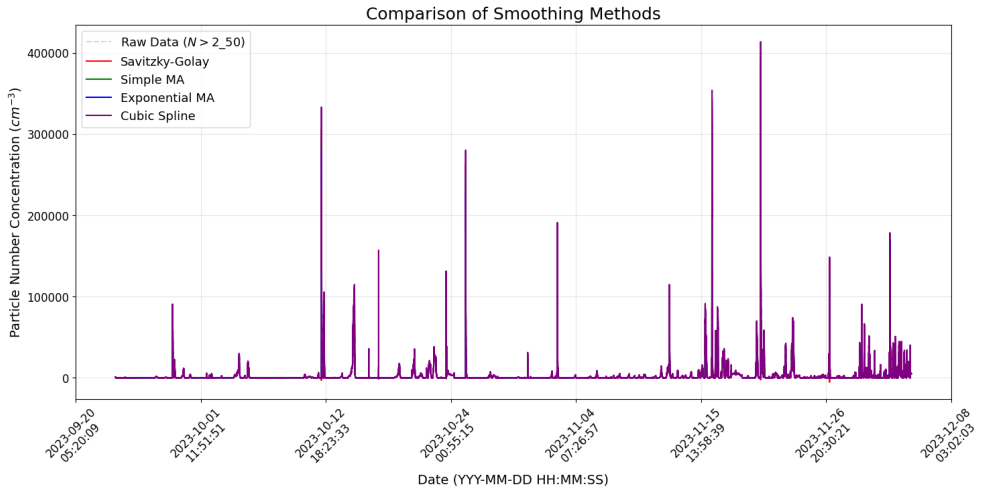
The *CoagulationSink* function is a pivotal component of the script, demonstrating a key methodological improvement from previous implementations by ensuring dimensional consistency. This function correctly calculates the coagulation sink term, which describes the loss of particles of a certain size, d_p, due to collisions and aggregation with all other particles in the chamber. This approach is a direct improvement upon previous implementations which relied on a dimensionally inconsistent formula for the coagulation sink, and the revised script meticulously corrects this by ensuring all terms conform to their physical units, resulting in a more robust calculation.

One significant enhancement in this function is the refined calculation of particle mass, where unlike previous works, the new script calculates a unique particle mass for each individual diameter bin within the distribution, which is a crucial step for accurately deriving size-dependent parameters, with the particle mass being determined from its diameter by calculating the particle's volume and multiplying it by the density. The calculation of the Brownian coagulation coefficient, $k_{\text{coag}}(d_p, d'_p)$ (Equation 2.7), within the function is meticulously detailed, including the sub-calculations for the mean thermal speed, $c(d_p)$ (Equation 2.8), the characteristic length parameter, $l(d_p)$ (Equation 2.9), and the dimensionless factor $g(d_p)$ (Equation 2.10). The summation over all relevant particle sizes is then rigorously performed using nested loops within the script (Equation 2.11). This robust and dimensionally consistent calculation provides a more accurate and scientifically sound basis for determining the coagulation sink.

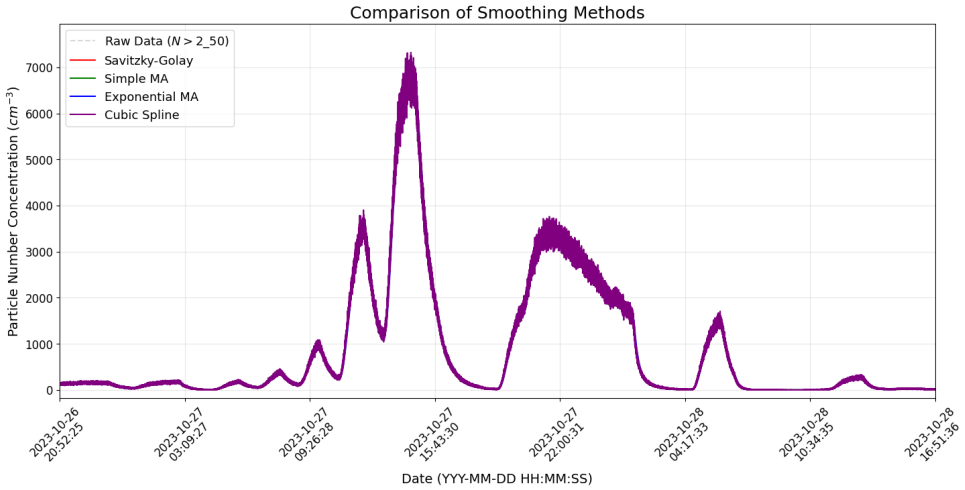
Lastly, the derivative function plays a crucial role in the overall calculation by accurately determining

the time rate of change of particle concentration for specific size thresholds (e.g., 'N>2_50', 'N>1_70'). Experimental data, particularly time-series measurements of particle number concentrations, often contain inherent noise, so accurately extracting the derivative from such noisy data is critical, as direct numerical differentiation can significantly amplify this noise, leading to erratic and unphysical results.

To identify the optimal denoising and differentiation methodology, a comparative analysis was performed on several filtering techniques, with their outputs shown in Figure 2.2.



(a) Full time series of the raw data alongside a cubic spline fit, an exponential moving average, a simple moving average (both with a window of 5 minutes) and the Savitzky-Golay filter.



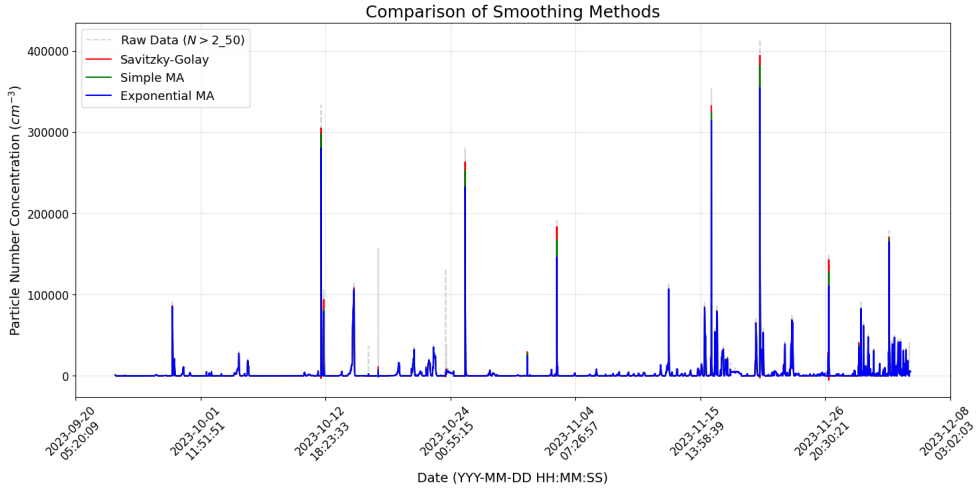
(b) A zoomed-in section provides a clearer view of the performance of the cubic spline fit.

Figure 2.2: Comparison of the Cubic Spline smoothing method against other filters and the raw data. The figure illustrates that the cubic spline method is unsuitable for this application, as it introduces unphysical oscillations and artifacts. (a) The plot shows the full time series of the raw particle number concentration data with the different smoothing methods overlaid, where the cubic spline method's erratic behavior is visible across the entire period. (b) A zoomed-in section of the plot highlights the oscillations of the cubic spline in greater detail, which misrepresent the underlying physical signal.

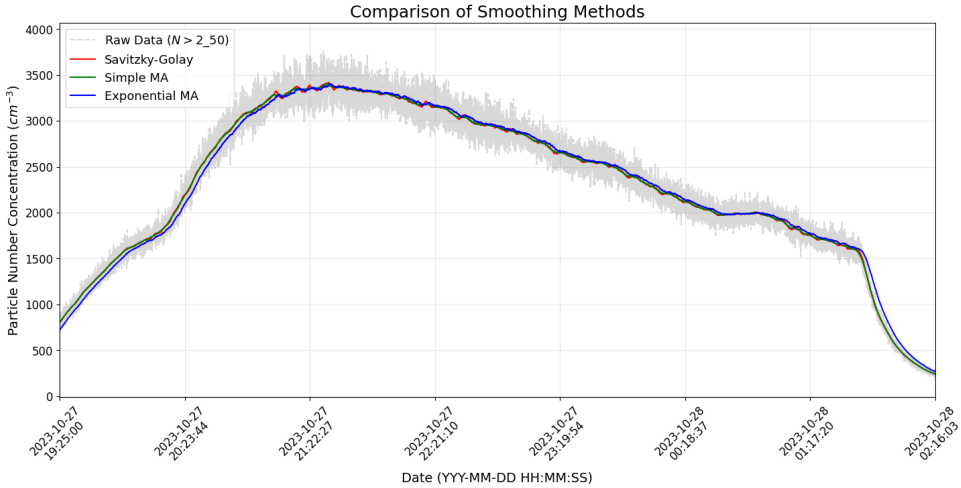
Initial analysis of these plots immediately demonstrated the unsuitability of the cubic spline method. As seen in Figure 2.2, the cubic spline fails to produce a smooth output and instead introduces oscillations and artifacts that do not represent the true signal. This behavior, which is particularly pronounced with noisy data, renders the method unreliable for accurate derivative calculation.

A closer inspection of the remaining methods in Figure 2.3 reveals a key difference: while both the

simple and exponential moving averages provide effective smoothing, they do so with a small but distinct temporal delay, as the exponential moving average in particular, exhibits a subtle lag behind the raw data, which can introduce inaccuracies when calculating time-sensitive derivatives.



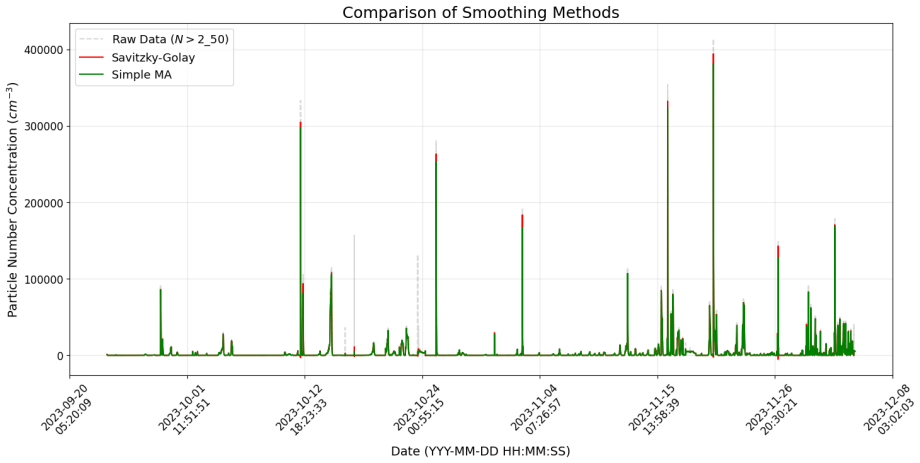
(a) Full time series of the raw data alongside an exponential moving average, a simple moving average (both with a window of 5 minutes) and the Savitzky-Golay filter.



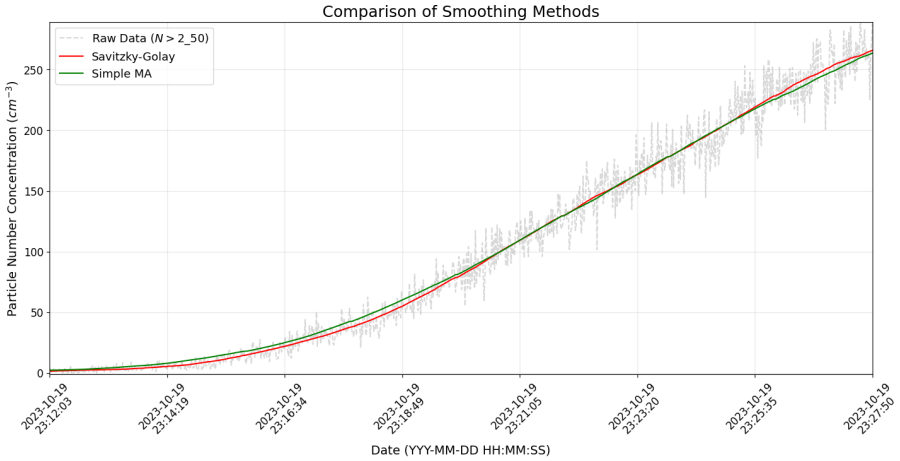
(b) A zoomed-in section provides a clearer view of the performance of the exponential moving average (with a window of 5 minutes).

Figure 2.3: A comparison of the smoothing methods used for the time-series particle concentration data (N>2_50) after excluding the unsuitable cubic spline. (a) The full time series plot shows the raw data alongside the Savitzky-Golay filter, a simple moving average, and an exponential moving average. (b) A zoomed-in section of the plot provides a clearer view of their performance, highlighting that the exponential moving average lags behind the raw data, a subtle but distinct temporal delay that can affect derivative calculations.

A closer inspection of the remaining methods reveals the superior performance of the Savitzky-Golay filter. As illustrated in Figure 2.4, while a simple moving average does smooth the data, it fails to follow the nuances of the raw data as accurately as the Savitzky-Golay filter.



(a) Full time series of the raw data alongside a simple moving average (with a window of 5 minutes) and the Savitzky-Golay filter.



(b) A zoomed-in section provides a clearer view of the performance of the simple moving average (with a window of 5 minutes) and the Savitzky-Golay filter.

Figure 2.4: A direct comparison of the smoothing performance of the Savitzky-Golay filter and a simple moving average. (a) The plot shows the full time series of the raw data alongside both filtering methods. (b) A zoomed-in section provides a clearer view of their performance, highlighting that while a simple moving average does smooth the data, it fails to track the signal as accurately as the Savitzky-Golay filter. This is particularly visible at the beginning of the plot, where the simple moving average rises before the raw data, while the Savitzky-Golay filter more closely follows the signal's true trend.

A crucial advantage of the Savitzky-Golay method is its integrated approach to differentiation, as it not only provides a high-quality smoothed signal but also allows for a robust and direct calculation of the derivative as part of its core methodology, making it the most efficient and reliable choice for this application. The output of this function directly contributes to the sum of terms that define the total particle formation rate.

2.2.2.6 Final Formation Rate Calculation and Output

The final step in the computational workflow is the rigorous summation of all individual loss and gain terms to yield the total new particle formation rate, J. This calculation synthesizes all the previously determined components — the time derivative of particle concentration, the dilution sink, the wall sink, and the coagulation sink, according to the fundamental formation rate equation (Equation 2.1). A significant advancement of this new script is its ability to calculate the formation rate not only for particles

greater than 2.5 nm, which was a limitation of previous work, but also for particles with diameters greater than 1.7 nm. Furthermore, the script is designed with the flexibility to compute the formation rate for any other specific size bin available from the nSMPS instrument.

The output of the script is structured for clarity and further analysis. The final formation rate, J, along with other relevant metadata, will be made available for real-time viewing and analysis at the CLOUD experiment, which ensures that the results are not only accurate and robust but also immediately accessible for scientific interpretation and visualization during experiments.

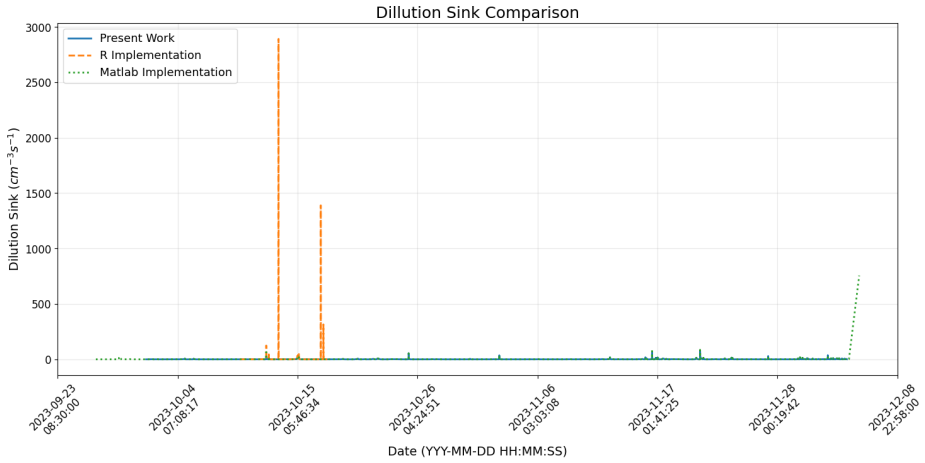
2.2.3 Results and Comparative Analysis

This section summarizes the key advancements achieved with the new Python script compared to previous approaches, emphasizing its enhanced transparency, modularity, computational robustness, and, most importantly, its scientific accuracy and dimensional consistency.

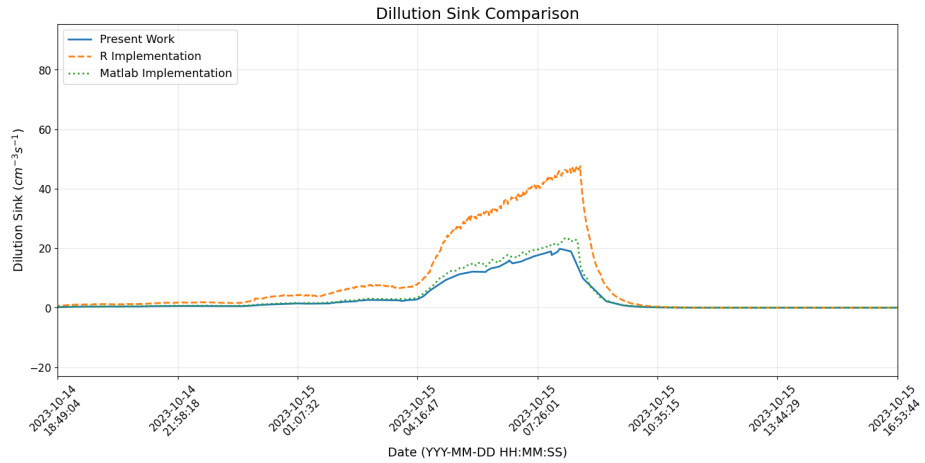
2.2.3.1 Presentation and Comparison of Results

The following figures illustrate the time series for each of the major sink terms—dilution, wall loss, and coagulation—along with the final calculated formation rate for particles greater than 2.5 nm. This visual presentation of the results highlights the contributions of each individual term to the overall rate, demonstrating the transparency and methodological rigor of the new script.

Starting with the Dilution Sink, as seen in Figure 2.5, a comparison of the dilution sink calculation from the new Python script ("Present Work") against previous R and Matlab implementations reveals several key insights. The R implementation exhibits several large, abnormal spikes that are not present in either the Matlab or the "Present Work" implementations. This suggests a potential flaw in the R script's handling of the flow data or its calculation of the dilution constant, a problem that is not observed in the other two scripts.



(a) Comparison of the full time series of the Dilution Sink calculation.

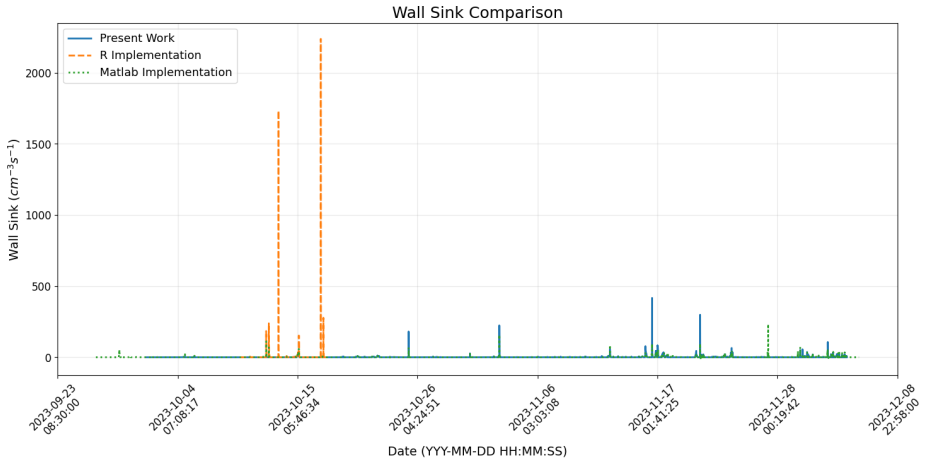


(b) Zoomed-in view showing the close agreement between the Present Work and Matlab implementations.

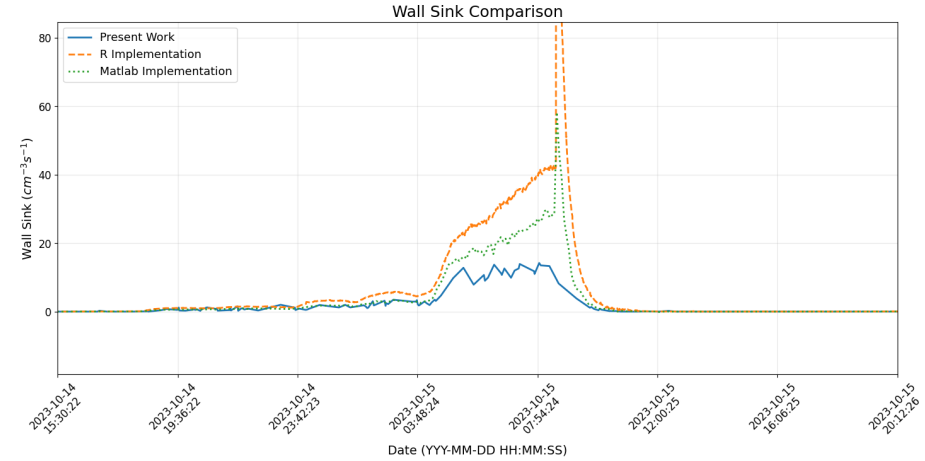
Figure 2.5: A comparison of the dilution sink calculation from the new Python script ("Present Work") against previous R and Matlab implementations. (a) The full time series plot illustrates the large, abnormal spikes present in the R implementation, which are absent in the other two scripts. (b) A zoomed-in view shows the close agreement between the Present Work and Matlab implementations, validating the new script's accuracy. The minor differences observed are attributed to the new script's use of real-time flow data, which provides a more physically representative result.

When examining a zoomed-in section of the data, it becomes clear that the new Python script and the Matlab implementation are in close agreement. The two scripts produce very similar results, validating the new Python script's accuracy. The minor differences observed are likely due to the new script's use of real-time flow data from the chamber's sensors, a key advancement over the previous versions that may have relied on a fixed-value dilution constant. This demonstrates that the new script provides results that are not only robust but also more physically representative of the dynamic chamber conditions.

A comparison of the wall sink calculation from the new Python script ("Present Work") with previous R and Matlab implementations reveals a similar pattern of consistency and discrepancy. As shown in Figure 2.6a, the R implementation once again produces abnormal, isolated spikes that are absent in the other two scripts. These spikes appear to be artifacts of the calculation, indicating a potential issue with the R script's methodology. The "Present Work" and Matlab implementations, however, show a high degree of agreement.



(a) Comparison of the full time series of the Wall Sink calculation.

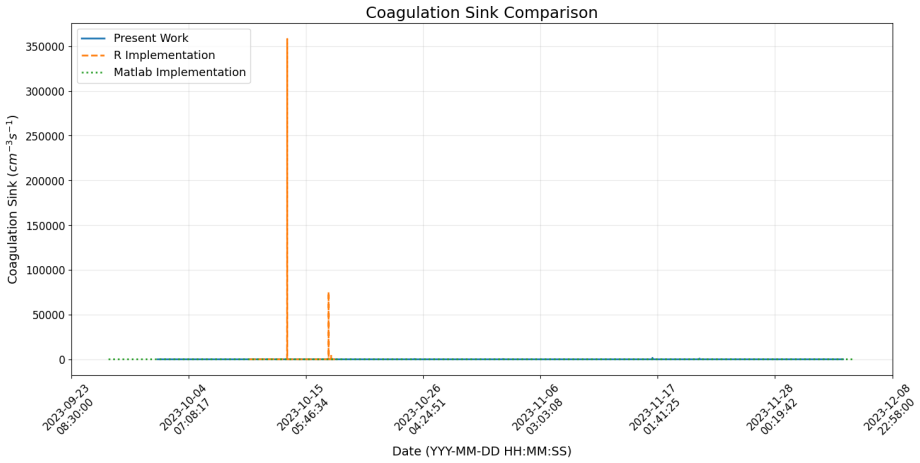


(b) Zoomed-in view showing the close agreement between the Present Work and Matlab implementations.

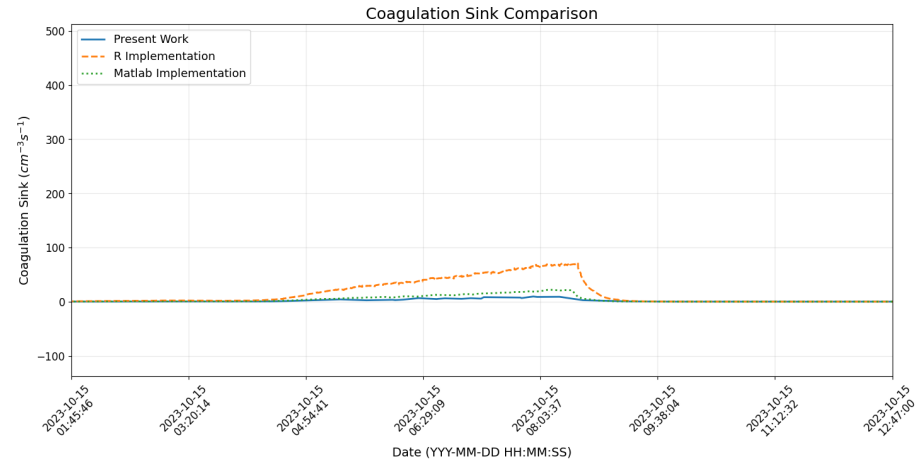
Figure 2.6: A comparison of the wall sink calculation from the new Python script ("Present Work") against previous R and Matlab implementations. (a) The full time series plot illustrates the large, abnormal spikes present in the R implementation, which are absent in the other two scripts. (b) A zoomed-in view shows the close agreement between the Present Work and Matlab implementations, validating the new script's accuracy.

A zoomed-in view in Figure 2.6b confirms that the two scripts follow the same trends and produce nearly identical results. This close correlation validates the robustness and accuracy of the new wall sink calculation, demonstrating its consistency with a known-good implementation while correcting the flaws of the R script.

A comparison of the coagulation sink calculation from the new Python script and previous R and Matlab implementations reveals significant discrepancies that highlight a key methodological advancement in this work.



(a) Comparison of the full time series of the Coagulation Sink calculation.



(b) Zoomed-in view showing the divergence between the new and previous implementations.

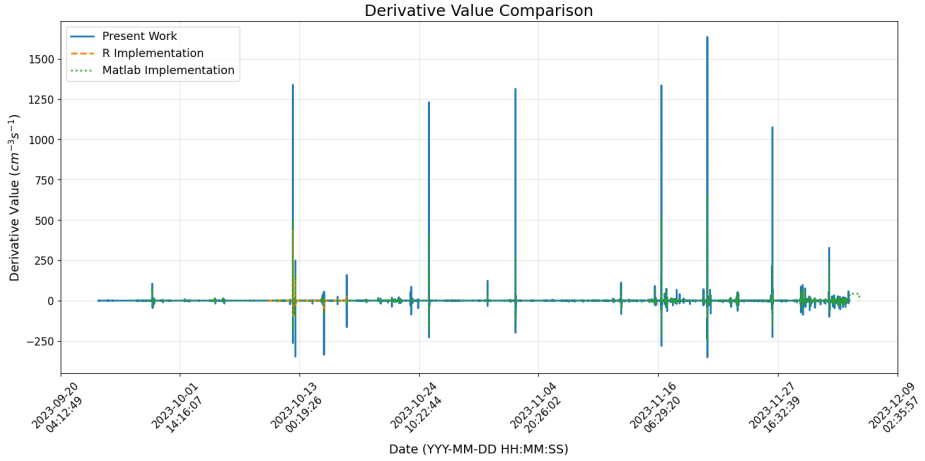
Figure 2.7: A comparison of the coagulation sink calculation from the new Python script ("Present Work") against previous R and Matlab implementations. (a) The full time series plot illustrates the abnormal, extremely large spikes present in the R implementation, which are absent in the other two scripts. (b) A zoomed-in view highlights the significant divergence between the present script and the previous implementations, which is a direct consequence of correcting a dimensionally inconsistent formula in the new work.

As seen in the full time series plot (Figure 2.7a), the R implementation produces abnormal and extremely large spikes that are orders of magnitude greater than the values from the other two scripts. The Matlab implementation also shows some less pronounced spikes and a non-zero baseline. In stark contrast, the new Python script ("Present Work") shows a much lower and more stable coagulation sink value. This difference is not a flaw but a direct consequence of the script's correction of a dimensionally inconsistent formula used in previous implementations.

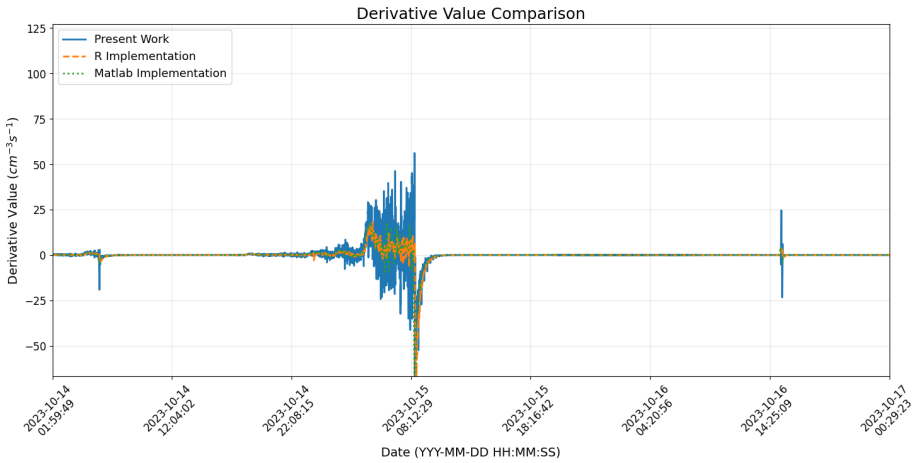
A zoomed-in view of the data in Figure 2.7b further illustrates this divergence. The previous implementations yield significantly higher coagulation sink values, whereas the new script's values are consistently lower and more representative of the physical process. This comparison serves as a powerful visual justification for the new script's improved accuracy, demonstrating that the corrected, dimensionally consistent formula leads to a more robust and physically sound calculation of the coagulation sink.

The final component needed for the formation rate calculation is the derivative of particle concentration. A comparison of the derivative values from the three implementations, as shown in Figure 2.8a, reveals significant differences in their stability and robustness. While the R and Matlab implementations

appear to produce a less noisy derivative, the "Present Work" shows a higher degree of fluctuation and numerous large spikes. This is a direct consequence of the Savitzky-Golay filter's sensitivity and its specific parameters, which, in this case, do not smooth the data as aggressively as the other methods.



(a) Comparison of the full time series of the derivative of particle concentration.



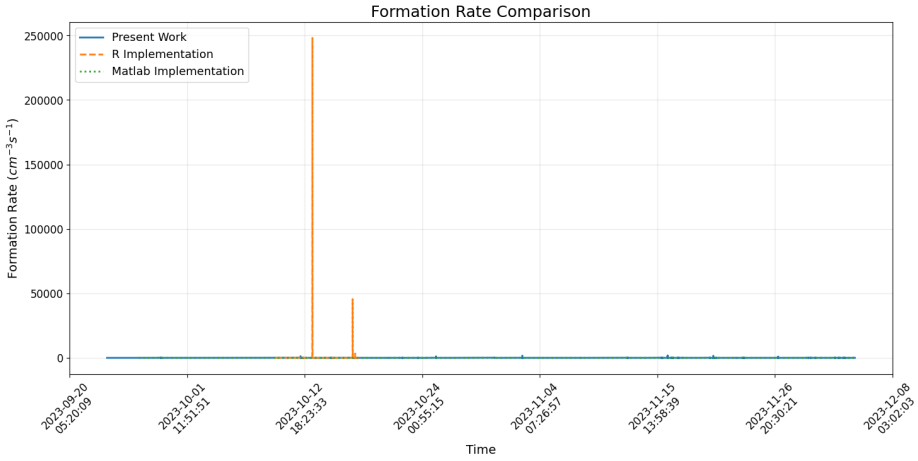
(b) Zoomed-in view showing the difference in stability and smoothness between the implementations.

Figure 2.8: A comparison of the derivative of particle concentration from the new Python script ("Present Work") against previous R and Matlab implementations. (a) The full time series plot highlights the stark difference in methodology, with the R and Matlab implementations appearing to produce a less noisy derivative, while the "Present Work" shows a high degree of fluctuation and numerous large spikes. (b) A zoomed-in view of the data illustrates that the new script is a more responsive and faithful representation of the raw data's changes, a direct benefit of the Savitzky-Golay filter's ability to extract the derivative without excessive smoothing or temporal lag.

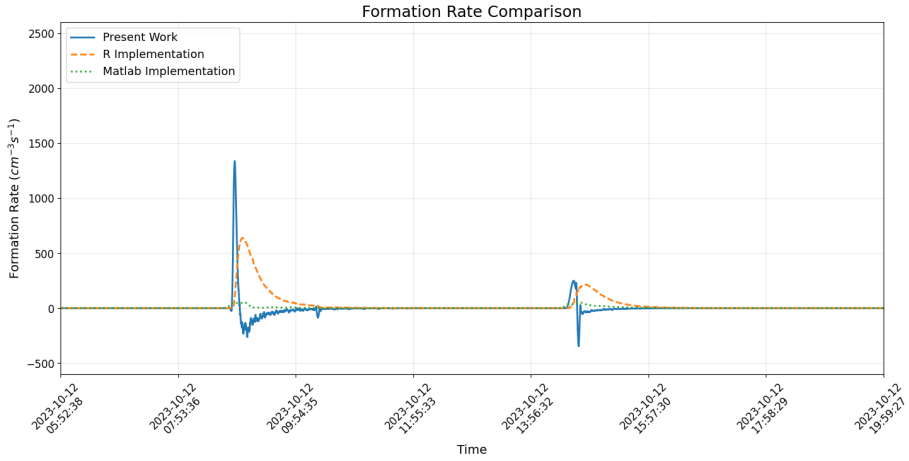
A closer inspection in Figure 2.8b highlights this trade-off. While the "Present Work" curve is the most noisy, its methodology avoids the temporal lag seen in other techniques and provides a robust, integrated method for calculating the derivative. The noise present in the "Present Work" is a known side-effect of its sensitivity and is considered an acceptable compromise for preserving the most accurate representation of the signal's original dynamics.

The final new particle formation rate for particles with a diameter greater than 2.5 nm is presented, synthesizing all the individual components—the dilution, wall, and coagulation sinks, as well as the derivative of particle concentration. This final rate serves as the ultimate output of the script and provides a comprehensive view of the results from the different implementations.

As shown in the full time series plot, the R implementation produces massive, isolated spikes that are clearly artifacts of its calculation methodology which are orders of magnitude larger than the rates calculated by the other two scripts. The "Present Work" and Matlab implementations, however, show a much closer agreement.



(a) Comparison of the full time series of the Final Formation Rate.



(b) Zoomed-in view showing the difference in the final formation rate between the new and previous implementations.

Figure 2.9: A comparison of the final new particle formation rate from the new Python script ("Present Work") against previous R and Matlab implementations. (a) The full time series plot highlights the stark contrast in methodology, with the R implementation producing massive, artifact-like spikes that are orders of magnitude larger than the other calculations. (b) A zoomed-in view of a period with active formation shows the significant differences in the final rate, with the "Present Work" revealing a more complex signal with both positive and negative values, a direct result of the script's dimensionally consistent calculations and responsive derivative method.

A closer look at a period with more activity reveals the subtle but significant differences that arise from the distinct methodologies. While the Matlab implementation appears smooth and produces only positive values, it follows an incorrect expression from the article, which results in a signal that is visually pleasing but scientifically inaccurate. In contrast, the "Present Work" implementation shows a more complex signal with both positive and negative values, which is a direct reflection of the new script's highly sensitive and dimensionally consistent calculation of each component. This outcome, while less visually smooth and showing periods of negative formation, is the most scientifically sound as it is derived from the robust and independently verified calculations of the derivative and sink terms discussed

previously. The differences observed in this final rate are the cumulative result of the methodological advancements and corrections introduced in the new script.

2.2.3.2 Discussion of Advancements, Robustness, and Future Prospects

These advancements collectively bring substantial benefits to CLOUD data analysis. The enhanced scientific accuracy and dimensional consistency, particularly in the coagulation sink, lead to greater confidence in the derived formation rates, and the modular and transparent design of the Python script also significantly improves the reproducibility of results, making it easier for other researchers to understand, verify, and build upon the methodology. Furthermore, the robust filtering and dynamic parameter handling contribute to the overall computational robustness, reducing the likelihood of errors and simplifying maintenance and debugging efforts.

Looking ahead, this re-implemented framework provides a solid foundation for future developments. The script's modularity and Python-based environment make it highly adaptable for real-time applications, and it will be integrated into dedicated software to provide a live view of the formation rates directly at the CLOUD chamber during experiments. This will further streamline the analysis of new particle formation events and provide immediate feedback on changing experimental conditions.

Chapter 3

Modeling and Prediction of Sulfuric Acid Concentration

This chapter details the development of predictive models designed to forecast the concentration slope of sulfuric acid within the CLOUD chamber experiments, leveraging data from the CLOUD16 campaign, and it's core objective is to move beyond a purely descriptive or explanatory analysis and establish a robust, data-driven framework for predicting this crucial atmospheric variable across a wide range of controlled conditions.

The methodology begins with a foundational theoretical background on sulfuric acid formation, grounding the subsequent analysis in established atmospheric chemistry principles, which is followed by a description of the data preparation and cleaning procedures necessary to ensure the integrity of the modeling process. The chapter then transitions into a two-tiered modeling approach: first, an explanatory modeling phase is conducted to understand the key drivers and relationships within the data. This stage is focused on interpretability, identifying the direct physical and chemical influences on sulfuric acid formation; second, and central to this chapter, is the predictive modeling phase, involving the development and rigorous optimization of both a regularized linear model (Elastic Net) as a robust baseline and a powerful non-linear ensemble method (XGBoost) to capture more complex data patterns.

The chapter gives a comprehensive comparative analysis of the final, optimized models, which relies on both quantitative performance metrics, such as Root Mean Squared Error, and a qualitative assessment of diagnostic plots to determine which model is most suitable for practical predictive applications. The final model is selected based on its superior ability to generalize accurately and demonstrate a robust fit to the data, thereby providing a valuable tool for understanding and anticipating atmospheric nucleation events.

3.1 Theoretical Background: Sulfuric Acid Formation

Sulfuric acid (H_2SO_4) plays a pivotal role in atmospheric chemistry, particularly as a key precursor for new particle formation (NPF) events, which significantly influence global climate by contributing to cloud condensation nuclei (CCN). Its importance stems from its extremely low volatility and high hygroscopicity, properties that enable it to readily condense from the gas phase and form stable molecular clusters, even at the very low concentrations found in the atmosphere [65], and these initial clusters are the foundational building blocks for new aerosol particles.

The primary pathway for sulfuric acid formation in the troposphere involves the oxidation of sulfur

dioxide (SO_2), a trace gas emitted from both natural sources (e.g., volcanic activity, oceanic emissions) and anthropogenic activities (e.g., combustion of fossil fuels). This multi-step process is predominantly initiated by the hydroxyl radical (OH), a highly reactive oxidant produced through a key photochemical reaction [66], and in this process, the absorption of ultraviolet (UV) radiation from sunlight by ozone (O_3) leads to the creation of the hydroxyl radical in the presence of water vapor, which makes the intensity of sunlight a critical factor in the rate of sulfuric acid production. The hydroxyl radical subsequently initiates the oxidation of SO_2, forming a short-lived intermediate radical:

$$SO_2 + OH \longrightarrow HSO_3 \tag{3.1}$$

Subsequently, the HSO_3 radical rapidly reacts with molecular oxygen (O_2) and water vapor (H_2O) to yield sulfur trioxide (SO_3), which then swiftly hydrates to form the stable sulfuric acid molecule:

$$HSO_3 + O_2 \rightarrow SO_3 + HO_2SO_3 + H_2O \rightarrow H_2SO_4 \tag{3.2}$$

Within the controlled environment of the CLOUD chamber, sulfuric acid is often considered the "workhorse" molecule for NPF studies, as its controlled introduction allows researchers to precisely investigate the fundamental mechanisms governing particle nucleation. While sulfuric acid itself is essential for forming initial molecular clusters, CLOUD experiments have consistently shown that it often requires the synergistic interaction with other trace gases, such as ammonia (NH_3) or various amines, to overcome the critical nucleation barrier and grow into stable, larger particles [10]. Therefore, accurately understanding, measuring, and predicting sulfuric acid concentration is paramount for elucidating the complex processes of NPF and their atmospheric implications.

3.2 Data Preparation and Feature Engineering

The success of any machine learning model is heavily dependent on the quality and structure of its input data, which is why in this chapter a comprehensive data preparation and feature engineering pipeline was developed to transform the raw, time-dependent data from the CLOUD chamber into a clean, well-structured dataset suitable for the subsequent modeling tasks. The primary goal of this process was to ensure that the dataset accurately reflects the underlying physical and chemical processes, thereby enabling the models to learn meaningful relationships and produce reliable predictions, so this section details the steps taken to define stable intervals of sulfuric acid concentration, construct the final dataset, and provide a comprehensive description of the variables utilized in the analysis.

3.2.1 Defining Stable Intervals of Sulfuric Acid Concentration

To ensure the reliability of the dataset for subsequent modeling, a dedicated algorithm was developed to systematically identify and extract "stable intervals" of sulfuric acid concentration. This process was designed to filter out periods of high fluctuation and focus on data segments where the concentration remained relatively constant, thereby minimizing noise and spurious trends, and this methodology can be applied to any time-dependent quantity.

First, the raw sulfuric acid concentration data (H_2SO_4) was pre-processed using a Savitzky-Golay filter to produce a smoothed time series, a crucial step as it reduces high-frequency noise while preserving the general shape and peaks of the data curve. After smoothing, a convolution-based approach was used to perform an initial, coarse identification of "flat" sections, as for each data point, the sum of the

smoothed values over a preceding and a subsequent time window using two kernels was calculated. This process is represented by two convolutions, $C_a(t)$ and $C_b(t)$, which capture the sum of the values before and after each point, respectively.

The absolute and relative differences between these two sums were then used to identify potential intervals of stability. This relative difference, $R(t)$, was computed as:

$$R(t) = \frac{|C_b(t) - C_a(t)|}{\min\{C_b(t), C_a(t)\}} \tag{3.3}$$

Smaller values of R indicate points that lie in a stable region, and intervals were only selected for further analysis if their value of R was below a defined threshold and they were at least 30 minutes in duration.

In the second phase, each of the initially identified intervals was subjected to a more rigorous validation using linear regression. A linear model was fit to the smoothed data within each interval. An interval was only considered truly "stable" if the absolute value of its linear slope, m, was below a pre-defined threshold, and to account for inherent variability within the data, a second, more robust criterion was applied: the relative slope, s, had to be below a separate threshold, which was calculated as follows:

$$s = \frac{m}{\sigma} \times \Delta t \tag{3.4}$$

where m is the slope of the linear fit, Δt is the length of the interval, and σ is the standard deviation between the original and smoothed data within the interval. Only intervals satisfying both stringent criteria were retained for the final dataset, ensuring a robust selection of stable periods for further analysis.

The robust identification of stable sulfuric acid concentration periods was therefore contingent upon the careful selection of hyperparameters throughout the data processing pipeline. These technical choices, summarized in Table 3.1, dictated the sensitivity and specificity of the stability criteria.

Table 3.1: **Parameters Used for Defining Stable Sulfuric Acid Concentration Intervals.**This table summarizes the crucial hyperparameters used in the algorithm for identifying stable intervals in the raw sulfuric acid concentration data.

Minimum Interval Time (minutes)	30
Convolution Size	30
R **Threshold**	0.04
m **Threshold (cm^{-3} s^{-1})**	10
s **Threshold**	1.5

The plot in Figure 3.1 provides a clear visual example of a stable interval identified by the algorithm, where the original, noisy data is shown in blue, while the linear fit is in green, and it demonstrates how the algorithm successfully isolates a period where the sulfuric acid concentration is nearly constant, which is confirmed by the near-zero slope of the linear fit. This visual confirmation underscores the effectiveness of the method in isolating high-quality data for subsequent modeling.

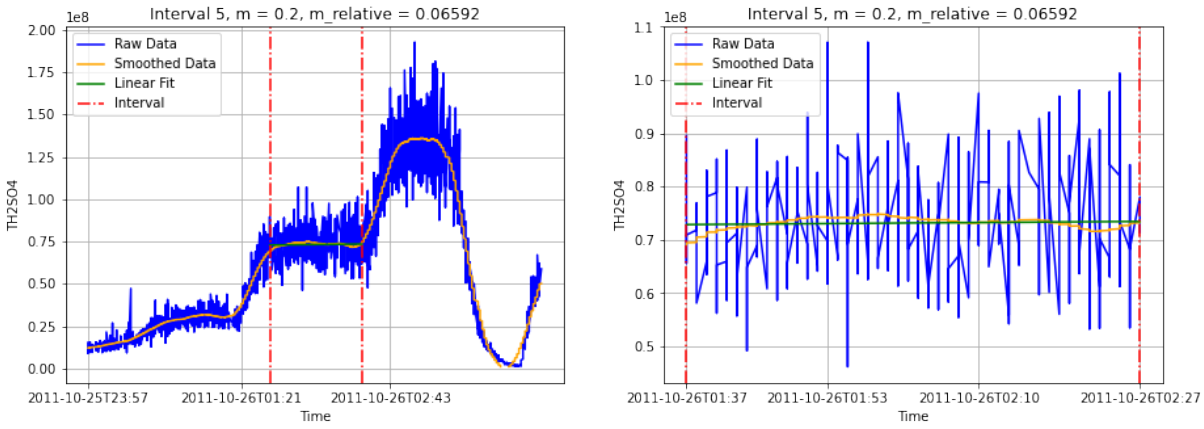


Figure 3.1: **Identification of a Stable Interval for Sulfuric Acid Concentration.** The left panel shows the full time series of sulfuric acid concentration with a representative stable interval highlighted by the red dashed lines. The raw data (blue) is processed with a Savitzky-Golay filter to produce a smoothed time series (orange). The right panel provides a zoomed-in view of the selected interval. The linear fit (green) within this period is nearly horizontal, visually confirming that the sulfuric acid concentration is consistent, and the algorithm has successfully isolated a high-quality data segment for subsequent analysis.

3.2.2 Dataset Construction

The construction of the final dataset involved several critical steps, including the careful selection of variables and the aggregation of data into meaningful samples, a process designed to transform the raw experimental data into a format suitable for machine learning, ensuring that each data point represents a stable and reliable observation. This section details the variables used and the core dataset compilation process.

3.2.2.1 Description of Variables

The dataset for this study was meticulously constructed from a variety of time-series measurements, each designed to capture distinct physical and chemical conditions within the CLOUD chamber during experimental runs. The selection of these variables was critical for building a robust model capable of understanding and predicting sulfuric acid concentrations.

The target variable for the predictive models was the average sulfuric acid (H_2SO_4) concentration measured during each identified stable interval.

The input features comprised a comprehensive set of parameters influencing atmospheric particle formation and growth: a proxy for the ultraviolet (UV) radiation intensity was included, as UV radiation is crucial for driving the photochemical reactions that produce sulfuric acid precursors. This value was derived from four UV lamps, which were kept continuously on to prevent damage from frequent cycling, where each lamp had a shutter that could be either open or closed, represented by a value of 1 or 0, respectively, so the final UV value for any given time point was determined by multiplying the lamp's intensity by its shutter state, and the maximum of these four values was then taken to represent the overall UV intensity.

To account for the chamber's internal dynamics, the speeds of the top and bottom fans were included. These values are vital for characterizing the mixing and turbulence within the chamber, which directly influence gas-phase concentrations and particle losses. The average chamber temperature, which was calculated as the mean of five separate temperature channels, was also included as a fundamental environmental parameter influencing reaction kinetics and gas-to-particle conversion processes, and the

average concentrations of ozone (O_3), sulfur dioxide (SO_2), and ammonia (NH_3) were also used as input features, as they are key reactants and intermediates in the atmospheric chemistry being studied.

A binary variable was also included to indicate whether the charged particle beam from the CERN Proton Synchrotron (PS) was active. This beam, which is a key part of CERN's accelerator complex [2], is used to simulate galactic cosmic rays and their potential influence on cloud formation [1]. Finally, both the average dew point and the average relative humidity were used as input features, as these parameters reflect the water vapor content, which is essential for sulfuric acid's hygroscopic properties and subsequent cluster formation. Each of these input features was averaged over its respective stable interval, ensuring consistency and reliability across the dataset.

3.2.2.2 Core Dataset Compilation

The core dataset, designated for the primary predictive modeling of sulfuric acid concentration, was meticulously compiled after the identification of stable intervals. The process involved extracting and aggregating key experimental parameters over these defined periods, ensuring that each data point represented a consistent and reliable snapshot of the chamber conditions.

Specifically, for each stable interval identified by the algorithm described in Section 3.2.1, the mean value of each relevant input feature was calculated. This included the average UV radiation intensity, average ozone concentration, average sulfur dioxide concentration, average fan speeds (top and bottom), average particle number concentration (N), average dew point and temperature. A few representative example rows of the dataset is shown in Table 3.2.

Table 3.2: **Sample of the Core Predictive Dataset. This table presents a small sample of the final dataset used for predictive modeling.** Each row represents an individual stable interval, with the input features averaged over that period. The table showcases the logarithm of the average sulfuric acid concentration as the target variable, along with examples of log-transformed and original input features. The log-transformed variables are expressed as $\log_{10}(C/C_0)$, where C_0 is the reference concentration of 1 cm^{-3}, rendering the value unitless.

Start Date	2011-10-25 10:06	2011-10-26 00:39	2011-10-26 01:37
End Date	2011-10-25 11:54	2011-10-26 01:14	2011-10-26 01:37
log(H_2SO_4) ± 0.15	6.11	7.48	7.86
log(SO_2) ± 0.10	1.37	1.50	1.51
log(O_3) ± 0.05	2.49	2.04	2.03
log(N) ± 0.20	1.76	1.63	2.86
Fan Speed (Bottom) (± 0.01 rpm)	343.87	343.87	343.87
Fan Speed (Top) (± 0.01 rpm)	343.90	343.89	343.87
Dew Point (± 0.01 ºC)	-7.78	-6.84	-8.40
Temperature (± 0.01 ºC)	5.30	5.29	5.29
UV (± 0.01 %)	0.00	10.00	25.00

A crucial step in preparing this dataset was the application of a logarithmic transformation to certain variables. This transformation was applied to the sulfuric acid concentrations, ozone, sulfur dioxide and particle number concentration. The primary purpose of this transformation was to normalize the data and mitigate the vast differences in orders of magnitude among these variables, as such normalization is essential for improving the performance of machine learning models, since it helps prevent features with larger numerical ranges from dominating the learning process. The compilation process ensured that the final dataset consisted of observations, each representing a stable period, with all features appropriately

scaled for robust model training.

3.2.2.3 Curve Fitting Methodology

A second dataset was constructed to specifically model the periods of transition where sulfuric acid concentration was observed to be rising. This dataset was used to explore the relationship between the averaged chemical and physical conditions and the parameters derived from a curve-fitting process, and to ensure the accuracy of the model, the starting and ending points of the sulfuric acid concentration rise for each transition were manually inspected and selected, as this was necessary to account for inconsistencies in the raw data. An example of this transition is shown in Figure 3.2.

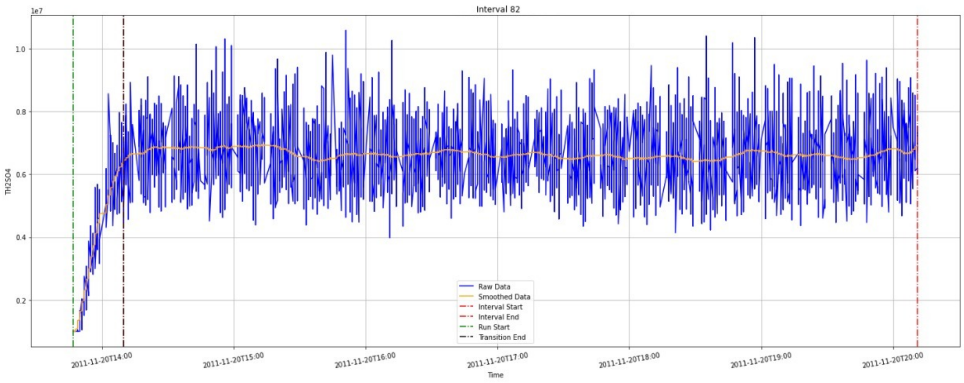


Figure 3.2: **Example of a Transition Period in Sulfuric Acid Concentration.** This figure illustrates a transition period selected for the second dataset. The plot shows the rise in the raw sulfuric acid concentration from a low stable baseline to a higher concentration, where the rate of gain significantly outweighs the rate of loss. The manually selected starting and ending points define the interval used to calculate the target variable, which represents the rate of new sulfuric acid formation under the averaged chamber conditions.

For this transition dataset, a subset of the variables from the stable interval dataset was used. The dew point, relative humidity, and the beam variable (which was always 1 and thus provided no analytical value) were removed. The speeds of the top and bottom fans were also combined into a single variable representing their mean speed. This resulted in a focused dataset containing only the most relevant variables for analyzing the chemical dynamics of the concentration rise.

To characterize the dynamics of these transitions, three different mathematical models were tested and compared: a linear, a power, and an exponential fit. Each fit modeled the sulfuric acid concentration as a function of time, providing a specific equation and set of fitted parameters. The linear fit, represented by the equation $y = mx + b$, where y is the sulfuric acid concentration and x is time, assumes a constant rate of change with m as the slope and b as the y-intercept. The power fit, given by the equation $y = ax^b$, is suitable for phenomena where the rate of change is proportional to a power of the variable, with a and b as the fitted parameters. Finally, the exponential fit, with the equation $y = ae^{bx}$, is used for processes that either grow or decay at a rate proportional to their current value, with a and b as the fitted parameters.

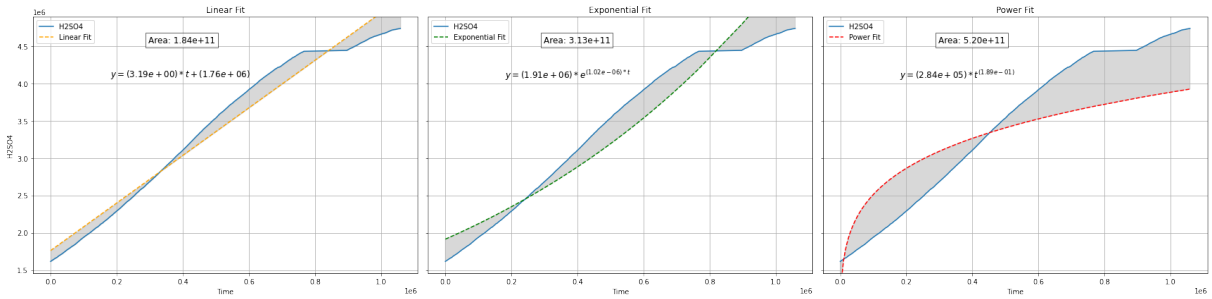


Figure 3.3: **Comparison of Linear, Power, and Exponential Fits for a Sulfuric Acid Concentration Rise.** The figure displays the raw sulfuric acid concentration data (blue points) during a transition period, along with fitted curves from three different models. The x-axis represents time, scaled from the start to the end of the specific transition in question, the green line represents the linear fit, the orange line the exponential fit, and the red dashed line the power fit, and the corresponding shaded areas represent the error bounds for each fit, with the wide grey area for the power fit indicating a poor model fit to the data. This comparison demonstrates that the linear and exponential models provide a better representation of the concentration rise.

After performing these fits, it was determined that the power model had the largest error, as indicated by the wide "grey area" surrounding the fitted curve in the plots, as it is clearly shown in figure 3.3. Furthermore, the exponential fit did not offer a significant improvement in reducing the overall error compared to the linear fit, so consequently, the linear model was chosen due to its simplicity and the ease with which its primary fitted parameter—the slope—could be interpreted as a direct measure of the sulfuric acid concentration rise rate. The final input features for this dataset were then created by averaging the key variables over the specified transition time frame, and the slope from the linear fit was used as the target variable for subsequent analysis. This rigorous process ensures that the model accurately captures the dynamics of the concentration changes, providing a quantitative basis for analyzing the experimental results.

3.3 Explanatory Modeling of Sulfuric Acid Formation

Building upon the meticulously prepared datasets, this section focuses on an explanatory modeling approach to unravel the underlying drivers of sulfuric acid concentration increases within the CLOUD chamber. While a stable interval of concentration is achieved when the rates of gain and loss are in approximate equilibrium, the focus of this analysis is on the preceding phase, where the gains in concentration are actively outweighing the losses, so by focusing on the linear slope observed during these periods of increase, the primary objective is to gain a deeper scientific understanding of the relationships between various physical and chemical parameters and the observed rate of sulfuric acid formation, which was achieved through a systematic process involving initial feature selection using an XGBoost model, followed by iterative linear regression analyses. These iterations progressively incorporated more complex feature sets, including non-linear transformations and interaction terms, to precisely identify how individual and combined factors influence the dynamics of sulfuric acid concentration changes.

3.3.1 Feature Engineering and Selection

The explanatory modeling process began with a systematic approach to feature engineering and selection, which was crucial for uncovering the complex relationships that govern sulfuric acid concentration changes. This process was executed in a staged manner, with each stage building upon the previous one.

The initial step was to identify the most influential variables from the base feature set, which included

the average concentrations of ozone (O_3), sulfur dioxide (SO_2), and ammonia (NH_3), the UV radiation proxy, the mean fan speed, and the average chamber temperature. An XGBoost model was trained on the transition dataset, and its feature importance functionality was leveraged to select the top 10 most significant predictors of the sulfuric acid slope. For this purpose, the model was allowed to overfit the data, as the objective was not to predict on new, unseen data but to thoroughly explore the relationships within the existing dataset to derive feature importance rankings.

The next stage involved engineering a comprehensive set of non-linear features from these base variables to account for potential non-linear dependencies in the underlying chemistry, where a variety of transformations were applied, including polynomial terms (squared and cubed, x^2, x^3), fractional powers (square root and cube root, $x^{1/2}$, $x^{1/3}$), inverse powers ($1/x$, $1/x^2$, $1/x^3$), the base-10 logarithm ($\log_{10}(x)$), and the absolute value ($|x|$). The exponential function (e^x) was initially considered but was not used due to numerical overflow issues with the dataset's value, and these transformations were selectively applied based on the data to avoid errors (e.g., roots and logarithms were only applied to non-negative values, while inverse functions were only applied to non-zero values).

Following this, the analysis was expanded to include interaction terms, which are critical for understanding how different variables work in synergy, which were comprised of all possible two-way combinations from the base features, created by multiplying every unique pair of features ($x_1 \cdot x_2$).

After each stage of feature engineering, the XGBoost model was re-run to determine the new top 10 most important features. This allowed for a clear view of how the importance scores shifted as more complex relationships were introduced into the model.

Table 3.3: **The Top 10 Most Important Features for Each Stage of Feature Engineering.** The importance scores for each feature were derived from a trained XGBoost model. The table shows how the ranking of features changed as non-linear and interaction terms were added to the feature set.

Rank	Base Features	Non-Linear Features	Two-Way Interactions
1	UV Intensity	Temperature	$\text{Temperature}^{-2} \times \log_{10}(SO_2)$
2	O_3	UV Intensity	UV Intensity $\times$ Temperature
3	Temperature	NH_3	UV Intensity $\times$ Fan Speed^{-1}
4	NH_3	Temperature^2	UV Intensity $\times$ Fan Speed^{-3}
5	SO_2	O_3	$SO_2 \times \text{Temperature}^{-3}$
6	Fan Speed	SO_2	$O_3^{-1} \times \text{Temperature}^{-1}$
7	-	Fan Speed	$\text{UV Intensity}^2 \times \text{Temperature}^{-2}$
8	-	Temperature^{-1}	$O_3^{-1} \times \text{Temperature}^3$
9	-	-	UV Intensity $\times$ NH_3^{-1}
10	-	-	UV Intensity $\times$ NH_3^2

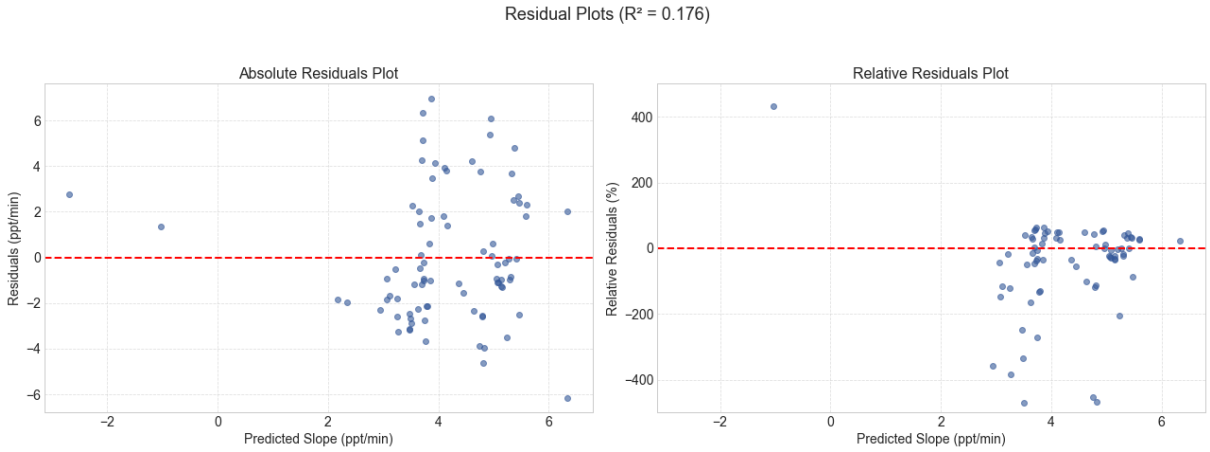
As shown in Table 3.3, the feature importance scores shifted significantly with each stage of engineering, revealing the most influential factors at different levels of complexity. The initial base features provided a foundational view, but as new non-linear and interaction terms were introduced, their importance often superseded that of the original variables, which suggests that the relationship between these factors and the rate of sulfuric acid increase is not purely linear but involves more complex, synergistic effects. Notably, some base features, like temperaute and the UV intensity, consistently appeared in the top rankings across different stages, while others were displaced by more descriptive non-linear or interaction terms. For the non-linear features and subsequent interaction terms, some features with a negligible importance score were excluded from the table. This systematic approach to feature engineering

and selection was essential for moving beyond simple correlations and identifying the key drivers of the chemical processes within the CLOUD chamber.

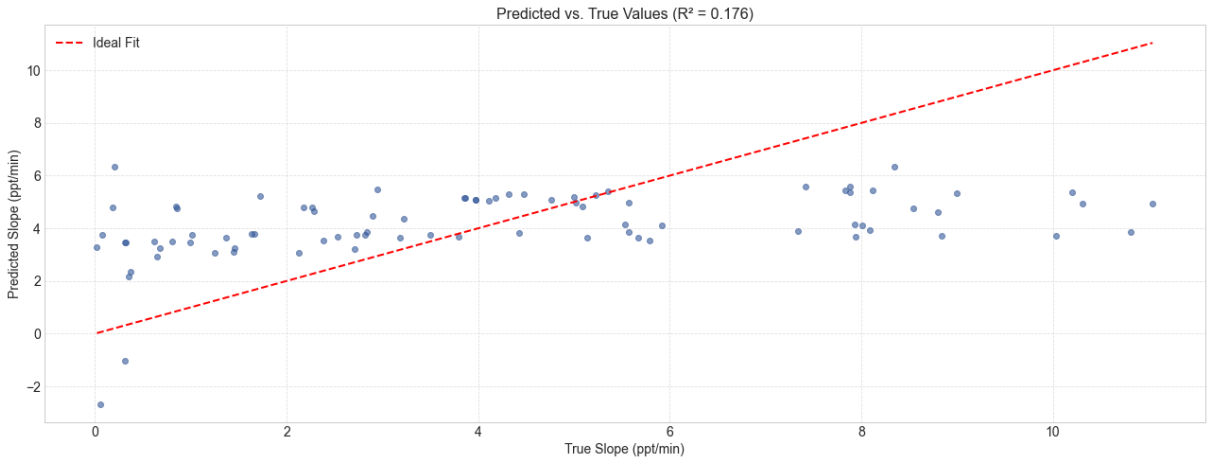
3.3.2 Explanatory Linear Regression Results

This section presents the results of the iterative explanatory modeling process, which moved from a basic linear model to more complex formulations that better capture the intricate relationships driving sulfuric acid concentration increases. Each stage of the analysis built upon the last, progressively revealing a deeper understanding of the system's underlying dynamics, where the linear regression model was trained exclusively on the variables from the corresponding column of Table 3.3, ensuring a clear and direct assessment of how each set of engineered features contributed to the model's explanatory power.

3.3.2.1 Baseline Model with Base Features



(a) **Absolute and Relative Residual Plots:** Plots showing the raw and normalized errors of the model's predictions. The non-random patterns and high error values indicate that key information is missing from the model.



(b) **Predicted vs. True Value Plot:** A scatter plot comparing the model's predictions to the actual measured values. The wide distribution of points around the ideal fit line confirms the model's limited explanatory power.

Figure 3.4: **Explanatory Linear Regression Model Performance Plots for Base Features ($R^2 = 0.176$).** This figure presents the diagnostic plots for the baseline linear regression model, which used only the base features to explain the sulfuric acid concentration slope. The low coefficient of determination (R^2) of 0.176 indicates that these features alone could explain only a small fraction of the total variance.

The initial linear regression model, which used only the base features identified by the XGBoost

analysis, served as a baseline for all subsequent models. This model yielded an R^2 of 0.176, indicating that these foundational features could explain approximately 17.6% of the variance in the sulfuric acid concentration slope, and while this provides a preliminary indication of a relationship between the core parameters and the rate of sulfuric acid increase, it also suggests that a significant portion of the variance remains unexplained.

As shown in Figure 3.4, the predicted versus true value plot reveals a scattered pattern of points around the ideal diagonal line, further confirming the model's limited explanatory power. The residual plots, which are crucial for diagnosing model fit, show both the absolute and relative residuals, the absolute residual plot displays the raw difference between the predicted and true values, while the relative residual plot normalizes these errors. The distribution of residuals in both plots reveals remaining structure and trends, indicating that the model is missing key information or that the relationships are not strictly linear.

3.3.2.2 Model with Non-Linear Features

The model's explanatory power was then evaluated after introducing the engineered non-linear features, which included polynomial, inverse, and logarithmic transformations. The inclusion of these terms led to a substantial improvement in the model's performance, with the R^2 increasing significantly to 0.665, which represents a significant leap from the baseline model, as the non-linear features now account for approximately 66.5% of the variance in the sulfuric acid concentration slope.

As shown in Figure 3.5, the visual evidence of this improvement is striking, as the predicted versus true value plot now displays points that are much more tightly clustered around the ideal diagonal line, indicating a stronger correlation between the model's predictions and the actual values, and the residual plots also show a distinct change, with the errors appearing more randomly distributed, which indicates that the non-linear transformations were crucial for addressing the systematic errors present in the baseline model and capturing the underlying non-linear relationships that govern the chemical processes within the chamber.

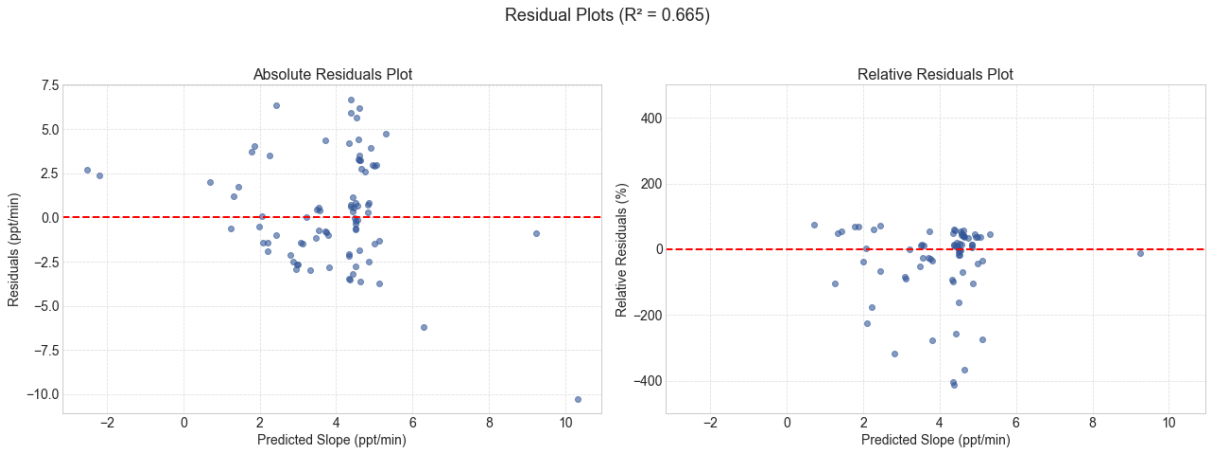
3.3.2.3 Model with Two-Way Interaction Terms

The final stage of the analysis involved introducing two-way interaction terms, which are critical for capturing the synergistic effects between variables, where this final model achieved the highest R^2 of 0.720, representing a further increase in explanatory power over the non-linear model. The progressive increase in the R^2 value from the baseline model to this stage underscores the importance of a comprehensive feature engineering approach.

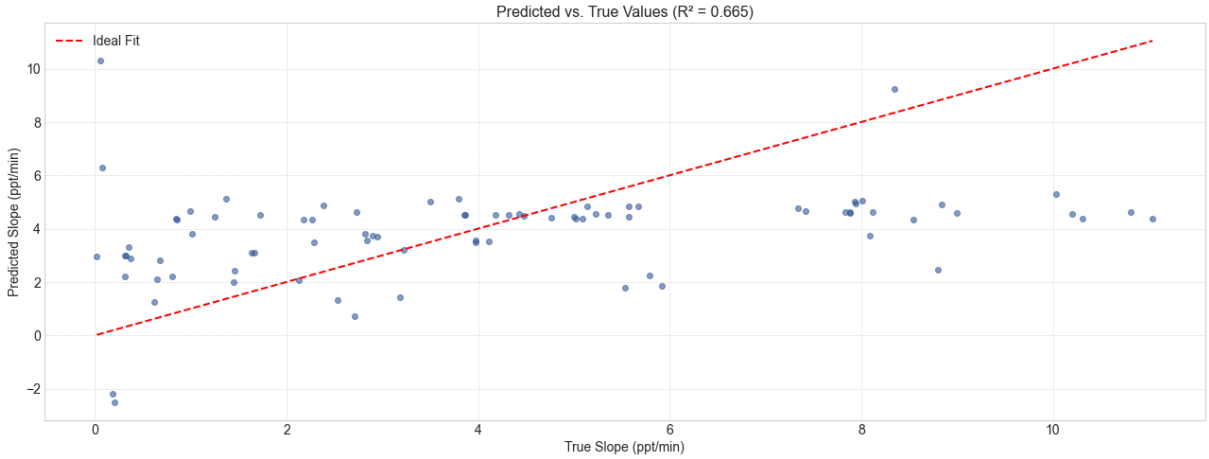
As shown in Figure 3.6, the predicted versus true value plot now exhibits a much tighter clustering of points along the diagonal line, indicating that the model's predictions are in close agreement with the true values. The residual plots also show a distribution of errors that is more random and centered around zero, suggesting that the interaction terms successfully accounted for much of the remaining systematic variance, and this strong fit indicates that the synergistic relationships between the measured parameters play a significant role in determining the rate of sulfuric acid formation.

3.3.2.4 Summary of Model Performance

The results from the iterative linear regression models are shown in Table 3.4 and demonstrate the increasing explanatory power of a systematic feature engineering approach, as the coefficient of determi-



(a) **Absolute and Relative Residual Plots:** Plots showing the raw and normalized errors of the model's predictions. The distribution of points in these plots is much more random than in the baseline model, indicating that the non-linear features successfully addressed the systematic errors present previously.



(b) **Predicted vs. True Value Plot:** A scatter plot comparing the model's predictions to the actual measured values. The points are now much more tightly clustered around the ideal fit line, confirming the substantial improvement in the model's predictive power.

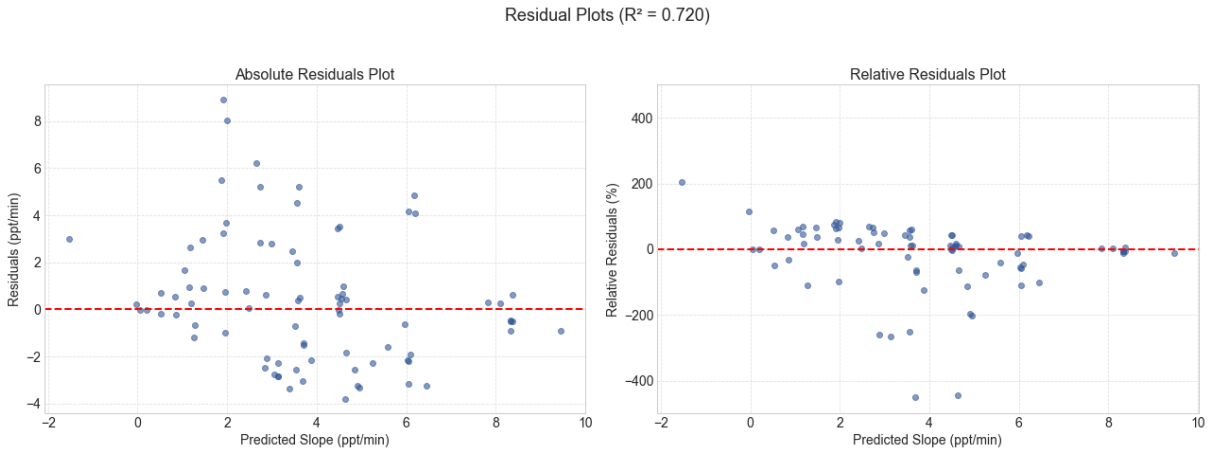
Figure 3.5: **Explanatory Linear Regression Model Performance Plots with Non-Linear Features ($R^2 = 0.665$).** This figure presents the diagnostic plots for the enhanced linear regression model, which includes non-linear feature transformations. The significantly higher coefficient of determination (R^2) of 0.665 indicates that these features were highly effective in capturing the variance in the sulfuric acid concentration slope.

nation (R^2) provides a quantitative summary of each model's ability to explain the variance in the sulfuric acid concentration slope.

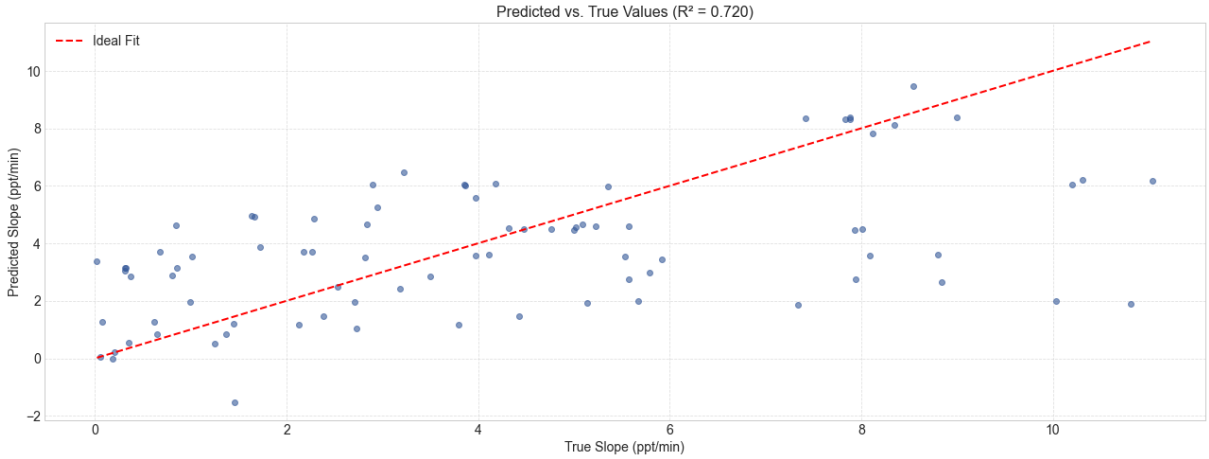
Table 3.4: **Summary of Linear Regression Model Performance.** This table summarizes the performance of the three linear regression models evaluated in this study, detailing their effectiveness in explaining the sulfuric acid concentration slope. The models are presented in the order of their complexity, starting with the base features and progressively adding non-linear and interaction terms. The performance is assessed using key statistical metrics: the Coefficient of Determination (R^2), which quantifies the proportion of variance in the dependent variable explained by the model.

Model Stage	R^2 Value
Baseline Model (Base Features)	0.176
Model with Non-Linear Features	0.665
Model with Two-Way Interactions	0.720

The progressive increase in the R^2 value from a baseline of 0.176 to a final value of 0.720 validates the hypothesis that non-linear and synergistic effects play a critical role in the chemical processes being



(a) **Absolute and Relative Residual Plots:** These plots show the model's errors after including the interaction features. The random scatter of points around the zero-line indicates that the model's errors are again unsystematic, suggesting that the model's assumptions have been met and that it is capturing most of the linear and non-linear patterns in the data.



(b) **Predicted vs. True Value Plot:** A scatter plot comparing the model's predictions to the actual measured values. The points are tightly clustered around the ideal fit line, with the lowest variance of all three models, which confirms the significant improvement in the model's explanatory power and its ability to accurately predict the sulfuric acid concentration slope.

Figure 3.6: **Explanatory Linear Regression Model Performance Plots with Two-Way Interactions ($R^2 = 0.720$).** This figure presents the final diagnostic plots for the most comprehensive linear regression model, which includes both non-linear and two-way interaction features. The high coefficient of determination (R^2) of 0.720 demonstrates that this model successfully captures the complex relationships and synergistic effects between the explanatory variables.

studied. The initial model with base features provided only a limited explanation, but the inclusion of non-linear transformations and, subsequently, two-way interaction terms, significantly improved the model's ability to capture the complex relationships between the measured parameters and the rate of sulfuric acid formation. This systematic analysis not only provided a robust model but also offered valuable insights into the underlying physical and chemical dynamics within the chamber.

3.4 Predictive Modeling of Sulfuric Acid Concentration

While the previous section focused on explanatory modeling to understand the underlying drivers of sulfuric acid concentration increases, this section shifts to a predictive modeling approach. The primary objective here is to develop a robust model capable of accurately forecasting future sulfuric acid concentration levels based on the measured physical and chemical parameters within the CLOUD chamber, and unlike explanatory models, which prioritize interpretability and causal inference, predictive mod-

els are optimized for forecasting performance on unseen data, making them valuable tools for real-time monitoring, early warning systems, and scenario simulations in atmospheric science.

3.4.1 Linear Regression with Regularization (Elastic Net)

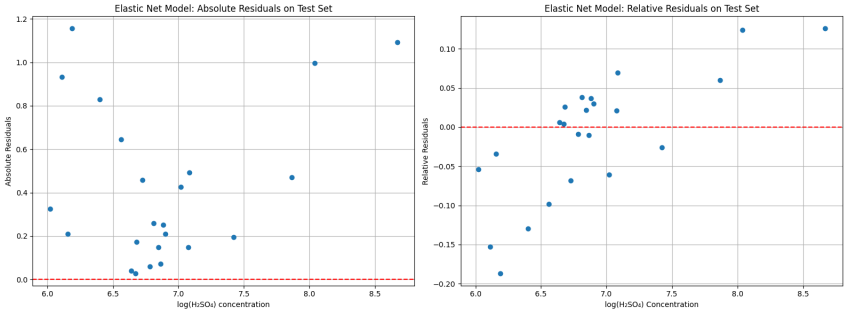
The first predictive model employed was an Elastic Net regression, a powerful linear method that mitigates the risk of overfitting by combining two distinct forms of regularization: L1 and L2. The L1 penalty promotes sparsity by driving some feature coefficients to zero, effectively performing automatic feature selection, while the L2 penalty stabilizes the model by shrinking all coefficients.

Model optimization was a critical step to achieve peak performance, so a systematic grid search with cross-validation was performed over a predefined hyperparameter space to find the optimal balance between regularization strength and feature mix. The initial search grid was deliberately broad to explore a wide range of values for the Alpha parameter (which controls the overall penalty strength) and L1 Ratio (which determines the blend of L1 and L2 regularization), and the specific ranges explored were:

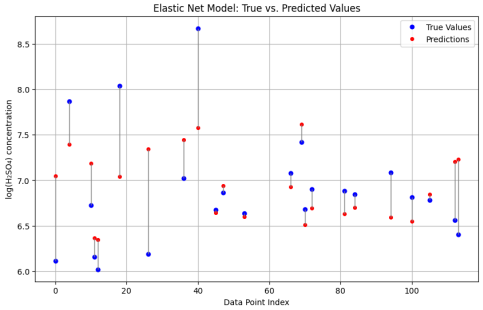
Table 3.5: **Grid Search Parameters for Elastic Net Hyperparameter Tuning.** This table outlines the initial hyperparameter grid used for tuning the Elastic Net model, which systematically tested combinations of two key parameters: Alpha, which controls the overall regularization strength, and L1 Ratio, which dictates the mix between L1 (Lasso) and L2 (Ridge) penalties. This process was performed to identify the optimal parameter combination that minimized the model's error on the validation data.

alpha	0.1	1.0	10.0	50.0	100.0	-
L1 Ratio	0.10	0.50	0.70	0.90	0.95	1.00

Following this exhaustive search, the best parameters were identified as {alpha: 0.1, L1 Ratio: 0.1}. This result indicated that the optimal model required a relatively small amount of regularization and a strong preference for the L2 penalty, and the final, optimized model achieved a Root Mean Squared Error (RMSE) of 0.5439 on the held-out test set, demonstrating a significant improvement over the base linear model and highlighting the effectiveness of regularization in improving predictive accuracy.



(a) **Residual Plots:**These plots show the model's errors against its predicted values. The Absolute Residuals plot (left) displays the raw prediction error, while the Relative Residuals plot (right) normalizes these errors by the true value. The distribution of these errors helps to diagnose potential issues like systematic bias in the model's predictions.



(b) **Predicted vs. True Values:** A visual representation of the model's accuracy on a per-data-point basis. For each observation (indexed on the x-axis), the vertical line connecting the true value to the predicted value represents the absolute error. The plot shows the overall distribution of errors and helps identify where the model's predictions deviate most from reality.

Figure 3.7: This figure presents the diagnostic plots for the initial Elastic Net model, optimized using a broad grid search. The plots illustrate the model's predictive performance on the unseen test set, serving as a baseline for further hyperparameter fine-tuning.

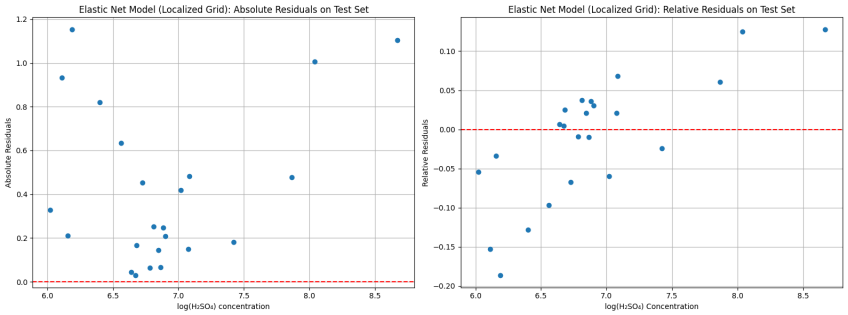
The diagnostic plots for this model's performance on the test set are presented in Figure 3.7. The Predicted vs. True Values plot provides a visual diagnostic of the model's accuracy on a per-data-point basis, where for each observation (indexed on the x-axis), a vertical line connects the true measured value to the model's predicted value, and the length of these vertical line segments directly represents the absolute error of the prediction. A visual inspection of this plot reveals that while most errors are small, some points exhibit significant deviation, particularly in areas of higher concentration values, and the Residuals plots further confirm the model's behavior, as the distribution of points appears more random and centered around zero, suggesting that the model's errors are not systematic. These findings warranted a more localized grid search in a subsequent step to determine if even more marginal gains in performance could be achieved with a finer resolution of the hyperparameters.

A refined localized grid search was then conducted to fine-tune the Elastic Net model's hyperparameters. This step was taken to explore a narrower range of values around the optimal parameters found in the initial broad grid search, aiming for a final, marginal improvement in performance, and a localized search grid was meticulously defined to center around the previously identified optimal region:

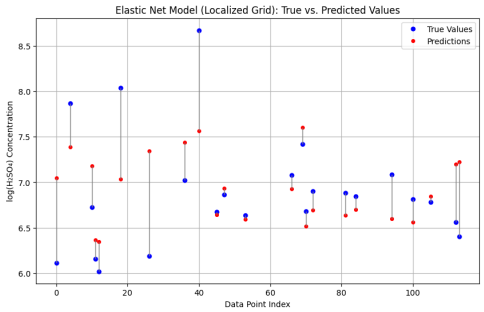
Table 3.6: **Grid Search Parameters for Elastic Net Fine-Tuning.** This table outlines the refined hyperparameter grid used to fine-tune the Elastic Net model, where the parameters were chosen based on the results of the initial broad grid search, focusing on a narrower range of values to precisely optimize the model's regularization strength (Alpha) and the blend of L1 and L2 penalties (L1 Ratio) for peak predictive performance.

alpha	0.01	0.05	0.1	0.5	1.0
L1 Ratio	0.01	0.05	0.1	0.5	1.0

The best parameters from this localized search were identified as {alpha: 0.05, L1 Ratio: 0.5}. This refinement led to a final model with a Root Mean Squared Error (RMSE) of 0.54326 on the test set, and while this represents a marginal reduction in error from the initial model's RMSE of 0.5439, it confirms the effectiveness of the fine-tuning process in extracting the maximum predictive performance from the Elastic Net model.



(a) **Residual Plots:** These plots show the model's errors against its predicted values. The Absolute Residuals plot (left) displays the raw prediction error, while the Relative Residuals plot (right) normalizes these errors by the true value. The distribution of points is more tightly clustered around the zero-line than in the previous model, confirming the improvement gained from the localized hyperparameter tuning.



(b) **Predicted vs. True Values:** A visual representation of the model's accuracy on a per-data-point basis. For each observation, the vertical line connecting the true value to the predicted value represents the absolute error. Compared to the non-localized model, a subtle but observable reduction in the length of these lines indicates a decrease in prediction errors.

Figure 3.8: This figure presents the diagnostic plots for the refined Elastic Net model, optimized using a localized grid search. The plots illustrate the model's enhanced predictive performance on the unseen test set after fine-tuning its hyperparameters.

The diagnostic plots for this refined model, presented in Figure 3.8, demonstrate the tangible impact of the localized tuning. A visual comparison of the Predicted vs. True Values plot with its predecessor reveals a subtle but noteworthy reduction in the overall length of the vertical line segments, indicating a decrease in prediction errors at the individual data point level. The Residuals plots also reinforce this improvement; while the distribution of points remains acceptably random, their clustering around the zero-line is slightly tighter and more compact, reinforcing the conclusion that the localized tuning process resulted in a more accurate and robust predictive model. This optimized Elastic Net model will serve as a strong baseline for comparison against the XGBoost model.

3.4.2 XGBoost Modeling

Building upon the linear baseline, an XGBoost (Extreme Gradient Boosting) model was also developed to explore the potential for improved predictive power by capturing complex, non-linear relationships. As a powerful ensemble method, XGBoost is particularly well-suited for structured data and is widely recognized for its high performance in regression tasks.

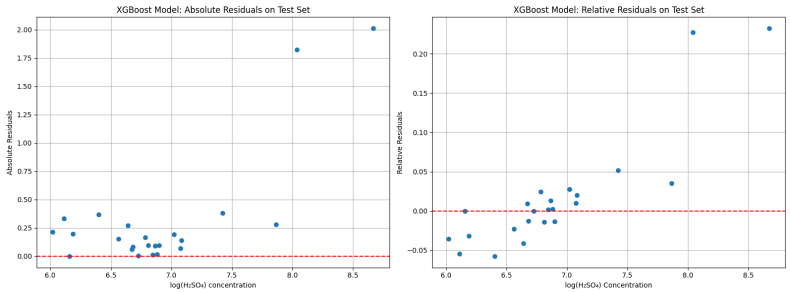
The XGBoost model underwent a comprehensive hyperparameter tuning process via a wide-ranging

grid search with cross-validation, as this approach systematically evaluated numerous combinations of parameters to identify the optimal configuration for the model. The full search grid included:

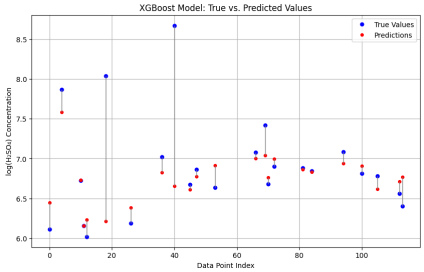
Table 3.7: **Grid Search Parameters for Initial XGBoost Hyperparameter Tuning.** This table outlines the comprehensive hyperparameter grid used to tune the initial XGBoost model. The grid search systematically explored a wide range of values for critical parameters, like the number of estimators, learning rate, and tree depth, with the goal of identifying the optimal configuration that minimizes the model's error on the validation set.

n_estimators	10	20	30	-
learning_rate	0.01	0.05	0.10	-
max_depth	3	4	5	-
gamma	0.0	0.1	0.2	-
subsample	0.8	0.9	1.0	-
colsample_bytree	0.8	0.9	1.0	-
min_child_weight	1	5	10	-
reg_alpha	0.0	0.1	0.5	1.0
reg_lambda	0.0	0.1	0.5	1.0

The grid search identified the best-performing hyperparameters as: {'colsample_bytree': 0.9, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 4, 'min_child_weight': 1, 'n_estimators': 30, 'reg_alpha': 0, 'reg_lambda': 0, 'subsample': 0.9}. This result indicates that the optimal model configuration favored a low gamma value (minimal splits) and no regularization, with a moderate learning rate and tree depth. The final, optimized XGBoost model achieved a Root Mean Squared Error (RMSE) of 0.5956 on the held-out test set. While this RMSE is slightly higher than the Elastic Net model's, it is important to note that the R-squared value for the XGBoost model on the test set was higher, indicating that it explained a larger proportion of the variance in the data.



(a) **Residual Plots:**These plots display the model's errors against its predictions. The Absolute Residuals plot (left) shows the raw prediction error, while the Relative Residuals plot (right) normalizes these errors by the true value. The random distribution of errors, with no clear patterns, confirms that the XGBoost model effectively captured the underlying relationships in the data.



(b) **Predicted vs. True Values:** A visual representation of the model's accuracy on a per-data-point basis. For each observation (indexed on the x-axis), a vertical line connects the true measured value to the model's predicted value. The shorter line segments indicate higher predictive accuracy for that specific data point.

Figure 3.9: This figure presents the diagnostic plots for the initial XGBoost model, which was optimized using a broad grid search. The plots illustrate the model's predictive performance on the unseen test set, demonstrating its ability to capture complex, non-linear patterns.

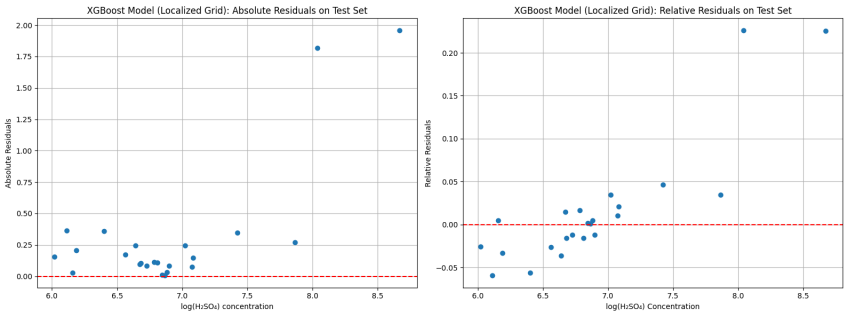
The diagnostic plots for the XGBoost model's performance on the test set are presented in Figure 3.9. The Predicted vs. True Values plot shows that the model's predictions are well-distributed around the ideal fit line, with a more random scatter than the Elastic Net. The Residuals plots further confirm this behavior, with the errors appearing to be more randomly distributed and with a less pronounced conical shape than the previous model, suggesting that the XGBoost model is better at capturing the complex, underlying patterns in the data.

To extract the maximal performance from the XGBoost model, a localized grid search was performed, which involved a more granular exploration of the hyperparameter space centered around the optimal values identified in the initial broad search. The localized search grid was meticulously defined as:

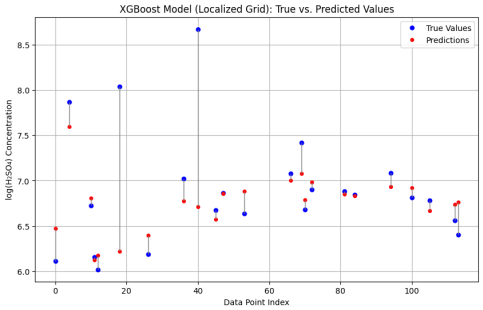
Table 3.8: **Localized Grid Search Parameters for Final XGBoost Model Tuning.** This table outlines the refined hyperparameter grid used to fine-tune the XGBoost model. This localized search was conducted to explore a more granular range of values for key parameters, aiming for a final, marginal improvement in the model's predictive performance beyond the initial broad grid search.

n_estimators	25	30	35
learning_rate	0.08	0.1	0.12
max_depth	3	4	5
gamma	0	0.05	-
subsample	0.85	0.9	0.95
colsample_bytree	0.85	0.9	0.95
min_child_weight	1	2	-
reg_alpha	0	0.001	0.01
reg_lambda	0	0..001	0.01

The best parameters from this localized search were identified as: {'colsample_bytree': 0.9, 'gamma': 0, 'learning_rate': 0.12, 'max_depth': 4, 'min_child_weight': 1, 'n_estimators': 30, 'reg_alpha': 0.001, 'reg_lambda': 0.01, 'subsample': 0.9}. This fine-tuning resulted in a final model that achieved a Root Mean Squared Error (RMSE) of 0.5855 on the test set. While this represents a small improvement over the non-localized model's RMSE of 0.5956, it demonstrates the effectiveness of the fine-tuning process in optimizing the model for a more precise fit to the data.



(a) **Residual Plots:** These plots display the final model's errors against its predictions. The Absolute Residuals plot (left) and the Relative Residuals plot (right) both show a tighter, more concentrated distribution of points around the zero-line. The overall compactness of the residual spread confirms that the final, fine-tuned model is more accurate and robust than its predecessor.



(b) **Predicted vs. True Values:** A visual representation of the final model's accuracy on a per-data-point basis. Compared to the initial XGBoost model, the vertical lines connecting the true and predicted values are subtly but consistently shorter, indicating a measurable reduction in prediction error achieved through the localized tuning.

Figure 3.10: **Final XGBoost Model Diagnostic Plots (Localized Grid).** This figure presents the final diagnostic plots for the XGBoost model, optimized through a localized grid search. The plots demonstrate the model's superior predictive performance on the unseen test set, showcasing the effectiveness of fine-tuning the hyperparameters.

The diagnostic plots for this final model are presented in Figure 3.10. A visual comparison with

the previous XGBoost plots reveals a subtle but noteworthy reduction in the overall magnitude of the prediction errors, and in the Predicted vs. True Values plot, the vertical line segments appear marginally shorter, indicating a decrease in prediction error on a per-point basis. The Residuals plots also show a more compact distribution of errors around the zero-line, confirming that the localized tuning successfully reduced the remaining prediction variance. This final optimized XGBoost model will now be used for the final comparative analysis and for making predictions on new, unseen data.

3.4.3 Comparative Analysis and Model Evaluation

This section presents a comparative analysis of the optimized Elastic Net and XGBoost models, evaluating their respective performances on the unseen test set to determine the most effective predictive tool for sulfuric acid concentration slope. The selection of the final model is based on a comprehensive assessment of quantitative metrics and qualitative insights derived from diagnostic plots.

The Elastic Net model, serving as our regularized linear baseline, demonstrated robust performance with a Root Mean Squared Error (RMSE) of 0.5433. This model effectively leveraged the combined strengths of L1 and L2 regularization to balance the bias-variance trade-off, resulting in a stable and interpretable linear predictor, and while its overall RMSE was slightly lower than the initial XGBoost model, it is crucial to consider the nature of the errors, as the diagnostic plots for the Elastic Net model, particularly the Predicted vs. True Values plot, showed a pattern where errors increased with the magnitude of the predicted slope.

In contrast, the XGBoost model, a non-linear ensemble method, achieved a final RMSE of 0.5855 after extensive localized hyperparameter tuning. Although its RMSE was marginally higher than the Elastic Net, the XGBoost model often provides a better overall fit for complex, non-linear relationships. The diagnostic plots for the XGBoost model revealed a more uniform and random distribution of errors in both the Predicted vs. True Values and Residuals plots, which suggests that the XGBoost model was more successful at capturing intricate patterns within the data, leading to errors that are less systematic and more evenly spread across the range of predictions. The absence of a clear trend in the residuals indicates a more robust and generalized predictive capacity across varying conditions.

Considering both quantitative metrics and the qualitative insights from the diagnostic plots, the XGBoost model is selected as the final predictive model. Despite a marginally higher RMSE, its ability to model non-linear relationships resulted in a more random and less structured error distribution, signifying a superior fit to the underlying data complexity, which is critical for a robust predictive tool in environmental science, where complex interactions between variables are common. The XGBoost model is therefore deemed more suitable for accurately forecasting sulfuric acid concentration slope in novel conditions.

Chapter 4

Conclusion and Final Remarks

This project successfully addressed key limitations of pre-existing scripts for calculating new particle formation (NPF) rates within the CLOUD experiment, ultimately creating a more transparent, robust, and scientifically accurate computational framework. The re-implementation in Python demonstrated significant advancements over the previous R and Matlab versions, particularly in ensuring dimensional consistency and providing a more responsive and faithful representation of raw data through the use of the Savitz-Golay filter. This new script avoids the massive, artifact-like spikes seen in the R implementation and shows strong agreement with the Matlab version for dilution and wall sinks, while providing a more physically accurate calculation of the coagulation sink.

Additionally, the project developed a reliable predictive framework for forecasting sulfuric acid concentration slopes. The explanatory modeling demonstrated that while a baseline linear model with core features could explain only a small fraction of the variance, the incorporation of non-linear features and interaction terms significantly increased the model's explanatory power, achieving an R^2 of 0.720. The XGBoost predictive model, after localized hyperparameter tuning, exhibited superior performance, effectively capturing complex, non-linear patterns in the data and demonstrating its potential for real-time predictive applications within the experiment. This work contributes to the broader goal of reducing uncertainties in climate models by providing a more reliable foundation for understanding aerosol-cloud interactions and the role of galactic cosmic rays.

Building upon the work accomplished, the next steps for this project involve both the practical application of the developed scripts and the conceptual expansion of the predictive framework.

First, the new Python scripts for NPF rate calculation and sulfuric acid prediction should be integrated into the CLOUD Data Acquisition System (DAQ). This integration would enable real-time monitoring of key atmospheric processes during experiments, allowing researchers to make immediate adjustments and optimizations to their experimental parameters. A key upgrade would involve adapting the script's output to provide final formation rates in real time directly at the CLOUD chamber. This would streamline the analysis of new particle formation events and provide immediate feedback on changing experimental conditions.

Second, the predictive framework for sulfuric acid concentration can be expanded to include other important atmospheric species, such as highly oxygenated organic molecules (HOMs) and amines. This would provide a more holistic predictive tool for understanding the complex chemistry within the chamber, moving beyond the current focus on sulfuric acid alone. Given the success of the XGBoost model in capturing complex, non-linear patterns in the sulfuric acid data, similar machine learning approaches could be applied to predict the concentrations of these other species.

Finally, an exciting new direction is the exploration of chemistry-informed machine learning mod-

els. While the current predictive models are data-driven, a chemistry-informed approach would incorporate fundamental physical and chemical laws—such as reaction kinetics and mass balance equations—directly into the model's architecture or loss function. This would constrain the model's predictions to be not only statistically accurate but also physically plausible, increasing their trustworthiness and generalizability. Such an approach would bridge the gap between purely empirical data fitting and foundational atmospheric chemistry, providing a powerful tool for advancing our understanding of these complex systems.

Bibliography

[1] *The CLOUD Experiment*. https://home.cern/science/experiments/cloud. Accessed: 2025-08-18. CERN, 2024.

[2] *Proton Synchrotron*. https://home.cern/science/accelerators/proton-synchrotron. Accessed: 2025-08-18. CERN, 2024.

[3] *CERN experiment sheds new light on cloud formation*. https://home.cern/news/news/experiments/cern-experiment-sheds-new-light-cloud-formation. Accessed: 2025-09-08. CERN, Aug. 2011.

[4] J. Kirkby et al.: "Role of sulphuric acid, ammonia and galactic cosmic rays in atmospheric aerosol nucleation", Nature, 4767361, pp. 429–433, 2011, DOI: 10.1038/nature10343.

[5] Francesco Riccobono et al.: "Oxidation Products of Biogenic Emissions Contribute to Nucleation", Science, 3446185, pp. 717–721, 2014, DOI: 10.1126/science.1243527.

[6] Eimear M. Dunne et al.: "Global atmospheric particle formation from CERN CLOUD measurements", Science, 3546316, pp. 1119–1124, 2016, DOI: 10.1126/science.aaf2649.

[7] Jasmin Tröstl et al.: "The role of highly oxidized organic molecules in atmospheric aerosol nucleation", Nature, 5337604, pp. 521–526, 2016, DOI: 10.1038/nature17953.

[8] Nianci Yao et al.: "Role of sesquiterpenes in biogenic new particle formation", Science Advances, 936, eadi5297, 2023, DOI: 10.1126/sciadv.adi5297.

[9] M. Boy et al. "Biogenic Sesquiterpenes and Atmospheric New Particle Formation: A Boreal Forest Site Investigation". In: *Nucleation and Atmospheric Aerosols*. Ed. by C.D. O'Dowd and P.E. Wagner. Dordrecht: Springer, 2007, pp. 97–116. DOI: 10.1007/978-1-4020-6475-3_70.

[10] Jasper Kirkby et al.: "Atmospheric new particle formation from the CERN CLOUD experiment", Nature Geoscience, 1611, pp. 948–957, 2023, DOI: 10.1038/s41561-023-01305-0.

[11] Stefan K. Weber et al.: "Data Acquisition System of the CLOUD Experiment at CERN", IEEE Transactions on Instrumentation and Measurement, 70, pp. 1–13, 2021, DOI: 10.1109/TIM.2020.3023210.

[12] TOFWERK AG. *Products*. 2025.

[13] R. Jones et al. "Monitoring of CERN's Data Interchange Protocol (DIP) System". In: *Proc. of the 16th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*. JACoW Publishing, 2017, pp. 1–4. DOI: 10.18429/JACoW-ICALEPCS2017-THPHA162.

[14] Jerry J. Jongerius. *TimeSyncTool*. 2024.

[15] Yung-Sung Cheng. “Condensation Particle Counters”. In: *Aerosol Measurement*. John Wiley Sons, Ltd, 2011. Chap. 17, pp. 381–392. ISBN: 9781118001684. DOI: https://doi.org/10.1002/9781118001684.ch17.

[16] William C. Hinds. *Aerosol Technology: Properties, Behavior, and Measurement of Airborne Particles*. Wiley-Interscience, 1999.

[17] A. G. Sutugin and N. A. Fuchs: “Coagulation rate of highly dispersed aerosols”, Journal of Colloid Science, 20, pp. 492–500, 1965.

[18] Z. Q. Zhang and B. Y. H. Liu: “Dependence of the performance of TSI 3020 condensation nucleus counter on pressure, flow rate and temperature”, Aerosol Science and Technology, 13, pp. 493–504, 1990.

[19] Bernard Vonnegut. “A continuous recording condensation nuclei meter”. In: *First National Air Pollution Symposium*. Los Angeles, CA, 1949, pp. 36–44.

[20] John Aitken: “On the number of dust particles in the atmosphere”, Proceedings of the Royal Society of Edinburgh, 35, pp. 1–19, 1888.

[21] L. M. Pollak and J. Daly: “An improved model of the condensation nucleus counter with stereophotomicrographic recording”, Geofisica Pura e Applicata, 41, pp. 211–216, 1958.

[22] R. K. Jaenicke and H. J. Kanter: “Direct condensation nuclei counter with automatic photographic recording, and general problems of absolute counters”, Journal of Applied Meteorology, 15, pp. 620–632, 1976.

[23] L. W. Pollak and A. L. Metnieks: “New calibration of photoelectric nucleus counters”, Geofisica Pura e Applicata, 41, pp. 201–210, 1958.

[24] D. Sinclair and G. S. Hoopes: “A continuous flow condensation nucleus counter”, Journal of Aerosol Science, 6, pp. 1–7, 1975.

[25] J. Bricard et al. “Detection of ultra-fine particles by means of a continuous flux condensation nuclei counter”. In: *Fine Particles*. Ed. by B. Y. H. Liu. New York: Academic Press, 1976, pp. 566–580.

[26] J. K. Agarwal and G. J. Sem: “Continuous flow, single-particle-counting condensation nucleus counter”, Journal of Aerosol Science, 11, pp. 343–357, 1980.

[27] M. R. Stolzenburg and P. H. McMurry: “An ultrafine aerosol condensation nucleus counter”, Aerosol Science and Technology, 14, pp. 48–65, 1991.

[28] S. V. Hering and M. R. Stolzenburg: “A method for particle size amplification by water condensation in a laminar thermally diffusive flow”, Aerosol Science and Technology, 39, pp. 428–436, 2005.

[29] S. V. Hering et al.: “A laminar-flow water-based condensation particle counter (WCPC)”, Aerosol Science and Technology, 39, pp. 659–672, 2005.

[30] W. Liu et al.: “Water-based condensation particle counters for environmental monitoring of ultrafine particles”, Journal of the Air Waste Management Association, 56, pp. 444–455, 2006.

[31] K. Okuyama, Y. Kousaka, and T. Motouchi: “Condensational growth of ultrafine aerosol particles in a new particle size magnifier”, Aerosol Science and Technology, 3, pp. 353–366, 1984.

[32] J. Wang et al.: “Fast mixing condensation nucleus counter: application to rapid scanning differential mobility analyzer measurements”, Aerosol Science and Technology, 36, pp. 678–689, 2002.

[33] TSI Incorporated. *NanoScan SMPS Nanoparticle Sizer Theory of Operation*. Application Note SMPS-005. TSI Incorporated, 2013.

[34] A. Medved et al.: "A New Corona-based Charger for Aerosol Particles", Journal of Aerosol Science, 31, S616–S617, 2000.

[35] E. Knutson and K. Whitby: "Aerosol Classification by Electric Mobility: Apparatus, Theory, and Applications", Journal of Aerosol Science, 6, pp. 443–451, 1975.

[36] S. H. Zhang et al.: "Radial Differential Mobility Analyzer", Aerosol Science & Technology, 23, pp. 357–372, 1995.

[37] Katrianne Lehtipalo et al. "Particle Size Magnifier as a tool to detect atmospheric aerosol particles, ions and clusters smaller than 2 nm". English. In: *Proceedings of CRAICC Annual Meeting 2011*. Report series in aerosol science. Unknown host publication ; Conference date: 01-01-1800. 2011, pp. 105–107.

[38] D. Wimmer et al.: "Performance of diethylene glycol-based particle counters in the sub-3 nm size range", Atmospheric Measurement Techniques, 6, pp. 1793–1804, 2013, DOI: 10.5194/amt-6-1793-2013.

[39] Sir William Thomson: "On the equilibrium of vapour at a curved surface of liquid", Philosophical Magazine, 42282, pp. 448–452, 1871.

[40] M. S. B. Munson and F. H. Field: "Chemical Ionization Mass Spectrometry. I. General Introduction.", Journal of the American Chemical Society, 8812, pp. 2621–2630, 1966.

[41] J. H. Gross. *Mass Spectrometry: A Textbook*. Springer-Verlag, 2017.

[42] R. C. Dougherty: "Negative Chemical Ionization Mass Spectrometry.", Analytical Chemistry, 534, 625A–636A, 1981.

[43] D. I. Carroll et al.: "Atmospheric pressure ionization mass spectrometry: Corona discharge ion source for use in a liquid chromatograph-mass spectrometer-computer analytical system.", Analytical Chemistry, 4714, pp. 2369–2373, 1975.

[44] Intergovernmental Panel on Climate Change (IPCC). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by V. Masson-Delmotte et al. Available from https://www.ipcc.ch/report/ar6/wg1/. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2021.

[45] K. S. Carslaw et al.: "Atmospheric aerosols in the Earth system: a review of interactions and feedbacks", Atmospheric Chemistry and Physics, 104, pp. 1701–1737, 2010, DOI: 10.5194/acp-10-1701-2010.

[46] H. Gordon et al.: "New Particle Formation in the Atmosphere: From Molecular Clusters to Global Climate", Atmospheric Chemistry and Physics, 221, pp. 1–52, 2022, DOI: 10.5194/acp-22-1-2022.

[47] J. Merikanto et al.: "Impact of nucleation on global CCN", Atmospheric Chemistry and Physics, 922, pp. 8601–8616, 2009, DOI: 10.5194/acp-9-8601-2009.

[48] B. Brunekreef and S. T. Holgate: "Air pollution and health", The Lancet, 3609341, pp. 1233–1242, 2002, DOI: 10.1016/S0140-6736(02)11274-8.

[49] W. Wang et al.: "Application of smog chambers in atmospheric process studies", Environmental Science and Ecotechnology, 6, p. 100085, 2021, DOI: 10.1016/j.ese.2021.100085.

[50] Lubna Dada et al.: "Formation and growth of sub-3-nm aerosol particles in experimental chambers", Nature Protocols, 153, pp. 1013–1040, 2020, DOI: 10.1038/s41596-019-0274-z.

[51] J. H. Seinfeld and S. N. Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. Wiley, 2012.

[52] R. Wagner et al.: "The role of ions in new particle formation in the CLOUD chamber", Atmospheric Chemistry and Physics, 17, pp. 15181–15197, 2017, DOI: 10.5194/acp-17-15181-2017.

[53] B. E. Poling, J. M. Prausnitz, and J. P. O'Connell. *The Properties of Gases and Liquids*. 5th ed. McGraw-Hill, 2001.

[54] P. H. McMurry and D. J. Rader: "Aerosol wall losses in electrically charged chambers", Aerosol Science and Technology, 4, pp. 249–268, 1985.

[55] Albert Einstein: "On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat", Annalen der Physik, 17, pp. 549–560, 1905.

[56] E. Cunningham: "On the velocity of steady fall of spherical particles through fluid medium", Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 83, p. 357, 1910.

[57] N. A. Fuchs. *The Mechanics of Aerosols*. Oxford: Pergamon Press, 1964.

[58] A. Savitzky and M. J. E. Golay: "Smoothing and Differentiation of Data by Simplified Least Squares Procedures", Analytical Chemistry, 368, pp. 1627–1639, 1964, DOI: 10.1021/ac60214a047.

[59] Pauli Virtanen et al.: "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", Nature Methods, 17, pp. 261–272, 2020, DOI: 10.1038/s41592-019-0686-2.

[60] Walter John. "Size Distribution Characteristics of Aerosols". In: *Aerosol Measurement: Principles, Techniques, and Applications*. John Wiley & Sons, Ltd, 2011. Chap. 4, pp. 41–54. DOI: 10.1002/9781118001684.ch4.

[61] P. G. Gormley and M. Kennedy: "Diffusion from a stream flowing through a cylindrical tube", Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences, 52, pp. 163–169, 1949.

[62] S. K. Friedlander. *Smoke, Dust, and Haze: Fundamentals of Aerosol Behavior*. New York: John Wiley & Sons, 1977.

[63] John E. Brockmann. "Aerosol Transport in Sampling Lines and Inlets". In: *Aerosol Measurement: Principles, Techniques, and Applications*. John Wiley & Sons, Ltd, 2011. Chap. 6, pp. 68–105. DOI: 10.1002/9781118001684.ch6.

[64] D. Stolzenburg et al.: "Enhanced growth rate of atmospheric particles from sulfuric acid", Atmospheric Chemistry and Physics, 20, pp. 7359–7389, 2020, DOI: 10.5194/acp-20-7359-2020.

[65] F. L. Eisele and D. J. Tanner: "Measurement of the gas phase concentration of H2SO4 and methane sulfonic acid and estimates of H2SO4 production and loss in the atmosphere", Journal of Geophysical Research: Atmospheres, 98D5, pp. 9001–9010, 1993, DOI: 10.1029/93JD00031.

[66] Stefanie Kremser et al.: “Stratospheric aerosol—Observations, processes, and impact on climate”, Reviews of Geophysics, 542, pp. 278–335, 2016, DOI: 10.1002/2015RG000511.

Appendix

Formation Rate Script

Below are showcased the *GatherData*, *DilutionSink*, *WallSink*, *CoagulationSink*, and *derivative* functions from the created Python script for the formation rate calculation, aswell as an example usage to calculate formation rate at 2.5 and 1.7 nm.

```
def gatherData(cloud_database, time_start, time_end, temperatures_ids, flows_ids,
               cpc=None, nSMPS=None, psm=None, temperatures=None, pressure=None, flows=None):
    if cpc is None:
        cpc = load_cpc(cloud_database, time_start, time_end)
    if nSMPS is None:
        nSMPS = load_nSMPS(cloud_database, time_start, time_end)
    if psm is None:
        psm = load_psm(cloud_database, time_start, time_end)
    if temperatures is None:
        temperatures = load_temperatures(cloud_database, time_s, time_e, temperatures_ids)
    if pressure is None:
        pressure = load_pressure(cloud_database, time_s, time_e)
    if flows is None:
        flows = load_flows(cloud_database, time_s, time_e, flows_ids)
    df = pd.DataFrame(nSMPS)
    df = pd.merge(df, cpc, on='time', how='inner')
    df = pd.merge(df, psm, on='time', how='inner')
    df = pd.merge(df, temperatures, on='time', how='inner')
    df = pd.merge(df, pressure, on='time', how='inner')
    df = pd.merge(df, flows, on='time', how='inner')
    pattern = r'^\d+_\d+$'  # Regular expression for columns like '2_05', '5_14', etc.
    diameters = [col for col in df.columns if pd.Series(col).str.match(pattern).any()]#List of the column names for the diameter counts '2_05','5_14',etc.

    #Convert from log distribution to particle counts
    i = 0
    for diam in diameters:
        if diam == '2_02':
            df[diam] = df[diam] * (np.log(float(diameters[i + 1].replace('_', '.'))* 10**(-7)) -
                                   np.log(float(diameters[i].replace('_', '.'))* 10**(-7))) #* 10**(-7)
        if diam == '63_8':
            df[diam] = df[diam] * (np.log(float(diameters[i].replace('_', '.'))* 10**(-7)) -
                                   np.log(float(diameters[i - 1].replace('_', '.'))* 10**(-7))) #* 10**(-7)
        if diam not in ['2_02', '63_8']:
            df[diam] = df[diam] * (np.log(float(diameters[i + 1].replace('_', '.'))* 10**(-7)) -
                                   np.log(float(diameters[i - 1].replace('_', '.'))* 10**(-7)))
        i += 1

    df = df.drop(columns = ['2_02','2_09','2_17','2_25','2_33','2_41','2_50','2_59','2_69','2_79','2_89','3_00','3_11','3_22','3_34','3_46','3_59',
                            '3_72','3_85','4_00','4_14','4_29', '4_45','4_61','4_78','4_96'], axis=1)
```

(a) GatherData (Part 1)

```
    df['N>5'] = df[['5_14','5_33','5_52','5_73','5_94','6_15',
              '6_38','6_61','6_85','7_10','7_37','7_64','7_91','8_20','8_51','8_82','9_14','9_47','9_82','10_2','10_6','10_9',
              '11_3','11_8','12_2','12_6','13_1','13_6','14_1','14_6','15_1','15_7','16_3','16_8','17_5','18_1','18_9','19_5',
              '20_2','20_9','21_7','22_5','23_3','24_1','25_0','25_9','26_9','27_9','28_9','30_0','31_1','32_2','33_4','34_6',
              '35_9','37_2','38_5','40_0','41_4','42_9','44_5','46_1','47_8','49_6','51_4','53_3','55_2','57_3','59_4','61_5',
              '63_8']].sum(axis=1, skipna=True)

    #Create 2_50 bin from CPC data
    df['2_50'] = np.where(df['N>2_50'] - df['N>5'] > 0, df['N>2_50'] - df['N>5'], 0)
    columns = list(df.columns)
    columns.remove('2_50')
    columns.insert(columns.index('time') + 1, '2_50')
    df = df[columns]

    #Create 1_70 bin from PSM data
    df['1_70'] = np.where(df['N>1_70'] - df['N>2_50'] > 0, df['N>1_70'] - df['N>2_50'], 0)
    columns = list(df.columns)
    columns.remove('1_70')
    columns.insert(columns.index('time') + 1, '1_70')
    df = df[columns]
    df = df.drop(['N>5'], axis=1)

    #Sample line correction
    Qvol = 5
    Ltube = 1.25
    Qvol = Qvol * 1/1000/60
    kB = 1.38 * 10**(-23) #Boltzmann Constant, J/K
    Lref = 66.4 #Reference Free Mean Path, nm
    S = 110.4 #Suntherland constant, K

    df['temperature_K'] = df['temperature_C'] + 273.15
    df['freeMeanPath_nm'] = Lref * (1013/df['pressure_hPa']) * (df['temperature_K']/293.15) * (1 + S/293.15)/(1 +  S/df['temperature_K'])

    diameters = [col for col in df.columns if pd.Series(col).str.match(pattern).any()]
```

(b) GatherData (Part 2)

Figure A.1: **Functions of Formation Rate Script (1/3).**

```
    for diam in diameters:
        diam_number = float(diam.replace('_', '.')) #get the float from the string
        df['Kn'] = 2 * df['freeMeanPath_nm']/diam_number #nm/nm, Knudsen Number

        a = 1.142
        b = 0.558
        g = 0.999

        Cc = 1 + df['Kn'] * (a + b * np.exp(-g/df['Kn'])) #Slip Correction
        Vref = 1.8325 * 10**(-5)
        Tref = 293.15

        df['viscosity_unit'] = Vref * (Tref + S)/(df['temperature_K'] + S) * (df['temperature_K']/Tref)**(3/2)
        DiamM = diam_number/1e+09 #diameter in meters
        df['B'] = Cc/(3 * math.pi * df['viscosity_unit'] * DiamM) #MECHANICAL MOBILITYY
        df['Ddiff'] = kB * df['temperature_K'] * df['B']
        df['gsi'] = math.pi * df['Ddiff'] * Ltube/Qvol
        df['Sh'] = 3.66 + 0.2672/(df['gsi'] + 0.10079 * df['gsi']**(1/3))
        df['etaeff'] = np.exp(-df['gsi'] * df['Sh'])
        df[diam] =  df[diam]/df['etaeff']

    df = df.drop(columns = ['Kn', 'B', 'Ddiff', 'gsi', 'Sh', 'etaeff'], axis = 1)

    return df
```

(c) GatherData (Part 3)

```
def DilutionSink(input_df, j_df, particle_diameter):
    j_df['k_dil'] = input_df['flow_air']/26.1
    j_df['dilution_sink'] = j_df['k_dil']*input_df['N>' + particle_diameter]
    j_df = j_df.drop(columns = ['k_dil'], axis=1)
    return j_df

def WallSink(input_df, j_df, particle_diameter):
    kB = 1.38 * 10**(-23)

    pattern = r'^\d+_\d+$'  # Regular expression for columns like '2_05', '5_14', etc.
    diameters = [col for col in input_df.columns if pd.Series(col).str.match(pattern).any()]
    diameters_after = [col for col in diameters if col > particle_diameter]

    for diam in diameters_after:
        diam_number = float(diam.replace('_', '.'))
        j_df['Kn_' + diam] = 2 * input_df['freeMeanPath_nm'] / diam_number
        j_df['Cc_' + diam] = 1 + j_df['Kn_' + diam] * (1.142 + 0.558 * np.exp(-0.999/j_df['Kn_' + diam]))
        j_df['D_' + diam] = 10**(4) * (kB * input_df['temperature_K'] * j_df['Cc_' + diam])/(3* 3.14 * input_df['viscosity_unit'] * diam_number * 10**(-9)
        j_df['k_wall_' + diam] = 0.0077 * np.sqrt(j_df['D_' + diam])

    j_df['wall_sink'] = 0
    for diam in diameters_after:
        j_df['wall_sink'] += input_df[diam]*j_df['k_wall_' + diam]
        j_df = j_df.drop(columns = ['Kn_' + diam, 'Cc_' + diam, 'D_' + diam, 'k_wall_' + diam], axis = 1)
    return j_df

```

(d) DilutionSink and WallSink calculation

```
def CoagulationSink(input_df, j_df, particle_diameter):
    kB = 1.38 * 10**(-23)

    pattern = r'^\d+_\d+$'  # Regular expression for columns like '2_05', '5_14', etc.
    diameters = [col for col in input_df.columns if pd.Series(col).str.match(pattern).any()]
    diameters_after = [col for col in diameters if col > particle_diameter]

    for diam in diameters_after:
        diam_number = float(diam.replace('_', '.'))
        j_df['Kn_' + diam] = 2 * input_df['freeMeanPath_nm'] / diam_number
        j_df['Cc_' + diam] = 1 + j_df['Kn_' + diam] * (1.142 + 0.558 * np.exp(-0.999/j_df['Kn_' + diam]))
        j_df['D_' + diam] = 10**(4) * (kB * input_df['temperature_K'] *
                                       j_df['Cc_' + diam])/(3* 3.14 * input_df['viscosity_unit'] * diam_number * 10**(-9))


    for diam in diameters_after:
        diam_number = float(diam.replace('_', '.'))
        diam_number = diam_number * 10**(-9)

        mass = (4/3) * math.pi * ((diam_number - (0.3 * 10**(-9))) / 2)**3 * 1000

        j_df['c_' + diam] = ((8 * kB * input_df['temperature_K'])/(math.pi*mass)) ** 0.5
        j_df['l_'+ diam] = (8 * j_df['D_' + diam] * 10**(-4))/(math.pi * j_df['c_' + diam])
        j_df['g_' + diam] = (1/(3 * diam_number * j_df['l_'+ diam]))*
        ((diam_number + j_df['l_'+ diam])**3 - (diam_number**2 + j_df['l_'+ diam]**2)**3/2) - diam_number

        for diam2 in diameters_after:
            diam2_number = float(diam2.replace('_', '.'))
            diam2_number = diam2_number * 10**(-9)

            mass = (4/3) * math.pi * ((diam2_number - (0.3 * 10**(-9))) /2)**3 * 1000
            j_df['c_' + diam2] = ((8 * kB * input_df['temperature_K'])/(math.pi*mass)) ** 0.5
            j_df['l_'+ diam2] = (8 * j_df['D_' + diam2]  * 10**(-4))/(math.pi * j_df['c_' + diam2])
            j_df['g_' + diam2] = (1/(3 * diam2_number * j_df['l_'+ diam2]))*
            ((diam2_number + j_df['l_'+ diam2])**3 - (diam2_number**2 + j_df['l_'+ diam2]**2)**3/2) - diam2_number
```

(e) CoagulationSink calculation (Part 1)

Figure A.1: **Functions of the Python Script (2/3).**

```
            if 'K_'+diam2+','+diam not in j_df.columns:
                j_df['K_'+diam+','+diam2] = math.pi*(diam_number*j_df['D_' + diam] * 10**(-4)
                                                     +diam2_number*j_df['D_'+diam2] * 10**(-4)
                                                     +diam_number*j_df['D_'+diam2] * 10**(-4)
                                                     +diam2_number*j_df['D_'+diam] * 10**(-4)) *
                (((diam_number + diam2_number)/(diam_number + diam2_number + 2*(j_df['g_' + diam]**2 + j_df['g_' + diam2]**2)**0.5)) +
                 ((8*(j_df['D_' + diam]  * 10**(-4) + j_df['D_' + diam2]  * 10**(-4)))/(((j_df['c_' + diam]**2 + j_df['c_' + diam2]**2)**0.5)*
                                                                                        (diam_number + diam2_number))))**(-1)


    j_df['coagulation_sink'] = 0


    for diam in diameters_after:
        for diam2 in diameters_after:
            if 'K_'+diam+','+diam2  in j_df.columns:
                j_df['coagulation_sink'] += input_df[diam]*input_df[diam2]*j_df['K_'+diam+','+diam2]
                j_df = j_df.drop(columns = ['K_'+diam+','+diam2], axis = 1)

    for diam in diameters_after:
        j_df = j_df.drop(columns = ['Kn_' + diam, 'Cc_' + diam, 'D_' + diam, 'c_' + diam, 'l_'+ diam, 'g_' + diam], axis = 1)

    j_df['coagulation_sink'] = j_df['coagulation_sink'] * 10**6

    return j_df

def derivative(input_df, j_df, particle_diameter):
    j_df['dN>' + particle_diameter +'/dt'] = input_df['dN>' + particle_diameter + '/dt']
    return j_df
```

(f) CoagulationSink calculation (Part 2) and Derivative calculation

```
pressure = load_pressure('cloud16', time_s, time_e)

temperatures = load_temperatures('cloud16', time_s, time_e,  [1,2,4])

cpc = load_cpc('cloud16', time_s, time_e)

psm = load_psm('cloud16', time_s, time_e)

flows = load_flows('cloud16', time_s, time_e, [1462, 54, 1637, 62, 1624])

nSMPS = load_nSMPS('cloud16', time_s, time_e)

pd.options.display.float_format = '{:.2e}'.format

df = gatherData('cloud16', time_s, time_e, [1,2,4], [1462, 54, 1637, 62, 1624],
                cpc = cpc, nSMPS = nSMPS, psm = psm, temperatures=temperatures, pressure=pressure, flows=flows)
#print(df)


j_2p5 = pd.DataFrame(df[['time']])
j_2p5 = DilutionSink(df, j_2p5, '2_50')
j_2p5 =  WallSink(df, j_2p5, '2_50')
j_2p5 = CoagulationSink(df, j_2p5, '2_50')
j_2p5 = derivative(df, j_2p5, '2_50')
j_2p5['j_2p5'] = j_2p5[['dN>2_50/dt','dilution_sink', 'wall_sink', 'coagulation_sink']].sum(axis=1, skipna=True)



#print(j_2p5)
j_2p5.to_csv('j_2p5.csv', index=False, float_format='%.2e')

j_1p7 = pd.DataFrame(df[['time']])
j_1p7 = DilutionSink(df, j_1p7, '1_70')
j_1p7 =  WallSink(df, j_1p7, '1_70')
j_1p7 = CoagulationSink(df, j_1p7, '1_70')
j_1p7 = derivative(df, j_1p7, '1_70')
j_1p7['j_1p7'] = j_1p7[['dN>1_70/dt','dilution_sink', 'wall_sink', 'coagulation_sink']].sum(axis=1, skipna=True)
#print(j_1p7)
j_1p7.to_csv('j_1p7.csv', index=False, float_format='%.2e')
```

(g) Main execution logic.

Figure A.1: **Structural Overview and Main Execution Logic of the Functions of Formation Rate Script (3/3)** This multi-part figure details the computational implementation of the aerosol population balance equation. (a–c) The `GatherData` routine handles the ingestion and temporal synchronization of raw data from multiple instruments, including the CPC, PSM, and nSMPS. (d–f) The specific sink functions—`DilutionSink`, `WallSink`, and `CoagulationSink`—quantify the particle loss rates due to chamber ventilation, wall deposition, and inter-particle collisions, respectively. (g) The `Main Execution Logic` illustrates the final workflow: it orchestrates the calculation by sequentially calling the aforementioned functions and summing the results with the time-derivative of the particle concentration. This automated pipeline ensures that the final NPF rate (J) is isolated from background losses, providing a physically accurate representation of new particle production within the chamber.