

Aerosol Particle Measurements: Strategies for Health-Relevant Data Collection and Analysis

Thesis by
Amanda Grantz Pansing

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended May 23, 2019

© 2019

Amanda Grantz Pansing

All rights reserved

ACKNOWLEDGMENTS

My graduate school experience was not a conventional one, if there is such a thing. I definitely took the scenic route, and I probably did not ask for enough help along the way, but somehow I managed to reach my destination. For this I know I owe special thanks to the following particularly supportive people.

My advisor, Rick Flagan, gets top billing in my thesis thank-you's for bearing with me and gently guiding me from start to finish, through all the ups and downs and starts and stops of creating this thesis document. It was quite the ride, but I think we pulled it together nicely in the end.

I would also like to thank John Seinfeld, Paul Wennberg, and Mark Davis for rounding out my thesis committee and lending their minds to pass final inspection on this work.

My unconventional ride through grad school would not have been possible in the official sense without the advocacy and accommodations of Felicia Hunt and Natalie Gilmore in the graduate studies office. The ride might have been possible, but not nearly as enjoyable, without a certain hodgepodge group of fellow students and friends who would identify with the ChEESE(G)PSer designation. We shared many fun lunches, birthday celebrations, softball games, and other shenanigans.

Finally, I owe very special thank-you's to my family. To my husband Craig, I don't know if I could ever thank you enough for all the love, support, and encouragement you show me every day, but I will always do my best to return the same to you. To my parents, of both Grantz and Pansing varieties, I also feel your love and support, and I thank you for always rooting for me.

ABSTRACT

Particulate matter (PM) is an important component of outdoor and indoor air pollution that can cause significant harm to human health. The present work, organized into two parts, introduces strategies for optimizing the collection and analysis of airborne particle measurements to inform PM health-effect research.

Part I focuses on the fundamental aerosol data analysis task of interpreting indirect measurements of particle size to reveal the distribution of sizes of particles in a sampled aerosol. An approach to this aerosol data inversion problem is developed that shows improved particle size distribution recovery compared to other common approaches described in the literature. This inverse solution method incorporates cubic spline interpolation to represent the particle size distribution within a discrete linear model of the inverse problem while placing no constraints on the number or spacing of solution points. The inverse problem setup can then interface with three established numerical methods for solution computation. The accuracy of this procedure is demonstrated through analysis of test-case data for differential mobility analyzer systems. Source code and supporting documentation are also provided to encourage researchers to use and adapt this inversion algorithm for analyzing data collected from existing as well as potential future measurement systems.

Part II of this work focuses on the retrieval of health-relevant information from aerosol particle measurement data. The inversion analysis introduced in Part I is incorporated into an extended analysis procedure for evaluating the metrics of PM exposure and respiratory dose that can be obtained from different measurement systems. Applying this evaluation procedure to a range of existing and potential future measurement techniques reveals that full characterization of particle size distributions need not be time and resource intensive and should be pursued for the great benefits this information would provide to health studies. Not only can size distribution information permit lung tissue dose estimates through a set of relatively simple calculations, but a single set of size distribution data can be analyzed and reanalyzed to provide dose estimates for human populations of interest by applying the appropriate respiratory tract deposition profiles. The measurement evaluation procedure developed here reveals target criteria for the particle characterization necessary to provide sufficient exposure and dose information for health studies. The intent is not to eliminate the current measurements and standards, but to help direct future developments in health-related aerosol particle measurement design.

TABLE OF CONTENTS

Acknowledgments	iii
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	xiv
Chapter I: Introduction	2
1.1 Data Inversion for Aerosol Spectrometers	3
1.2 Particle Measurements and Health Metrics	4
I Data Inversion for Aerosol Spectrometers	
Chapter II: Inversion of Aerosol Size Distribution Data using Cubic Spline Quadrature	8
2.1 Introduction	8
2.2 Overview of Inverse Problems	10
2.3 Review of Inversion Methods for Aerosol Spectrometers	14
2.4 Present Approach	22
2.5 Methods	29
2.6 Results and Discussion	33
2.7 Conclusions	39
Chapter III: Analysis Software for Differential Electrical Mobility Classifiers .	43
3.1 Introduction	43
3.2 Inversion Code	45
3.3 Discussion	68
3.4 Conclusions	72
II Particle Measurements and Health Metrics	
Chapter IV: Particle Measurements Relevant to Aerosol Health Studies I: Evaluation Methodology	75
4.1 Introduction	75
4.2 Methods	80
4.3 Results and Discussion	93
4.4 Conclusions	100
4.5 Additional Measurement Evaluation Results	101
Chapter V: Particle Measurements Relevant to Aerosol Health Studies II: Current Abilities and Future Requirements	105
5.1 Introduction	105
5.2 Classification of Measurement Systems	108
5.3 Evaluation of Measurement Systems	126

5.4 Results and Discussion	127
5.5 Conclusions	134
5.6 Additional Measurement Evaluation Results	139
Chapter VI: Conclusion	146
6.1 Summary	146
6.2 Future Work	147
Bibliography	149
Appendix A: Measurement Simulations	156
Appendix B: Lung Deposition Model	169
Appendix C: Spline Interpolation	170
Appendix D: Kernel Matrix Setup	180
Appendix E: Inversion Code	194

LIST OF FIGURES

<i>Number</i>	<i>Page</i>
2.1 Histogram, linear spline, and cubic spline representations of a bimodal test-case aerosol previously presented in Russell et al. (1995) and Collins et al. (2002). Each interpolation method was applied to 20 known distribution function values indicated as the 'Interpolation Nodes.' Also shown are values of percent relative error quantified as $e_j = \left(f_j^{\text{interp}} - f_j^{\text{true}} \right) / f_j^{\text{true}} \times 100$ for individual interpolation points over the entire particle size range.	21
2.2 Relative error in reconstruction, e_R , shown as a function of the number of knots used for the interpolation of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002).	35
2.3 Inversion results obtained using the simple, histogram, linear spline, and cubic spline quadrature methods to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002). All inversion analyses were performed on noise-free measurements simulating moderate resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 5$).	37
2.4 Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the DMA operating resolution.	38
2.5 Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of n_{sub} , the number of secondary integration points used per interval of integration in the kernel matrix setup. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	38
2.6 Recovery of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$) (top); and relative error in reconstruction, e_R , for inversion analyses of the same test-case particle size distribution, shown as a function of the DMA operating resolution (bottom).	40
2.7 Recovery of a pathological multimodal test-case aerosol presented in Talukdar and Swihart (2003) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	41

2.8	Recovery of an atmospheric test-case aerosol size distribution from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	41
2.9	Recovery of unimodal test-case aerosol size distributions with $d_{pg} = 0.1\mu\text{m}$ and varying σ_g : $\sigma_g = 1.1$ (top), $\sigma_g = 1.5$ (middle), and $\sigma_g = 2.0$ (bottom). All inversion analyses were performed on noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	42
3.1	Raw count data, shown as particle number versus voltage setting, to be analyzed to recover the particle number size distribution.	53
3.2	Particle number size distribution recovered from inversion analysis.	53
3.3	Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the number of charges accounted for in the multiple charge correction of the kernel function. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	70
3.4	Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the ratio of the equivalent length value used for signal simulation to that used in calculating the inversion kernel matrix. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	70
3.5	Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the theta value, θ , used to define measurement channels. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	72
3.6	Convergence behavior for the NNLS, TNNLS, and Regularization optimization programs to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	73
3.7	Convergence behavior for the NNLS, TNNLS, and Regularization optimization programs to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).	73

4.1	Source-to-Health Effects paradigm for investigating the health effects associated with airborne particulate matter. Research activities can be broadly categorized as exposure assessment (color-coded here in blue) or effects assessment (color-coded here in red). The overlap in these focus areas (color-coded here in purple) reflects the importance of exposure and dose information to both sets of activities. Adapted from Liroy (1990), Sexton et al. (1992), and the National Research Council (2004, p. 26).	77
4.2	Procedural flow illustration for evaluating measurement performance. Define an input particle distribution, simulate a given measurement scheme, and process the recovered particle characterization to estimate dose to the respiratory tract. From the completely defined input, an ideal estimate of dose can also be calculated. By comparing the information provided by simulated measurements to the ideal estimates of dose, this evaluation procedure can reveal limits or biases in measurement data that may constrain the scope and impact of PM health-effect studies.	81
4.3	Anatomical regions of the human respiratory tract, shown schematically along with deposition efficiency curves describing the fraction of inhaled particles of a given diameter that deposit in the different lung regions (ICRP 1994).	83
4.4	Particle number size distributions simulated using the Ion-UHMA model (Leppä et al. 2009) to reproduce an atmospheric nucleation burst typical of many that occurred in the highly polluted atmosphere of New Delhi during October and November 2002 (Mönkkönen et al. 2005).	86
4.5	Map of the mass median diameter and geometric standard deviation parameter space for the lognormal size distributions simulated to represent occupational aerosol populations. Ovals represent particle size ranges reported in the literature for several occupational aerosol types.	87
4.6	Particle size selective criterion for a PM _{2.5} sampler, shown with two example particle mass size distributions. Essentially the entire mass of the size distribution from the nucleation simulation will be captured, while some of the lognormal distribution will be lost. . . .	89
4.7	Normalized response functions for a stepping-mode DMA system, simulated to approximate typical high resolution operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$). The model system is set to operate with 36 voltage steps from 10 V to 10 kV, equally spaced on a logarithmic scale. . . .	92
4.8	Correlation of PM _{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions. . . .	96

4.9	Correlation of PM _{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the map of lognormal distributions.	97
4.10	Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	98
4.11	Percent bias in dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) compared to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the map of lognormal distributions.	99
4.12	Correlation of PM _{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	101
4.13	Correlation of PM _{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the map of lognormal distributions.	102
4.14	Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	103
4.15	Percent bias in dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) compared to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the map of lognormal distributions.	104
5.1	Procedural flow illustration for the quantitative evaluation of health-relevant information provided by any given measurement strategy. Simulated experimental information is evaluated by drawing comparisons to estimates of respiratory dose that would actually occur, given knowledge of the ambient aerosol that is not biased by limitations of the measurement methods. This comparison to a “true dose” creates the opportunity for quantitative assessments. Relying upon synthetic data and ideal models of the measurement methods under consideration enables control over any uncertainties associated with the different methods and ensures fair assessments and inter-comparisons of measurement strategies with potentially very different operating principles.	109

5.2	Summary of currently accepted particle size-selective criteria for aerosol sampling of inhalable, thoracic, respirable, and PM _{2.5} fractions (Vincent 2005). A collection efficiency curve is also shown for a theoretical 100 nm cut-point.	114
5.3	Counting efficiency curve for the TSI Model 3025A UCPC, determined from exponential fit of experimental data presented by Wiedensohler et al. (1997).	115
5.4	Model deposition fractions representing the information captured by the TSI NSAM sampler as a function of particle diameter for the tracheobronchial and alveolar regions of the human respiratory tract. Deposition fractions were determined using the ICRP (1994) model for a reference worker.	117
5.5	Model deposition fractions for the RDS sampler developed by Koehler and Volckens (2013), shown as a function of particle diameter for each of three regions in the human respiratory tract, as well as total deposition. Deposition fractions were determined using Hinds' (1999) parameterization for the average of adult males and females under conditions of sitting, light exercise, and heavy exercise.	118
5.6	Transfer functions for a 15-stage ELPI, with cut-points ranging from 6 nm to 10 μ m. The individual peaks cover the range in particle diameter sizes collected on a given stage.	121
5.7	Transfer functions for a simulated DMPS system operating with 36 measurement channels. Peaks indicate the range of aerosol sizes that contribute to the measured responses for a given channel. The distinct waves of peaks demonstrate the effect of multiple charging on the response function. Multiple charging is not significant at the smaller sizes, but does become important at the larger sizes.	122
5.8	Correlation of PM _{2.5} , CPC, and (theoretical) total surface area measurements to values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	128
5.9	Correlation of (theoretical) PM _{100 nm} , PN _{100 nm} , and PSA _{100 nm} measurements to values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	130
5.10	Correlation of dose values collected with regional deposition samplers (RDS, ideal sampling efficiency and detection of specific particle metrics) to values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	132
5.11	Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as "true" mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	133

5.12	Percent bias in dose values estimated from DMA measurements collected under a range of operating conditions compared to those calculated as “true” hourly deposited number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	136
5.13	Percent bias in dose values estimated from DMA measurements collected under a range of operating conditions compared to those calculated as “true” 24-hour deposited number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	137
5.14	Correlation of dose values estimated from low resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 3$) to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.	138
5.15	Correlation of PN_{total} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	140
5.16	Correlation of PSA_{total} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	140
5.17	Correlation of $PN_{100\text{ nm}}$ measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	141
5.18	Correlation of $PSA_{100\text{ nm}}$ measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	141
5.19	Correlation of mass dose collected with the regional deposition sampler (RDS) to values calculated as “true dose” for the nucleation simulation time-series of size distributions.	142
5.20	Correlation of number dose collected with a theoretical RDS to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	142
5.21	Correlation of surface area dose collected with a theoretical RDS to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	143

5.22	Correlation of dose values estimated from low resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 3$) to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	144
A.1	Non-diffusing transfer function, Ω_{nd} , showing the probability of transiting the DMA for a particle with electrical mobility $Z_p = \tilde{Z}_p \cdot Z_p^*$ where Z_p^* is the centroid of the transfer function.	158
A.2	Typical shapes of the functions contributing to the definition of the integration kernel Ψ_i for measurement channel i of a DMA.	160
A.3	Transfer functions, Ω , as a function of dimensionless particle mobility, \tilde{Z}_p , for a flow ratio of $\beta = \frac{1}{10}$ ($\delta = 0$) at DMA centroid diameter $D_p^* = 10$ nm.	163
A.4	Counting efficiency curve for the TSI Model 3025A UCPC, determined from exponential fit of experimental data presented by Wiedensohler et al. (1997).	166
C.1	Basis functions (B-splines) $B_{j,d}$ of degree $d = 0$ (top), $d = 1$ (middle), $d = 3$ (bottom).	173

LIST OF TABLES

<i>Number</i>	<i>Page</i>
3.1 Example of the measured data provided as input to the inversion analysis routine in the “datamatrix” wave.	48
3.2 Input parameters to describe DMA system setup.	50
3.3 Parameter values to be referenced in inversion calculations.	50
3.4 Input parameters to define reponse matrix calculations.	50
4.1 Correlation coefficients (Pearson’s r) for $PM_{2.5}$ measurements with the values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	95
4.2 Correlation coefficients (Pearson’s r) for estimates of dose from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) with the values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	100
5.1 Performance characteristics for various approaches to measuring particle exposure and regional lung dose.	111
5.2 Correlation coefficients (Pearson’s r) for $PM_{2.5}$ and other integral measures of ambient exposure with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	129
5.3 Correlation coefficients (Pearson’s r) for theoretical integral measures of ambient exposure with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	131
5.4 Correlation coefficients (Pearson’s r) for dose measures from regional deposition samplers (RDS, ideal sampling efficiency and detection of specific particle metrics) with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	134

5.5	Correlation coefficients (Pearson's r) for estimates of dose from DMA measurements with the values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.	135
A.1	Steady State distribution for "air" ions and conductive particles at 101325 Pa and 298.15 K. Fit coefficients are shown for charge states of negatively charged particles with $\nu = 1$ to 6 charges.	165
B.1	Parameters defining the reference worker used for lung deposition calculations.	169

Chapter 1

INTRODUCTION

Particulate matter (PM) consists of microscopic pieces of condensed phase matter that can be emitted by a source or formed from vapor-phase precursors in the atmosphere. As a complex, air-suspended mixture of particles with varying physical and chemical properties, PM can enter the human airways during breathing and may deposit within the respiratory tract to elicit varying types and degrees of health effects. Epidemiological evidence from around the world links human exposure to PM with diseases and causes of death ranging from asthma and lung cancer to heart attack and stroke, even diabetes, neurological disorders, and adverse birth and pregnancy outcomes.

Controlled exposure studies and in vitro mechanistic work suggest that observed PM health impacts stem from the body's biological response to the dose of depositing particles delivered to different anatomical regions of the respiratory tract, from the head and neck airways through to the deepest regions of the lungs. Ideally then, efforts to understand and mitigate PM health impacts would have access to measures of both external PM exposure and internal, biologically effective, regional particle dose for the human population or subpopulations of interest.

Actually obtaining such a comprehensive data set, however, seems to be more of the exception than the rule in PM health-effect research today. Not all aerosol particle measurement schemes are optimized to provide sufficient exposure and dose information to facilitate rigorous assessments of PM health effects. Moreover, researchers may fail to recognize when the available measurement data is incomplete and how its limitations or biases will constrain the scope and impact of PM health-effect studies. The present work discusses these challenges in depth and introduces strategies for optimizing the collection and analysis of aerosol particle measurements for health effects research.

These discussions and strategic contributions are presented in two parts. Part I focuses on the fundamental aerosol data analysis task of interpreting indirect measurements of particle size to reveal the distribution of sizes of particles in a sampled aerosol. An approach to this aerosol data inversion problem is developed that shows improved particle size distribution recovery compared to other common approaches

described in the literature. This inversion procedure provides a contribution to aerosol research in general, but it also serves to optimize the exploration of lung tissue dose that becomes possible with access to aerosol particle size measurements. This retrieval of health-relevant information from aerosol particle measurement data is discussed in greater depth in Part II of this work. Quantitative analyses reveal criteria for the particle characterization necessary to provide sufficient exposure and dose information for health studies.

1.1 Data Inversion for Aerosol Spectrometers

Inverse problems arise across a broad range of disciplines, from the most abstract and pure mathematics to almost all fields of physical science and practical engineering. In a general, qualitative sense, any attempt to use a set of observations to discern the causal factors that produced them can be classified as an inverse problem. In a more rigorous, mathematical sense, an inverse problem is a mathematical framework used to obtain information about a physical object or system from observed measurements.

Since the scientists and mathematicians that study these problems can have diverse backgrounds and goals, it can seem as though each discipline has developed its own unique approach and underlying theory for finding inverse solutions. In reality, though, inverse problems encountered across disciplines possess strong and fundamental similarities. A coherent big picture understanding of inverse problems is presented in Chapter 2, meant as an accessible discussion for engineers and scientists looking for a clear description of what inverse problems are, what makes them difficult, and what specialized tools are used to solve them.

The focus of Chapter 2 then narrows to the interpretation of indirect sensing measurements in atmospheric science, specifically the inversion of measurements obtained from particle sizing instruments to characterize the distribution of sizes of particles in a sampled aerosol. An important component of any given approach to this aerosol spectrometer inversion problem is the mathematical representation of the particle size distribution for numerical evaluations. Much of what determines both the complexity and the effectiveness of a given approach lies in these mathematical details.

Size distribution representations commonly used in inversion methods include multimodal lognormal representations, nodal representations in which the size distribution is represented as a collection of delta functions at different sizes, and histogram or linear spline representations. The inverse solution method developed here incor-

porates cubic spline interpolation to represent the particle size distribution within a discrete linear model of the inverse problem while placing no constraints on the number or spacing of solution points to recover. The inverse problem setup can then interface with three established numerical methods for solution computation. The accuracy of this procedure is demonstrated through analysis of test-case data for differential mobility analyzer (DMA) systems, and the solutions obtained with the cubic spline technique are shown to be superior to those obtained with other common approaches to this inversion problem.

Supporting documentation for this aerosol inversion routine is provided in Chapter 3 to encourage its use and adaptation for analyzing data collected from existing as well as potential future measurement systems. Source code is included in Appendix E. Data analysis for aerosol size distribution measurements will continue to benefit from advancements in mathematical techniques only as long as these advancements are documented coherently enough to enable practical implementation. When an inversion algorithm comes across as poorly documented or incomprehensibly complex, it is likely to be treated as a black box tool. This is not ideal because the algorithm is then at risk for being inappropriately applied, if it is used at all. To avoid having the inversion routine developed here turn into a black box tool, Chapters 2 and 3 balance practical performance demonstrations with accessible descriptions of the functional details of the solution method.

1.2 Particle Measurements and Health Metrics

Determining the most appropriate approach to measuring exposure to airborne PM is a daunting task. Typical data collection rarely possesses the resolution to efficiently explore the entire array of PM characteristics are possibly significant to the human health response. Measurements that do possess this level of resolution are generally too expensive and too operationally complex to encourage widespread use. The result is that, too often, the metric most relevant to a given case remains uncertain.

A procedure for evaluating the metrics of PM exposure and respiratory dose that can be obtained from present measurement techniques as well as potential future ones is introduced in Chapter 4. This procedure incorporates the aerosol inversion routine from Part I for analyzing any measurement techniques that provide particle size information. The overall measurement evaluation methodology provides a straightforward yet powerful tool for recognizing limits or biases in measurement data that will constrain the scope and impact of PM health-effect studies.

This measurement evaluation procedure is demonstrated in Chapter 4 on two well-known measurement systems: the regulatory metric of $PM_{2.5}$ and the more involved measurements provided by DMA systems. The operational theories behind these two measurement systems are well-established, and they happen to represent the extreme ends of the spectrum in terms of operational complexity and data resolution. The intent is not to dwell on the strengths and weaknesses of these two measurement systems, but simply to use them as examples to demonstrate the important role that particle measurements play in assessing health effects. Other measurement options are available, with more middle-ground capabilities.

A more complete, representative sample of measurement systems is investigated in Chapter 5 to evaluate the range in utility of present measurement techniques as well as potential future ones. This analysis reveals what information existing technology can provide and what information future technology should target. Overviews of PM measurements have been done before, but not in a quantitative manner and not by addressing various metrics of human health effects.

The *in silico* analyses presented here illustrate how measurement systems that use a single metric to characterize an aerosol, be it mass, surface area, or number, may provide data that correlate well with deposited dose of the targeted physical metric, but the data will fail to reflect other measures of dose. This will similarly limit the perceived links between particle measurements and health impacts. Instead, researchers should seek measurements that characterize an aerosol with enough size-resolved detail to estimate various physical metrics of exposure and dose with appropriate assumptions about particle shape and density. Measurement systems based on DMA technology have great potential for this approach.

While state-of-the-art DMA systems, typical of present-day, high-resolution operation, can readily capture the different exposure or dose measures, such instruments are generally too expensive and too operationally complex to encourage widespread use. Fortunately, the present simulations reveal that much simpler, lower resolution mobility analyzers can provide data with sufficient resolution to serve the needs of PM health-effect researchers. Relaxing the requirements on measurement system performance would enable design of simpler systems still capable of recovering sufficient size information to infer lung dose by number, surface area, or mass metrics.

Air pollution regulation and mitigation efforts need access to PM measurements more informative than a single metric concentration, yet simple enough to be re-

covered from instruments that can populate a dense, even personal, monitoring network. The strategies for data collection and analysis presented here promote this perspective of striking the right compromise between operational complexity and information resolution. This approach will lead to future developments in health-related aerosol particle measurement design that emphasize exploring various physical metrics of particle dose delivered to the human lungs.

Part I

**Data Inversion for Aerosol
Spectrometers**

*Chapter 2***INVERSION OF AEROSOL SIZE DISTRIBUTION DATA USING
CUBIC SPLINE QUADRATURE**

Inverse problems are widely encountered in the field of atmospheric remote sensing, as well as many other spheres of interest. We take a broad view of existing inversion techniques and applications to revisit the problem of recovering aerosol particle size distribution information. A common feature for particle size distribution measurement systems is that the relation between collected observations and the size distribution function can be expressed as a Fredholm integral equation of the first kind. Previous inversions have represented the particle size distribution within this response equation as (i) delta-function nodes, (ii) histogram distributions, or (iii) linear splines. We describe an approach to the aerosol inversion problem that incorporates cubic spline interpolation to represent the particle size distribution within a discrete linear model of the inverse problem while placing no constraints on the number or spacing of solution points to recover. We then interface this model with three established numerical methods for solution computation. We demonstrate the accuracy of this procedure through analysis of test-case data for differential mobility analyzer (DMA) systems, and show that the solutions obtained with the cubic spline representation are superior to those obtained by other common approaches to this inversion problem.

2.1 Introduction

Inverse problems are everywhere. They are at the heart of scientific inquiry and technological development in many fields. Any time we hope to use a set of observations or measurements to calculate information about the physical object or system that produced them, we are attempting to solve an inverse problem. Inverse solutions tell us about parameters or properties that we cannot quantify directly. This makes inverse problems some of the most important mathematical problems encountered in many areas of science and technology.

Early publications on inverse problems related to physics, geophysics, astronomy, and other areas of science date back to the first half of the 20th century. Since then, the topic has been revisited repeatedly in the literature. New approaches and concepts are constantly emerging for how to solve inverse problems in particular

fields of application.

In the present study, we draw inspiration from the work presented in diverse fields to revisit the inversion problem as it relates to interpreting indirect sensing measurements in atmospheric science. We focus specifically on the inversion of measurements obtained from particle sizing instruments to characterize the distribution of sizes of particles in a sampled aerosol. This is just one example of the many inverse problems encountered in atmospheric remote sensing, but it is a key element with respect to understanding and managing aerosol effects on health, visibility, and climate. Thus, determining the particle size distribution function is a basic task in aerosol research. One goal of this study is to improve the recovery of the particle size distribution from aerosol measurements via data inversion; our approach is to take a broad view of inverse problems and incorporate mathematical and analytical techniques from other fields of practical application.

Another goal of this study is to provide a clear description and thorough documentation for our inversion approach. Data analysis for aerosol size distribution measurements will continue to benefit from advancements in mathematical techniques, but only as long as these advancements are documented coherently enough to enable practical implementation. It is an unfortunate observation that many of the inversion routines presented in the literature seem to have not been extensively implemented in practice. In some cases, these algorithms have been perceived as too mathematically complex to allow straightforward interpretation of the results or adaptation of the code to specific applications. Another unfortunate observation is that many of the inversion algorithms that are actually used much for practical data analysis seem to be documented only poorly. Surprisingly few publications are available that specify their algorithm in detail, particularly regarding the implementation of numerical approximations.

When an inversion algorithm comes across as poorly documented or incomprehensibly complex, it is likely to be treated as a black box tool. This is not ideal because the algorithm is then at risk for being inappropriately applied, if it is used at all. Here, we discuss the inversion problem in a manner that aims to shed some light on this black box nature of inverse solution methods. We describe the development and implementation of a new inversion algorithm while balancing demonstration of its practical performance with clarity in its functional details. In doing so, we fully acknowledge the fundamental mathematical basis of inverse problems, however, our discussion is not designed to satisfy mathematicians. This presentation is meant to

be more accessible for engineers and scientists looking for a transparent discussion of application-oriented inverse problems and their solution methods.

We begin with a discussion of the general characteristics of inverse problems, followed by a review of the methods typically employed to tackle the inversion of particle size distribution measurements. Then we describe our own inversion algorithm, which incorporates cubic spline quadrature with no assumptions about knot spacing and interfaces with three established numerical methods for solution. Finally, we demonstrate the performance of our method and make quantitative comparisons to other methods currently in use.

2.2 Overview of Inverse Problems

The core of any inverse problem is the mathematical model of the corresponding forward problem. Informally, forward problems consist of finding an effect from a cause, and inverse problems consist of recovering a cause when given the effect. Most often, inverse problems arise from the need to extract information about a desired unknown function from indirect and incomplete physical measurements taken from an instrument.

We construct a general mathematical model of indirect linear measurements as follows. We are interested in a continuous function f but cannot measure its values directly. However, f is connected to another physical quantity \mathbf{g} , which is available for measurement. If we adequately understand the governing physics of our measurement system, we can specify a linear operator \mathcal{A} to model the connection between f and \mathbf{g} such that

$$\mathbf{g} = \mathcal{A}f + \mathbf{e} \quad (2.1)$$

where f is the desired unknown function and $\mathbf{g} \in \mathbb{R}^m$ is a vector of m numbers given by a measurement device. The vector $\mathbf{e} \in \mathbb{R}^m$ models errors coming from measurement noise, which is inevitable in practical situations. The error can be modeled as a random variable with certain statistics, or it can be considered as a deterministic but unknown error found by calibration of the measurement device.

The linear operator \mathcal{A} can take on many forms. In the most general context, \mathcal{A} describes the *forward map*, or the transformation of f into \mathbf{g} , and must be established using parameters and physical laws that characterize the measurement system under study.

We will call Eq. (2.1) the *continuum* model. The notation of this expression suggests

the function f may contain functions of a continuous variable, such as time or space, and so may have infinitely many degrees of freedom. This describes many inverse problems that we encounter which attempt to map a finite number of data onto an infinite dimensional model.

For practical extraction of information from indirect linear measurements using computational models, we need to introduce a finite dimensional approximation $\mathbf{f} \in \mathbb{R}^n$ to the function f and to build a matrix model \mathbf{A} for the linear operator \mathcal{A} . This constructs a discrete linear inverse problem expressed as

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \mathbf{e} \quad (2.2)$$

where $\mathbf{g} \in \mathbb{R}^m$ and $\mathbf{f} \in \mathbb{R}^n$. Moreover, \mathbf{A} is a matrix of size $m \times n$ (m rows and n columns). Discretizing a continuous inverse problem is most appropriate when the continuous functions under consideration are smooth enough compared to the sampling length, or if the functions can conveniently be described by their development on a truncated basis.

The advantage of a discretized point of view for problems involving functions of a continuous variable is that the mathematics is easier. As suggested by Eq. (2.2), the calculations for a discrete inverse problem describing a linear system rely on the theory of vectors and matrices rather than on the somewhat more complicated theory of continuous functions and operators.

The disadvantage of approximating continuous inverse problems by their discretized versions is that there is a certain arbitrariness in the discretization process. Making a finite dimensional approximation to a continuous function is always both approximate and, to some degree, arbitrary. In discretizing the continuum model of Eq. (2.1) to give the discrete model of Eq. (2.2), we must choose the dimension n of the discrete vector approximation \mathbf{f} , and we must also establish the mathematical methods used to build the matrix model \mathbf{A} . Different approaches will correspond to different model vectors and different matrices relating the data \mathbf{g} to the model \mathbf{f} . The current literature does not sufficiently demonstrate how these differences propagate through to the final inverse solution. This is a major motivator for our study. Here we quantitatively compare the solutions calculated for continuous linear inverse problems when using different common discretization techniques.

Once the computational model of Eq. (2.2) is constructed, it is tempting to try to solve the inverse problem by the naive reconstruction

$$\hat{\mathbf{f}} \approx \mathbf{A}^{-1} \mathbf{g} \quad (2.3)$$

However, in most relevant applications, this approach will fail. Problems stem from evaluation of the inverse of the matrix model \mathbf{A} . For a matrix to be invertible, its rows must be linearly independent of each other, i.e., no row can be expressed as a linear combination of the other rows. Otherwise, the matrix is said to be noninvertible or singular. For many indirect measurements modeled by Eq. (2.2), the rows of the matrix model \mathbf{A} are “quasi-dependent” on each other, so that the matrix is “nearly” singular and \mathbf{A}^{-1} becomes very large. The quasi-dependency arises due to overlap within the measurement model, which reflects the fact that there is not a strict one-to-one correspondence between individual measurement values of \mathbf{g} and certain sub-ranges of the unknown function f . So, while discretization allows an approximate inverse of \mathbf{A} to be obtained, the large values of \mathbf{A}^{-1} will make the solution very sensitive to any measurement errors. To understand this, we reconsider the naive reconstruction attempt:

$$\hat{\mathbf{f}} \approx \mathbf{A}^{-1} \mathbf{g} \approx \mathbf{A}^{-1} (\mathbf{A} \mathbf{f} + \mathbf{e}) = \mathbf{f} + \mathbf{A}^{-1} (\mathbf{e}) \quad (2.4)$$

If there were no measurement error, the naive reconstruction $\hat{\mathbf{f}}$ would be mathematically equivalent to the true solution \mathbf{f} . However, the presence of measurement error will translate into uncertainty in the solution and will be amplified by large values of \mathbf{A}^{-1} . This makes the solution unstable in that a small error in measurement can lead to very large error in solution.

Solution instability is a fundamental characteristic of “ill-posed” problems. A problem is called ill-posed if any one of the following conditions is violated: (i) existence of a solution, (ii) uniqueness of the solution, and (iii) continuity of the solution on the input data. The first two conditions seem rather straightforward, and the third condition can be understood as a question of stability, requiring that small changes to the input (e.g., due to noise or error) do not produce arbitrarily large changes to the output. Inverse problems generally fail to fulfill at least one, and often all three, of these conditions. We must often draw on methods from mathematical optimization to find successful and noise-robust solutions to inverse problems.

From our overview so far, we identify three major components to tackling commonly encountered continuous linear inverse problems: (i) establish a mathematical model of the forward problem using parameters and physical laws that characterize the system under study; (ii) discretize the continuum model to construct a discrete linear inverse problem suitable for numerical evaluation; and (iii) solve the mathematical inverse problem by applying an appropriate computational technique.

Forward modeling tasks associated with component (i) are not the primary focus of this study. Given a particular application, it is not necessarily straightforward to define the mathematical model of indirect measurements described by Eq. (2.1). Constructing the forward map \mathcal{A} must be considered as a nontrivial mathematical modeling task in itself. The forward model must capture the physical processes that govern the measurement theory, as well as technical properties of the measurement device and possible limitations in the data sets. We will address how these critical elements are modeled when we describe specific example applications of our developed inversion algorithm, but we do not claim to have developed novel measurement theory. We will reference application-specific studies for established measurement models based on appropriate theory or calibration data.

Computational solution techniques associated with component (iii) are also not the primary focus of this study. An array of techniques have been developed by other researchers to solve the mathematical inverse problem (Kandlikar and Ramachandran 1999). We did not develop a novel optimization routine to tackle this problem for this study. Rather, we examine how our inversion algorithm interfaces with three established numerical methods to evaluate the final solution of given inverse problems.

Our focus here will be on (i) converting the continuum measurement model of Eq. (2.1) to a form suitable for numerical evaluation, and (ii) applying an appropriate numerical method to handle the solution computation. In the next section, we narrow our focus onto the inversion of measurements obtained from particle sizing instruments to characterize the particle size distribution in a sampled aerosol. We concentrate especially on some of the mathematical methods commonly used to approximate the unknown function f as a discrete vector \mathbf{f} for solution. This will set the stage for a comparison to the methods used in our own inversion algorithm.

2.3 Review of Inversion Methods for Aerosol Spectrometers

The distribution of particle sizes is a fundamental property of any aerosol. An array of instruments, which we will refer to collectively as “aerosol spectrometers,” have been developed for determining aerosol particle size distributions. Aerosol spectrometers classify an ensemble of particles into a number of channels based on various measures of particle size. These measures include the aerodynamic diameter (impactors), optical-equivalent diameter (multi-angle or multi-spectral nephelometers and so-called optical particle counters), diffusion-equivalent diameter (diffusion batteries), electrical mobility-equivalent diameter (electrostatic classifiers), or any other size parameter derived from the behavior of the individual particles or the aerosol in an energy or force field.

A common feature for particle size distribution measurement systems is that the relationship between the discrete collected observations and the unknown size distribution function (i.e., the relationship that must be modeled by the linear operator \mathcal{A} discussed above) can be mathematically expressed by means of an integral equation (Nguyen and Cox 1989; Voutilainen and Kaipio 2000; Wang 2007; Wang and Yang 2008; and Wang 2008). This is formulated as

$$g_i = \int_a^b K_i(x) f(x) dx + e_i, \quad i = 1, 2, \dots, m \quad (2.5)$$

where $f(x)$ now represents the particle size distribution we wish to determine, which is a function of the particle size parameter x , and g_i represents the i th measurement out of m total recorded observations. The variable i may label the i th stage in the series of a cascade impactor, or the i th channel of a differential electrical mobility classifier, etc. The function $K_i(x)$ then describes the response of the i th measurement stage/channel/etc. to a particle of size x , which is determined from calibration data or theoretical models and can have the character of a deposition function, a spectral function, an angular scattering pattern, a transmission efficiency, etc. The integration limits $[a, b]$ define the finite x region wherein the integrand is nonzero. Note also that the data g_i are contaminated with errors e_i .

We call the function $K_i(x)$ in Eq. (2.5) the *kernel* function. There are different mathematical meanings for the term *kernel*; we use it to mean the representation of a linear operator. Here, the kernel function $K_i(x)$ plays the same role as the continuous linear operator \mathcal{A} from the continuum model of Eq. (2.1). That is,

$K_i(x)$ represents a collection of the governing equations related to a particular measuring instrument to ultimately describe the transformation of $f(x)$ into g_i .

The formulation in Eq. (2.5) results in a set of m so-called Fredholm integral equations of the first kind. Given a discrete set of measurements $\{g_i\}_{i=1}^m$, each with a corresponding kernel function $K_i(x)$ and measurement error e_i , the inversion problem for aerosol spectrometers amounts to solving this set of equations for the unknown particle size distribution function $f(x)$. As discussed above for inversion problems in general, a common strategy for extracting this information involves making discrete approximations to the continuous functions of the measurement model. Numerical *discretization* describes this process of transferring continuous functions, models, and equations into discrete counterparts suitable for numerical evaluation. For the set of integral equations defined by Eq. (2.5), this process involves numerical *quadrature*, which refers to any method used to numerically approximate the value of a definite integral.

Generally, the continuum model of Eq. (2.5) is discretized by approximating the integral as a sum such that

$$g_i \approx \sum_{j=1}^n K_{ij} f_j + e_i, \quad i = 1, \dots, m \quad (2.6)$$

where n is the number of discrete points at which the particle size distribution function $f(x)$ is to be obtained. This constructs a system of m linear equations in n unknowns, the unknowns being the discrete values f_j to comprise the particle size distribution solution. Defining the $n \times 1$ vector $\mathbf{f} = (f_j)$, along with the $m \times 1$ vectors $\mathbf{g} = (g_i)$ and $\mathbf{e} = (e_i)$, and the $m \times n$ matrix $\mathbf{A} = (K_{ij})$ constructs the single matrix equation model we're after: $\mathbf{g} = \mathbf{A}\mathbf{f} + \mathbf{e}$.

This discrete model is a defining element of any approach to the aerosol spectrometer inversion problem. Much of what determines both the complexity and the effectiveness of a given approach lies in the mathematical methods used to construct the discrete model. Details behind these methods define the intermediate steps leading from the continuum model of Eq. (2.1) to the discrete model of Eq. (2.6). These details dictate how to evaluate the entries K_{ij} of the matrix model \mathbf{A} , and they establish how to interpret the solution values f_j .

To understand this, let's discuss these discretization details more explicitly using mathematical expressions. For the moment, as we focus on the mathematical

methods typically employed to approximate the integral operator in Eq. (2.5), we will ignore the measurement error term. Consider the following summary of steps:

$$g_i = \int_a^b K_i(x) f(x) dx = \sum_{j=1}^n \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) f(x) dx \quad (2.7)$$

$$= \sum_{j=1}^n f(x_j) \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx \quad (2.8)$$

$$= \sum_{j=1}^n K_{ij} f_j \quad (2.9)$$

Each of these tidy summary steps involves certain underlying details. We review these details here to set the stage for comparing the mathematical methods most commonly employed in tackling the aerosol spectrometer inversion problem to those we use in our own inversion algorithm.

Selection of Solution Points

In the first step of the discretization process, as summarized by Eq. (2.7), we break the integral over $[a, b]$ into n subintervals and define the limits of integration for these subintervals with a set of $n + 1$ points we will call the *primary integration points*. These $n + 1$ points establish size bin boundaries around the n discrete output nodes $(x_j)_{j=1}^n$ at which the particle size distribution function $f(x)$ is to be obtained. We define these size bin boundaries to be the geometric mean values between successive output nodes, i.e., $x_{j\pm\frac{1}{2}} = \sqrt{x_j x_{j\pm 1}}$. At the endpoints, we use $x_0 = 2x_1 - x_2$ and $x_{n+1} = 2x_n - x_{n-1}$ to evaluate $x_{-\frac{1}{2}}$ and $x_{n+\frac{1}{2}}$.

The discrete output nodes $(x_j)_{j=1}^n$ partition the size interval $[a, b]$ such that $a \leq x_1 < \dots < x_n \leq b$. To completely characterize this sequence of points, we must specify (i) the targeted upper and lower size bounds $[x_1, x_n]$ for our inverse solution, (ii) the number n of output nodes to use, and (iii) the spacing $(x_{j+1} - x_j)$ between successive output nodes. These specifications are a few examples of the important details within the model discretization process that will affect the complexity and the effectiveness of the inversion approach.

For example, with respect to node spacing, the output nodes $(x_j)_{j=1}^n$ are most often defined to be evenly spaced such that

$$x_j = x_1 + \Delta \cdot (j - 1), \quad j = 1, \dots, n \quad (2.10)$$

where

$$\Delta = (x_n - x_1) / (n - 1) \quad (2.11)$$

This uniform approach can make subsequent calculations for setting up the discrete model slightly more mathematically convenient. However, this even-spacing constraint is not always necessary, and we do not impose it within our inversion algorithm. We remove any point-spacing restrictions from our discrete model setup, and the resulting data analysis is more flexible and adaptable to any given application.

As for the targeted lower and upper size bounds $[x_1, x_n]$ and the number n of output nodes to use for the inverse solution, these details are largely application specific. Since the number of output nodes sets the column dimension of the $m \times n$ matrix model \mathbf{A} , it has an important connection to not only the amount of information provided by the m measured data points, but also the level of detail available for evaluating the kernel function within the matrix model \mathbf{A} . In constructing the system of linear equations with m measured data points and the $m \times n$ matrix model \mathbf{A} , the number of unknowns n need not necessarily coincide with the number of equations m . The only requirement is that n be large enough to approximate all size distributions of interest as closely as needed and to capture the linear independence of the kernel functions.

It is difficult to justify choosing $n < m$ such that fewer points are recovered for the size distribution as there are raw data points. This would be eliminating information and restricting the solution set. We will not explore this option further.

Often, $n = m$, giving an equal number of discrete values in the computed size distribution as in the raw measured data. If the size intervals corresponding to these points exactly overlap (i.e., the upper and lower size bounds $[x_1, x_n]$ and all of the included size points at which the particle size distribution function is to be recovered coincide exactly with the centroid size-equivalent parameters of the measurement channels), this approach utilizes the actual information content of the measured data without additional assumptions. This is the approach of many early inversion algorithms for differential mobility analyzer (DMA) systems (Knutson 1976; Hoppel 1978; Hagen and Alofs 1983).

In some cases, end effects may arise in the inversion solution if the discretization of the kernel integral is restricted to the measurement range. For such situations, it is advantageous for the targeted lower and/or upper size bounds $[x_1, x_n]$ to extend beyond the minimum and/or maximum measured size parameter. The typical approach here is to let $n > m$, such that the size channels of the computed size distribution exactly overlap those of the raw measured data within the measurement range, but the kernel discretization is also extrapolated beyond the endpoints of the measurement range. There are then two options for the overall setup of the discrete model before solving the inversion problem: (i) leave the matrix model \mathbf{A} with dimensions $m \times n$ where $n > m$ (underdetermined, with more columns than rows) and solve for the n vector \mathbf{f} , given the m vector \mathbf{g} ; or (ii) extrapolate both the raw measured data and the response matrix so that the response matrix \mathbf{A} has dimensions $m_{extrap} \times n$ where $n = m_{extrap} > m$ (square matrix, with equal number of columns and rows) and solve for the n vector \mathbf{f} , given the extrapolated m_{extrap} vector \mathbf{g} ; the extrapolated bins can be discarded after the inversion.

These different options for defining the size bounds $[x_1, x_n]$ and node number n can have unique impacts on size distribution recovery. Our inversion algorithm can handle any definition of the solution points. We will compare and contrast the options outlined here in greater depth in Chapter 3. For all of the inversion demonstrations included in the present study, we will use one-to-one correspondence between the size channels of the computed size distribution and those of the raw measured data, with extrapolation so that $n = m_{extrap} > m$.

Representation of the Size Distribution Function

Once the dimension n and the corresponding integration points have been defined, the next step in developing the discrete model, as summarized by Eq. (2.8), addresses the representation of the size distribution function within each of the subinterval integrals. The most common approach is to represent the size distribution function as a histogram so that we may place a discrete value of $f(x)$ outside of the integral. This is the approach represented in Eq. (2.8) as

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) f(x) dx = f(x_j) \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx$$

The discrete value $f(x_j)$ becomes the unknown represented by f_j to be recovered from the inversion analysis. If x_j corresponds to the point c_j in $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ where

$f(x)$ achieves its mean value for that subinterval (i.e., $f(x_j) = f(c_j)$), then this is a simple application of the first mean value theorem for definite integrals.¹ However, the location of this c_j cannot be precisely known a priori, so it is often taken to represent some midpoint value for the j th subinterval. Here, we simply use the discrete output nodes $(x_j)_{j=1}^n$ around which we have defined our size bin intervals.

Relying on a single value for the size distribution function $f(x)$ over the size bin interval corresponding to x_j can be a major source of error in inverse calculations. This histogram method provides no indication of the shape of the distribution within channel boundaries. Figure 2.1(a) illustrates this lack of detail, with a histogram representation overlaying a simulated particle number size distribution.

More generally, the size distribution function $f(x)$ in the response integral can be approximated as a higher order piecewise polynomial, or *spline* function. A spline function consists of polynomial pieces on subintervals joined together with certain continuity conditions. The polynomial pieces interpolate between control points that correspond to locations called knots, and the continuity conditions depend on the degree of the polynomial pieces. Splines tend to be more stable than fitting a polynomial through all control points, with less possibility of wild oscillations between points. Mathematical details behind spline interpolation are presented in Appendix C.

After the histogram method, the next-higher order approximation for any given continuous function involves linear interpolation. This defines a linear, or first-

¹The first mean value theorem for definite integrals states that for a continuous function f in $[a, b]$, there exists c in (a, b) such that

$$\int_a^b f(x) dx = f(c)(b - a)$$

and since the mean value of f on $[a, b]$ is defined as

$$\frac{1}{b - a} \int_a^b f(x) dx$$

we can interpret the conclusion as f achieves its mean value at some point c in (a, b) . In general, for a continuous function f and a nonnegative integrable function g that does not change sign on $[a, b]$, there exists a c in (a, b) such that

$$\int_a^b f(x) g(x) dx = f(c) \int_a^b g(x) dx$$

degree, spline function. Figure 2.1(b) illustrates the improvement in capturing details of the simulated particle number size distribution that comes with using a linear spline representation.

Second-degree, or quadratic, splines are generally not used in applications. An even smoother, and often more realistic, approximation for a given particle size distribution can be achieved with cubic, or third-degree, spline interpolation. The goal of cubic spline interpolation is to get an interpolation formula that is continuous in both the first and second derivatives, both within the intervals and at the interpolating nodes. In general, if the function to be approximated is smooth, cubic splines will do better than piecewise linear interpolation. Figure 2.1(c) illustrates how closely a cubic spline representation approximates the simulated particle number size distribution.

Throughout the different fields concerned with inversion of aerosol spectrometer data, we most frequently encounter the histogram method for approximating the particle size distribution function within calculations. There is more limited use of linear spline interpolation. A notable example is the numerical approach described by Wolfenbarger and Seinfeld (1990) for representing the size distribution as a linear spline with equally spaced knots.

Higher order spline approximations are least commonly encountered in the literature. Studies that do apply higher order quadrature are mostly concentrated in optical fields. These applications will usually represent the size distribution function as a cubic spline, with the constraint of defining equally spaced knots.

As we will describe in more detail in the next section, the inversion algorithm we developed for this study is capable of performing numerical calculations with the size distribution function represented as a cubic spline with no constraints on knot spacing. We will demonstrate the improvement in size distribution recovery provided by this approach over approximating the size distribution function with either histogram or linear spline interpolation.

Evaluation of Kernel Matrix Elements

The final step in the overall discretization process, as summarized Eq. (2.9) evaluates the integral of the kernel function over the j th subinterval to define the entries K_{ij}

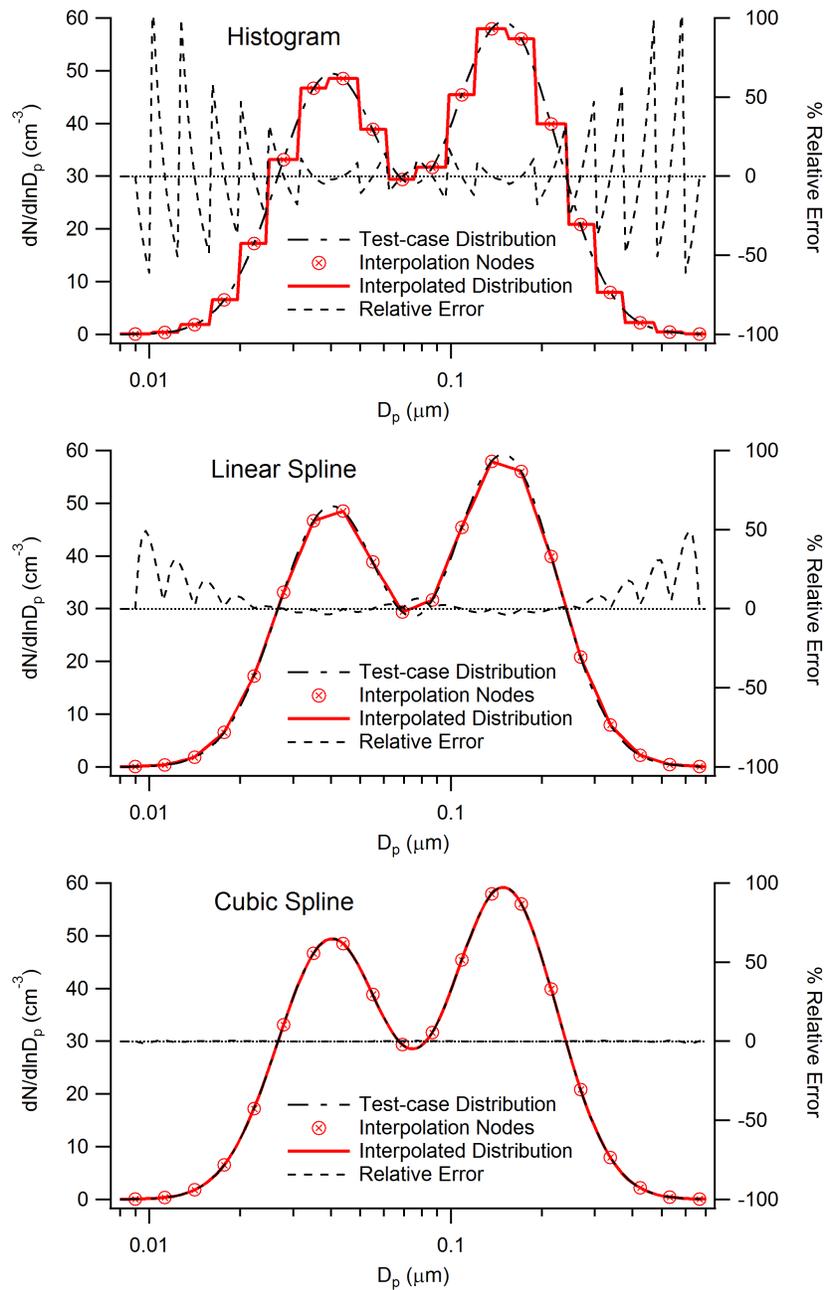


Figure 2.1: Histogram, linear spline, and cubic spline representations of a bimodal test-case aerosol previously presented in Russell et al. (1995) and Collins et al. (2002). Each interpolation method was applied to 20 known distribution function values indicated as the 'Interpolation Nodes.' Also shown are values of percent relative error quantified as $e_j = \left(f_j^{\text{interp}} - f_j^{\text{true}} \right) / f_j^{\text{true}} \times 100$ for individual interpolation points over the entire particle size range.

of the matrix model \mathbf{A} ,

$$K_{ij} = \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx, \quad i = 1, \dots, m, j = 1, \dots, n \quad (2.12)$$

The simplest approximation to this integration is to represent $K_i(x)$ by only a single value evaluated at the point x_j within the j th subinterval such that

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx = K_i(x_j) \Delta x_j \quad (2.13)$$

where $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. Ideally, $K_i(x_j)$ represents the true bin-average value for the given subinterval so that this is another application of the first mean value theorem for definite integrals (i.e., $K(x_j) = K(c_j)$). However, $K_i(x)$ is often simply evaluated at the same midpoint value x_j chosen for the interval as defined above.

Just as for the size distribution function $f(x)$, relying on a single value for the kernel function $K_i(x)$ over the size bin interval corresponding to x_j can be a source of error in inverse calculations. It is important to take full advantage of the detail available for the kernel function definition. If the kernel function can be defined with finer resolution than the width of the size bin intervals, then we should be using this resolution to evaluate the integral of the kernel function over the size bin interval. This requires a much different quadrature rule than the simple approximation of Eq. (2.13), but it effectively improves the recovery of the size distribution information without increasing the number of overall size bins. This is the approach we take in our inversion algorithm, which we describe in detail in the next section.

2.4 Present Approach

Our success in converting raw measurements into true particle number size distributions depends largely on how accurately we are able to represent the continuum measurement model of Eq. (2.5) within our analytical inverse calculations. In general, we must implement a number of numerical approximations to transfer a given measurement model, including properties of the sampled aerosol, to a form suitable for numerical evaluation within our inversion algorithm. In this section, we describe the functional details of our approach to (i) discreteizing the continuum response integral and (ii) incorporating our quadrature method into our kernel matrix setup.

Our success with a given inverse problem further depends, at least to some extent, on the computational technique used to evaluate the final inverse solution. To address this, we also describe in this section our approach to (iii) applying an appropriate mathematical optimization method to handle the final inverse solution computation.

Numerical Discretization

Any quadrature method relies on evaluating a given integrand on a finite set of points (called the *abscissas* or *quadrature points*), then processing these evaluations somehow to produce an approximation to the value of the integral. The goal is to determine which points to evaluate and how to process the integrand around the evaluated points so as to maximize performance over a broad class of integrands.

The integrand of the continuum model of Eq. (2.5) consists of the product of two functions: the kernel function $K_i(x)$, and the particle size distribution function $f(x)$. The accuracy of the discrete model relies on the resolution with which these two functions are represented within the chosen quadrature method. The kernel function $K_i(x)$ is ideally known with fine enough resolution from detailed calibration or theoretical data that it can be evaluated within the size bin intervals defined for the discrete model. The particle size distribution function $f(x)$, on the other hand, is unknown, and we are hoping to solve for just a finite number of representative values (i.e., one for each of the defined size bin intervals). To represent $f(x)$ with finer resolution in the discrete model calculations then, we define expressions to interpolate between the individual solution values. Our strategy is to approximate the particle size distribution function $f(x)$ as a cubic spline function.

The histogram representation of the particle size distribution described above can be thought of as a spline function of degree zero, consisting of piecewise constant functions. Linear interpolation involves polynomial pieces of degree one, or first-degree. These zero- and first-degree splines are not so useful for actual applications because their low-order derivatives are discontinuous. For first-degree splines, the slope of the spline may change abruptly at the knots. Zero-degree splines are entirely discontinuous.

Higher-degree splines are useful whenever more smoothness is needed in the approximating function. Cubic splines are the most popular. They produce an interpolated function that is continuous through to the second derivative. We represent the continuous particle size distribution function as a piecewise cubic spline function within our discrete inverse calculations. This cubic spline representation is a distinguish-

ing feature of our inversion algorithm. Nevertheless, we will keep the mathematical expressions we introduce here general for spline functions of degree d .

By definition, a spline function of degree d with knot sequence \mathbf{t} , which we denote as $S_{\mathbf{t}}^d(x)$, is a linear combination of special spline functions called B-splines, or basis functions, such that

$$S_{\mathbf{t}}^d(x) = \sum_j c_j B_{j,d}(x) \quad (2.14)$$

where the $B_{j,d}$ represent the constituent B-splines of degree d , and the c_j are real numbers called the spline coefficients. An in-depth discussion of B-splines and spline function approximations is included in Appendix C, but we summarize a few key properties here to aid the immediate discussion. These properties will prove essential to understanding the mathematical setup of our discrete inverse calculations, in which we represent the continuous particle size distribution function as a piecewise spline function.

1. **Knot points.** A spline function is defined with respect to a certain set of points, called the control points, that correspond to locations (t_j) called knots.
2. **Local knots.**
 - a) The j th B-spline $B_{j,d}$ depends only on the $d + 2$ knots $t_j, t_{j+1}, \dots, t_{j+d+1}$.
 - b) If x lies in the interval $[t_j, t_{j+1})$ then only the $d+1$ B-splines $B_{j-d,d}, \dots, B_{j,d}$ are nonzero.
3. **Local support.** If x is outside the interval $[t_j, t_{j+d+1})$ then $B_{j,d}(x) = 0$.
4. **Positivity.** If $x \in (t_j, t_{j+d+1})$ then $B_{j,d}(x) > 0$.

To construct a spline function representation of our discretized size distribution, working with n output nodes $(x_j)_{j=1}^n$, we must then also include the endpoints $x_0 = 2x_1 - x_2$ and $x_{n+1} = 2x_n - x_{n-1}$. We will need these points to provide full support to our first and last subinterval integrals in our inverse calculations. We then need $n + 2d + 2$ spline function knot points $\mathbf{t} = (t_j)_{j=-d}^{n+d+1}$ to correspond exactly to the n output nodes $(x_j)_{j=1}^n$ of our discretized size distribution, with some extrapolation beyond the upper and lower bounds (i.e., $(t_j)_{j=1}^n = (x_j)_{j=1}^n$). On this knot vector, we define $n + d + 1$ B-splines $\{B_{j,d}\}_{j=-d}^n$ of degree d , and each B-spline will have a corresponding coefficient value $\mathbf{c} = (c_j)_{j=-d}^n$.

From property 2(b), we can express any interpolant piece $S_j^d(x)|_{x \in [x_j, x_{j+1}]}$ of the spline function (i.e., any of the piecewise polynomials between adjacent knot points $[t_j, t_{j+1}] \equiv [x_j, x_{j+1}]$) as a linear combination of $d+1$ B-splines and their coefficients, such that

$$S_j^d(x) = \sum_{k=j-d}^j c_k B_{k,d}(x) \quad (2.15)$$

For example, for cubic splines where $d = 3$, this gives

$$\begin{aligned} S_j^3(x) &= \sum_{k=j-3}^j c_k B_{k,3}(x) \\ &= c_{j-3} B_{j-3,3}(x) + c_{j-2} B_{j-2,3}(x) + c_{j-1} B_{j-1,3}(x) + c_j B_{j,3}(x) \end{aligned} \quad (2.16)$$

This is the expression we use to represent the size distribution function $f(x)$ within our inversion calculations. Inserting the B-spline representation of Eq. (2.15) into the continuum measurement model of Eq. (2.5) changes the summary of our discretization process as follows

$$\begin{aligned} g_i &= \int_a^b K_i(x) f(x) dx = \sum_{j=1}^n \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) f(x) dx \\ &= \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) S_{j-1}^d(x) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) S_j^d(x) dx \right\} \\ &= \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) \left[\sum_{k=j-1-d}^{j-1} c_k B_{k,d}(x) \right] \right. \\ &\quad \left. + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) \left[\sum_{k=j-d}^j c_k B_{k,d}(x) \right] dx \right\} \end{aligned} \quad (2.17)$$

where now we plan to represent the both the size distribution function and the kernel function with much finer resolution within the size bin intervals. This final expression can still be simplified, however, with some algebraic rearrangement to collect like terms for our kernel matrix setup.

Kernel Matrix Setup

When we expand all sums in Eq. (2.17), paying close attention to indexing, and we collect terms involving common spline coefficients c_j , we find

$$\begin{aligned}
 g_i &= \int_a^b K_i(x) f(x) dx = \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) \left[\sum_{k=j-1-d}^{j-1} c_k B_{k,d}(x) \right] \right. \\
 &\quad \left. + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) \left[\sum_{k=j-d}^j c_k B_{k,d}(x) \right] dx \right\} \\
 &= \sum_{j=-d}^n c_j \int_{x_j}^{x_{j+d+1}} K_i(x) B_{j,d}(x) dx \\
 &= \sum_{j=-d}^n K_{ij} c_j \tag{2.18}
 \end{aligned}$$

We have not made explicit here the details of the necessary algebraic steps, but they can be found in Appendix D. The takeaway from Eq. (2.18) is that we now designate the set of spline coefficients $\mathbf{c} = (c_j)_{j=-d}^n$ as the discrete unknown values to be recovered from the inversion analysis. For zero- and first-degree spline functions, these unknown coefficients are uniquely defined by the interpolation conditions, and so their values correspond to the magnitude of the interpolated function. For the higher degree cubic splines, there is some arbitrariness in choosing the coefficient values. In fact, none of the coefficients is uniquely determined by the interpolation conditions. Once we solve for the set of spline coefficients with our chosen mathematical optimization method, we must use these coefficients with our constituent B-splines in Eq. (2.14) to construct our final cubic spline function solution.

To accompany our $(n + d + 1) \times 1$ solution vector $\mathbf{c} = (c_j)_{j=-d}^n$, we must complete the setup of our $m \times (n + d + 1)$ kernel matrix model $\mathbf{A} = (K_{ij})$. We express the response matrix elements generally as

$$K_{ij} = \int_{\mathcal{I}} K_i(x) B_{j,d}(x) dx \tag{2.19}$$

where \mathcal{I} is the interval where the product $K_i(x) B_{j,d}(x)$ is nonzero. Recall from above that the support of the j th B-spline the interval $[t_j, t_{j+d+1})$; outside of this

interval $B_{j,d}(x) = 0$. We evaluate this integral using the trapezoidal rule over each size bin interval:

$$\int_{x_j}^{x_{j+1}} K_i(x) B_{j,d}(x) dx = \sum_{p=0}^{n_{sub}} \beta_p K_i(x_{j,p}) B_{j,d}(x_{j,p}) \Delta x_{j,p} \quad (2.20)$$

where n_{sub} is the number of subintervals to use in evaluating the integral over the j th size bin, $x_{j,p}$ represents the secondary integration point at which to evaluate the kernel function within the j th bin, $\Delta x_{j,p}$ is the width of the n_{sub} subinterval bins, and $\beta_p = 1/2$ at the endpoints of the summation and 1 elsewhere. The set of $n_{sub} + 1$ points $(x_{j,p})_{p=0}^{n_{sub}}$ that define the secondary integration points (i.e., subinterval boundaries) are evaluated as

$$x_{j,p} = x_j + \Delta x_{j,p} \cdot p, \quad p = 0, \dots, n_{sub} \quad (2.21)$$

with

$$\Delta x_{j,p} = \frac{\Delta x_j}{n_{sub}} \quad (2.22)$$

$$\Delta x_j = x_{j+1} - x_j \quad (2.23)$$

This leads to the matrix representation of the response equation and system kernel where

$$\begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} = \begin{bmatrix} \int_{\mathcal{D}} K_1(x) B_{-d,d}(x) dx & \cdots & \int_{\mathcal{D}} K_1(x) B_{0,d}(x) dx & \cdots & \int_{\mathcal{D}} K_1(x) B_{n,d}(x) dx \\ \int_{\mathcal{D}} K_2(x) B_{-d,d}(x) dx & \cdots & \int_{\mathcal{D}} K_2(x) B_{0,d}(x) dx & \cdots & \int_{\mathcal{D}} K_2(x) B_{n,d}(x) dx \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \int_{\mathcal{D}} K_m(x) B_{-d,d}(x) dx & \cdots & \int_{\mathcal{D}} K_m(x) B_{0,d}(x) dx & \cdots & \int_{\mathcal{D}} K_m(x) B_{n,d}(x) dx \end{bmatrix} \begin{bmatrix} c_{-d} \\ \vdots \\ c_0 \\ \vdots \\ c_n \end{bmatrix} \quad (2.24)$$

The advantage of the approach described here (i.e., discretizing the continuum response integral by approximating the size distribution function as a piecewise spline function and then incorporating this quadrature method into our kernel matrix setup) is that we need only compute the response matrix elements K_{ij} once for a given set of measurement conditions. We can then use this kernel matrix to analyze multiple sets of measured data $\mathbf{g} = (g_i)$ collected under equivalent conditions. We simply solve our linear system to yield unique sets of coefficients $\mathbf{c} = (c_j)_{j=-d}^n$ with which the cubic spline function approximation of the size distribution can be evaluated at any desired values of x .

Numerical Solution Methods

Finally, solving an inverse problem involves implementing a computational algorithm that recovers useful information from measured data. The word “useful” can best be understood in the context of a particular application. The aerosol size distribution inversion problem is an example of an ill-posed problem which is notoriously difficult to solve (Wolfenbarger and Seinfeld 1990). The solution technique must be capable of handling ill-posed problems and difficulties with conditioning of the kernel matrix (overlap of kernel functions, failure of inverse to exist) and should employ reasonable constraints (not seemingly arbitrary). An array of techniques have been developed to tackle this problem. Kandlikar and Ramachandran (1999) compiled a critical review on inverse methods.

No single solution technique can be considered universally superior to the others and hence prescribed as the magic bullet for all types of aerosols and for all measurement devices. Algorithm comparisons should be done with the task and context in mind. “The particular form of inversion selected is not important provided that reasonable solutions can be found through its application. Alternative solutions found using other techniques to analyze the same data merely serve to illustrate the lack of uniqueness in this problem.” (Steele 1997) The focus of this study is not on the performance of specific mathematical optimization algorithms, except to show that the ones employed here are sufficient.

Our kernel matrix setup can interface with three numerical solution methods: (i) the non-negative least squares (NNLS) method of Lawson and Hanson (1987); (ii) an interior-point gradient method for large-scale totally non-negative least squares problems (TNNLS) from the work of Merritt and Zhang (2005); and (iii) a regularization method (Ilavsky and Jemian 2009).

The non-negative least squares algorithms (NNLS and TNNLS) are constrained least squares methods that require every component of the particle size distribution function, being a concentration value, to be non-negative. NNLS is an active-set method that requires matrix factorization and updates and can become overly expensive for very large-scale problems. TNNLS is a gradient-type method that requires only matrix-vector multiplications.

TNNLS can have very slow convergence, but is the method of choice for very large-scale applications when high-accuracy solutions are not necessary. Interior-point algorithms have many good characteristics, such as low memory usage and the ability to solve large problems quickly. However, solutions can be slightly less accurate

than those from other algorithms. The reason for this potential inaccuracy is that the (internally calculated) barrier function keeps iterates away from inequality constraint boundaries. For most practical purposes, this inaccuracy is usually quite small. There are a couple approaches to reduce the inaccuracy: use smaller tolerances; or run a different algorithm, starting from the interior-point solution. The TNNLS algorithm also incorporates measurement uncertainties into the inversion solution. This is important because when the accuracy of the data is overestimated then too much confidence has been placed in the data's ability to determine a highly structured solution and features may be smoothed out. When the accuracy is underestimated then confidence is lost that the structure observed in the solution really exists.

Regularization methods overcome ill-posedness by replacing the problem with a nearby well-posed problem (e.g., Wolfenbarger and Seinfeld 1990). They involve smoothness criteria and a regularization parameter which controls the degree of smoothing applied. These algorithms allow a free form for the instrument response matrix, but require a careful choice of constraining parameters and weights of penalty functions. Literature examples include generalized cross-validation for selecting the regularization parameter (Crump and Seinfeld 1982); the L-curve method; and the discrepancy principle.

We include a zeroth-order regularization approach (i.e., discrepancy principle) as a numerical solution method. This approach provides a solution that matches the measurements to just within expected experimental error. It has the advantage of being computationally simple and very time-efficient. However, this approach imposes a high standard for the reported errors, and it can tend to oversmooth the solution, corresponding to a very large value of the regularization parameter.

For simplicity and consistency, all of the inversion demonstrations included here use the TNNLS algorithm for final solution computations.

2.5 Methods

We use synthetic data to demonstrate and quantitatively evaluate our inversion analysis. For a specific example application, we investigate the inversion problem encountered with data collected using a long column differential mobility analyzer (DMA), the most commonly used instrument system for recovering aerosol particle size distributions. For our purposes, the principal advantages of using synthetic data over experimentally measured data are that (i) the true size distribution is known, so that the retrieved size distribution can be checked against it, and (ii) there are

no measuring errors in the data so that errors can be simulated as desired. We calculated measured response values for various test size distributions, then used these “artificial measurements” as input for the inversion algorithm, and finally compared the inversion result with the initial test particle size distribution. The following general procedure describes our numerical evaluations:

1. A particle number size distribution function is defined over a particle size range of interest.
2. All nonzero response values are calculated for a model instrument system.
3. The simulated measurement data is processed and analyzed with our inversion algorithm to retrieve a size distribution solution.
4. The *reconstructed* distribution is compared to the *true* size distribution using suitable error parameters.

We describe our approach in more detail here.

Generation of Noisy Measurements

With direct analogy to Eq. (2.5), we now adopt nomenclature more specific to aerosol remote sensing to define the following relationship between measured responses R_i (analogous to g_i) and the aerosol size spectrum $n_N(\ln D_p)$ (analogous to $f(x)$)

$$R_i = \int_0^{\infty} n_N(\ln D_p) \Psi(i, D_p) d \ln D_p, \quad i = 1, 2, \dots, I \quad (2.25)$$

Here, the particle number size distribution $n_N(\ln D_p) = \frac{dN}{d \ln D_p}$ [particles/cm³] is expressed in terms of the logarithm of diameter, $\ln D_p$, to facilitate the recovery of distributions that span several decades in particle size. The kernel $\Psi(i, D_p)$ (analogous to $K_i(x)$) relates the sampled size distribution to the measured response values. Details on the DMA kernel function and overall system setup for simulation are included in Appendix A.

To simulate measured response values, we express the integral in Eq. (2.25) as a discrete sum, covering a finite region from $\ln D_{p1}$ to $\ln D_{pJ}$, in which the integrand is nonzero, subdivided into $J - 1$ intervals. Using the extended trapezoidal rule

(Abramowitz and Stegun 1972) gives

$$R_i \cong \sum_{j=1}^J \beta_j n_N (\ln D_{pj}) \Psi(i, D_{pj}) \Delta \ln D_{pj} \quad (2.26)$$

where J is the number of quadrature points, D_{pj} is the particle diameter corresponding to the j th bin, $\Delta \ln D_{pj}$ is the width of the j th bin, and $\beta_j = 1/2$ at the endpoints of the summation and 1 elsewhere.

Unless otherwise noted, the input particle number size distributions n_N analyzed here are created as superpositions of p normal and p' lognormal distributions, as defined by

$$\begin{aligned} n_N (\ln D_p) = & \sum_{i=1}^p \frac{N_i}{(2\pi)^{1/2} \ln \sigma_i} \exp \left(-\frac{(\ln D_p - \ln \bar{D}_{pi})^2}{2 \ln^2 \sigma_i} \right) \\ & + \sum_{i=1}^{p'} \frac{D_p N'_i}{(2\pi)^{1/2} \sigma'_i} \exp \left(-\frac{(D_p - \bar{D}'_{pi})^2}{2 (\sigma'_i)^2} \right) \end{aligned} \quad (2.27)$$

where $N_i, \bar{D}_{pi}, \sigma_i$ and $N'_i, \bar{D}'_{pi}, \sigma'_i$ are the number concentration, median diameter, and standard deviation of the i th lognormal and normal distributions, respectively.

A dangerous pitfall in algorithm development and testing is the act of committing an inverse crime. This occurs when we obtain a great reconstruction because the simulated data resonates in some helpful way with the reconstruction algorithm. For example, using the same computational grids for data simulations and inversion analyses sometimes results in perfect reconstruction from noise-free data. To avoid committing an inverse crime, we define a set of $(D_{pj})_{j=1}^J$ to use in Eq. (2.26) that is much more finely resolved than the set of particle diameters that correspond to the solution nodes for a reconstructed particle size distribution. While the inverse solution nodes are defined within our inversion algorithm based on user input, we can define any other size grid by subdividing a target size range $[D_{p1}, D_{pJ}]$ into J points spaced equally on a logarithmic scale using the expression

$$D_{pj} = D_{p1} e^{(j-1)\Delta}, \quad j = 1, 2, \dots, J \quad (2.28)$$

where

$$\Delta = (\ln D_{pJ} - \ln D_{p1}) / (J - 1) = \Delta \ln D_p \quad (2.29)$$

We calculate response values from Eq. (2.26) using $J = 2001$ discrete size points from $D_{p1} = 0.1 \text{ nm}$ to $D_{pJ} = 10 \text{ }\mu\text{m}$.

Analyzing noise-free data is also not realistic. We can obtain excellent inversion results this way, but these are not representative of any realistic inversion problem, since noise is present in any experimental setting. Actual measurements are subject to a varying degree of uncertainty. For measurements based on the counting of events, the uncertainty is governed by a Poisson distribution. Therefore, we impose Poisson counting statistics on the simulated instrument particle counts and estimate the uncertainties in inverted size distribution results by applying a Monte Carlo method. For this, we draw sets of instrument responses as random samples from the calculated instrument responses, assuming a Poisson distribution, and use these as input for the inversion algorithm. We repeat this procedure two hundred times, storing each inversion result, and then statistically evaluate the stored inversion results to obtain a mean result and its uncertainty. We then compare this result to the original size distribution using suitable error parameters.

Evaluation Metrics

It is important to have a quantitative measure available for the closeness of an inversion reconstruction to the actual measured object. Of course, such a measure makes sense only with simulated data, when the original object is known. In this way, we can isolate and quantify the error caused by assumptions and approximations made in the inversion setup that do not represent the instrument or measurement process accurately.

A number of statistical metrics can be used as this quantitative measure of performance. The best statistical metrics should not only provide a performance measure, but also produce a representation of the error distribution. Sometimes multiple metrics are required for completeness. Statistical measures, by definition, condense a large number of data into a single value, removing a lot of information. Any single metric provides only one projection of the model errors and, therefore, only emphasizes a certain aspect of the error characteristics. Thus, a combination of metrics is often required to provide a complete picture of error distribution. We momentarily return to a more general mathematical nomenclature to define the following two evaluation metrics.

For a sense of the relative error distribution over the size range of interest of x , we

evaluate a simple, discrete measure of relative error defined by

$$e_j = \frac{f_j^R - f_j^t}{f_j^t} \quad (2.30)$$

where the f_j^t are discrete values from the true size distribution and the f_j^R are discrete values from the reconstructed size distribution. Thus, e_j provides a measure of relative error for each of the discrete values $f_j^R = f^R(x_j)$ evaluated for the set of size parameters $(x_j)_{j=1}^J$.

For an assessment of the overall size distribution recovery from our inversion analysis, we use the parameter defined by Ramachandran and Leith (1992) as a relative error in reconstruction:

$$e_R = \sqrt{\frac{\int_a^b [f^t(x) - f^R(x)]^2 dx}{\int_a^b [f^t(x)]^2 dx}} \quad (2.31)$$

where $f^t(x)$ is the true size distribution function and $f^R(x)$ is the reconstructed size distribution function. This provides a single quantitative measure of agreement between these two functions.

2.6 Results and Discussion

In the demonstrations that follow, we evaluate the relative error metrics introduced above, setting $f^t(x) = \left(\frac{dN}{d \ln D_p}\right)_{true}$ to describe the size distribution simulated as input to a DMA system and $f^R(x) = \left(\frac{dN}{d \ln D_p}\right)_{inv}$ to describe the size distribution recovered from our inversion analysis. We compare the two functions by evaluating each at the same $J = 301$ discrete size points spaced equally on a logarithmic scale from $D_{p1} = 1 \text{ nm}$ to $D_{pJ} = 1 \text{ }\mu\text{m}$. For $\left(\frac{dN}{d \ln D_p}\right)_{true}$, this simply involves evaluating Eq. (2.27) for the corresponding set of size parameters $(D_{pj})_{j=1}^J$ defined by Eq. (2.28). For $\left(\frac{dN}{d \ln D_p}\right)_{inv}$, this requires interpolation between the inverse solution nodes. This is straightforward with our cubic spline approach, since the inverse solution is a set of coefficients $\mathbf{c} = (c_j)_{j=-d}^n$ that can be used with the constituent cubic B-splines in Eq. (2.14) to construct a final cubic spline function solution at any desired values of $x = \ln D_p$. For the other methods that use delta-function node, histogram, or linear spline approximations within the inversion analysis, the discrete inverse solution values correspond to the magnitude of the approximated

size distribution function so that we must choose how to interpolate between these values.

As we will discuss next, using cubic B-splines for simple interpolation has great advantages. So that these advantages do not overshadow the benefits of incorporating cubic B-splines *within* our inverse solution method, we choose to apply cubic spline interpolation to the solution nodes retrieved with any of the inversion methods considered here.

Advantage of Cubic Spline Interpolation

We have already briefly discussed one of the major advantages of using higher order piecewise spline functions when approximating a continuous aerosol particle number size distribution: the higher the degree of the spline function, the smoother the approximating function. We included the discrete evaluation metric of Eq. (2.30) in the plots of Figure 2.1 to aid our visual comparison of the histogram, linear spline, and cubic spline interpolation methods. Of these three methods, cubic spline interpolation provides the smoothest and most accurate representation of the given size distribution function, across the entire size range of interest.

A further advantage of using higher order spline functions in interpolation applications is that, in general, the higher the degree of the spline used for interpolation, the fewer the number of knots needed to attain a given level of precision. We can demonstrate this for the simulated particle number size distribution shown in Figure 2.1. When we vary the number of knots (i.e., interpolation nodes) used to perform the histogram, linear spline, and cubic spline interpolation calculations and then evaluate the overall measure of relative error defined by Eq. (2.31) for each case, we trace the trend shown in Figure 2.2. Cubic spline interpolation consistently provides the most accurate representation of the true size distribution, and it even performs as well with only 10 knots as linear spline interpolation performs with 20 knots.

The advantages afforded by using higher order spline functions (i.e., smoother, more accurate approximations needing fewer knot points) extend to applications beyond simple interpolation. Next, we use our inversion algorithm to demonstrate that this trend also applies to size distributions recovered from inverse calculations.

Size Distribution Recovery: Kernel Matrix Setup

We compare various approaches to the mathematical setup of the kernel matrix for discrete inverse calculations with respect to the impact on size distribution recovery.

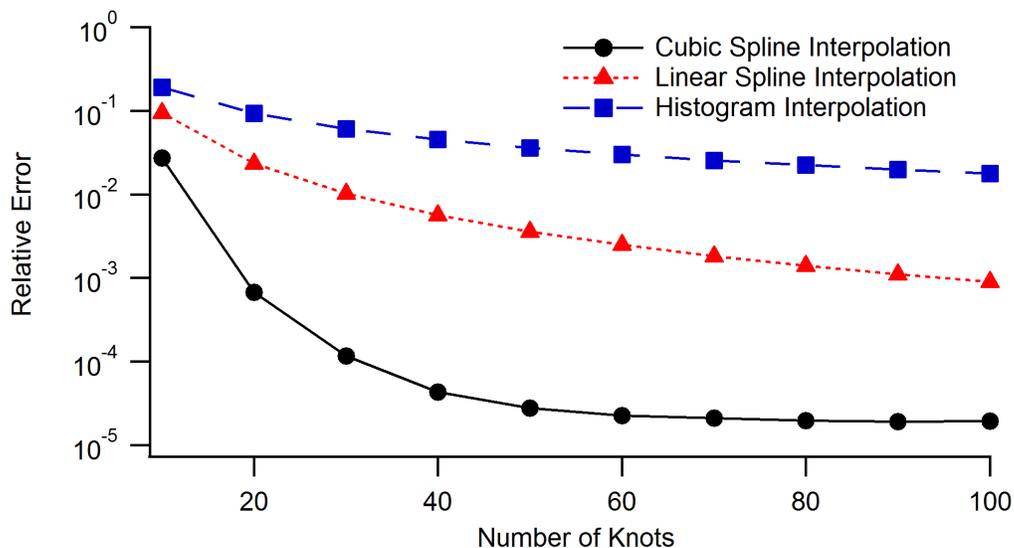


Figure 2.2: Relative error in reconstruction, e_R , shown as a function of the number of knots used for the interpolation of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002).

Specifically, we consider each of the following:

1. Assuming constant kernel function elements and size distribution over the discretized size bins provides the simplest approximation which we will call “simple quadrature.”
2. Employing a histogram representation of the size distribution with subdivisions for evaluating kernel elements provides a slightly improved approximation which we will call “histogram quadrature” since the size distribution function is still represented as a histogram.
3. Employing a linear spline representation of the size distribution with subdivisions for evaluating kernel elements provides an even more improved approximation which we will call “linear spline quadrature.”
4. Employing a cubic spline representation of the size distribution with subdivisions for evaluating kernel elements provides the most improved approximation which we will call “cubic spline quadrature.”

Mathematical details for these different quadrature methods are described in Appendix D. To investigate the impact on size distribution recovery, we perform inversion analyses on the same bimodal test-case aerosol used for the interpolation

investigations above. This test distribution was introduced by Russell et al. (1995) for inversion analyses, and it was replicated for comparison analyses by Collins et al. (2002). The bimodal number distribution is composed of two lognormal distributions with peaks of $N_1 = 50 \text{ cm}^{-3}$ at $\bar{D}_{p1} = 40 \text{ nm}$ and $N_2 = 60 \text{ cm}^{-3}$ at $\bar{D}_{p2} = 150 \text{ nm}$, both with standard deviations of $\sigma_1 = \sigma_2 = 1.5$.

Inversion results obtained with the different quadrature methods under noise-free conditions are illustrated in Figure 2.3. The discrete measure of relative error, e_j , is also shown with each solution. The particle size distribution recovered using cubic spline quadrature within our inversion analysis is shown in Figure 2.3(d) to have overall improved accuracy compared to the solutions recovered using the delta-function node, histogram, or linear spline approximations. The size distribution recoveries shown correspond to moderate resolution DMA operating conditions, which we will define here by the ratio of the particle-free sheath flow rate to that of the incoming aerosol sample, i.e., $R_{ideal} = \frac{Q_{sh}}{Q_a}$. See Appendix A for further details on the DMA measurement simulations.

The measurement range of the DMA under given operating conditions is indicated in Figure 2.3 with an increased thickness of the relative error line to highlight the targeted particle size range. Each of the inverse solution methods underestimates the test-case distribution quite abruptly as the particle size approaches the lower limit of detection. The raw particle counts measured here are low or zero due to poor measurement system efficiency and low sampled aerosol concentration in this size range. Throughout the rest of the measured size range, and even slightly beyond, the inverse solution obtained with the cubic spline technique remains well within $\pm 10\%$ relative error. The other solution methods show larger deviations again as the particle size approaches the upper limit of detection.

Another advantage of the cubic spline technique, as seen above for simple interpolation, is that a given quality of reconstruction can be obtained with fewer data points. We demonstrate this in Figure 2.4, which shows the measure of overall agreement between true and reconstructed size distributions, e_R , as a function of DMA resolution. In our simulated DMA measurements, the system operating resolution dictates the setting of measurement channels, which then correspond to the discrete size distribution solution nodes within our inversion routine. That is, fewer solution points are recovered to reconstruct a given particle size distribution under lower DMA operating resolutions. The cubic spline technique outperforms the other methods under all operating conditions considered, only converging to comparable

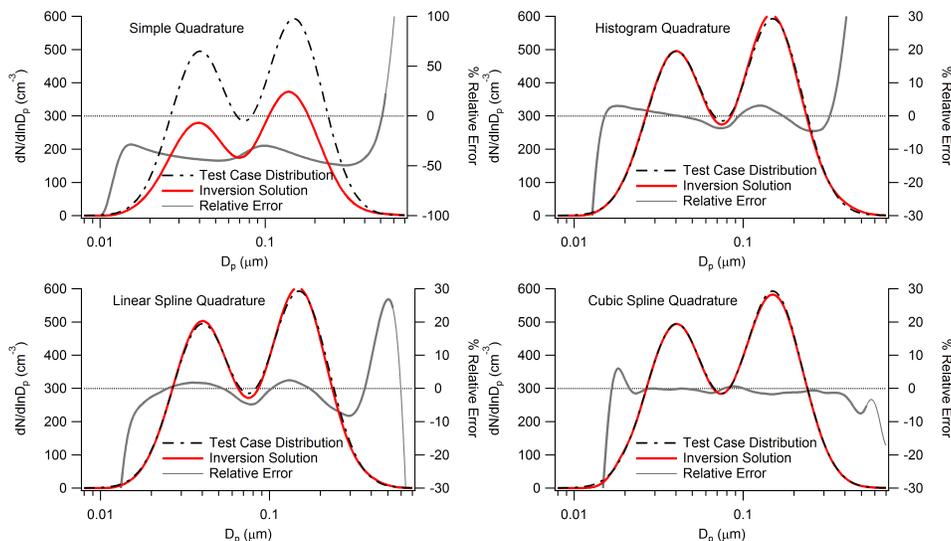


Figure 2.3: Inversion results obtained using the simple, histogram, linear spline, and cubic spline quadrature methods to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002). All inversion analyses were performed on noise-free measurements simulating moderate resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 5$).

inverse solution quality at the lowest of DMA resolutions.

Definition of Integration Points

We also used our inversion algorithm to consider how the number of secondary integration points defined per subinterval, n_{sub} , affects the recovered size distribution and optimize this number to balance performance and complexity of the cubic spline calculations. Notice how each quadrature method simplifies to the histogram method when finer kernel function resolution is not used in the calculations, i.e., when $n_{sub} = 1$. For our remaining analyses, we set $n_{sub} = 10$.

Additional Test Case Aerosols

We challenge our inversion algorithm with various test case distributions while imposing Poisson counting statistics on the simulated instrument particle counts. We estimate the uncertainties in the inverted size distribution results as described above. In the plots of size distribution recovery shown in Figures 2.6 through 2.9, the thick dashed black line represents the true size distribution. The cloud of dots represents the range of distributions recovered from the Monte Carlo solutions. The gray band around the size distribution shows ± 1 standard deviation around the

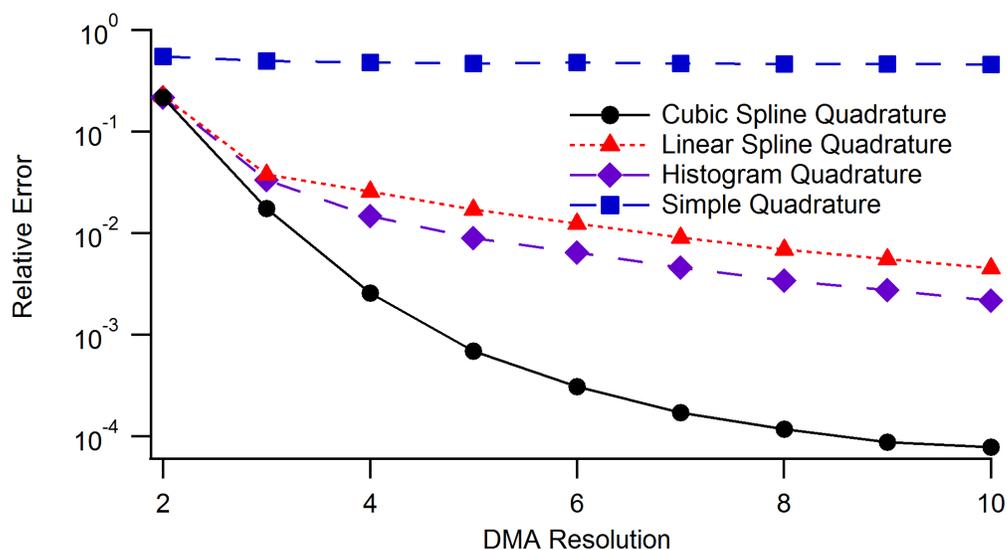


Figure 2.4: Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the DMA operating resolution.

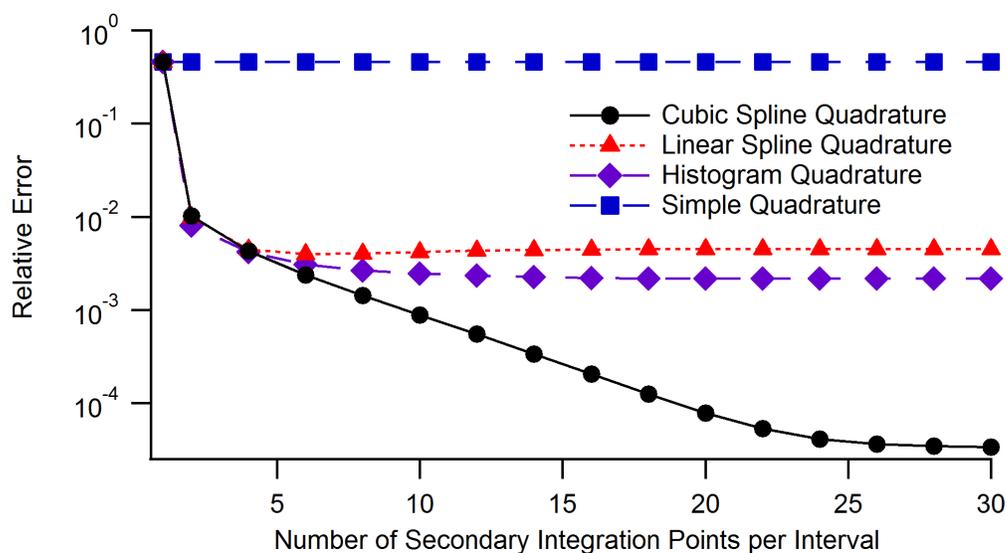


Figure 2.5: Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of n_{sub} , the number of secondary integration points used per interval of integration in the kernel matrix setup. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

average recovered distribution, which is shown as the solid red line. The discrete measure of relative error e_j is also included for the average solution over the full size range, plotted against the right axis along with a surrounding gray band to again highlight the DMA measurement size range and to show the spread in relative error for the ± 1 standard deviation solutions. The size distribution recoveries shown correspond to high resolution DMA operating conditions ($\frac{Q_{sh}}{Q_a} = 10$), while a summary of performance over a range of DMA operating resolutions is also shown in Figure 2.6.

2.7 Conclusions

Many DMA inversion algorithms have been proposed in the literature, each valid for a certain experimental setup (e.g., Alofs and Balakumar 1982; Collins et al. 2002; Stratmann et al. 1997). This reflects not only the common desire to develop and continually refine inversion algorithms to be somehow simultaneously robust and easy to use, it also demonstrates the many difficulties encountered in doing so.

The inversion analysis presented here performs well in analyzing uni- or multi-modal size distributions, with narrow or wide peaks, whether the sampled distribution is contained within or extends beyond the measurement size range. In this chapter and the next, we balance these performance demonstrations with clarity in the functional details of this inversion routine so that it may be used and adapted in the future. This inversion analysis can in principle be applied to other particle sizing instruments, such as cascade impactors, optical counters, new developments in mobility analysis technology, etc., provided the performance characteristics of the measurement devices are well understood.

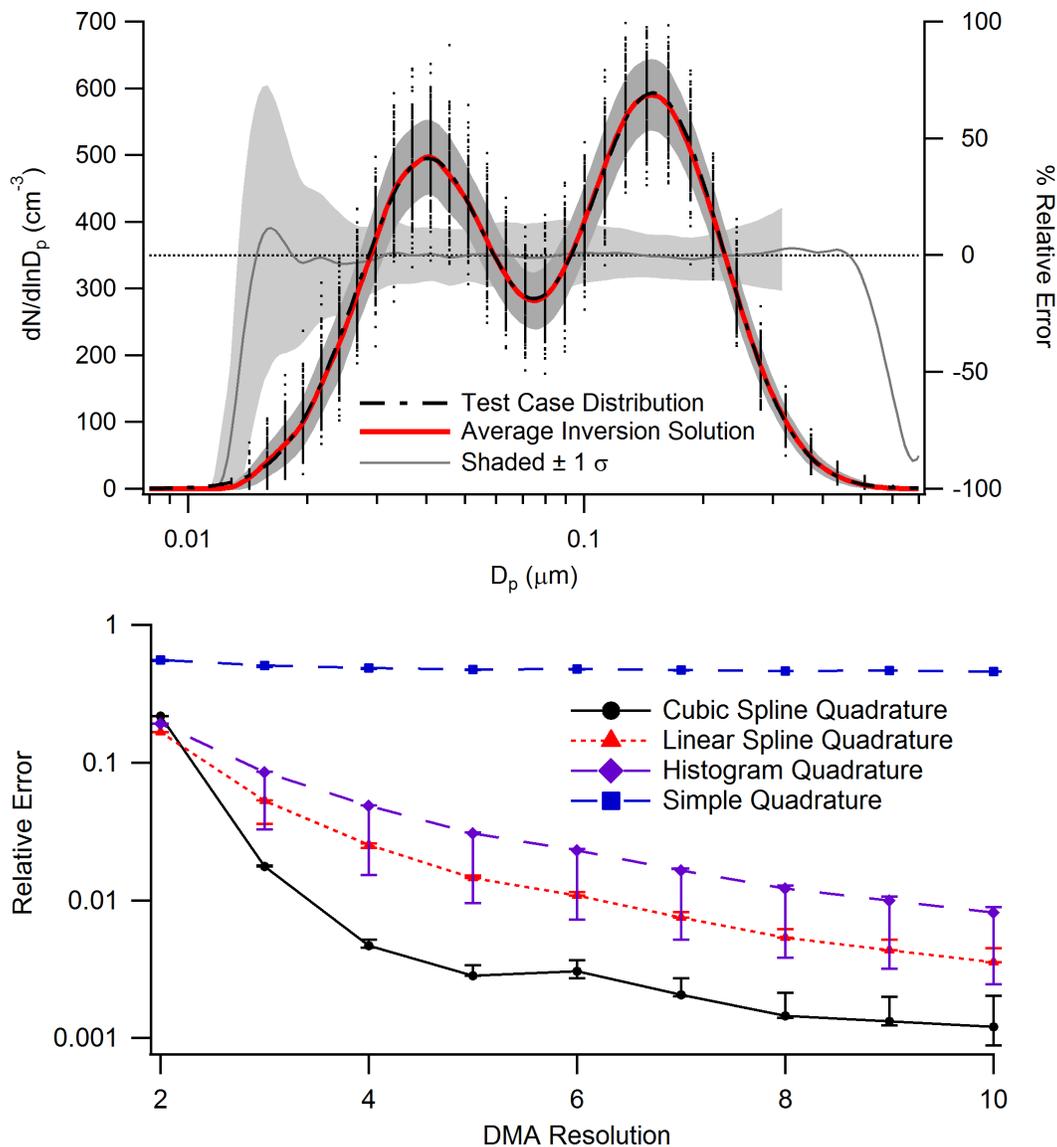


Figure 2.6: Recovery of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$) (top); and relative error in reconstruction, e_R , for inversion analyses of the same test-case particle size distribution, shown as a function of the DMA operating resolution (bottom).

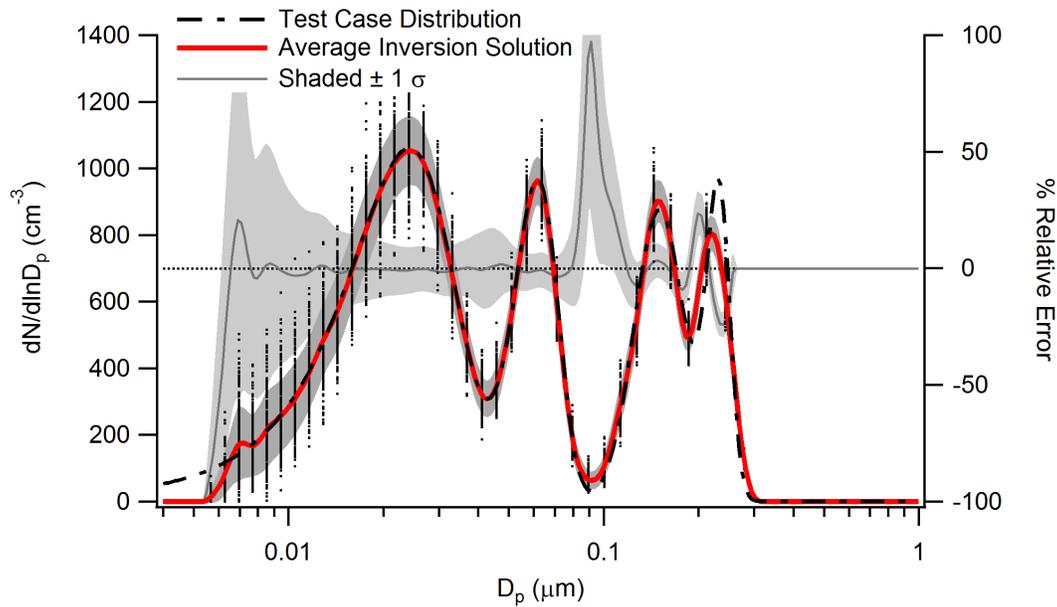


Figure 2.7: Recovery of a pathological multimodal test-case aerosol presented in Talukdar and Swihart (2003) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

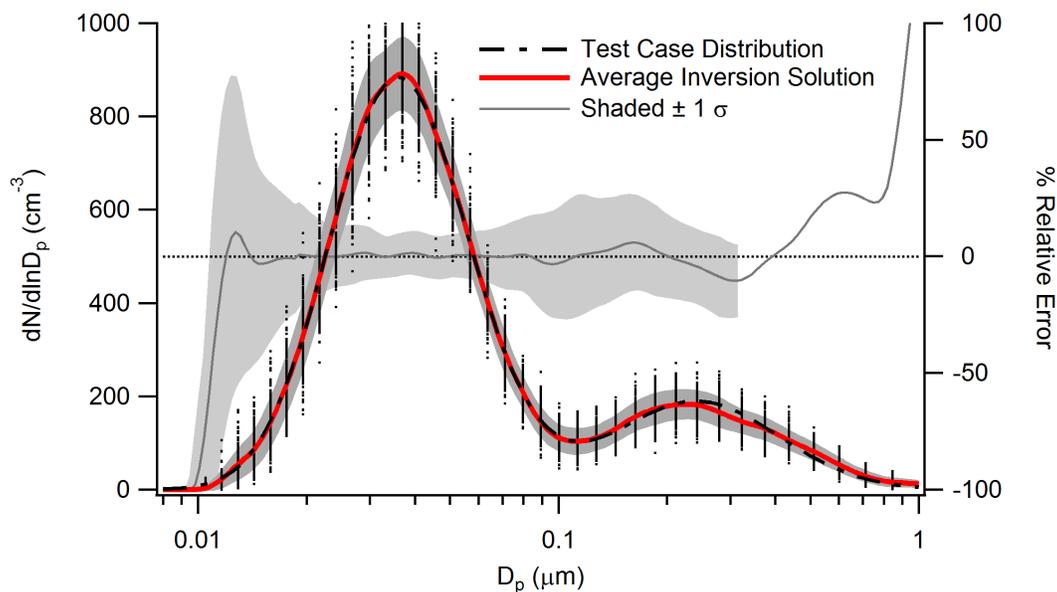


Figure 2.8: Recovery of an atmospheric test-case aerosol size distribution from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

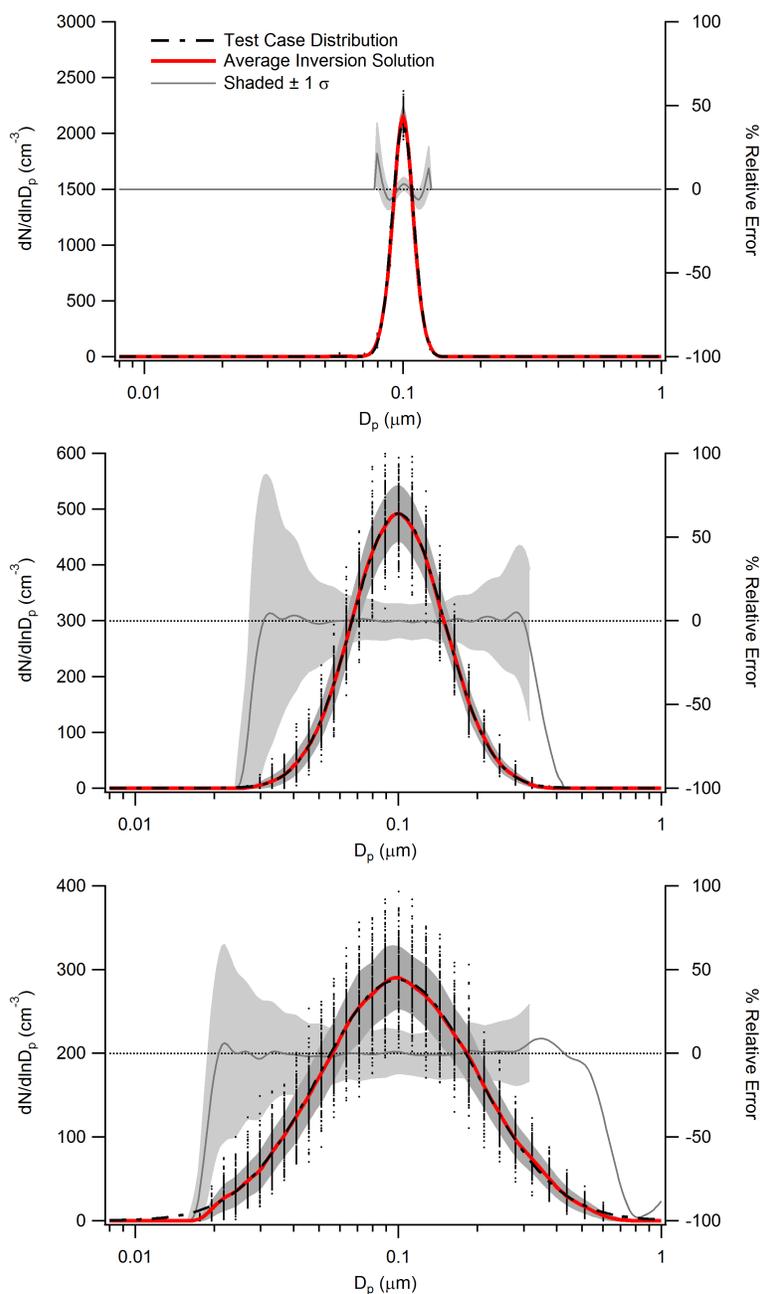


Figure 2.9: Recovery of unimodal test-case aerosol size distributions with $d_{pg} = 0.1 \mu\text{m}$ and varying σ_g : $\sigma_g = 1.1$ (top), $\sigma_g = 1.5$ (middle), and $\sigma_g = 2.0$ (bottom). All inversion analyses were performed on noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

*Chapter 3***ANALYSIS SOFTWARE FOR DIFFERENTIAL ELECTRICAL MOBILITY CLASSIFIERS**

A measurement strategy can be described as a framework for defining the assortment of parameters relevant to a given measurement campaign. The parameters in question here are not limited to details of data collection (e.g., what substance will be measured, at which location, when, for how long, and by what methods). Aspects of data processing and analysis are also integral to the overall measurement campaign. Since the objectives of particular measurement campaigns can vary broadly, the foundational measurement strategy for any given campaign should be designed to be consistent with its study objectives. We focus here on the familiar objective of recovering particle number size distribution information using differential mobility analyzer (DMA) systems. Data analysis considerations for DMA measurements revolve around the setup and solution of this inversion problem. We address various aspects of raw data processing and analysis that impact recovered size distributions, in an attempt to harmonize DMA measurement strategy as much as possible.

3.1 Introduction

Gathering useful information about airborne particulate matter involves extensive data collection and analysis (Knutson 1976; Hoppel 1978). The number concentration and size distribution of atmospheric particles are of particular importance in understanding and managing aerosol impacts on health, visibility, and climate. Aerosol particle size distribution data is often collected remotely, using deployable instruments. Collected data then need to be processed in a structured way. “The reliability of the information obtained depends not only on the quality of the measurements but also on the method of data processing.” (Bashurova et al. 1991) We focus here on the many aspects of raw data processing and analysis that impact recovered size distributions.

Since measurement systems rarely provide direct measures of the quantity of interest, characterizing the distribution of sizes of particles in a sampled aerosol usually implies an inversion of particle size data. Thorough consideration of the equations which will finally be inverted is an important step in any planned measurement campaign. The particle size distribution retrieval algorithm needs to be considered

as integral to the overall sampling system; the measuring system consists of the analysis algorithm, as well as the instruments.

Data analysis for any given measurement system must consider: (i) how to handle measurement errors, (ii) how to accurately capture system operation in equations (as reflected in the measurement kernels), (iii) the nature of the inversion algorithm applied, and (iv) the computational effort involved to implement algorithms. We address these data analysis considerations here, to the task of determining aerosol particle size distributions from differential mobility analyzer (DMA) measurements.

Differential mobility analysis is a powerful method for determining particle size distributions. A number of instrument systems apply the theory of differential electrical mobility classification to study atmospheric particles (Knutson and Whitby 1975; Kousaka et al. 1986; McMurry 2000). As this technology has gained broader use, it becomes necessary to detail technical standards and guidelines with respect to measurement collection and data analysis in order to enhance the mutual comparability of worldwide atmospheric aerosol measurements. Wiedensohler et al. (2012) detailed many such recommendations for mobility particle size spectrometers to ensure high data accuracy, comparability, transparency, and traceability of the measured particle number size distributions. They addressed the harmonization of data analysis with an intercomparison of various commercial and custom-made multiple-charge inversion routines. Deviations revealed with respect to the resulting particle number concentrations were attributed to different physical constants and charging probabilities used, different solutions to the matrix inversion problem, and different approaches of how to discretize, pre-process, and post-process the data. Limited descriptions of the analyzed inversion routines are provided, but these do not include full details on the potentially influential differences. In fact, surprisingly few publications are available that specify their algorithm in detail or the occupation of the matrix (Pfeifer 2014). We seek to do just that here.

Many DMA inversion algorithms have been proposed in the literature, each valid for a certain experimental setup (e.g., Alofs and Balakumar 1982; Collins et al. 2002; Stratmann et al. 1997). Kandlikar and Ramachandran (1999) reviewed numerous approaches that have been proposed to solve this problem. These algorithms seem to have not been sufficiently implemented in practice, perhaps because they have been perceived as too complex (hindering practical implementation) and/or numerically demanding. Most of the algorithms described contain elements of arbitrariness, such as how to smooth data and/or penalize deviations from prescribed size distribution

shapes.

The inversion code historically used for much of the data collected in the Flagan and Seinfeld research groups has been edited and amended by several researchers, but to other users, it is much like a black box. The code itself contains some minor errors, and one of its biggest draw-backs is the use of nonlinear iterative methods (e.g., Twomey iterations) for the inversion solution. Solutions from this method depend on the initial guess, which is not ideal behavior. Crump and Seinfeld (1982) discuss this solution method and challenge its use for solving the aerosol data inversion problem, stating, “the Twomey nonlinear algorithm does not confront the main problem associated with inversion of aerosol size distribution data, which is the ill-posed nature of the linear inverse problem.”

Ideally, inversion routines should produce positive and realistically smooth results that are easily understood and interpreted, and they should accurately reconstruct the true distribution. In addition, these routines should be easily adapted to a given situation. Currently, the better routines available which do not assume a functional form have a number of limitations. The routines are too often mathematically complex so that interpretation of the results or modification of the code may not be simple. It is an unfortunate observation that the majority of inversion algorithms that are actually used much in practice seem to be documented only poorly.

For specific demonstrations here, we investigate the data collected using a long column DMA, the most commonly used instrument system for recovering aerosol particle size distributions. We use synthetic data so that (i) the true size distribution is known, so that the retrieved size distribution can be checked against it, and (ii) there are no measuring errors in the data so that errors can be simulated as desired. We calculated measured responses, essentially “artificial measurements”, and used these as input for our data analysis. We compare the analyzed particle number size distributions to those taken as ideally known from the generated input distributions.

3.2 Inversion Code

Code for analyzing data collected from existing measurement systems, specifically the common cylindrical differential mobility analyzer (DMA), was prepared using the *Igor Pro* commercial scientific analysis application, version 6.37 (Wavemetrics, 2015). Final versions of full scripts are available in a repository with this thesis, and Appendix E archives printouts of the critical components of the cubic spline inverse solution method.

The following provides further information and context about the separately submitted *Igor* files necessary for others to continue to apply these prepared inversion packages. Contained here is an example application of the DMA data inversion routine, along with descriptions for all of the critical functions involved. These functions are shown in Appendix E as coded.

Recommended Mobility Particle Size Spectrometer Setup

The included analyses are set up to evaluate measurement systems which satisfy the recommended mobility particle size spectrometer setups described by Wiedensohler et al. (2012).

The quality of the measurements of mobility particle size spectrometers depends essentially on the stability of the aerosol and sheath flow rates as well as the performance of the detector. Correct determination of instrument air flow is important. Sheath air flow should be circulated in a closed loop, with dryers included to reduce RH in the sample and sheath air flows. Temperature and absolute pressure in the instrument should also be monitored.

We use equivalent pipe length to describe particle losses, e.g., inside of pipes, the DMA, aerosol dryer and bipolar charger. Equivalent pipe lengths of different device and plumbing elements aligned in sequence can be simply added if they are traversed by the same rate of aerosol flow. We describe particle losses by diffusion on a straight pipe by analytical formulas derived for the laminar flow regime (Hinds 1999).

The size-dependent counting efficiency of an individual condensation particle counter (CPC) used as the measurement system's particle detector may depend on geometry or actually supersaturation profile inside the condenser. Use experimental data or apply manufacturer's calibration curve with caution. Calibrate CPCs individually against a reference instrument. Particle losses inside CPCs are implicitly included in the measured counting efficiency; they can be estimated using diffusional deposition models (Stolzenburg and McMurry 1991).

Data Input and Parameter Specification

Defining various inputs is the first step in our DMA data analysis task. These inputs include the measured size data, along with the measurement conditions and system setup parameters, and the user's selection of parameters to define the response matrix calculations. All of these inputs are contained within waves in *Igor* that can

be viewed and edited within tables that appear as those of Tables 3.1 through 3.4.

Measured Data

Data recorded from a given measurement campaign must be organized into a format the inversion routine can read. This is illustrated in Table 3.1, where there is one column for each of the following:

1. Scan index; If analyzing multiple scans of data, the scan index will group data for an individual scan together for analysis.
2. Aerosol sample inlet flow rate, Q_a , in units of lpm.
3. Classified aerosol outlet flow rate, Q_m , in units of lpm.
4. Clean sheath air inlet flow rate, Q_{sh} , in units of lpm.
5. Excess air outlet flow rate, Q_{ex} , in units of lpm.
6. DMA voltage sequence; Instruments generally have equally spaced size channels (on a log scale).
7. Raw count data, expressed particle number count.
8. Sample time, in seconds; longer times will result in more accurate measurements, due to particle statistics, but the data will then take longer to collect.
9. Ambient temperature, in K.
10. Ambient pressure, in Pa.
11. Temperature difference between saturator and condenser in CPC detector, in degrees Celsius.
12. Detector flow rate, Q_{det} , in units of lpm.

The setup of this table is generated, with column labels, by executing the function **makedatamatrix()**. This function takes no inputs and makes a 2D matrix to hold recorded measurement data. It can hold as many rows as needed, to analyze multiple sets of measured size data. Each set should be grouped by its scan index, in the *iScan* column. The indices should count up in increments of 1, but can begin with any integer (i.e., 0 or 1 or 19...).

Table 3.1: Example of the measured data provided as input to the inversion analysis routine in the “datamatrix” wave.

iScan	Qa_lpm	Qm_lpm	Qsh_lpm	Qex_lpm	HVset_V	Counts	tsample_s	Tamb_K	Pamb_Pa	dTdet_C	Qdet_lpm
0	1.003	1.003	5.000	4.996	10	0.0955959	1	298	101325	26.7	0.999
0	1.002	1.005	4.996	4.995	15.0131	0.654082	1	298	101325	26.7	0.989
0	0.996	0.994	5.005	4.992	22.5393	3.44535	1	298	101325	26.7	0.994
0	0.994	1.000	5.007	4.998	33.8386	13.7772	1	298	101325	26.7	0.993
0	0.999	1.008	4.999	4.992	50.8022	41.4001	1	298	101325	26.6	1.000
0	1.005	0.995	5.006	5.001	76.2699	92.7005	1	298	101325	26.6	1.002
0	1.005	1.006	5.004	4.991	114.505	153.95	1	298	101325	26.6	1.008
0	1.010	1.010	5.008	4.994	171.907	191.091	1	298	101325	26.6	1.006
0	1.009	1.003	5.006	4.996	258.086	189.434	1	298	101325	26.5	0.998
0	0.997	1.007	4.999	5.000	387.467	191.828	1	298	101325	26.5	0.994
0	1.002	1.008	5.002	5.000	581.709	260.246	1	298	101325	26.5	0.990
0	0.996	1.004	5.002	4.999	873.326	378.678	1	298	101325	26.5	1.001
0	1.001	1.006	4.996	4.996	1311.13	438.209	1	298	101325	26.6	1.003
0	0.998	1.003	5.006	4.996	1968.42	364.253	1	298	101325	26.6	1.016
0	1.004	1.001	5.003	4.995	2955.21	208.265	1	298	101325	26.6	1.007
0	1.009	1.002	4.997	4.984	4436.69	76.596	1	298	101325	26.6	1.002
0	1.005	1.005	4.998	5.001	6660.85	16.3451	1	298	101325	26.7	1.005
0	1.011	1.014	5.000	5.000	10000	1.78967	1	298	101325	26.7	1.003

DMA system

Parameters describing the DMA system setup are to be input to the “DMAparams2specify” and “diffLossMatrix” data waves illustrated in Table 3.2. The values specified here will either be read and passed directly, or read and used in calculations, to assign the values in the “DMAmeasParams” and “DMAgeomParams” waves illustrated in Table 3.3. Subsequent calculations call from these waves to use these parameters, using the dimension labels.

The **makeWvDMAparams2specify()** function takes no input and makes a 1D input wave to specify parameters for DMA operation.

The **makeDiffLossMatrix()** function also takes no input, and it makes a 2D matrix to hold equivalent pipe lengths and flow parameters for system components where flow rate may differ from DMA flow rate (i.e., sample line splitting to different instruments). Losses through the DMA inlet/outlet, and other components with the same DMA flow rate, should be accounted for in the “DMAparams2specify” input wave. See Wiedensohler et al. (2012) for a discussion of using equivalent pipe lengths for DMA systems.

Response Matrix

Parameters defining the response matrix calculations are to be input to the “calcreponsematrixParams” data wave and the “CalcTFformula” and “InvMethod” text waves illustrated in Table 3.4. Executing the function **makeWvcalcreponsematrixParams()** again with no input makes these three input waves. Subsequent calculations call from these waves to use these parameters, using the dimension labels.

makeWvcalcreponsematrixParams ()

Input to “calcreponsematrixParams” data wave:

ndias2nchanRatio usually 1, for one-to-one correspondence between diameter bins of inverse solution and those of measurement channels

ZfactorLower factor to extend lower limit of kernel integration diameter range; 1.5 is good, extends to size with 50% greater mobility than that of the smallest measured size

Table 3.2: Input parameters to describe DMA system setup.

DMAparams2specify		diffLossMatrix		
Label	Input	Component	Leq_m	Q_lpm
R1dma_m	0.00937			
R2dma_m	0.01958			
Ldma_m	0.4444			
Leq_DMA_m	7.1			
Leq_other_m	6			
particleDensity_kgm3	1400			

Table 3.3: Parameter values to be referenced in inversion calculations.

DMAmeasParams		DMAgeomParams	
Label	Value	Label	Value
Qa_lpm	1.003	R1dma_m	0.00937
Qsh_lpm	4.999	R2dma_m	0.01958
betaDMA	0.201	Ldma_m	0.4444
deltaDMA	-0.001	f_dma	0.707
tsample_s	1	G_dma	2.42
dTdet_C	27	Leq_DMA_m	7.1
Qdet_lpm	1.000	Leq_other_m	6
Tamb_K	298		
Pamb_Pa	101325		
particleDensity_kdm3	1400		

Table 3.4: Input parameters to define reponse matrix calculations.

calcreponsematrixParams		CalcTFformula	InvMethod
Label	Input	Text Input	Text Input
ndias2nchanRatio	1	stolz	TNNLS
ZfactorLower	1.5		
ZfactorUpper	3		
nSub4Kernel	6		
chargemax4Kernel	3		
extrapMeasChan	1		

ZfactorUpper factor to extend upper limit of kernel integration diameter range; 3 is good, extends to size whose mobility with 3 charges equals that of the largest measured size with 1 charge

nSub4Kernel number of subdivisions for evaluating integral over size bin intervals

chargemax4Kernel maximum number of charges to account for in kernel function

extrapMeasChan option to extrapolate raw counts and kernel matrix before performing inversion; enter 1 for yes; 0 for no

Input to “CalcTFformula” text wave:

reference to chosen form of transfer function (i.e., stolz, nd, or diff)

Input to “InvMethod” text wave:

reference to chosen solution computation algorithm (i.e., NNLS, TNNLS, or reg)

Inversion Procedure

After executing each of the above setup functions and filling in the necessary data and parameter values, the single function to call to convert the raw count data of Figure 3.1 to the size distribution solution of Figure 3.2 is **DMAinversion()**. Thus, the function lineup for inversion analyses is as follows:

makedatamatrix ()

Fill with measured data.

makeWvDMAparams2specify ()

Specify DMA operating parameters.

makeDiffLossMatrix ()

Account for additional diffusion losses through system components.

makeWvcalcreponsematrixParams ()

Specify parameters for response matrix setup.

DMAinversion (datamatrix, diffLossMatrix, DMAparams2specify, calcreponsematrixParams, CalcTFformula, InvMethod, Kernel_precalc)

This function can analyze a single scan, or multiple scans, of DMA data.

Input:

datamatrix data recorded from measurement campaign
diffLossMatrix equivalent pipe lengths and flow parameters for system components
DMAparams2specify instrument geometry and Leq ; particle density
calcresponsematrixParams options for kernel matrix setup
CalcTFformula text wave to call form of transfer function
InvMethod text wave to call solution computation algorithm
Kernel_precalc can input a previously calculated kernel matrix to save computation time; otherwise input empty set $\{0\}$

Functions Called:

DMAanalysis (...)

Output:

invsolnMeas $[m \times nScans]$ $dN/d \ln D_p [\text{cm}^{-3}]$ evaluated over measurement range
dp_meas $[m \times nScans]$ $D_p [m]$ corresponding to measurement range
invsolnCoeffs $[ncoeff \times nScans]$ inverse solution coefficients for entire scan

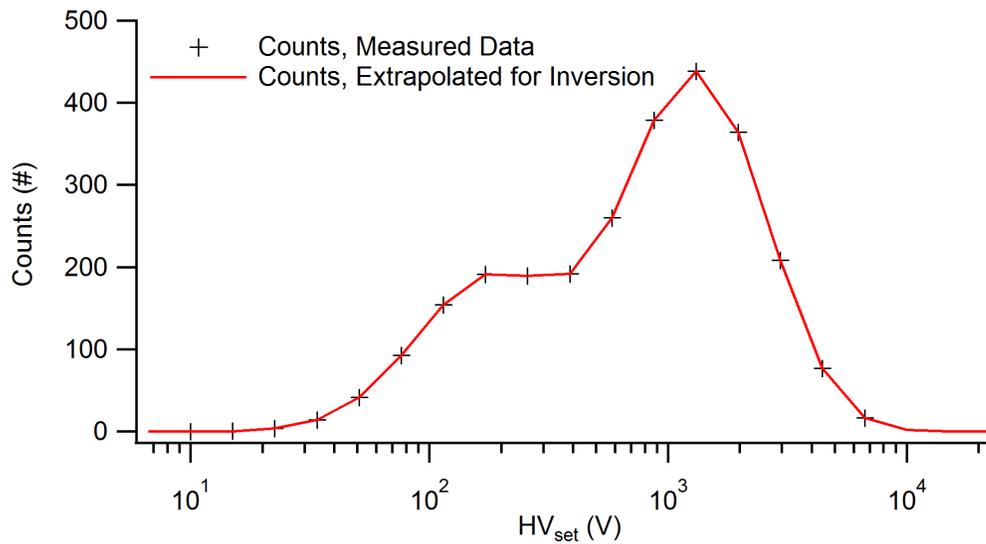


Figure 3.1: Raw count data, shown as particle number versus voltage setting, to be analyzed to recover the particle number size distribution.

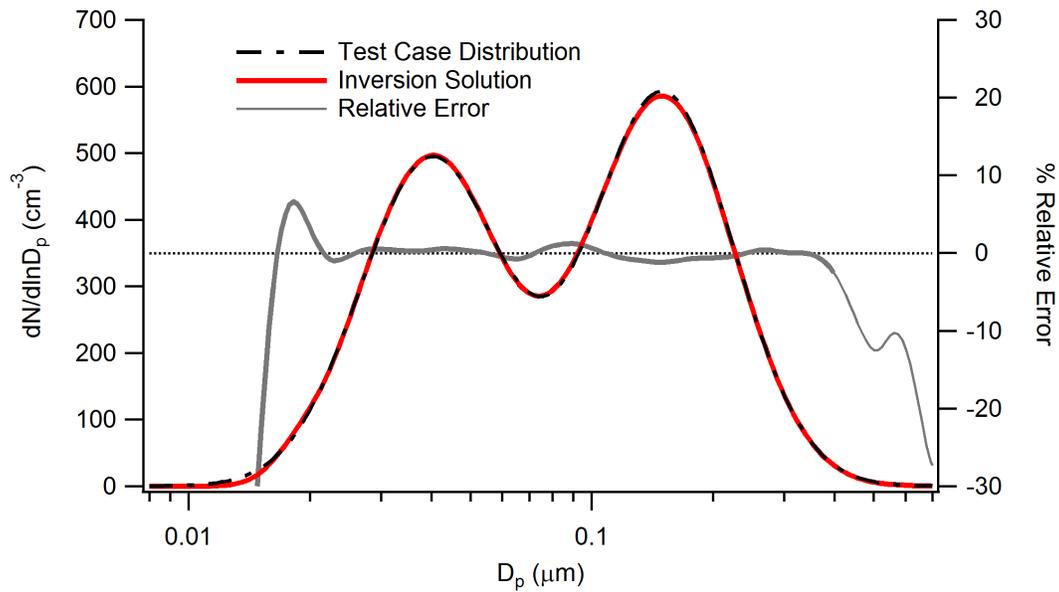


Figure 3.2: Particle number size distribution recovered from inversion analysis.

Supporting Code

The supporting code critical to the **DMAinversion(...)** function is organized here within descriptive subsections. These subsections group supporting functions together by their contributions to different key aspects of the inversion routine.

DMA Data Analysis

The following functions are called directly by the **DMAinversion(...)** function and deal with higher level data processing and analysis.

DMAanalysis (datamatrix, diffLossMatrix, DMAparams2specify, calcresponsematrixParams, CalcTFformula, InvMethod, Kernel_precalc)

Analyzes a single scan of DMA data.

Input:

datamatrix data recorded from measurement campaign

diffLossMatrix equivalent pipe lengths and flow parameters for system components

DMAparams2specify instrument geometry and Leq ; particle density

calcresponsematrixParams options for kernel matrix setup

CalcTFformula text wave to call form of transfer function

InvMethod text wave to call solution computation algorithm

Kernel_precalc can input a previously calculated kernel matrix to save computation time; otherwise input empty set $\{0\}$

Functions Called:

DMAparamSet (...)

calcresponsematrix (...)

extrapolateCounts_exp (...)

invsoln_Bspline (...)

Output:

$dN/d \ln D_p [\text{cm}^{-3}]$ size distribution inverse solution(s) in “invsoln_XXX” waves

D_p [m] corresponding particle diameters in “dpwave_XXX” waves

DMAparamSet (datamatrix, DMAparams2specify)

Reads input datamatrix for a single scan of DMA data and sets measurement and instrument parameters for subsequent calculations.

Input:

datamatrix data recorded from measurement campaign

DMAparams2specify instrument geometry and Leq; particle density

Functions Called:

makeWvDMAmeasParams ()

makeWvDMAgeomParams ()

Output:

DMAmeasParams

DMAgeomParams

makeWvDMAmeasParams ()

Makes a 1D wave to hold parameters describing operating conditions for a given measurement campaign. Subsequent calculations call from this wave to use these parameters, using dimension labels.

makeWvDMAgeomParams ()

Makes a 1D wave to hold parameters describing instrument setup for a given measurement campaign. Subsequent calculations call from this wave to use these parameters, using dimension labels.

extrapolateCounts_exp (ChanCounts, ChanMobility, ExtrMobility, numFitPts)

Extrapolates raw count data. See Hagen and Alofs (1983), Eqs. (6) and (7).

Input:

ChanCounts raw measured count data

ChanMobility measurement channel targeted mobility values corresponding to raw count data; input as inverse mobility

ExtrMobility mobility values for extrapolated range; input as inverse mobility

numFitPts number of endpoints to use in establishing extrapolation function

Output:**ExtrCounts**

invsoln_fineEval (SplineCoeffMatrix, SplineControlptKnots, SplineKnotWave, diam0_m, diamN_m, numdps)

Evaluates inverse solution(s) over size range of interest.

Input:

SplineCoeffMatrix [ncoeff x nSolns] coefficients of cubic spline size distribution solution

SplineControlptKnots [nDias x nSolns] binning for control points of size distribution solution

SplineKnotWave [(ncoeff+d+1) x nSolns] wave of knot values that define the cubic B-splines

diam0_m set lower bound for size distribution solution

diamN_m set upper bound for size distribution solution

numdps set number of points to evaluate for size distribution solution; can be very fine

Function Called:

cubicBspline (...)

Output:

invsolnFinal [numdps x nSolns] $dN/d \ln D_p [\text{cm}^{-3}]$ evaluated over size range of interest

dp_final [numdps x nSolns] $D_p [\text{m}]$ corresponding to final solution size range

Kernel Matrix Calculations

The following functions together create the inversion kernel matrix, incorporating cubic spline quadrature.

calcreponsematrix (precalcMatrix, numdias2numchan_ratio, Zfactor_lower, Zfactor_upper, nSub4CalcKernel, chargemax, extrapYorN, TFformula)

Calculates the kernel matrix.

Input:

- precalcMatrix** can input a previously calculated kernel matrix to save computation time; otherwise input empty set {0}
- numdias2numchan_ratio** usually 1, for one-to-one correspondence between diameter bins of inverse solution and those of measurement channels
- Zfactor_lower** factor to extend lower limit of kernel integration diameter range; 1.5 is good, extends to size with 50% greater mobility than that of the smallest measured size
- Zfactor_upper** factor to extend upper limit of kernel integration diameter range; 3 is good, extends to size whose mobility with 3 charges equals that of the largest measured size with 1 charge
- nSub4CalcKernel** number of subdivisions for evaluating integral over size bin intervals
- chargemax** maximum number of charges accounted for in kernel function
- extrapYorN** option to extrapolate raw counts and kernel matrix before performing inversion; enter 1 for yes; 0 for no
- TFformula** string to call form of transfer function

Functions Called:

CalcKernel_cubicBspline (...)

Output:

response matrix “Kernel_inv” and Bspline knot waves

CalcKernel_cubicBspline (ChanZ, ChanV, Dpbin, nSub, bbeta, delta, chargemax, TFformula)

Input:

- ChanZ** [m] vector of mobility values corresponding to channel measurements; $Z \left[\frac{\text{m}^2}{\text{V} \cdot \text{s}} \right]$
- ChanV** [m] vector of voltage values corresponding to channel measurements; V
- Dpbin** [nDias] vector of diameter values to define “model” size distribution solution nodes; should be input as $\ln(D_p)$ for log-linear channel spacing; D_p in m

nSub number of subdivisions for evaluating integral over size bin intervals

bbeta flow ratio

delta flow balance

chargemax maximum number of charges accounted for in kernel function

TFformula string to call form of transfer function

Functions Called:

define_binning (...)

MakeBsplineKnotWave (...)

KernelFcn_DMA (...)

BsplineC1 (...)

BsplineC2 (...)

BsplineC3 (...)

BsplineC4 (...)

Output:

KernelS3 [m x ncoeff] kernel matrix

BsplineKnotWave [nDias+2d+2]

BsplineCoeffKnots [ncoeff=nDias+d+1]

BsplineControlptKnots [n=nDias+2]

B-spline functional forms

BsplineC1 (xi, xk, xk1, xk2, xk3, xk4)

BsplineC2 (xi, xk, xk1, xk2, xk3, xk4)

BsplineC3 (xi, xk, xk1, xk2, xk3, xk4)

BsplineC4 (xi, xk, xk1, xk2, xk3, xk4)

MakeBsplineKnotWave (ControlptBin, degree)

Makes the wave of knot values needed to calculate B-splines of given degree.

Input:

ControlptBin [n]

degree for cubic splines, degree = 3

Output:

BsplineControlptKnots [n=nDias+2] binning for control points of size distribution solution; duplicate of input wave “ControlptBin”

BsplineKnotWave [n+2d] wave of knot values needed to calculate B-splines

BsplineCoeffKnots [ncoeff=n+d-1] binning for coefficients of size distribution solution

BsplineCoeffKnots_j [ncoeff=n+d-1] binning for coefficients of size distribution solution, corresponding to j index

define_binning (wavetobin)

Defines spline control points and bounds for integrations.

Input:

wavetobin [nDias] any vector of diameter values to define size distribution nodes; should be input as $\ln(D_p)$ for log-linear channel spacing; D_p in m

Output:

deltavals_geom [nDias] to set width of diameter bins

binvals_geom [nDias+1] to set bounds of size bin intervals

binvals_plusuplodiff [nDias+2=n] to set control points

Cubic Spline Inversion

The following functions together evaluate the inverse solution.

invsoln_Bspline (Kernelmatrix, ModelDataBinning, MeasDataBinning, SplineCoeffKnots, SplineControlptKnots, SplineKnotWave, precalcBdata, Buncertainties, InvSolnMethod)

Solves inverse solution with cubic B-spline setup.

Input:

Kernelmatrix [m x ncoeff] kernel matrix, describing measurement collection

ModelDataBinning [nDias] binning for size distribution solution, model

MeasDataBinning [m_raw] binning for measured signals (raw data)

SplineCoeffKnots [ncoeff=n+d-1] binning for coefficients of size distribution solution

SplineControlptKnots [n=nDias+2] binning for control points of size distribution solution

SplineKnotWave [n+2d] wave of knot values needed to calculate B-splines

precalcBdata [m] vector of [extrapolated] measured signals (raw data)

Buncertainties [m] vector of uncertainty/error values associated with [extrapolated] measured signals (raw data)

InvSolnMethod string to call solution computation algorithm

Functions Called:

cSplineInv (...)

cubicBspline (...)

Output:

$dN/d \ln D_p [\text{cm}^{-3}]$ size distribution inverse solution(s) in “invsoln_XXX” waves

D_p [m] corresponding particle diameters in “dpwave_XXX” waves

cSplineInv (AmatrixInput, SplineCoeffKnots, SplineControlptKnots, SplineKnotWave, ModelWaveOutput, ModelDataBinning, BvectorInput, Bvector_uncertainties, SolnMethod)

Solves the inverse problem with cubic B-spline setup.

Input:

AmatrixInput [m x ncoeff] kernel matrix, describing measurement collection

SplineCoeffKnots [ncoeff=n+d-1] binning for coefficients of size distribution solution; number of points must match y-dimension of Kmatrix

SplineControlptKnots [n] binning for control points of size distribution solution

SplineKnotWave [n+2d] wave of knot values needed to calculate B-splines

ModelWaveOutput [any] wave to hold size distribution solution; any number/spacing

ModelDataBinning [any] binning for size distribution solution; number of points must match ModelWaveOutput

BvectorInput [m] vector of measured signals (raw data)

Bvector_uncertainties [m] vector of uncertainty/error values associated with measured signals (raw data)

SolnMethod string to call solution computation algorithm

Functions Called:

NNLS (...)

IPG_TNNLS (...)

Regularization (...)

cubicBspline (...)

Output:

solution in "ModelWaveOutput"

cubicBspline (SplineCoeffWave, SplineControlptKnots, SplineKnotWave, SplineOutputBin, cSplineOutputWv)

Evaluates cubic spline interpolation, given coefficient values from an inverse solution.

Input:

SplineCoeffWave [ncoeff=n+d-1] coefficients of cubic spline size distribution solution

SplineControlptKnots [n] binning for control points of size distribution solution

SplineKnotWave [n+2d] wave of knot values that define the cubic B-splines

SplineOutputBin [any] binning for size distribution solution; number of points must match cSplineOutputWv

cSplineOutputWv [any] wave to hold size distribution solution; any number/spacing

Functions Called:**FindKnotPt** (...)**BsplineC1** (...)**BsplineC2** (...)**BsplineC3** (...)**BsplineC4** (...)**dBsplineC1** (...)**dBsplineC2** (...)**dBsplineC3** (...)**dBsplineC4** (...)**Output:****solution** in “cSplineOutputWv”**FindKnotPt** (KnotPtWave, ControlPtWave, InterpFinePtWave)

Finds primary knot point to correspond to secondary (fine) knot point.

Input:**KnotPtWave** [n+2d] wave of knot values that define the cubic B-splines**ControlPtWave** [n] binning for control points of size distribution solution**InterpFinePtWave** [any] binning for size distribution solution**Output:****KnotPt****KnotPt_indx****B-spline** functional forms; first derivatives**dBsplineC1**(xi, xk, xk1, xk2, xk3, xk4)**dBsplineC2**(xi, xk, xk1, xk2, xk3, xk4)**dBsplineC3**(xi, xk, xk1, xk2, xk3, xk4)**dBsplineC4**(xi, xk, xk1, xk2, xk3, xk4)

DMA Transfer Function

The following functions together define the transfer function for the DMA measurement system.

KernelFcn_DMA ()

Evaluates the transmission probability through the DMA.

Input:

- Dpart** some measure of particle diameter
- ChanZ** measurement channel targeted mobility
- ChanV** measurement channel voltage setting
- bbeta** flow ratio
- delta** flow balance
- chargemax** maximum number of charges accounted for in kernel function
- TFformula** string to call form of transfer function

Functions Called:

- ZpCalc_DpDMA** (...)
- chargeprob** (...)
- TransferFcn_DMA** (...)
- diffLossMatrix** [][]
- eta_pen** (...)
- counteff_CPC3025** (...)

Output: omega value

TransferFcn_DMA (Zp, ncharges, ChanZ, ChanV, bbeta, delta, TFformula)

Calculates value of transfer function for particle with mobility Z_p when DMA is set to voltage ChanV, with associated targeted mobility ChanZ.

Input:

- Zp** particle mobility, with single charge
- ncharges** number of charges on particle
- ChanZ** measurement channel targeted mobility
- ChanV** measurement channel voltage setting

bbeta flow ratio

delta flow balance

chargemax maximum number of charges accounted for in kernel function

TFformula string to call form of transfer function

Functions Called:

omegaDMAcalc (...)

Output: omega value

omegaDMAcalc (sigmap, zzz, bbeta, delta, TFformula)

Calls one of the “omegaDMA_xxx” functions to calculate omega values.

Input:

sigmap dimensionless diffusion parameter

zzz dimensionless mobility parameter

bbeta flow ratio

delta flow balance

TFformula string to call form of transfer function

Functions Called:

omegaDMA_xxx (...)

Output: omega value

omegaDMA_stolz (sigmap, zzz, bbeta, delta)

Calculates the DMA diffusive transfer function, as developed by Stolzenburg.

Input:

sigmap dimensionless diffusion parameter

zzz dimensionless mobility parameter

bbeta flow ratio

delta flow balance

Output: omega value

omegaDMA_nd (zzz, bbeta, delta)

Calculates the DMA ideal, non-diffusing transfer function.

Input:**zzz** dimensionless mobility parameter**bbeta** flow ratio**delta** flow balance**Output:** omega value**omegaDMA_diff** (sigmap, zzz, bbeta, delta)

Calculates a Gaussian approximation to the DMA diffusive transfer function.

Input:**sigmap** dimensionless diffusion parameter**zzz** dimensionless mobility parameter**bbeta** flow ratio**delta** flow balance**Output:** omega value**chargeprob** (dp_m, echarges)

XcodeS3

counteff_CPC3025 (dp_m)**eta_pen** (dp_m, Leff, Q)**Functions Called:****diffusion_coeff** (...)**diffusion_coeff** (dp)**Functions Called:****mobility** (...)**mobility** (dp, q)**Interconversions between particle diameter, particle mobility, and DMA voltage setting****Functions to call parameters and constants**

Solution Computation Algorithms

Non-Negative Least-Squares Method

An active-set method can be employed as described by Lawson and Hanson (1987). Least-squares methods overdetermine the problem by making the number of unknowns less than the number of measurements, e.g., by representing the unknown distribution parametrically, or by using more measurements than the number of sizes at which the size distribution is determined. This can lead to false ease of convergence to a solution.

NNLS (AmatrixInput, BvectorInput, ModelWaveOutput)

Input:

AmatrixInput [m x n] kernel matrix, describing measurement collection

BvectorInput [m] vector of measured signals (raw data)

ModelWaveOutput [n] wave to hold size distribution solution; number of points must match y-dimension of AmatrixInput

Output solution in “ModelWaveOutput”

Interior-Point Gradient Method

The interior-point gradient method for large-scale totally non-negative least squares problems implements the work of Merritt and Zhang (2005). Uncertainties are used in this method to identify a sufficiently good solution, i.e., to calculate χ^2 . Whereas the non-negative least-squares method requires matrix factorizations and updates, gradient-type methods only require matrix-vector multiplications. This makes gradient-type methods the method of choice for very large-scale applications when high-accuracy solutions are not necessary. Potential inaccuracies with the interior-point algorithm are due to the internally calculated barrier function keeping iterates away from inequality constraint boundaries.

IPG_TNNLS (AmatrixInput, ModelWaveOutput, BvectorInput, Bvector_uncertainties, ApproachParameter, MaxNumIterations)

Input:

AmatrixInput [m x n] kernel matrix, describing measurement collection

ModelWaveOutput [n] wave to hold size distribution solution; number of points must match y-dimension of AmatrixInput

BvectorInput [m] vector of measured signals (raw data)

Bvector_uncertainties [m] vector of uncertainty/error values associated with measured signals (raw data)

ApproachParameter “step” - needs to be smaller than 1, usually 0.6 is good; reasonable range seems to be 0.3-0.99

MaxNumIterations limit to sensible number; depends on complexity of problem

Output solution in “ModelWaveOutput”

Regularization

Another class of algorithm uses constrained regularization of the inverse problem (e.g., Wolfenbarger and Seinfeld 1990). Regularization selects the smoothest solution subject to fitting the measured data. The goodness of fit criteria χ^2 (sum of squared standardized residuals) must be close to the number of measured data points used in the analysis, subject to an additional constraint of maximizing the entropy or smoothness of the solution. This imposes a high standard for the reported errors on the measured data. If errors are not correct, it is likely that artifacts in the derived size distribution will result.

Regularization algorithms allow a free form for the instrument response matrix, but require a careful choice of constraining parameters and weights of penalty functions. Literature examples include generalized cross-validation for selecting the regularization parameter (Crump and Seinfeld 1982); the L-curve method; and the discrepancy principle (which provides a solution that matches the measurements to just within expected experimental error and is computationally simple, with the advantage of being very time-efficient; its disadvantage is over smoothing the solution, corresponding to a very large value of the regularization parameter, lambda).

Regularization (AmatrixInput, ModelWaveOutput, BvectorInput, Bvector_uncertainties)

Input:

AmatrixInput [m x n] kernel matrix, describing measurement collection

ModelWaveOutput [n] wave to hold size distribution solution; number of points must match y-dimension of AmatrixInput

BvectorInput [m] vector of measured signals (raw data)

Bvector_uncertainties [m] vector of uncertainty/error values associated with measured signals (raw data)

Internal Calculations

MakeHmatrix ()

H_matrix [n x n] constraint matrix; here done for second derivative

CalculateBVector ()

B_vector [n] calculated from Kmatrix, Bvector, and errors

CalculateDMatrix ()

D_matrix [n x n] calculated from Kmatrix and errors

CalculateAmatrix(aValue)

A_matrix [n x n] = D[n x n] + a*H[n x n]

FindOptimumAvalue (Evaluate)

for the fitting itself; Evaluate is internal precision parameter; a lower value requires χ^2 to be closer to target

CalculateChisquared ()

chi squared sum of the difference value between the fit and measured data

RegularizationMinRatio = 1e-4

Regularization is forced to have at least this x max of SD in each bin to avoid negative values

Output solution in “ModelWaveOutput”

3.3 Discussion

Data Analysis Recommendations

In general, multiply charged particles will be sampled by the DMA that would have an electrical mobility less than the minimum mobility of singly charged particles that are classified by the DMA. For aerosols with small d_g and or small σ_g , the response of the DMA at this maximum mobility (higher voltage settings) goes to zero. If the response is not zero for such multiply charged, large particles, more information is needed before the inversion can proceed. A lack of super-micrometer size distribution information could lead to a drastic overestimation of the particle number size distribution in the upper size range of the mobility spectrometer.

Disturbances could only be completely avoided if a mobility spectrometer were able to measure the PNSD across the entire range until the concentration reaches zero at the upper end. Due to technical reasons, however, the range of a mobility particle size spectrometer cannot be extended far beyond $1\text{-}2\mu\text{m}$. One approach has been to extrapolate the measured electrical particle mobility distribution into larger diameters. This might be appropriate in the case of a continuously decreasing number concentration towards larger particles, but usually not when a significant coarse particle mode is present.

One possibility is to use another instrument to determine concentrations at these upper sizes (Collins et al. 2002). A second possibility is to install a device upstream of the DMA to remove all particles larger than that associated with the minimum classified mobility. This is often accomplished by using an impactor at the DMA inlet to exclude particles that are too large to be classified in their singly charged state. Just assuming the size distribution goes to zero at these upper sizes leads to truncation errors.

The smaller the maximum number of charges accounted for in the inversion analysis, the lesser the computational burden, but this can then lead to errors in the reconstructed particle size distribution. In searching for a point of compromise, Figure 3.3 reveals that limiting the number of charge states to 6 for a given inversion analysis has barely any effect on solution quality, but limiting to 3 causes noticeable errors in the size distribution solution. This analysis can, however, depend on the sampled particle size range, as larger particles can acquire considerably more charges.

Another important element of DMA data analysis is accurately quantifying the diffusion losses through a measurement system (as through inlet/outlets and connecting tubing). Figure 3.4 shows how over- or under-estimating the effective length values used to capture diffusion losses within kernel function calculations can quickly lead to errors in reconstructed particle size distributions.

Data Collection Considerations

System operating conditions affect collected and recovered data. For example, flow rates and resolution affect the measurement range. The size range of interest needs to be identified in order to design data collection using the correct instrument or combination of instruments. Operational parameters also influence the number and location of measured data points, as would be determined by stepping or scanning through the measurement range. The impact of these parameters should be explored

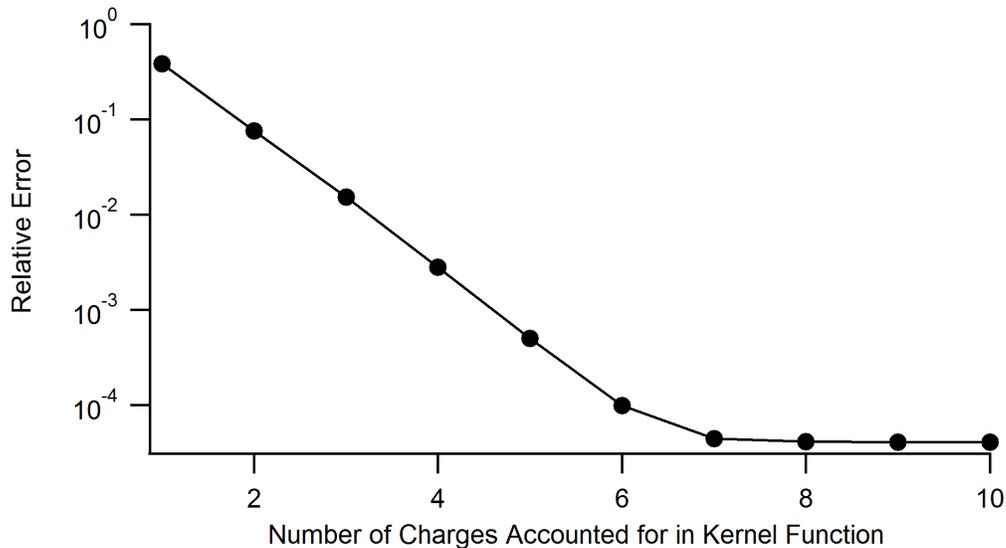


Figure 3.3: Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the number of charges accounted for in the multiple charge correction of the kernel function. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

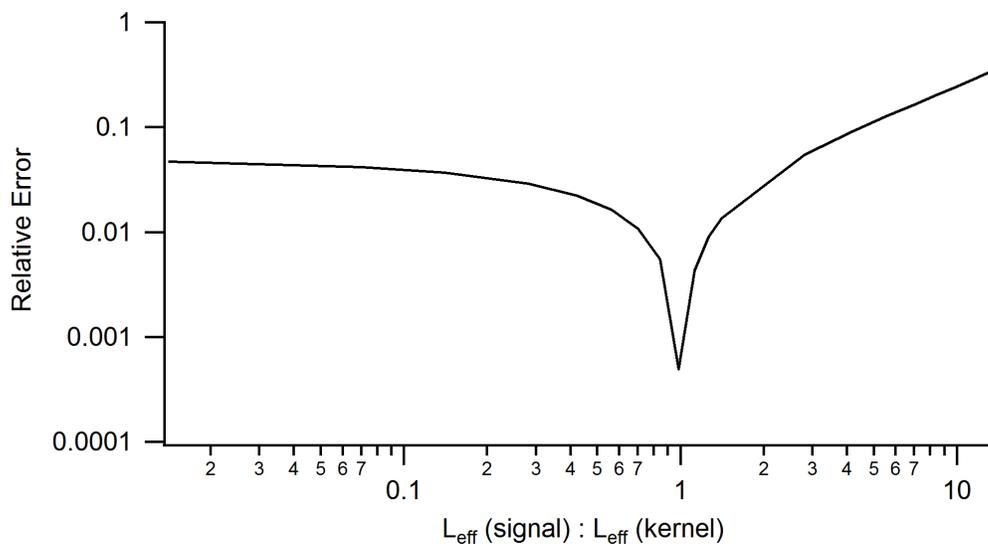


Figure 3.4: Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the ratio of the equivalent length value used for signal simulation to that used in calculating the inversion kernel matrix. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

as data collection design considerations.

For operation under given conditions, the accuracy of particle number size distribution recovery improves with increasing number of sample channels, until a number of channels is reached such that the diameter range of interest is covered by some nonzero values of the transfer function. This is reached with fewer channels for lower resolution. Very high resolution transfer functions leave gaps in the collected measurements if too few channels are employed. Changing system flow rates affects the resolution of particle classification, which in turn affects the coverage of the instrument transfer functions for a given number and spacing of sampling channels. Thus, the number and spacing of sample channels, and the system flow rates which determine sampling resolution, will affect the quality of the collected data.

Optimization of sample channel number and spacing is quantitatively explored in Figure 3.5, where a parameter referred to as the theta value has been defined. This parameter describes the amount of overlap in sampled particle mobilities from one measurement channel to the next. The theta value, θ , sets the ratio of width to spacing for measurement channels based on the ideal non-diffusing DMA transfer function (see Appendix A).

$$\theta = \frac{\text{nonzero width of } \Omega_{nd}}{\text{space between } Z_{p,i}^*} \quad (3.1)$$

For a given θ , the number of voltage steps to use for a given DMA simulation is then determined by modifying Eq. A.3 to the following

$$n_{channels} = \begin{cases} \left[\left[\frac{\ln\left(\frac{Z_{min}(1-\beta)}{Z_{max}(1+\beta)}\right)}{\ln\left(\frac{1-\beta}{1+\beta}\right)} \right] \times \theta - (\theta - 1) \right] & \text{if } \theta \geq 1 \\ \left[\left[\frac{\ln\left(\frac{Z_{min}(1-\beta)}{Z_{max}(1+\beta)}\right)}{\ln\left(\frac{1-\beta}{1+\beta}\right)} \right] \times \theta \right] & \text{if } \theta < 1 \end{cases} \quad (3.2)$$

where now the bracketed quantities are to be rounded to the nearest integer. Figure 3.5 suggests that setting sample channels with at least a bit of overlap (i.e., $\theta \approx 1.5$) may be more effective than either over- or under-sampling.

Computational Solution Method Comparisons

The final inverse solution depends on the mathematical optimization algorithm used. When counts are not extrapolated (underdetermined kernel matrix), the upper tail of distribution is more accurate when using NNLS than other solution methods because particles beyond upper measurement range are set to zero.

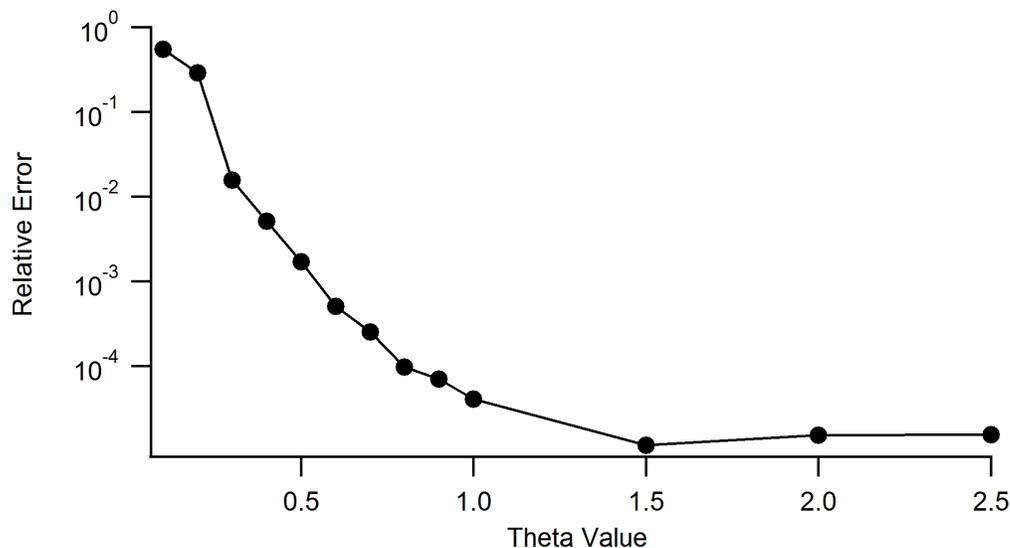


Figure 3.5: Relative error in reconstruction, e_R , for inversion analyses of the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002), shown as a function of the theta value, θ , used to define measurement channels. All inversion analyses were performed on noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

The zeroth-order regularization approach (i.e., discrepancy principle) tends to over-smooth the solution, corresponding to a very large value of the regularization parameter. This means it doesn't perform as well as the others if no measures of uncertainty are provided, because the solution is held to lower standards in terms of acceptable error.

The convergence behavior of these three numerical solution methods is explored in Figures 3.6 and 3.7.

3.4 Conclusions

Data analysis involves numerous decisions surrounding how to represent an instrument or measurement process accurately. The particle size distribution retrieval algorithm needs to be considered as integral to the overall sampling system. When evaluating analysis procedures, the computational effort also involves the amount of effort to implement the algorithm.

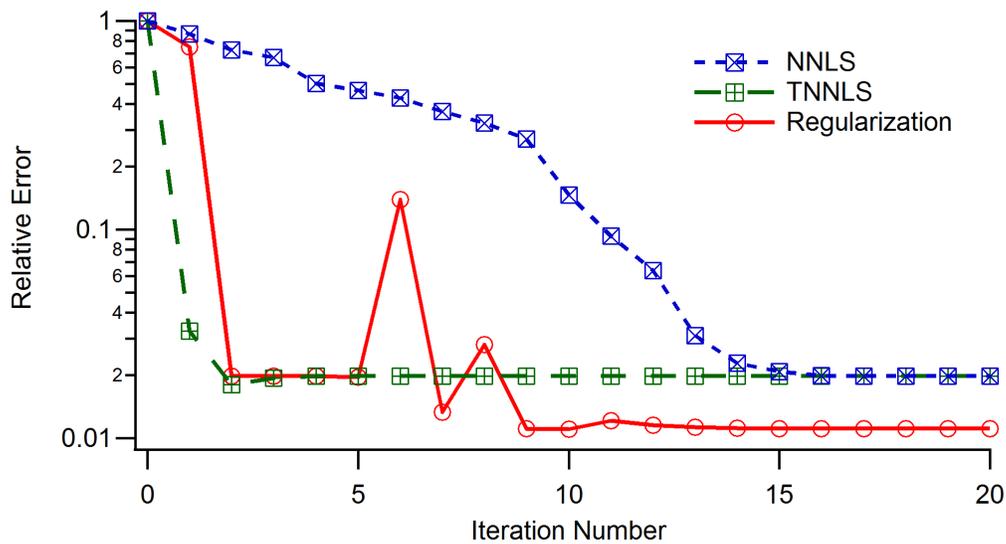


Figure 3.6: Convergence behavior for the NNLS, TNNLS, and Regularization optimization programs to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noise-free measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

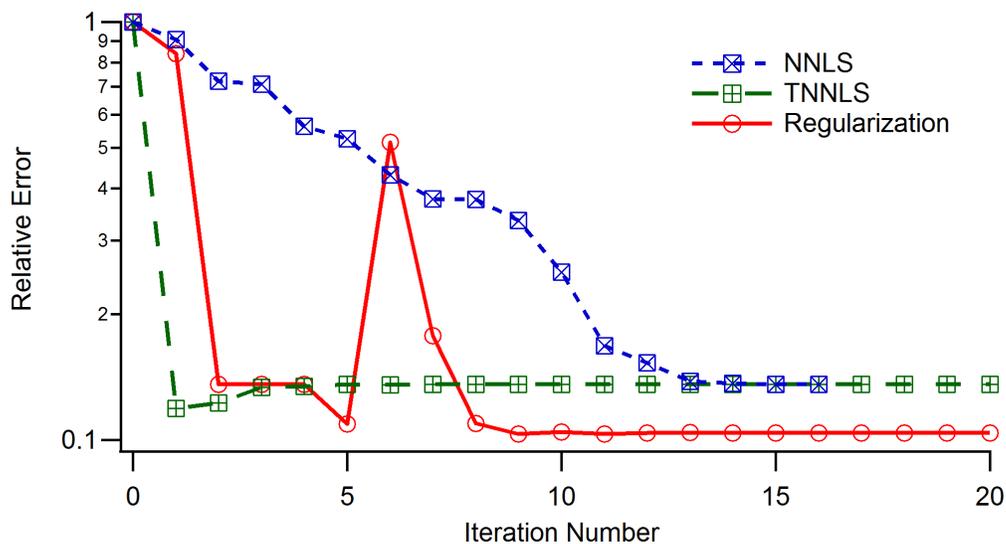


Figure 3.7: Convergence behavior for the NNLS, TNNLS, and Regularization optimization programs to recover the bimodal test-case aerosol presented in Russell et al. (1995) and Collins et al. (2002) from noisy measurements simulating high resolution DMA operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$).

Part II

**Particle Measurements and Health
Metrics**

*Chapter 4***PARTICLE MEASUREMENTS RELEVANT TO AEROSOL
HEALTH STUDIES I: EVALUATION METHODOLOGY**

Epidemiological evidence associates human exposure to airborne particulate matter (PM) with a variety of adverse health outcomes, influenced by the deposition of inhaled particles within different compartments of the human respiratory tract. A critical component of epidemiological studies is the quantitative determination of exposure and dose information, provided by relevant particle measurements. Yet not all particle measurements are optimized to inform rigorous assessments of PM health effects, and health studies may neglect to question the relevance of the characterization provided. The purpose of the present study is (i) to motivate researchers to evaluate health-related aerosol particle measurements in terms of estimation of regional lung dose metrics, and (ii) to demonstrate a procedure for such evaluation. We use synthetic data to demonstrate our methodology on two common sources of particle measurement data: PM_{2.5} samplers, and differential mobility analyzer (DMA) systems. Simulated measurements are generated for model aerosol exposure scenarios to provide input to a lung deposition model. We calculate total and regional deposited dose based on particle number, mass, and surface area concentration metrics. Dose values estimated based on simulated measurements are compared to assumed dose values calculated directly from the fully characterized input particle size distributions. The evaluation methodology presented here can be used to assess the performance of any existing, or proposed, measurement system in providing health-relevant metrics of human exposure and lung deposited dose.

4.1 Introduction

Particulate matter (PM) is an important component of air pollution and workplace exposures that has been associated with substantial health impacts. Numerous studies have established links between PM and such adverse health endpoints as cardiopulmonary morbidity and mortality, as well as reproductive and developmental effects. Links between PM and various adverse health endpoints are established primarily through epidemiological analyses that draw on available observational data to correlate health outcomes with PM exposure measures. Too often, the available

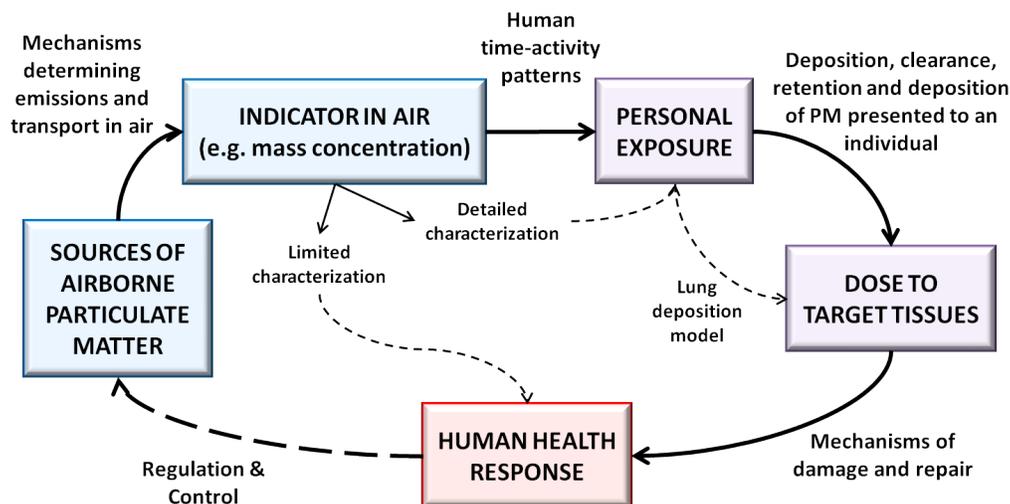
dose or exposure measures are limited in depth of characterization, or quantity, or both, and the incomplete data constrain the scope and impact of PM health-effect studies.

For example, compliance measurements are often referenced in epidemiological studies since they are designed to probe measures previously found to impact human health and welfare. Yet these measures must also be sufficiently robust for routine use, and they must be within the budget and operation capabilities of personnel employed by regulatory agencies. As such, these compliance measurements may confirm and elaborate upon associations that were expected, or perhaps even detected in earlier studies, but they are unlikely to extend understanding of the biological mechanisms underlying these associations. In contrast, research-grade instruments enable detailed resolution of aerosol particle characteristics, but the collection of such measurements is generally much more complex and expensive and requires unique skill sets for successful operation and analysis of data.

A general framework for the interplay of particulate matter research along the continuum from emission of a contaminant to a health effect is illustrated schematically in Figure 4.1. The outer circle represents the chain of events that serves as the conceptual basis for understanding and evaluating environmental and occupational health. The dotted-line connections shown within this circle hint at pathways for the flow of information provided by particle measurements. Limited particle characterization, as is widely provided by many regulatory metrics, is often used to draw direct connections to human health response, short-circuiting the Source-to-Health Effects paradigm. Meanwhile, the less accessible, more detailed resolution of particle characteristics, as provided by the complex instruments discussed above, would allow estimation of personal exposure and subsequent lung dose.

The Source-to-Health Effects paradigm of Figure 4.1 highlights the importance of accurately quantifying human PM exposure and dose within the context of creating a stable cycle of information to direct policy makers on specific sources of PM to target to effectively mitigate health effects. This applies to both occupational health and air pollution control efforts. The integrity of this cycle of information relies on the overlapping focus areas of exposure assessment and effects assessment. Exposure assessment involves investigating the emissions of particles from sources and examining the pathways by which particles approach and enter the body. Effects assessment then attempts to correlate an observed adverse health effect with dose or exposure measures. Thus, accurate health effects assessments require appropriate

Exposure Assessment



Effects Assessment

Figure 4.1: Source-to-Health Effects paradigm for investigating the health effects associated with airborne particulate matter. Research activities can be broadly categorized as exposure assessment (color-coded here in blue) or effects assessment (color-coded here in red). The overlap in these focus areas (color-coded here in purple) reflects the importance of exposure and dose information to both sets of activities. Adapted from Liroy (1990), Sexton et al. (1992), and the National Research Council (2004, p. 26).

exposure assessments. The overlap between exposure and effects assessment, shown in Figure 4.1, reflects the importance of exposure and dose information to both focus areas.

Aerosol particles present potential health impacts through ingestion, inhalation, or dermal contact, though it is the inhalation route that is of prime importance when considering aerosol measurement for exposure and dose information. Exposure measures are generally related to the probability of penetration of particles to the respiratory tract. A number of studies have recognized that measurements of exposure alone are imprecise for probing the mechanisms that underlie observed adverse health outcomes and measures of dose to different regions of the respiratory tract would be more effective (Esmen et al. 2002). This is because PM health effects are associated with the human body's biological response to particles deposited in the respiratory tract during breathing, and not all inhaled particles deposit within the

respiratory tract. Many inhaled particles remain suspended in the air to be exhaled without depositing. Those that are retained deposit in different regions of the airways depending upon their size. Exposure is only relevant if deposition is known; dose is more important, i.e., what quantity of particulate matter deposits within the respiratory tract, and where it deposits.

In practice, regionally deposited dose is very difficult to evaluate experimentally. Respiratory dose depends on two factors: (i) the properties of the atmospheric or workplace aerosol; and (ii) the way those properties determine where and with what efficiency particles deposit in the respiratory tract. Real-life analyses rely on measurements to infer the characteristics of airborne particles that influence lung dose and subsequent health risks. This information is used to estimate dose values using mathematical models based on present understanding of regional deposition patterns (ICRP 1994; Vincent 2005). In this way, particle measurements introduce a major complexity in linking PM to health (Russell and Brunekreef 2009). Limitations or biases in these measurements will affect the conclusions drawn in PM health effect studies; insufficient information may conceal major effects.

As mentioned above, much of the data cited for PM exposure assessments in air pollution epidemiology studies are regulatory measures, such as $PM_{2.5}$ and PM_{10} , the mass concentrations of particles with a fifty percent cutoff diameter at $2.5 \mu m$ and $10 \mu m$, respectively (Brunekreef and Forsberg 2005; Pope and Dockery 2006; Vedal 1997). These measures were developed as compliance metrics in response to concerns about particles in the ambient atmospheric environment associated with adverse health outcomes in the general population (Vincent 2005). $PM_{2.5}$ and PM_{10} data are widely measured to document compliance with the National Ambient Air Quality Standards (NAAQS), making these measures of exposure available for epidemiological studies. However, these single cut, mass-based metrics provide limited particle characterization and do little to help guide efforts to understand the deposition, fate, and effects of inhaled particles.

At the other end of the measurement spectrum, in terms of operational complexity and data resolution, are state-of-the-art aerosol mobility analyzers capable of recovering sufficient information to calculate good estimates of various physical metrics of deposition dose (Oberdörster et al. 2005). A number of researchers have demonstrated the value of the information provided by these instruments to health effects research in occupational settings (Elihn et al. 2011) as well as in hot spots of atmospheric exposure (e.g., near roads with busy traffic) (Kozawa et al.

2009; Zhu et al. 2002, 2004, among others). Because ultrafine particles deposit primarily by Brownian diffusion (Rissler et al. 2012), and the electrical mobility of an aerosol particle is proportional to its diffusivity, mobility analysis measures the property that determines where and at what rate particles will deposit in the airways. However, few health effects studies can employ such large, expensive, and complex instruments. This begs the question: How good must a measurement be to represent an adequate exposure assessment for health studies?

Whether the concern is environmental or occupational, there is an unmet need for health effects researchers to answer this question in a quantitative way. A number of researchers have assembled qualitative reviews of the various measurement strategies employed for occupational and environmental aerosol exposure assessments (Vincent 2005; Maynard and Pui 2007; Brouwer et al. 2009; Vincent, 2012). These reviews commentate on the exposure metrics and instrumentation currently available, and some even explore future requirements for technology yet to be developed. These reviews do not, however, evaluate existing or proposed aerosol exposure measurement approaches in a quantitative manner. The present study outlines a procedure for carrying out such quantitative evaluations.

Our evaluation method involves a series of calculations that can be applied to any given measurement method to assess its suitability for health effects research. These calculations ultimately quantify how the particle characterization provided by a measurement affects the conclusions drawn about particulate dose delivered to the human respiratory tract. This is rooted in the concept that, as noted above, exposure is only relevant if deposition is known; dose is more important.

Previous investigators have employed numerical models of lung deposition to calculate values of respiratory dose from experimental descriptions of aerosol properties (Alföldy et al. 2009; Elihn et al. 2011; Kristensson et al. 2013). We augment this approach by comparing the dose estimates based upon experimental measurements with those that would actually occur, given knowledge of the ambient aerosol that is not biased by limitations of the measurement methods. This comparison to a “true dose” creates the opportunity for quantitative assessments. To ensure fair assessments and inter-comparisons of measurement strategies with potentially very different operating principles, we employ ideal models of the methods under consideration to simulate experimental measurements. By relying upon synthetic data in this way, we can control for any uncertainties associated with the measurement method under consideration.

We describe two example applications of our evaluation procedure to demonstrate the performance of the most common measurement techniques and begin to set the stage for the future of health-relevant particle measurements. We evaluate the regulatory metric of $PM_{2.5}$ as well as the more involved measurements provided by differential mobility analyzer (DMA) systems. The operational theories between these two measurement systems are well-known, and they happen to represent the extreme ends of the spectrum in terms of operational complexity and data resolution. Our intent is not to dwell on the strengths and weaknesses of these two measurement systems, but simply use them as examples to demonstrate the important role that particle measurements play in assessing health effects. Other measurement options are available, with more middle-ground capabilities; the evaluation methodology described herein can and should be applied to them.

4.2 Methods

Our measurement evaluation strategy is illustrated in Figure 4.2. From an unbiased description of input aerosol, we employ an accepted model of lung deposition to calculate the regional particle deposition and call this the “true dose.” At the same time, a chosen measurement technology is modeled, including measurement uncertainties and statistics, to produce raw signals comparable to those obtained in actual measurements. The resulting synthetic data are analyzed using the same methods commonly applied in air quality studies or aerosol exposure measurements to infer the nature of the aerosol. The same model of lung deposition is then applied to the analyzed synthetic data to estimate the regional dose. The comparison between this estimate and that which we calculated as the “true dose” reveals the efficacy and biases of the measurement technology. We consider each component of this analysis separately in the discussion that follows.

Particle Deposition in the Respiratory Tract

The human body responds to particles deposited in the respiratory tract during breathing. Various health outcomes (e.g., sinusitis versus bronchitis versus pneumonia), and their underlying biological mechanisms, are influenced by particle deposition within different compartments of the human respiratory tract. Knowledge of this regional deposition would allow health effect studies to explore the pathways by which PM exposures cause disease.

Lung deposition is governed by a complex set of parameters, including breathing pattern, particle characteristics, flow dynamics, and morphological structure of the

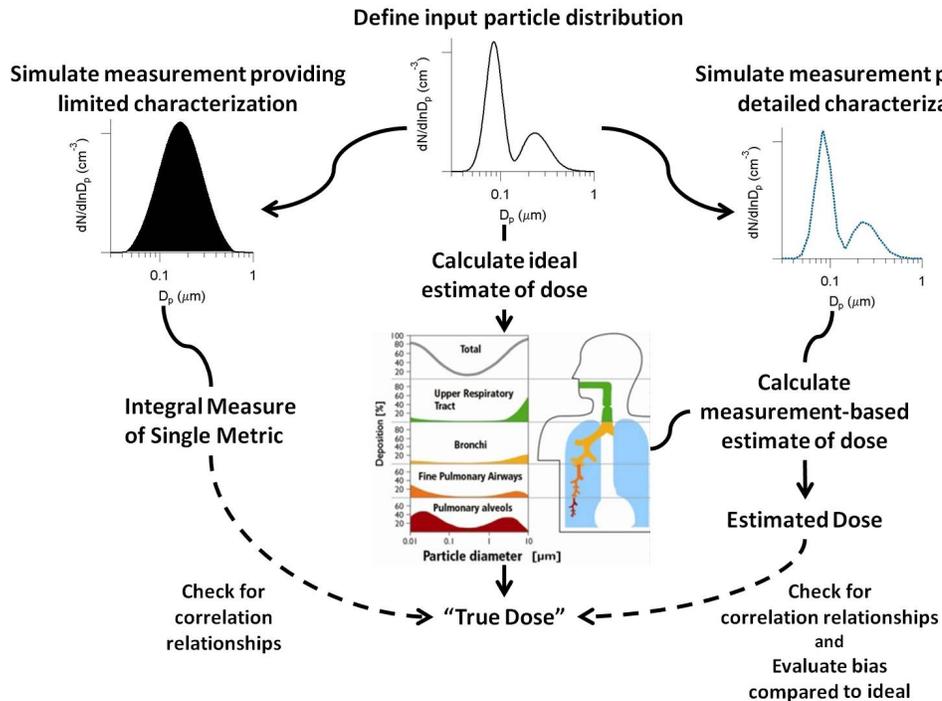


Figure 4.2: Procedural flow illustration for evaluating measurement performance. Define an input particle distribution, simulate a given measurement scheme, and process the recovered particle characterization to estimate dose to the respiratory tract. From the completely defined input, an ideal estimate of dose can also be calculated. By comparing the information provided by simulated measurements to the ideal estimates of dose, this evaluation procedure can reveal limits or biases in measurement data that may constrain the scope and impact of PM health-effect studies.

lung. Experimental challenges make studies of regional deposition in actual human subjects difficult and prone to a lack of coherence in reported data. Experimental studies are often complemented by mathematical and numerical modeling to create accessible tools for predicting particle deposition in the respiratory tract. The International Commission on Radiological Protection (ICRP) developed an empirical model to predict size-dependent particle penetration and deposition within different anatomical regions of the human lungs (ICRP 1994). The ICRP reviewed experimental data from many laboratories in several countries to create a well-established framework for deposition calculations. We employ the ICRP model for our analysis here.

The ICRP model evaluates the fraction of inhaled particles of a given diameter ($1 \text{ nm} < D_p < 100 \text{ μm}$) that deposit in different regions of the respiratory tract,

accounting for effects of gender, age, and breathing characteristics. For its calculations, the model considers the respiratory tract as four anatomical regions as illustrated in Figure 4.3: (1) the extrathoracic region (ET), comprising the anterior nose (ET1) and the posterior nasal passages, larynx, pharynx, and mouth (ET2); (2) the bronchial region (BB), comprising the trachea through the bronchi; (3) the bronchiolar region (bb), comprising the bronchioles and terminal bronchioles; and (4) the alveolar/interstitial region (AI), comprising the respiratory bronchioles, the alveolar ducts and sacs with their alveoli, and the interstitial connective tissue.

The deposition efficiency curves for these regions, shown in Figure 4.3, reveal distinct modes of particle deposition within the various regions of the human respiratory tract. Large particles, $D_p > 1 \mu\text{m}$, deposit most efficiently in the extrathoracic regions due to their inertia, though a small fraction of supermicron particles, $D_p < 5 \mu\text{m}$, penetrate into the alveolar region where they deposit by gravitational sedimentation. Smaller particles penetrate through the head airways and deposit in the bronchial, bronchiolar, and alveolar-interstitial regions, primarily by Brownian diffusion. For the smallest particles, $D_p < 10 \text{ nm}$, diffusional deposition even occurs within the head and neck airways. This filtering prevents these smallest particles from penetrating to the deepest regions of the lung and leads to a distinct peak in the alveolar deposition efficiency at $D_p \sim 10\text{--}100 \text{ nm}$. Less than 20% of particles 150–500 nm in size are retained for deposition anywhere within the respiratory system. Particle size and the complex mechanisms of aerosol deposition control the deposition location of particles within the respiratory tract, which in turn determines their health impacts. Understanding the regional deposition of inhaled particles would aid further investigations into the underlying biological mechanisms of different size fractions of PM exposure. We do not address specific health impacts or their underlying biological mechanisms here, but rather focus on our ability to assess dose to the different regions in order to inform health effect studies.

For each respiratory tract region considered here, we estimate particle deposition in terms of number, mass, and surface area concentrations. Since the most effective measure of dose for evaluating a particular adverse health outcome remains unclear, we incorporate calculations for all three of these physical metrics into our measurement evaluations for a broad performance assessment.

Mathematically, the deposited dose of a certain particle size-fraction ($D_{p1} - D_{p2}$)

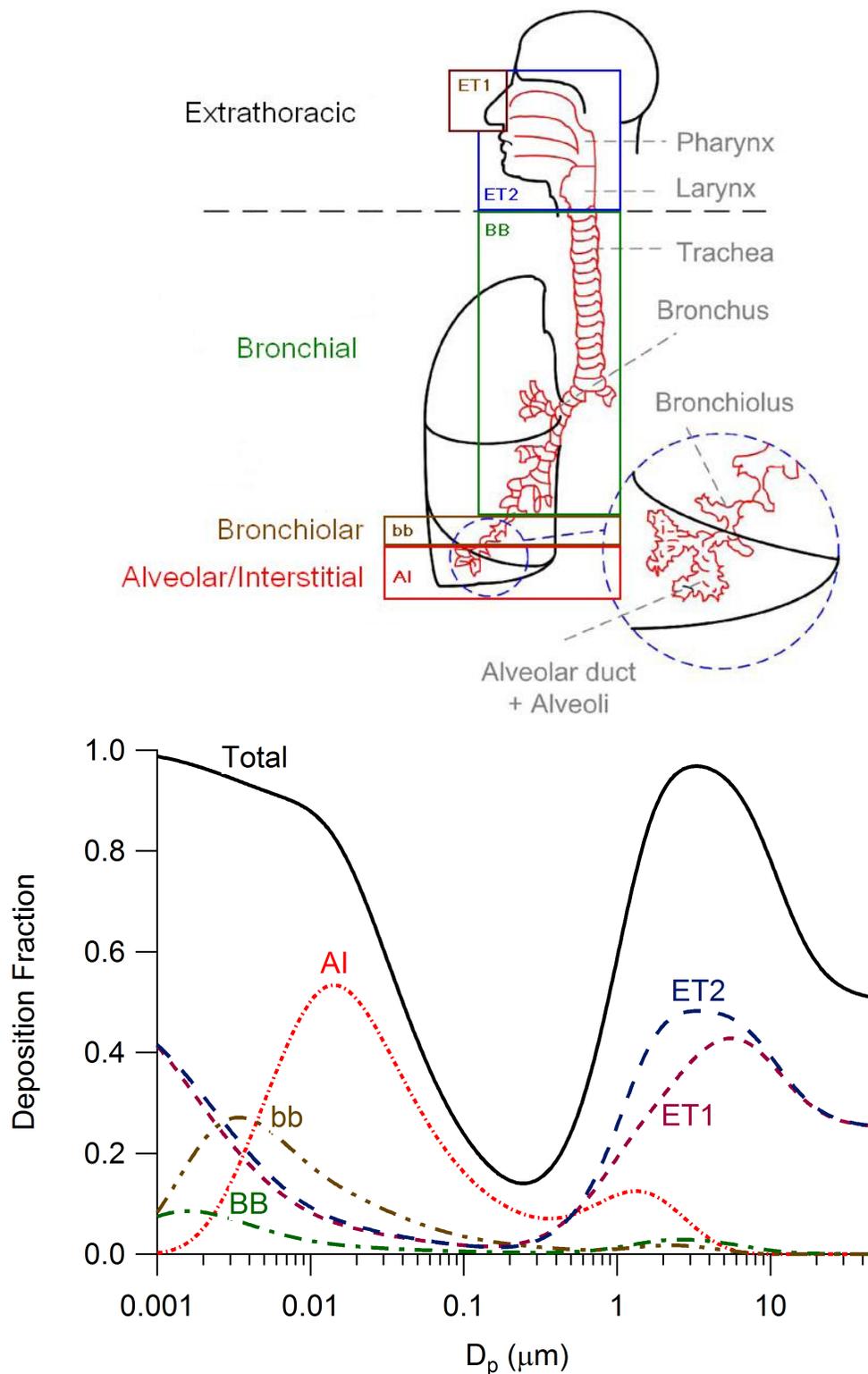


Figure 4.3: Anatomical regions of the human respiratory tract, shown schematically along with deposition efficiency curves describing the fraction of inhaled particles of a given diameter that deposit in the different lung regions (ICRP 1994).

can be described as

$$Dose_{\nu} = \int_{t_1}^{t_2} \int_{D_{p1}}^{D_{p2}} V_E \cdot \eta_{dep} \cdot n_N \cdot \nu \cdot d \ln D_p \cdot dt \quad (4.1)$$

where V_E [m^3/hr] is the ventilation rate, or the volume of air inhaled, η_{dep} [—] is the deposition fraction of aerosol particles in the respiratory system, and $n_N = dN/d \ln(D_p)$ [particles/ m^3] is the particle number size distribution. Both η_{dep} and n_N are functions of $\ln(D_p)$ where D_p is the particle diameter. Also a function of particle diameter, ν describes which dose metric is being evaluated, i.e., number (N ; $\nu = 1$), surface area (S ; $\nu = \pi D_p^2$), or mass (M ; $\nu = \frac{\pi}{6} D_p^3 \rho_p$). For present purposes, particles are assumed to be spherical and to have specific gravity of 1.4, which is typical of secondary organic aerosols that comprise much of the atmospheric aerosol (Ahlm et al. 2012).

Equation (4.1) reveals the pieces of information needed for our analysis. We first define a “reference worker” to receive our calculated dose values, using parameters matching those established by Phalen (1999). These parameters are listed in Appendix B. We use the ICRP model with these parameters to evaluate the deposition fraction of aerosol particles in the respiratory system, η_{dep} [—]. We evaluate the dose from Eq. (4.1) using the ventilation rate of $V_E = 1.5 \text{ m}^3/\text{hr}$ over different exposure time periods $\Delta t = t_2 - t_1$. Daily, or 24-hour, average exposure is commonly examined in epidemiological studies (Vedal 1997). Studies have also shown important health impacts associated with shorter averaging periods, e.g., 10 minutes and one hour (Peters et al. 2015). Most previous studies have assumed the input particle number size distribution, n_N , to be lognormal, or consist of multiple lognormal modes. We will take a more general approach as described below.

Representative Input Particle Distributions

A number of studies have focused on exposures to specific particle populations of interest, such as diesel exhaust aerosols (Alföldy et al. 2009; Rissler et al. 2012), or airborne particles found in particular workplace environments (Elihn et al. 2011). These studies reveal isolated PM effects by excluding many other possible exposures. We seek to analyze a broad range of particle populations in order to resolve source- and time-related variations in PM exposure and dose. Other studies have approached this aim by employing size distribution data collected as continuously monitored field measurements to evaluate regional lung deposition of ambient aerosols (Hussein et al. 2013; Kristensson et al. 2013). While this approach provides the natural

range of time-evolving particle populations we seek to consider, particle measurements from the field are subject to truncation and counting efficiency biases. Thus, measured size distributions are limited by the incomplete coverage of instruments.

We seek unbiased exposure scenarios as input to our evaluation methodology and thus define synthetic sets of test particle size distributions. We generate two categories of test aerosol particle size distributions for input in order to approximate both atmospheric particle populations and occupational aerosol types.

Atmospheric Particle Populations

A wide range of particle populations exist in the atmosphere that may be inhaled and deposited in the lungs. Relevant particles span orders of magnitude in size, from “inhalable coarse particles” $2.5 - 10 \mu\text{m}$ in diameter, to “ultrafine” particles $< 0.1 \mu\text{m}$ in diameter. Within a given air basin, exposure varies dramatically with time and location due to local and intermittent sources as well as transients associated with the diurnal air quality cycle. We are interested here in demonstrating appropriate particulate matter exposure and dose analyses for a representative range of atmospheric particle populations.

Atmospheric nucleation bursts are common, well studied, and include a spectrum of fine particles that extends to sizes below those that can routinely be probed. For our analysis, we simulate the time evolution of an aerosol particle population during such a new particle formation event. The nucleation event chosen for our analysis is typical of many that occurred in the highly polluted atmosphere of New Delhi during October and November 2002 (Mönkkönen et al. 2005). We use the Ion-UHMA model (University of Helsinki Multicomponent Aerosol model for neutral and charged particles; Leppä et al. 2009) with input parameters from air quality data to reproduce the observed aerosol evolution of the particle populations for a 24 hour period with 10 minute time resolution, as is shown in Figure 4.4. The original simulation covers a diameter range of $1.6\text{nm} < D_p < 1 \mu\text{m}$ using 240 size sections spaced equally on logarithmic scale. For a fair challenge of $\text{PM}_{2.5}$ measurements, in the present study we extend this size range to cover particle diameters up to $10 \mu\text{m}$ with an additional 85 size sections of the same spacing. We use linear extrapolation on a log-log scale to extrapolate the tails of the simulated distributions.

For the time-series simulation, the particle number size distribution is initiated as forming a background aerosol and then updated every ten minutes within the 24-hour period. Beginning at midnight, particle concentrations gradually diminish

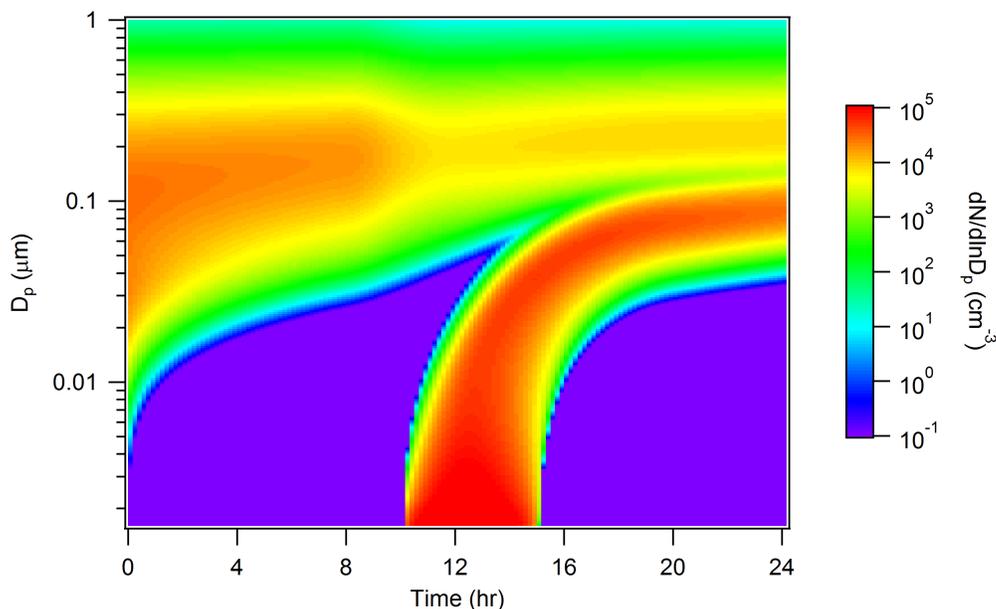


Figure 4.4: Particle number size distributions simulated using the Ion-UHMA model (Leppä et al. 2009) to reproduce an atmospheric nucleation burst typical of many that occurred in the highly polluted atmosphere of New Delhi during October and November 2002 (Mönkkönen et al. 2005).

between 07:00 and 11:00, due to the increase in the boundary layer height. By 09:00, the particle concentration, and concomitant scavenging of vapors has diminished to the point that homogeneous nucleation begins; new particles continue to form until about 15:00. This episode was selected as representative of recent extreme events that have occurred in a number of megacities because it includes exposures to both aged secondary organic aerosols and freshly nucleated particles. This single simulation day captures the natural evolution of ambient aerosols and resultant transient exposures in an extreme urban environment, and spans the entire fine and ultrafine particle size range. This allows us to evaluate biases in dose estimates evaluated for different exposure time periods.

Occupational Aerosol

We also analyze a myriad of aerosol size distributions representing particle size ranges reported in the literature for various occupational aerosol types. We generate a map of lognormal size distributions with mass median diameters (MMDs) ranging from 0.005 – 10 μm and geometric standard deviations (GSDs) ranging from 1.1 – 4. Total number concentration is held constant at 1000 cm^{-3} , and the distributions are evaluated over the diameter range 1.6 nm < D_p < 10 μm using 325 size sections

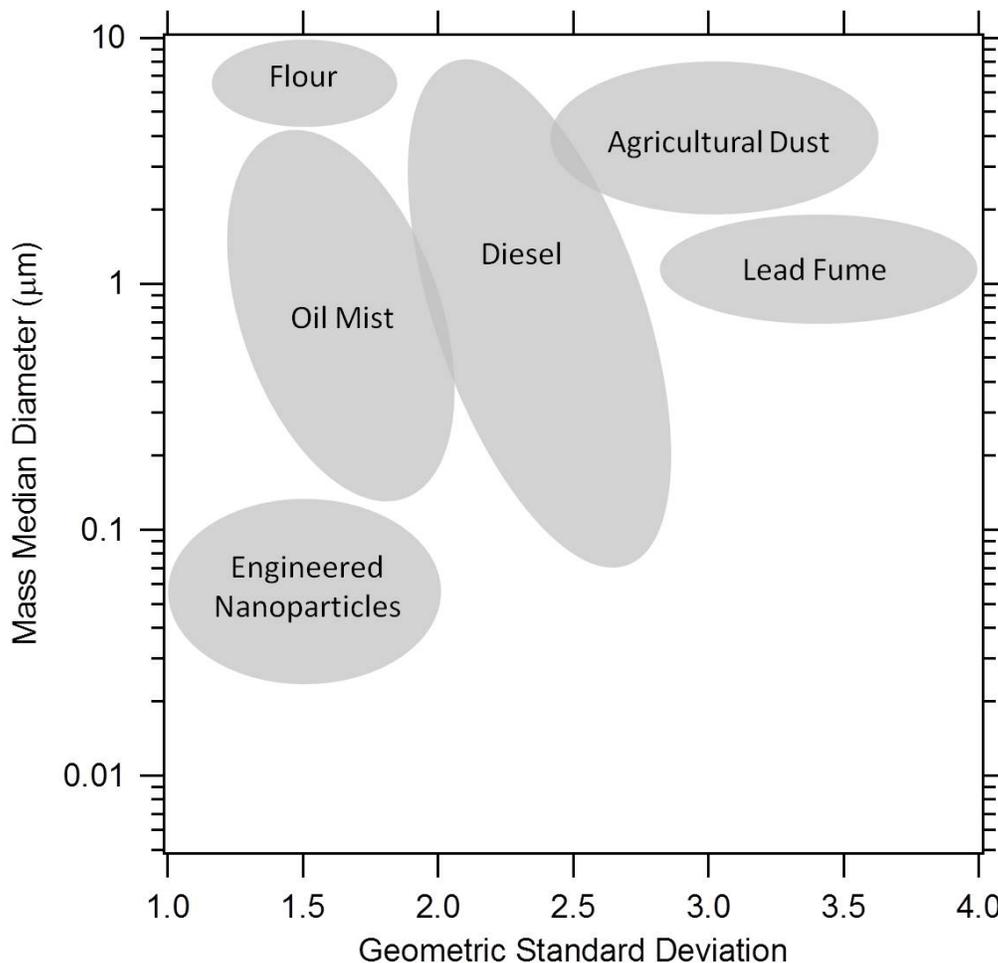


Figure 4.5: Map of the mass median diameter and geometric standard deviation parameter space for the lognormal size distributions simulated to represent occupational aerosol populations. Ovals represent particle size ranges reported in the literature for several occupational aerosol types.

spaced equally on logarithmic scale. This size range is defined to be consistent with that of the nucleation event simulation. From this map of input size distributions, we can create a map of corresponding dose bias values that could be used to predict a measurement system's performance for a given application. Figure 4.5 shows the MMD vs. GSD parameter space for the lognormal distributions analyzed in this study. Overlaid on the figure are shaded regions that represent common aerosol size distributions reported in the literature.

Particle Measurement Simulations

We focus on the performance of two, quite different, measures to demonstrate our measurement analysis strategy: the regulatory $PM_{2.5}$ metric and particle size dis-

tribution information recovered from a differential electrical mobility classification system. In this way, we explore the left- and right-hand sides of Figure 4.2, as we consider an example of a measurement that provides a single piece of information integrated over size and time and progress from there to a measurement that provides more detailed resolution with respect to size and time. These measurement schemes are sufficiently well understood for the *in silico* experiments to be applied.

PM_{2.5} Sampler

To evaluate PM_{2.5} measurements, we model a sampler that follows the particle size-selective criterion described by Vincent (2005) for PM_{2.5} in which

$$\eta_{\text{PM}_{2.5}} = 1 - F(x) \quad (4.2)$$

where $F(x)$ is the cumulative probability function of a standard normal variable (x) given by

$$x = \frac{\ln\left(\frac{d_{ae}}{\Gamma_x}\right)}{\ln(\Sigma_x)} \quad (4.3)$$

in which $\Gamma_x = 2.5 \mu\text{m}$ and $\Sigma_x = 1.50$. The result is that the function reaches 0.5 at $d_{ae} = 2.5 \mu\text{m}$. This criterion for PM_{2.5} is shown in Figure 4.6, together with two particle mass size distributions representative of the challenge aerosols in our analyses.

To simulate the measured response of a PM_{2.5} sampler, then, we calculate the integral

$$R_{\text{PM}_{2.5}} = \int_0^\infty \eta_{\text{PM}_{2.5}} \cdot n_M \cdot d \ln D_p \quad (4.4)$$

where $n_M = dM/d \ln(D_p)$ [$\mu\text{g}/\text{m}^3$] is the particle mass size distribution, calculated from an input number size distribution, n_N , as $n_M = \frac{\pi}{6} D_p^3 \rho_p n_N$. We include an estimate of systematic uncertainty for our simulated PM_{2.5} measurements, calculated as $\pm 10\%$ of the measured concentration. This is based upon what is stated by the EPA in Section 2.3.1.1 of Appendix A to Part 58 for PM_{2.5} methods: “The goal for acceptable measurement uncertainty is defined as 10 percent coefficient of variation (CV) for total precision and plus or minus 10 percent for total bias.”

DMA System

Differential mobility analysis is a powerful method for determining particle size distribution.

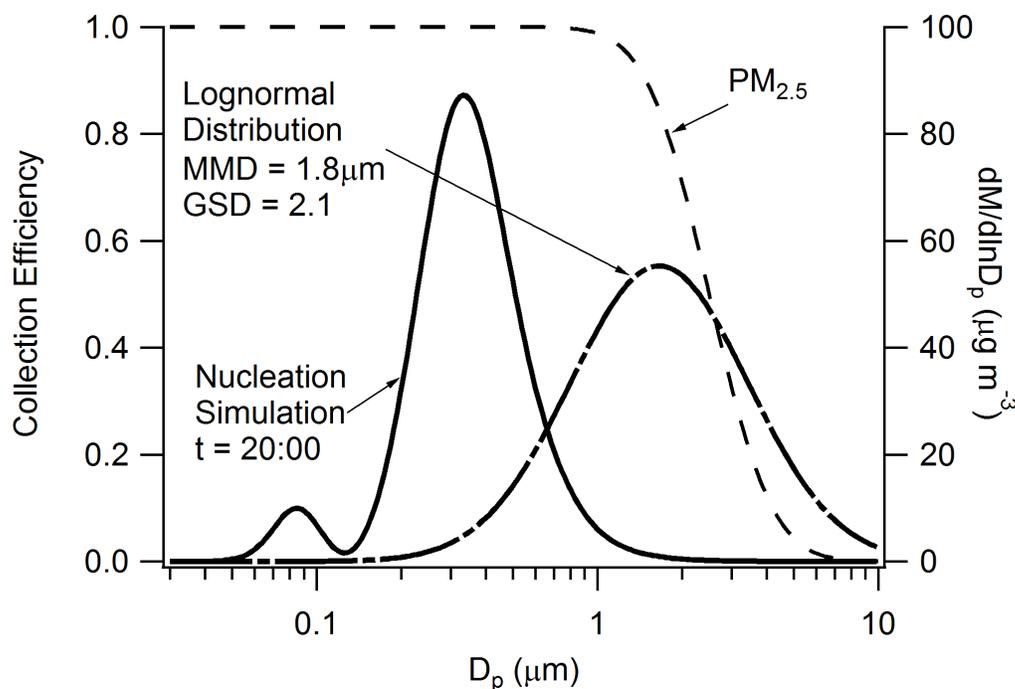


Figure 4.6: Particle size selective criterion for a $PM_{2.5}$ sampler, shown with two example particle mass size distributions. Essentially the entire mass of the size distribution from the nucleation simulation will be captured, while some of the lognormal distribution will be lost.

A DMA system operates by charging particles in an aerosol sample, typically by passing them through a charge conditioner that generates a bipolar ion cloud using a radioactive source. The sample then passes through a well-defined electrostatic field. Charged particles in that sample are induced to migrate across a parallel flow of particle-free air by an applied electrical field. Particles with a specific electrical mobility are sampled from the exit of the electrodes, and counted. By scanning or stepping the voltage between the electrodes, particles with electrical mobilities corresponding to a range of particle diameters can be counted sequentially (typically using a condensation particle counter, or CPC), allowing the aerosol size distribution to be determined.

To evaluate DMA measurements in our study, we model the performance of the long column DMA (e.g., TSI Model 3081, TSI Inc., USA), which is widely used in the form of the so-called scanning mobility particle sizer (SMPS). We do not simulate scanning operation, however, as a number of additional challenges are introduced to the analysis of scanning DMA data by the slow response time of many commonly used CPCs. Without an accurate description of the nonunique relationship between

the time a particle exits a scanning-mode DMA and the time it is ultimately detected, inverted distributions will be distorted. The simulations presented here consider stepping the DMA voltage instead, which avoids the additional distortions. Some CPCs do indeed have fast enough response times that SMPS measurements closely approximate stepping-mode DMA operation.

Our model system is considered to consist of a sequential setup of a bipolar diffusion charge conditioner, a long column DMA ($L = 444$ mm, inner radius $R_1 = 9.38$ mm, outer radius $R_2 = 19.58$ mm), and a CPC. To approximate the present state-of-the-art operation of the DMA, typical of present-day, high-resolution measurements with this technique, all flows are considered balanced and strictly laminar, with constant aerosol inlet and sample outlet flow rates of $Q_a = Q_s = 1$ lpm, constant particle-free sheath and particle-laden excess flow rates of $Q_{sh} = Q_{ex} = 10$ lpm, and a detector flow of $Q_{det} = 0.03$ lpm through a TSI Model 3025A Ultrafine CPC.

Measurements are simulated according to the principles described by Wiedensohler et al. (2012). The relationship between the measured responses and the sampled aerosol size spectrum is given by

$$R_i = \int_0^{\infty} n_N (\ln D_p) \Psi (i, D_p) d \ln D_p, \quad i = 1, 2, \dots, I \quad (4.5)$$

where R_i is the response for a given mobility-determining voltage setting of the DMA, I is the total number of measurements made, $n_N (\ln D_p) d \ln D_p$ is the number concentration [cm^{-3}] of aerosol particles in the diameter range $\ln D_p$ to $\ln D_p + d \ln D_p$, and $\Psi (i, D_p)$ is the response of the i th measurement channel to a particle of diameter D_p .

The response of the DMA system is described by a semi-analytical transfer function that was derived by Stolzenburg (1988) (see also Flagan 1999), along with the charging probability from the bipolar diffusion charger (Fuchs 1963; Hoppel and Frick 1986; López-Yglesias and Flagan 2013; Wiedensohler 1988) and empirical, instrument specific transmission and counting efficiency curves. Incorporating these functions into a definition for $\Psi (i, D_p)$ gives

$$\Psi (i, D_p) = Q_a \sum_{\nu=1}^{\infty} W_{\nu} \eta (D_p) \phi_{\nu} (D_p) \Omega (\nu, i, D_p) \quad (4.6)$$

where Q_a denotes the aerosol volume flow rate, ν denotes the number of electric charges on the aerosol particle, W_{ν} denotes the factor relating the sensor response

to the flux [particles/sec] of particles carrying ν charges, $\eta(D_p)$ denotes the collection of empirical, instrument specific transmission and counting efficiency terms, $\phi_\nu(D_p)$ denotes the fraction of particles of diameter D_p carrying ν charges, and $\Omega(\nu, i, D_p)$ denotes the transfer function of the DMA, i.e., the probability that a particle of size D_p carrying ν charges will pass through the DMA when it is set at voltage V_i .

For our simulated DMA, we calculate responses at 36 voltage steps from 10 V to 10 kV, equally spaced on a logarithmic scale. This range of voltages extends from the minimum setting necessary for a DMA to provide reasonable resolution in size classification, up to the maximum setting before electrostatic breakdown occurs. We define the number and spacing of sample channels within this range such that the channels are equally spaced on a logarithmic scale for particle mobility, with overlap in the ideal triangular transfer functions to sample all mobilities within the measurement range. Figure 4.7 shows a graph of the response function $\Psi(i, D_p)$ for our simulated DMA. The regions where $\Psi(i, D_p)$ is nonzero indicate the range of aerosol sizes that contribute to the measured response R_i . The distinct waves of peaks demonstrate the effect of multiple charging on the response function. Multiple charging is not significant at the smaller sizes, but does become important at the larger sizes.

Extracting information from DMA-CPC measurements is one important example of the many inverse problems encountered in atmospheric remote sensing. The relationship between measured responses and the aerosol size spectrum, as defined in Eq. (4.5), results in a set of Fredholm integral equations of the first kind. With the response values R_i and response function $\Psi(i, D_p)$ known, the inversion problem amounts to solving this set of equations for the unknown sampled size distribution $n(\ln D_p)$. The governing equations and methods of inversion related to DMA measurements have been explored in detail in Chapter 2. We employ the routine developed and described there to solve any inverse problems here in this study.

We impose Poisson counting statistics on the simulated instrument particle counts to produce signals representative of what would be obtained using real instruments. To estimate the uncertainties in inverted size distribution results, we apply a Monte Carlo method. For this, we draw sets of instrument responses as random samples from the calculated instrument responses, assuming a Poisson distribution, and use these as input for the inversion algorithm. We repeat this procedure two hundred times, storing each inversion result, and then statistically evaluate the stored inversion

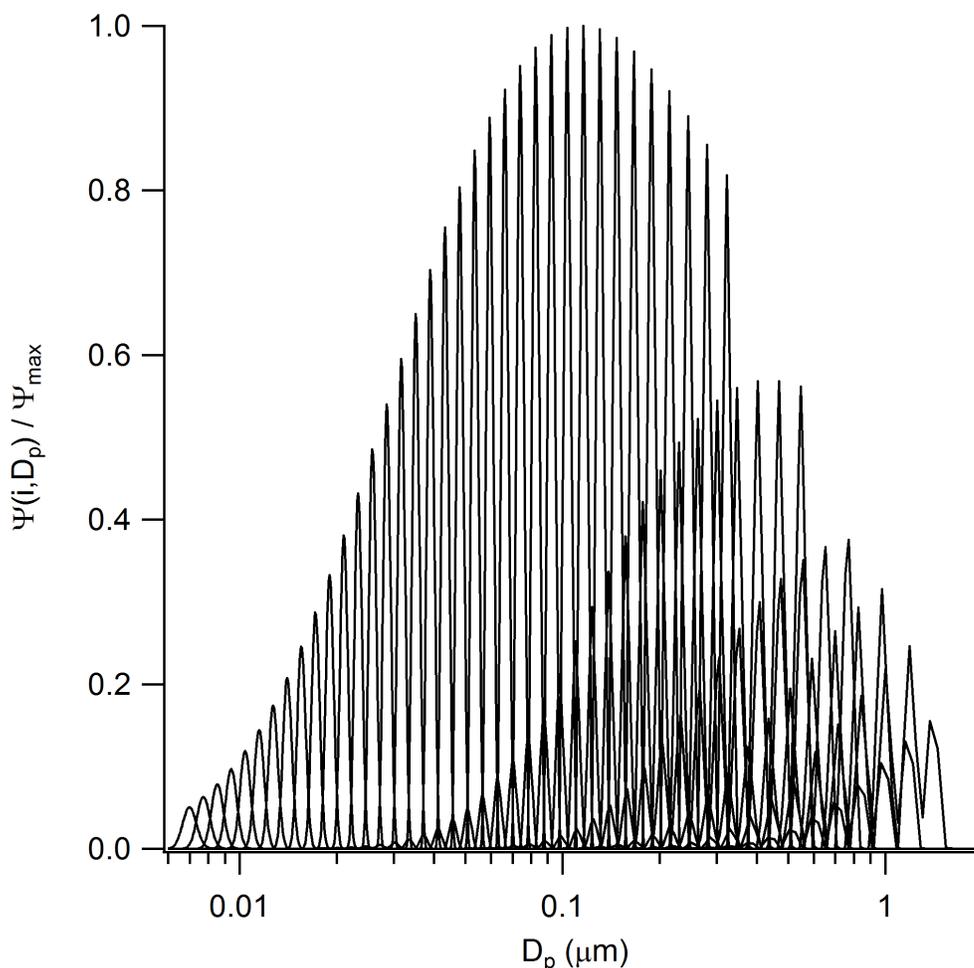


Figure 4.7: Normalized response functions for a stepping-mode DMA system, simulated to approximate typical high resolution operating conditions ($R = \frac{Q_{sh}}{Q_a} = 10$). The model system is set to operate with 36 voltage steps from 10 V to 10 kV, equally spaced on a logarithmic scale.

results to obtain a mean result and its uncertainty.

Quantitative Comparisons

Our measurement analysis strategy examines the utility of PM measurement schemes by comparing the information recovered from simulated measurements to the information calculated as “true dose.” As demonstrated in Figure 4.2, appropriate comparisons should be made with respect to the level of characterization provided by the measurement.

For measurements providing limited particle characterization (e.g., $PM_{2.5}$) it is ap-

appropriate to compare measured responses (e.g., values calculated from 4.4) directly to the “true dose” to uncover any possible correlation relationships. Though these measurements alone cannot offer insight into how different size particles may deposit in different anatomical regions of the respiratory tract, the information recovered may still correlate well with calculations of true deposited dose. It is possible to reveal that a measurement can be used as proxy for the deposited dose.

At the other end of the spectrum, for measurements providing detailed particle characterization, it is appropriate to apply lung deposition calculations to the full extent possible with the information recovered. With the size distribution information recovered from the DMA measurement simulations, we calculate total and regional deposited dose estimates based on particle mass, surface area, and number metrics, using the ICRP lung deposition model and Eq. (4.1). Then, for each combination of physical metric and lung region (i.e., mass dose to the alveolar region), we calculate the bias in the estimated dose value as the percent difference from the “true dose” value.

$$\% \text{ Bias} = \frac{\text{estimated dose} - \text{"true dose"}}{\text{"truedose"}} \times 100 \quad (4.7)$$

4.3 Results and Discussion

We calculate ideal estimates of hourly average particle dose to the extrathoracic, bronchial, bronchiolar, and alveolar/interstitial regions based on number, surface area, and mass metrics for the 24 hour exposure scenario shown in Figure 4.4, as well as for the map of lognormal distributions covering the parameter space of Figure 4.5. We compare these ideal dose estimates to the information recovered from the simulated measurements of PM_{2.5} and DMA systems. Figures 4.8 and 4.9 compare the hourly average mass, surface area, and number of particles deposited in the alveolar region with hourly average PM_{2.5} mass concentration exposures for the nucleation simulation and the lognormal distributions, respectively. The plots illustrate the regression relationships between the sets of data, with the correlation coefficient (i.e., r and not r^2) provided for each case. A 1:1 line is also included to quickly assess if data points are above, below, or straddling the 1:1 line. Ideal values are presented on the Y-axis, while the measured values are presented on the X-axis. Similar comparisons for other regions of the respiratory tract are shown in the supplementary information.

The true mass dose correlates well with the PM_{2.5} mass exposure (linear correlation coefficients, $r = 0.99$ and $r = 0.94$ for the time-series and lognormal distributions,

respectively). Small deviations from the perfect correlation occur for the time-series measurements when the atmospheric aerosol under consideration is in either the very early or very late hours of the simulation day. During both periods, the aerosol is an aged one, concentrated in the $50 \text{ nm} < D_p < 500 \text{ nm}$ size regime where many of the inhaled particles are exhaled rather than deposited. During the nucleation burst, from about 10:00 through 15:00, the mass deposited in the alveolar region is reasonably well correlated with $\text{PM}_{2.5}$.

Deviations occur in the positive direction for measurements of the lognormal distributions when the MMD of the sampled distribution falls in that same $50 \text{ nm} < D_p < 500 \text{ nm}$ size regime where many of the inhaled particles are exhaled rather than deposited. Deviations occur in the negative direction when the sampled distribution has a large enough MMD that the entire mass is not collected by the $\text{PM}_{2.5}$ sampling criterion.

The correlation of deposited surface area with $\text{PM}_{2.5}$ is slightly weaker for the time-series measurements (linear correlation coefficient, $r = 0.93$). Over the first eight hours of the simulation day, particles grow, but the number concentration and total surface area decrease due to coagulation, leading to trends in number and surface area dose that are orthogonal to that of $\text{PM}_{2.5}$. As the inversion layer lifts between 07:00 and 11:00, the mass and number concentrations diminish, as does the surface area. When nucleation commences, the surface area increases more rapidly than does $\text{PM}_{2.5}$. Thus, mass and surface area data follow a rough general trend, but they show systematic differences. For measurements of the lognormal distributions, on the other hand, the correlation of deposited surface area with $\text{PM}_{2.5}$ is quite strong (linear correlation coefficient, $r = 0.99$). The surface area median diameter seems to shift in just the right ways.

In contrast, the number dose is uncorrelated with $\text{PM}_{2.5}$ (linear correlation coefficients, $r = -0.56$ and $r = -0.54$ for the time-series and lognormal distributions, respectively). $\text{PM}_{2.5}$ mass measurements do not represent number concentrations well when the particles sampled are very small in size. This occurs in the nucleation time-series simulation during the nucleation burst, and it is the case for many of the lognormal distributions considered. Table 4.1 summarizes the correlation coefficients for $\text{PM}_{2.5}$ with various metrics of deposition within in different regions of the lungs. The $\text{PM}_{2.5}$ measure of ambient exposure fails to correlate with measures of dose outside the targeted mass metric. This may limit the perceived links between PM exposure and health impacts. A more detailed characterization of the sampled

Table 4.1: Correlation coefficients (Pearson’s r) for $PM_{2.5}$ measurements with the values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	AI	TB	Head	Total
Mass Dose	0.997	0.996	0.999	0.999
Surface Area Dose	0.936	0.933	0.998	0.966
Number Dose	-0.567	-0.582	-0.581	-0.589

aerosol, such as that provided by a DMA system, may avoid such limitations.

We obtain the measured dose estimates shown in Figure 4.10 by simulating the measurements of a TSI long column DMA, inverting the data as discussed in detail in Chapter 2, and applying Eq. (4.1) to the recovered particle number size distributions. Error bars are derived from repeating the data simulation and inversion several hundred times while including simulated statistical noise in the measurements due to small count number. When all instrument efficiencies are appropriately accounted for, the number of particles transmitted through the DMA in any given measurement channel may be small, giving rise to large relative uncertainties in measured counts. We quantify these measurement uncertainties so that we can consider them separately from the systematic errors arising from the overall measurement process as dictated by instrument performance.

Figure 4.10 shows that the DMA can capture the different measures of dose quite accurately for the nucleation simulation distributions. The measurement range of the DMA simulated covers the size range of these distributions where particle concentrations are high. Many of the lognormal distributions considered, however, are concentrated at sizes beyond the measurement range of the DMA. This results in large negative biases for DMA estimates of dose values, as demonstrated in Figure 4.11 for MMDs greater than $1 \mu\text{m}$.

Table 4.2 summarizes the correlation coefficients for estimates of dose from DMA measurements with calculated true values of dose within in the lungs. That DMA measurements accurately capture dose estimates to all regions in all metrics is not surprising since particle size distribution recovery from DMA measurements is highly accurate under the operating conditions considered here.

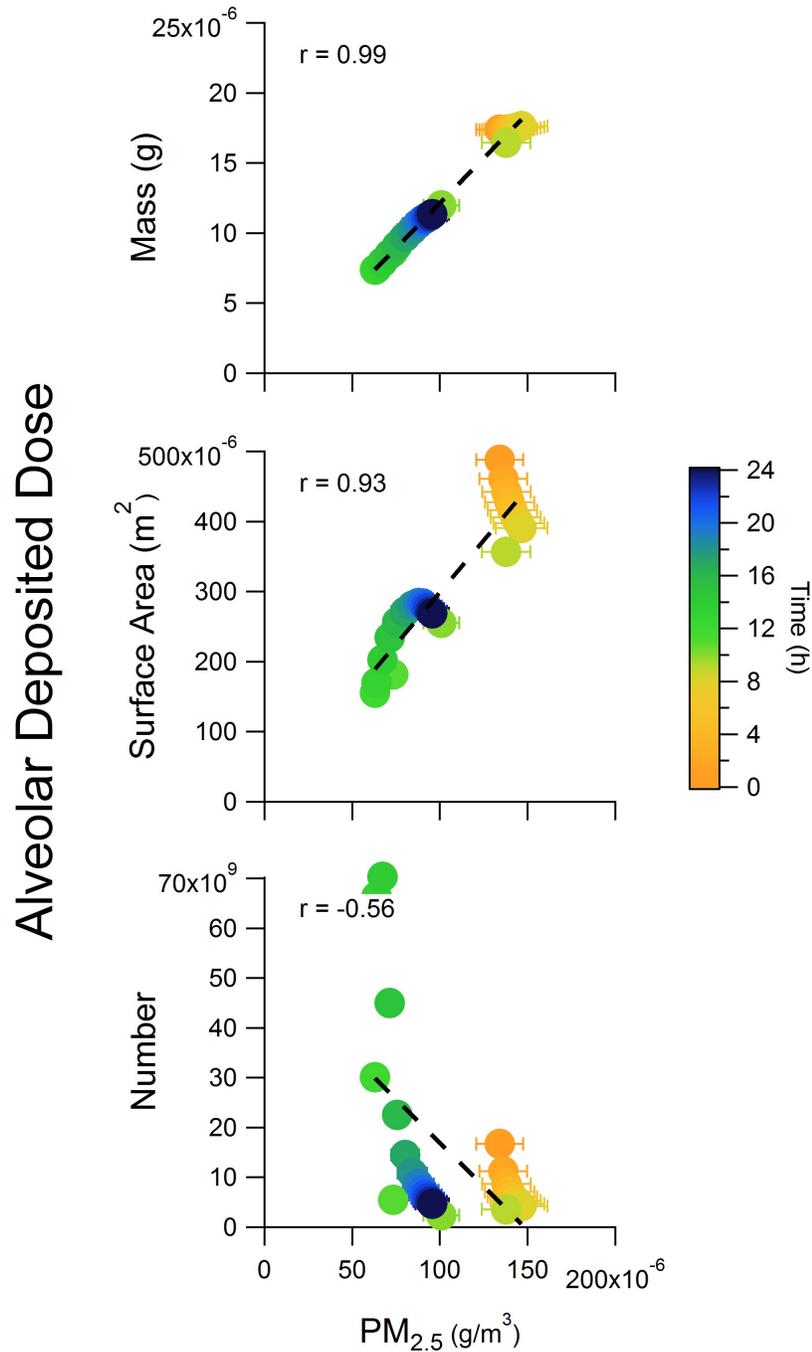


Figure 4.8: Correlation of $PM_{2.5}$ measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

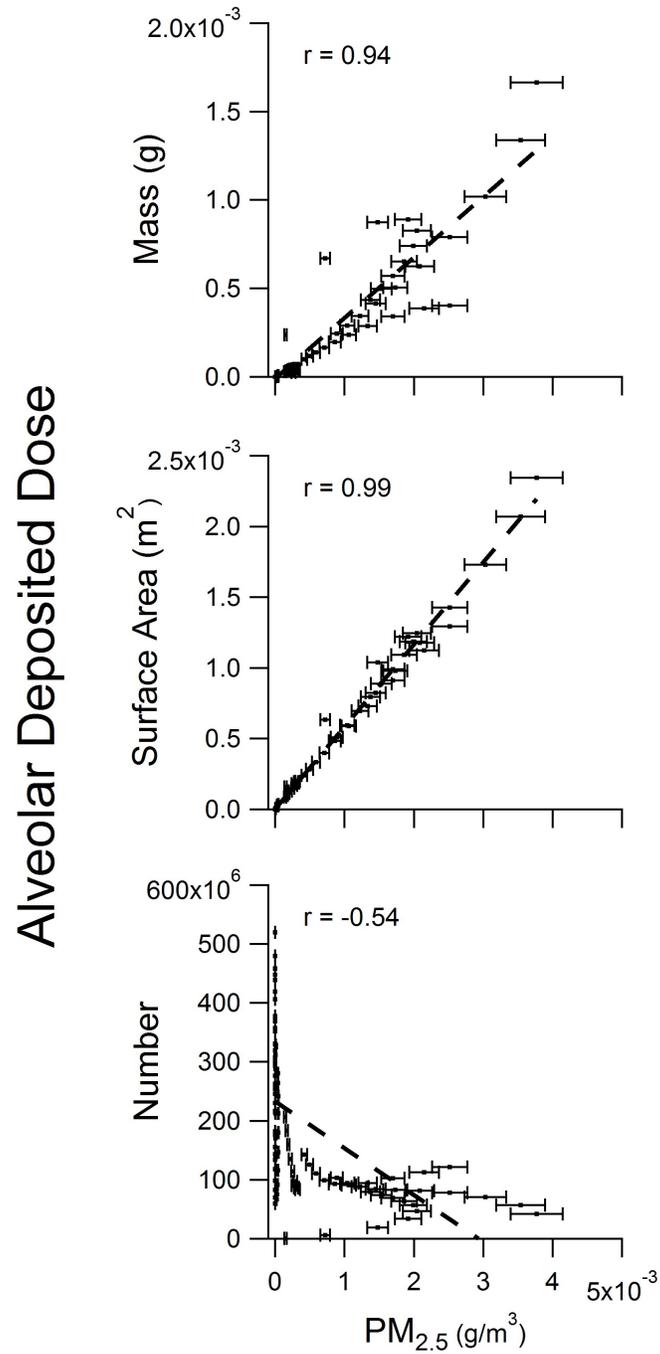


Figure 4.9: Correlation of $PM_{2.5}$ measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the map of lognormal distributions.

Alveolar Deposited Dose
DMA Estimate ($R=10$) vs True Dose

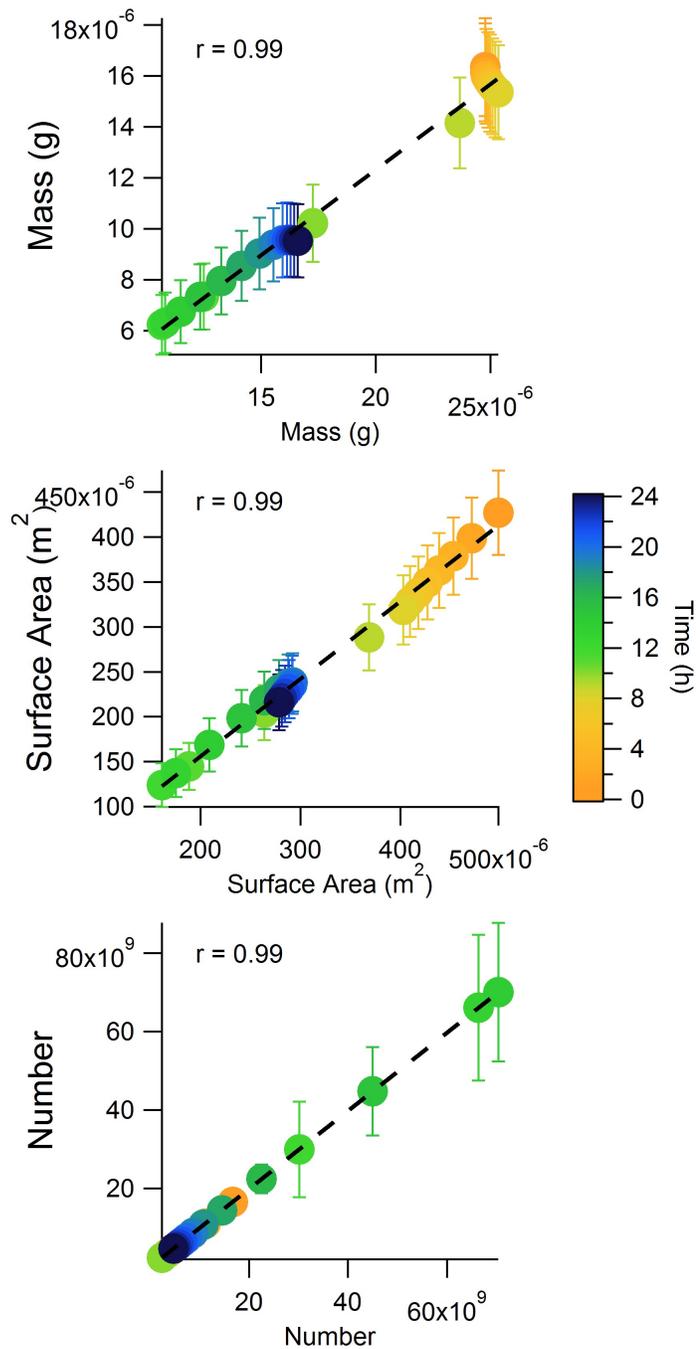


Figure 4.10: Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

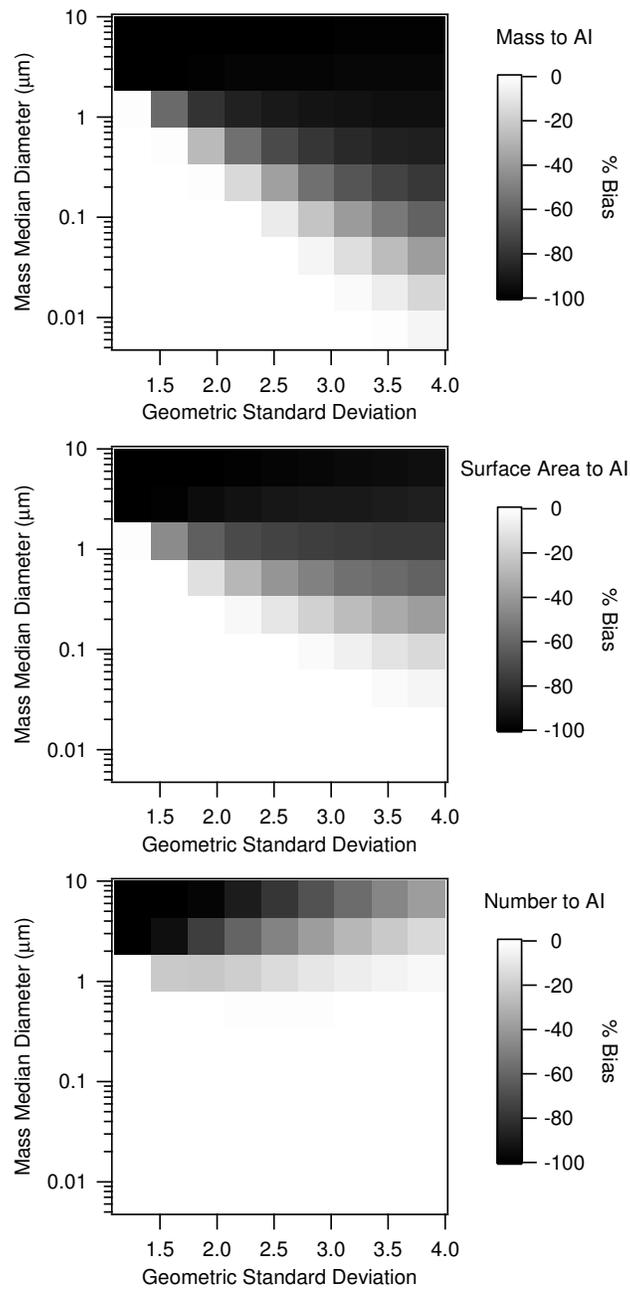


Figure 4.11: Percent bias in dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) compared to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the map of lognormal distributions.

Table 4.2: Correlation coefficients (Pearson's r) for estimates of dose from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) with the values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	AI	TB	Head	Total
Mass Dose	0.995	0.997	0.997	0.994
Surface Area Dose	0.997	0.998	0.989	0.992
Number Dose	0.999	0.999	0.999	0.999

4.4 Conclusions

The present study has outlined an approach to assessing the suitability of different measurement methods for health effects research, considering two select examples. We employed a well-studied type of air pollution event, as well as a myriad of lognormal distributions, to consider a wide range of exposures in probing the uncertainties introduced by different measurement scenarios. Indeed, any representative challenge aerosol can be used as input to this evaluation methodology; one need only be able to describe an ideal representation of the size distribution of interest.

This flexibility extends to the measurement systems that can be evaluated with this procedure. Any measurement system of interest can be simulated for evaluation; one need only be able to describe the operating principles, and data analysis if necessary. Provided the performance characteristics of the measurement devices are well understood, this scenario enables quantitative estimations of the measurement errors and uncertainties.

We have employed the ICRP model of lung deposition here to examine total and regional deposited dose in terms of mass, surface area, and number of particles. This is yet another flexible element of our evaluation procedure. Lung deposition can be modeled using the state-of-the-art model of choice. This is important for inevitable improvements in present models.

This evaluation methodology is straightforward but powerful and should be employed for planning measurement strategies for aerosol health studies. Many other measurement options are available; this evaluation can and should be applied to them. We encourage aerosol health researchers to question and evaluate the particle data available to them and, whenever possible, seek particle measurements that fo-

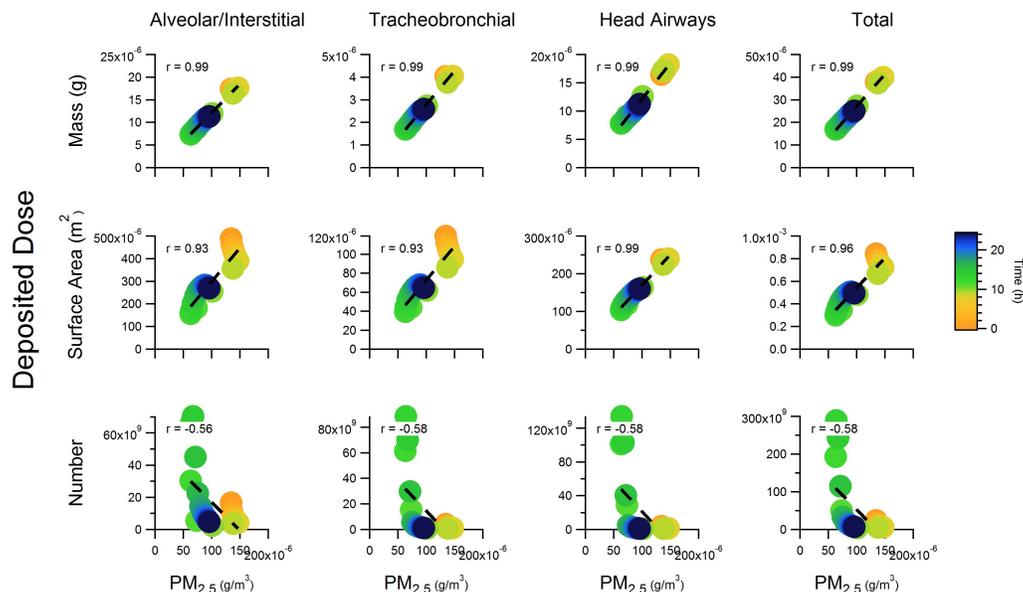


Figure 4.12: Correlation of PM_{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

cus on exploring particulate dose delivered to the human lungs. The methodology described here can be applied to the development of such strategy for any particular measurement campaign.

4.5 Additional Measurement Evaluation Results

The respiratory tract is often considered to consist of three anatomically and functionally distinct units: (a) the head/throat region, from the mouth and nose to the larynx; (b) the tracheobronchial region from the larynx through the conducting airways; and (c) the pulmonary/alveolar region, or the gas exchange zone. Lung regions defined in this way (combining the bronchial and bronchiolar regions of the original ICRP model into one tracheobronchial region) are used for the ICRP grand average deposition model based on Hinds’ (1999) parameterization, on which the regional deposition sampler (RDS) is based. Additional dose comparisons for these regions of the respiratory tract are shown in Figures 4.12 through 4.15.

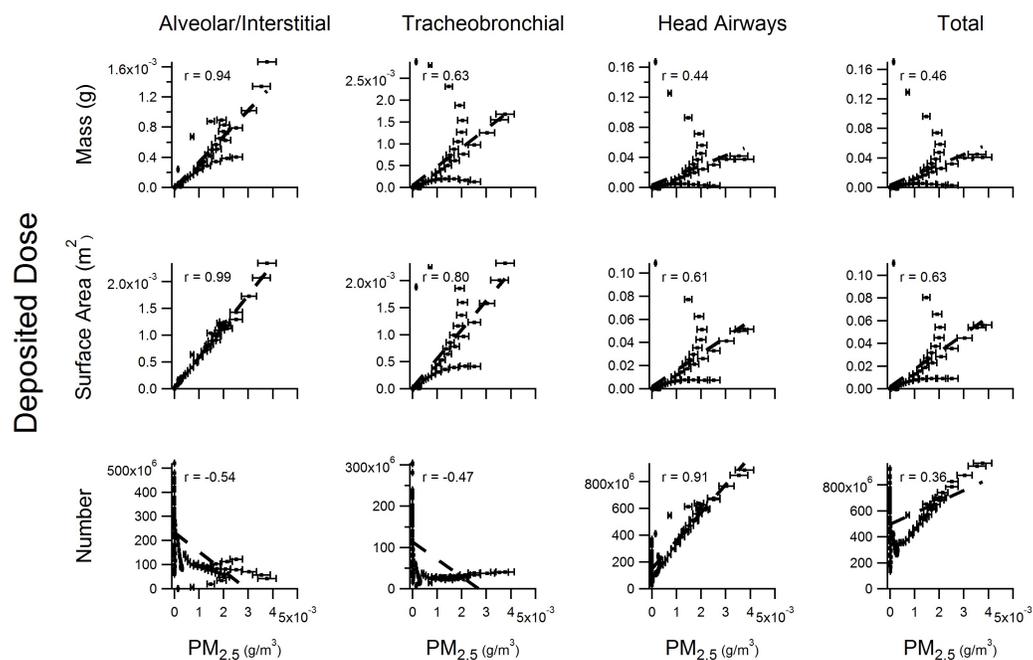


Figure 4.13: Correlation of PM_{2.5} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the map of lognormal distributions.

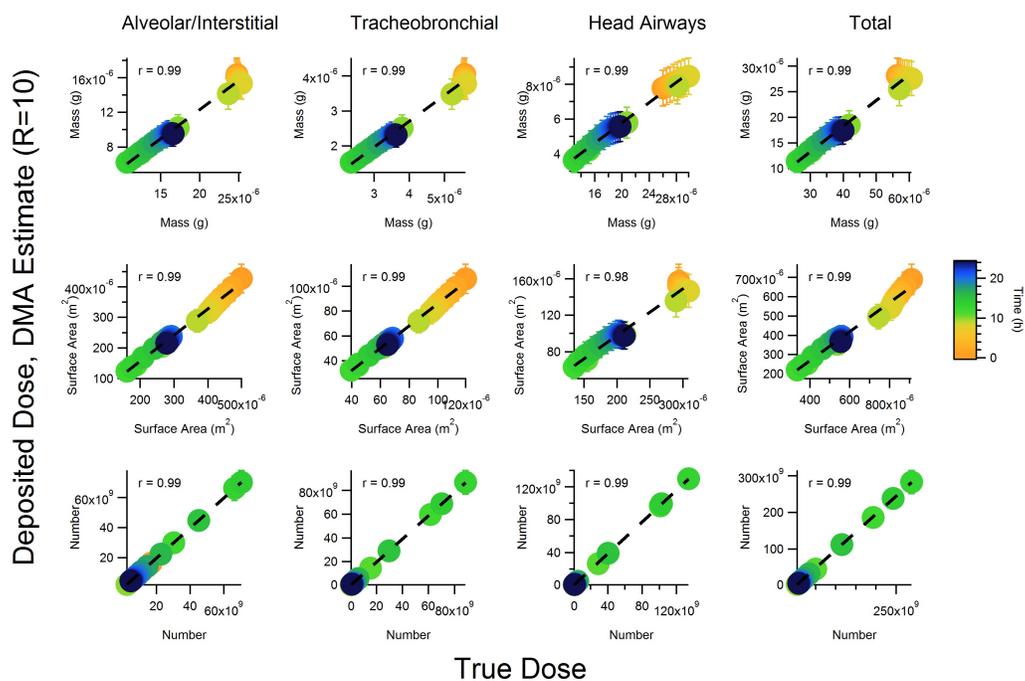


Figure 4.14: Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

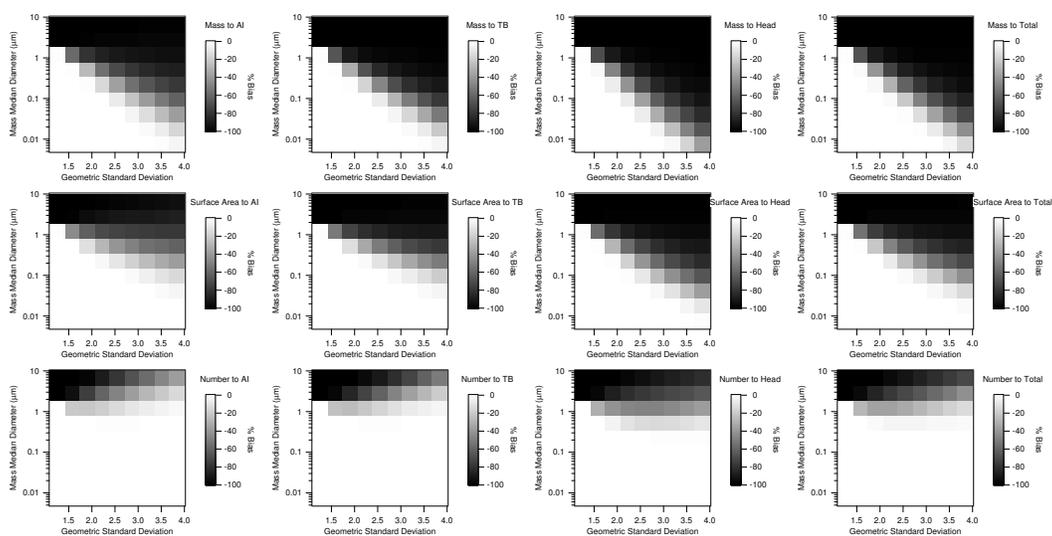


Figure 4.15: Percent bias in dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) compared to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the map of lognormal distributions.

*Chapter 5***PARTICLE MEASUREMENTS RELEVANT TO AEROSOL
HEALTH STUDIES II: CURRENT ABILITIES AND FUTURE
REQUIREMENTS**

A variety of adverse health effects have come to be associated with human exposure to airborne particulate matter. Previous studies have suggested that various metrics for this exposure and for the subsequent lung-deposited particle dose may be relevant to observed health outcomes. The metric most relevant to any given case remains uncertain, and typical data collection rarely possesses the resolution needed to efficiently explore the different options. The purpose of this study is to assess the nature of the health-related information that existing technology can provide and demonstrate the performance criteria that future technology should target. We do this by employing the methodology described in a previous study to evaluate the performance of a range of measurement schemes in estimating particle dose delivered to the human lungs. Our analysis demonstrates the need for, and validity of, a new paradigm for health-related aerosol particle measurements that prioritizes estimation of regional lung dose metrics. It is important to collect enough information from airborne particle measurements that various physical metrics of particle dose to the human lungs can be inferred. However, the measurement approach must also be accessible enough to encourage widespread use. We explore how the technology of differential electrical mobility classification has great potential for application in a measurement system capable of collecting data more informative than a single metric concentration, yet still simple enough to be used in developing a dense monitoring network.

5.1 Introduction

Health-related aerosol exposure measurements have long been characterized in terms of the mass of material per unit volume of air (CEN 1993; ISO 1995). Early epidemiological studies showed a good correlation between particulate matter (PM) mass concentration and adverse health effects in exposed human populations (Chamberlain et al. 1970). The mass concentration of particles is the mass of particles of different and unknown sizes per unit of air and is a relatively easy quantity to measure accurately. This led to the paradigm that potential health impacts can be

associated with the mass composition of inhaled material (Maynard and Jensen 2001). That paradigm now dominates health-related aerosol sampling.

In many ways, the mass paradigm can be seen as a pragmatic solution, combining effectiveness in curbing the impact of what were at one point dominant occupational aerosols, with the simplicity and cost-effectiveness afforded by filter samples. Yet recent research has begun to challenge this paradigm, suggesting that the potential for some classes of materials to cause harm might not be well represented by mass concentration measurements alone (Dockery et al. 1993; Oberdörster 1995; Peters et al. 1997; Pope and Dockery 2006).

In particular, there is a considerable knowledge gap about the effect of particles with an aerodynamic diameter of less than 100 nm on human health (Brouwer et al. 2009; Maynard and Aitken 2009). Currently, these particles are classified into two major categories based on their sources. Ultrafine particles (UFPs) refer to the particles that are incidentally generated in the environment, whereas nanoparticles are manufactured through controlled engineering processes (ASTM International 2006). Although there are many differences in the physicochemical composition of UFPs and nanoparticles, one common feature is their extremely small size and the corresponding “weightless” nature of these particles. The mass concentration of UFPs and nanoparticles is usually small and difficult to measure directly. There is considerable uncertainty over how exposure to these materials should be measured (Oberdörster et al. 2005).

Some studies have suggested that health effects of airborne nanoparticles may be more sensitive to number concentration than to mass (Oberdörster 1995). Studies also suggest that the health impacts of nontoxic engineered nanoparticles may be strongly associated with the surface area deposited within the respiratory tract (Oberdörster et al. 2007). These effects relate to mechanisms of action in the body, with hypotheses ranging from “particle overloading” of macrophage clearance mechanisms, to surface reactions for adsorption of blood proteins after they deposit in the respiratory tract. A number of studies have recognized that the most effective method for probing the mechanisms that underlie observed adverse health outcomes is to quantify the particle dose delivered to different regions of the respiratory tract (Esmen et al. 2002).

Total deposited dose can be measured experimentally by using human subjects to monitor inhaled and exhaled particle concentrations (Rissler et al. 2012). However, there can be considerable variability between individual subjects, and it is very

difficult to address experimentally how the measured total dose may be regionally distributed within the respiratory tract. Thus, dose values are typically estimated using mathematical models based on present understanding of regional deposition patterns (ICRP 1994; Vincent 2005). Even if quantitative dosimetry were possible, the question of what metric for dose should be measured remains open (Oberdörster et al. 2005). The range of possibly significant particle characteristics to consider, combined with the uncertainty about the most relevant metric of exposure and lung dose, makes exposure measurement a complex issue within the scope of current knowledge.

Though atmospheric exposure measurements are dominated by regulatory PM mass concentrations, many additional techniques and devices have been developed for measuring both atmospheric and occupational exposures and drawing relations to lung deposited dose values (Maynard and Aitken 2009). These devices can be classified according to the principles on which they are based, such as inertial or diffusional deposition, light scattering, and so on. Categories can also be defined for the different approaches based on their targeted metrics and information resolution. One purpose of the present study is to present an alternative method of classification for aerosol measurements, according to the type and quantity of information provided about potential particle dose delivered to the human lungs. The physical principles on which the detectors are based are of secondary importance in this classification scheme. Of primary importance is the consistent evaluation of any given measurement technique or available data when strategizing for health studies.

In Chapter 4 we discussed in depth this need for health effects researchers to scrutinize the available particle measurement data. We then outlined an approach for quantitatively evaluating different measurement methods with respect to estimating particulate exposure and lung dose. This approach compares the information provided by experimental measurements to predicted regional dose values that can be determined given knowledge of the aerosol population that is not biased by limitations of the measurement methods.

Our dose comparison strategy is illustrated in Figure 5.1. We define an unbiased description of input aerosol and calculate an ideal estimate of regional particle dose by employing an accepted mathematical model of lung deposition. We call this the “true dose.” At the same time, a chosen measurement technology is modeled, including measurement uncertainties and statistics, to produce data comparable to that obtained in actual measurements. As we will discuss below, different categories

of measurements provide different levels of information. For measurements of size-distributed information, the same model of lung deposition can be applied to the synthetic data to estimate regional dose, which can be compared to that which we calculated as the “true dose” to reveal the efficacy and biases of the measurement technology with respect to estimating particulate exposure and lung dose. For measurements of size-integrated information, the collected data can be directly compared to the “true dose” to evaluate whether the measurement can serve as proxy for the deposited dose.

In the present study, we use our evaluation methodology to investigate the performance of a range of existing PM measurement approaches. In our review of current techniques, we reveal the need for new expectations for health-related aerosol particle measurements that prioritize regional lung dose estimation, and we discuss an approach to developing new exposure monitoring instruments that address the challenges being presented today. Our intent is not to eliminate the current measurements and standards, but to promote support of a novel perspective for future developments in health-related measurement design: one that would lead to more sharply defined research focused on exploring particulate dose delivered to the human lungs.

5.2 Classification of Measurement Systems

Some of the more widely used techniques for studying aerosols and their health effects are listed in Tables 5.1(a) and (b), where they are classified in terms of their performance characteristics. The second column of the table shows whether the instrument classifies particles according to size and also how fine the resolution. The third column shows the time response behavior of the system, that is, whether the instrument responds to single particles or to short- or long-term averages. The fourth column shows the resolution of regional lung deposition estimates possible from collected data. Also indicated are the physical metrics that can be either directly quantified or inferred from the collected measurements.

The first device listed, the “universal aerosol monitor,” is an idealized instrument capable of sizing and analyzing individual particles to provide number, surface area, and mass metrics for any sampled aerosol. Since the monitor responds to individual particles, the time resolution is also perfect. Its perfect size classification and time resolution are represented by the solid coverage of the size distribution curve and the timeline graphics shown in the size and time columns, respectively. From the

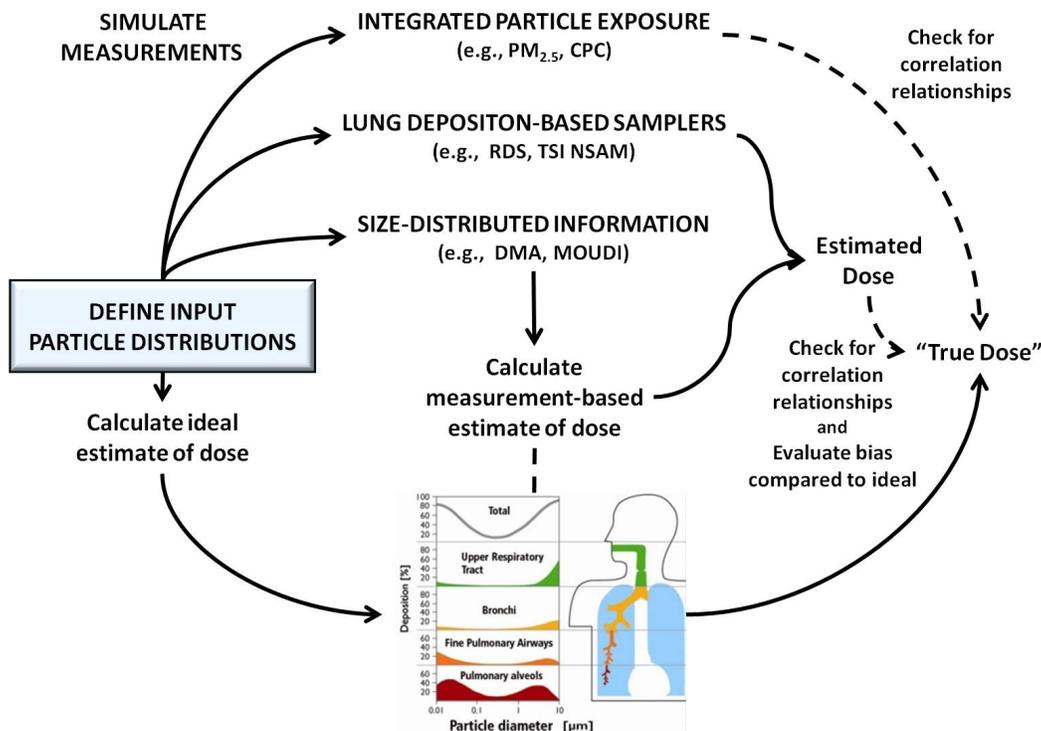


Figure 5.1: Procedural flow illustration for the quantitative evaluation of health-relevant information provided by any given measurement strategy. Simulated experimental information is evaluated by drawing comparisons to estimates of respiratory dose that would actually occur, given knowledge of the ambient aerosol that is not biased by limitations of the measurement methods. This comparison to a “true dose” creates the opportunity for quantitative assessments. Relying upon synthetic data and ideal models of the measurement methods under consideration enables control over any uncertainties associated with the different methods and ensures fair assessments and inter-comparisons of measurement strategies with potentially very different operating principles.

detailed information collected, regional lung deposition can be estimated with the fullest resolution currently available. And, of course, this instrument is small and cheap enough for widespread personal monitoring.

The next several devices listed in Tables 5.1(a) and (b) are real, existing instruments employing measurement schemes that range in operational complexity and information resolution. We analyze this sample of techniques here, progressing from measures of integrated exposure, to proposed samplers that reveal integrated regional lung dose, and finally to particle size distribution measurements from which regional deposition can be calculated. These measurements are sufficiently well understood for our *in silico* experiments; the size and time resolution of the tech-

niques considered, along with the physical metrics characterized, are defined from operational theories. It is the classification with respect to regional lung deposition estimates that calls upon our measurement evaluation methodology, illustrated in Figure 5.1.

Applying our measurement evaluation procedure to this range of existing measurement techniques makes it possible to see how far health-related particle measurement technology has developed and also indicates where further developments should be directed in the future. We have considered possible future developments and included as the final entry in Table 5.1(b) a proposed, more realistic, approximation of the ideal universal aerosol monitor. The design of this system would involve compromises in the operational complexity and information resolution exhibited by the other measurement techniques analyzed.

In the following sections, we describe the operational theories and applications of these various measurement techniques in greater depth.

Integral Measures of Particle Exposure

Several devices listed in Table 5.1(a) measure integral functions of certain distributed properties of aerosol particles. These devices lump together all particle size classes in a certain range. Since all particles in the given size range contribute to the measured metric without distinguishing among them, there is no resolution with respect to size as indicated by the lack of particle size distribution information in the size resolution column. Instead of revealing the form of the particle size distribution, these instruments provide integral measures of given physical properties, with efficiencies determined by the instrument transfer functions. Sketches of these transfer functions are shown in the size resolution column.

The $PM_{2.5}$ sampler, listed below the universal aerosol monitor, is the primary method used for assessing integrated ambient fine particle exposure today. This sampler measures the mass concentration of particles with diameters smaller than $2.5 \mu\text{m}$. $PM_{2.5}$ data are widely measured to document compliance with the National Ambient Air Quality Standards (NAAQS), making these measures of exposure widely available for epidemiological studies (Brunekreef and Forsberg 2005; Pope and Dockery 2006; Vedal 1997).

As a regulatory metric, the $PM_{2.5}$ criterion was born of the desire to target a fine particle fraction derived in large measure from combustion processes. It does not relate to the particle size dependency of how particles are deposited within

Table 5.1: Performance characteristics for various approaches to measuring particle exposure and regional lung dose.

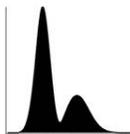
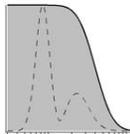
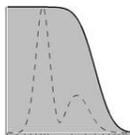
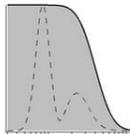
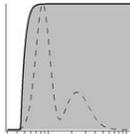
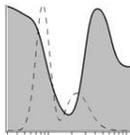
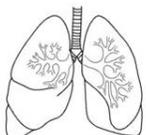
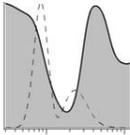
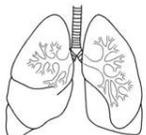
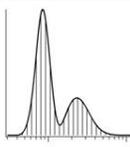
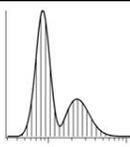
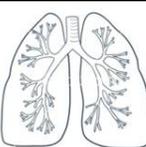
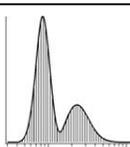
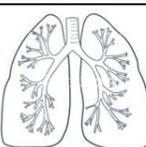
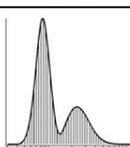
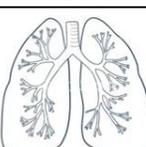
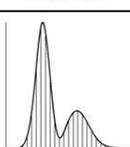
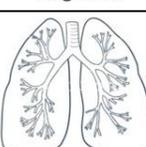
Device	Measured Quantity and Resolution		
	Size	Time	Lung Deposition
Idealized Universal Aerosol Monitor	 Continuous	All physical metrics  Instantaneous	 Detailed Regional
PM _{2.5}	 Integral	Mass  Hours	 Total
Lung Deposition-Based Criteria Samplers	 Integral	Mass  Hours	 Total
Tapered Element Oscillating Microbalance (TEOM)	 Integral	Mass  Seconds	 Total
Condensation Particle Counter (CPC)	 Integral	Number  Seconds	 Total
Regional Deposition Sampler (RDS)	 Integral	Mass  Hours	 Fixed Regional
TSI Nanoparticle Surface Area Monitor (NSAM)	 Integral	Surface Area  Seconds	 Fixed Regional

Table 5.1: Continued summary of performance characteristics for various approaches to measuring particle exposure and regional lung dose.

Device	Measured Quantity and Resolution		
	Size	Time	Lung Deposition
Nano MOUDI	 Discrete	Mass  Hours	 Regional
Electrical Low Pressure Impactor (ELPI)	 Discrete	Estimate N, SA, M  Minutes	 Regional
Aerodynamic Particle Sizer (APS)	 Discrete	Estimate N, SA, M  Minutes	 Regional
Electrical Mobility Analyzer (e.g., SMPS)	 Discrete	Estimate N, SA, M  Minutes	 Regional
Future Technology	 Discrete	Estimate N, SA, M  Minutes	 Regional

the respiratory tract, but rather to the source-related properties of atmospheric aerosol. These measurements are generally collected from stationary community air monitoring sites and can thus fail to describe the variability of individual exposures.

To better quantify the mass concentration of individual exposures, size-selective personal samplers are often used in workplaces. Particle size fraction definitions for health-related sampling follow definitions proposed by the American Conference of Governmental Industrial Hygienists (ACGIH 1985) and are accepted internationally by such organizations as the European Standards Organization (CEN 1993) and the International Organization for Standardization (ISO 1995). Standards exist for penetration-based criteria for the inhalable, thoracic, and respirable aerosol

fractions, as shown in Figure 5.2. The mathematical expressions used to reproduce these criteria are included in Appendix A.

The particle size-selective criterion for $PM_{2.5}$ samplers is also shown in Figure 5.2. For this present study, we focus our measurement evaluation on the $PM_{2.5}$ criterion over the respirable fraction criterion since the shapes of these curves are very similar and provide nearly identical measured data for our challenge aerosol populations. A number of samplers have been developed to measure the respirable fraction, including cyclone-separator-based instruments (Gorner et al. 2001); these samplers can be considered to provide the same quality of data as $PM_{2.5}$ measurements, but with the advantage of quantifying exposure at the human receptor.

Whether considering measurements from a $PM_{2.5}$ sampler or a respirable personal sampler, the data provided is averaged over the sampling volume or time of operation with resolution typically on the order of hours, as indicated by the divisions on in the timeline graphic in Table 5.1. These measurements thus fail to describe the variability of an individual's exposure over time. This is problematic for studying atmospheric new particle formation events that lead to exposures to ultrafine particles. Such events may occur over large areas, but due to rapid growth and losses of ultrafine particles, are often brief, so the concentrations vary rapidly with time and with distance from sources such as highways.

The Tapered Element Oscillating Microbalance (TEOM, e.g., Series 1400a, Rupprecht & Patashnick Co. Inc., USA) provides integrated mass concentration measurements with faster time response. The short-term or real-time response is illustrated as finer divisions on the timeline graphic. The TEOM is a static device with a high flow rate primarily used to measure the aerosol component of atmospheric pollution. The instrument provides true time-differentiated mass measurements by detecting the change in oscillating frequency of a collection substrate as it becomes loaded with particles. With a suitable size selective inlet, the TEOM can be configured to measure a range of particle sizes. We explore the theoretical performance of this technology with a size-selective inlet capable of excluding particles larger than approximately 100 nm in diameter. Such a setup would allow real-time mass concentration measurements of nanoparticles. Although there are no commercial devices of this type currently available, there is no reason in principle why they could not be developed in the future. Particle size-selective inlets based on diffusion would make appropriate pre-selectors for such devices. For our model instrument, we use the 100 nm cut-point curve shown in Figure 5.2.

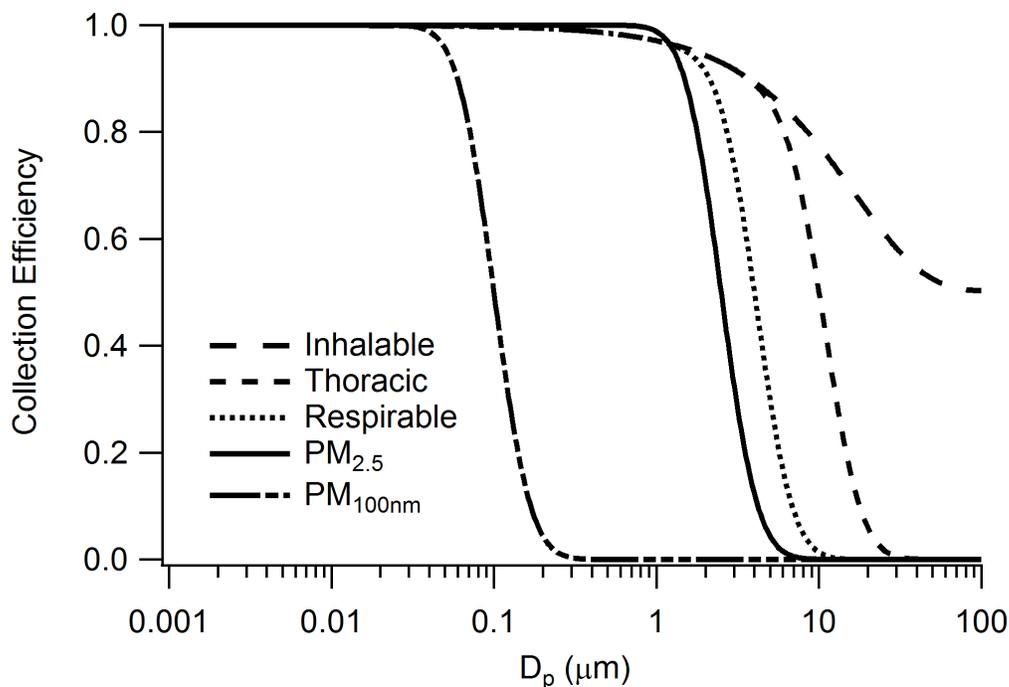


Figure 5.2: Summary of currently accepted particle size-selective criteria for aerosol sampling of inhalable, thoracic, respirable, and PM_{2.5} fractions (Vincent 2005). A collection efficiency curve is also shown for a theoretical 100 nm cut-point.

For real-time measurement of particle number concentration, condensation particle counters (CPCs) are the most widely used type of instrument. CPCs measure the total aerosol number concentration larger than some minimum detectable size. In all particle number concentration measurements, the integration limits over which a particular instrument operates are critical in understanding the reported results. CPCs become increasingly insensitive to particles smaller than 10–20 nm. Concentrations measured with instruments with different sensitivities may therefore differ substantially, particularly if the particle count median diameter is close to or within this range. For the present study, we model an ultrafine CPC (UCPC, e.g., TSI Model 3025A, TSI Inc., USA) with the counting efficiency curve shown in Figure 5.3.

To simulate the integral measurements described here, we calculate the integral

$$R_{device} = \int_0^{\infty} \eta_{device} \cdot n_N \cdot v \cdot d \ln D_p \quad (5.1)$$

where R_{device} is the measured response of the modeled device, η_{device} [–] is the collection efficiency of the device, $n_N = dN/d \ln(D_p)$ [particles/m³] is the par-

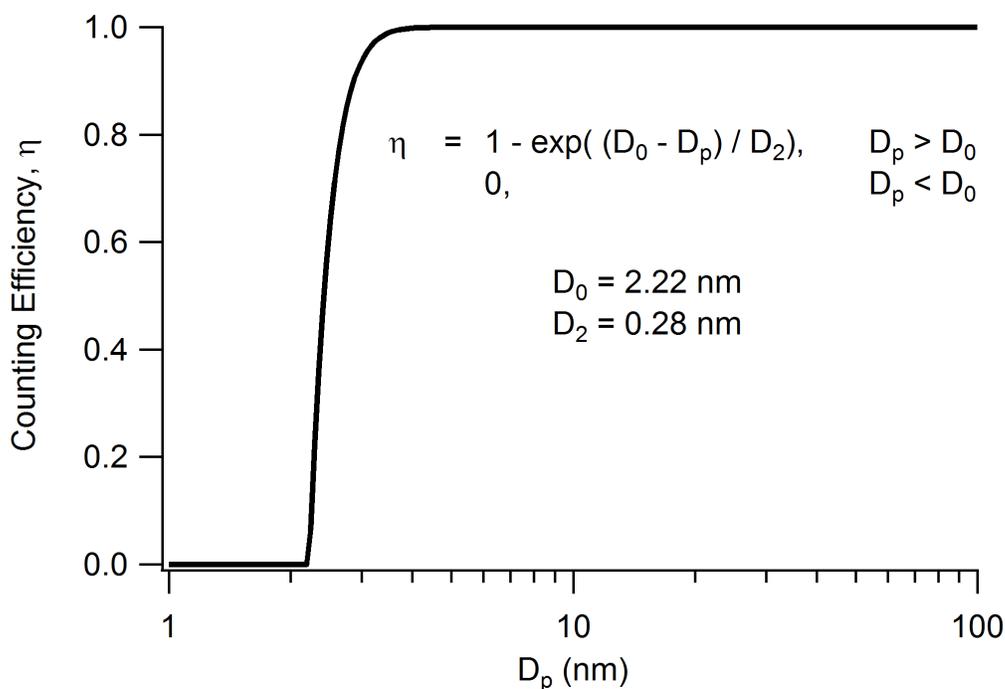


Figure 5.3: Counting efficiency curve for the TSI Model 3025A UCPC, determined from exponential fit of experimental data presented by Wiedensohler et al. (1997).

particle number size distribution, and ν describes which metric is being evaluated, i.e., number (N ; $\nu = 1$), surface area (S ; $\nu = \pi D_p^2$), or mass (M ; $\nu = \frac{\pi}{6} D_p^3 \rho_p$). For present purposes, particles are assumed to be spherical and to have specific gravity of 1.4, which is typical of secondary organic aerosols that comprise much of the atmospheric aerosol (Ahlm et al. 2012). We include estimates of systematic uncertainty for our simulated measurements, calculated as $\pm 10\%$ of the measured concentrations.

Whether time-averaged or real-time, each of the above integral measures of mass (or number) concentration may provide information on total inhaled mass (or number) of particles, but offers no insight into how different size particles may deposit in different anatomical regions of the respiratory tract. This is illustrated with the uniform, unresolved gray coloring of the lung graphics in the lung resolution column of Table 5.1(a). These measures provide little, if any, resolution of particle size. Lacking knowledge of the particle size distribution, the internal, or absorbed, particulate matter dose cannot be estimated. This does little to help guide efforts to understand the deposition, fate, and effects of inhaled particles. Indeed, mass-based standards for particle concentrations in air implicitly assume the same risk to health

from all particles of a given size (Council 2004). This assumption fails in many respects. Not all inhaled particles deposit within the respiratory tract; many particles are exhaled without depositing. Those that are retained deposit in different regions of the airways depending upon their size.

Lung Deposition-Based Samplers

The criterion for the respirable aerosol fraction discussed earlier is based on the penetration of particles to the alveolar region. It does not refer to what is actually deposited, and so does not strictly relate to the health-related dose. This criterion has the advantage that it is simple in form, and so suggests relatively easy options for sampling instrumentation. However, many exposure assessment situations would benefit from exposure information more closely associated with dose. The ISO (2012) has released recommendations for sampling conventions based on particle deposition (rather than exposure) in adult males and females engaged in activities of sitting, light exercise, and heavy exercise. The ISO estimates of deposition were determined using the International Commission of Radiological Protection (ICRP 1994) human respiratory tract model.

Thus, one approach to improving estimates of personal lung deposition dose is to design samplers with particle size-selective criteria that closely match respiratory tract deposition curves. We consider here two examples of such respiratory dose measuring devices that have been developed based upon different physical principles and targeting different metrics of dose.

Fissan et al. (2006) provide a measure of “lung deposited nanoparticle surface area.” They employ a unipolar (single polarity) charger to bring sampled aerosol to a charged state where the electrical current carried by the charged particles is proportional to the cumulative surface area of spherical particles that deposit within the alveolar region of the respiratory tract. They note that, even for the agglomerate particles produced in many high temperature processes, the charging probability scales with particle surface area. An ion trap tunes the signal to match the size dependent lung deposition profile. Commercial instruments have been developed using this approach for inhaled surface area dose measurements. For our analysis here, we simulate operation of the TSI Nanoparticle Surface Area Monitor (NSAM, Model 3550, TSI Inc., USA).

The TSI NSAM indicates the surface area of the fraction of particles that deposit in either the tracheobronchial or alveolar regions of the respiratory tract. The target

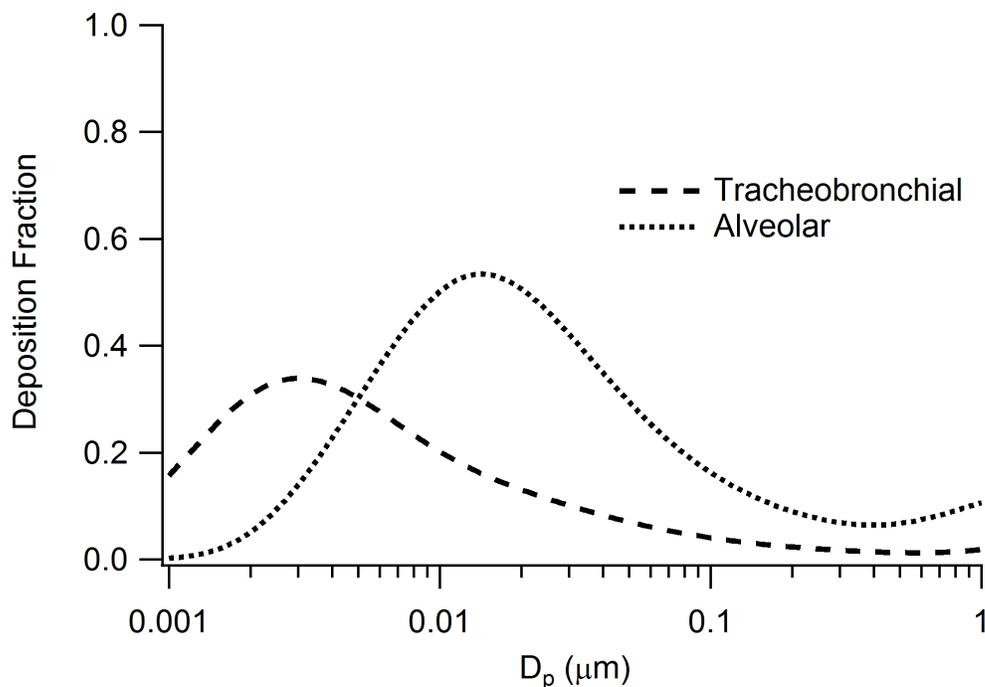


Figure 5.4: Model deposition fractions representing the information captured by the TSI NSAM sampler as a function of particle diameter for the tracheobronchial and alveolar regions of the human respiratory tract. Deposition fractions were determined using the ICRP (1994) model for a reference worker.

performance of an ideal NSAM is to match the criteria for lung deposition of particles predicted by the ICRP model for a reference worker. These curves are shown in Figure 5.4 as a function of particle diameter. While the Model 3550's performance is well characterized up to 400 nm, the instrument's response for larger particles is not as clearly understood.

Kuo et al. (2005) developed an inlet that reproduced a portion of the ICRP (1994) lung deposition pattern, allowing downstream instruments to collect samples representative of the aerosol that penetrates to the lower airways. Koehler and Volckens (2013) adapted this approach to develop a sampler, the regional deposition sampler (RDS), for determination of the lung particle mass dose at a personal level. The RDS was designed to assess particle deposition in the head airways, tracheobronchial, and alveolar regions using mass-based sampling. The target performance of an ideal RDS is to match the criteria for lung deposition of particles predicted by Hinds' (1999) parameterization for the average of adult males and females at three levels of exercise (i.e., sitting, light exercise, and heavy exercise). These curves are shown in Figure 5.5 as a function of particle diameter.

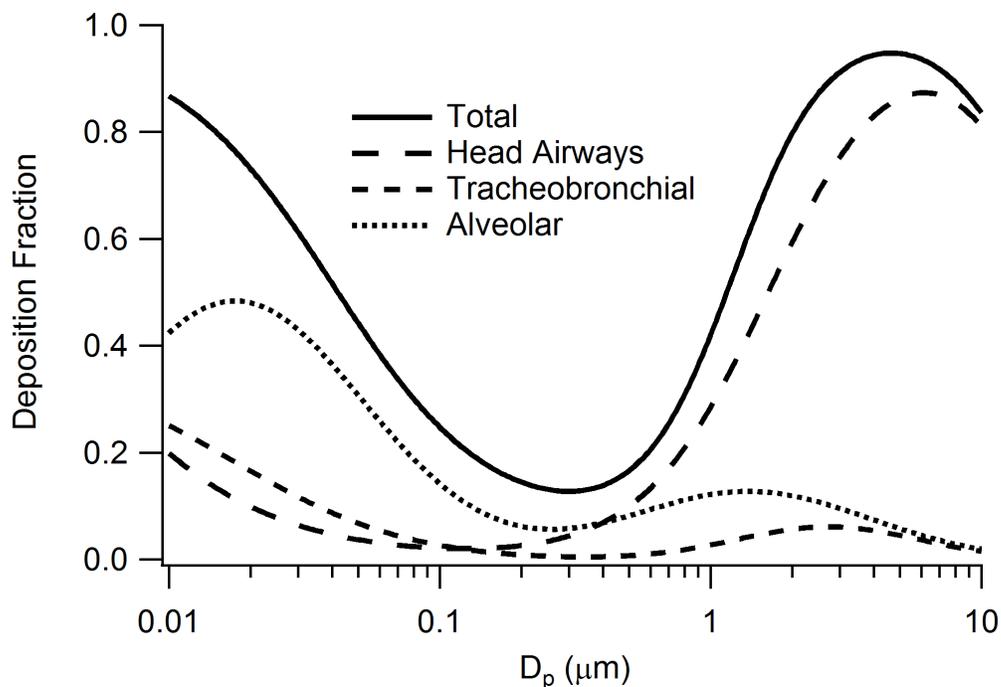


Figure 5.5: Model deposition fractions for the RDS sampler developed by Koehler and Volckens (2013), shown as a function of particle diameter for each of three regions in the human respiratory tract, as well as total deposition. Deposition fractions were determined using Hinds' (1999) parameterization for the average of adult males and females under conditions of sitting, light exercise, and heavy exercise.

The resolution of regional deposition capable with the RDS and the NSAM is illustrated in Table 5.1(a) with lung graphics containing moderate detail. The information captured by these samplers can be most ideally represented by applying the respective modeled deposition fractions directly to the simulated aerosol exposure size distribution to calculate the human lung-deposited surface area of particles as

$$Dose_{\nu} = \int_{t_1}^{t_2} \int_{D_{p1}}^{D_{p2}} V_E \cdot \eta_{dep} \cdot n_N \cdot \nu \cdot d \ln D_p \cdot dt \quad (5.2)$$

where V_E [m^3/hr] is the ventilation rate, or the volume of air inhaled (we used $V_E = 1.5 \text{ m}^3/\text{hr}$), η_{dep} [—] is the deposition fraction of aerosol particles in the respiratory system (as determined from the ICRP model or Hinds' parameterization for the region of interest), $n_N = dN/d \ln(D_p)$ [particles/ m^3] is the particle number size distribution, and ν describes the dose metric being evaluated, i.e., number (N ; $\nu = 1$), surface area (S ; $\nu = \pi D_p^2$), or mass (M ; $\nu = \frac{\pi}{6} D_p^3 \rho_p$). Additional parameters and details for the models used are provided in Appendices A and B.

For our purposes here in demonstrating the performance of lung deposition-based sampling, we focus our analysis on the theoretical ideal operation of the RDS. The modeled and experimental results presented by Koehler and Volckens (2013) show decent agreement with the simplified deposition models described by Hinds (1999), so that assuming perfect recreation of Hinds' parameterization for the RDS collection efficiency allows us to evaluate the most optimistic view of any potential application of this device. We carry our optimism one step further and also consider the theoretical application of the RDS to providing measures of all three of the physical metrics of dose listed above. We again include estimates of systematic uncertainty for the simulated measurements calculated as $\pm 10\%$ of the measured concentrations.

When one uses a single metric, be it mass, surface area, or number, to characterize the aerosol, even the deposited dose of aerosol, one presumes that the measured quantity bears the strongest association with the health effects under study. Narrowing the accessible information in this way may conceal major effects. Bartley and Vincent (2011) suggests designing arrays of samplers or sampler stages to estimate the different metrics of deposition in the different regions of the respiratory tract in order to avoid perceived difficulties in characterizing the full aerosol particle size distribution. We investigate instruments capable of such characterization here to demonstrate that full characterization of particle size distributions need not be time and resource intensive, and should be pursued for the great benefits this information would provide to health studies. Together with records of individual breathing profiles, size distribution information would permit accurate estimation of tissue doses through a set of relatively simple calculations, and would allow a single data set to be applied to different populations by applying the appropriate respiratory tract deposition profiles.

Measures of Distributed Properties

Finally, we consider the approach of measuring the distribution of particle properties with respect to size. Instruments capable of size-discretized measurements are identified in Table 5.1(b) by drawing interval divisions within the size distributions in the size resolution column.

A number of instruments enable aerosol mass concentration to be measured as a function of particle size, down to nanometer diameters. For time-integrated, off-line analysis of mass concentration as a function of particle size, the most prominently

used instrument is the Nano MOUDI (Micro Orifice Uniform Deposit Impactor, Applied Physics Co., USA). It is a cascade impactor consisting of thirteen stages collecting progressively smaller particles. The stages have cut-off diameters between 10 nm and 10 μm , making this one of the few instruments that spans the respirable particle size range.

For direct, on-line measurements, the Electrical Low Pressure Impactor (ELPI, Dekati Ltd., Finland) is a static aerosol sampler capable of measuring particle size distribution and mass concentration in the size range 7 nm to 10 μm , again spanning the respirable range. Particles sampled into the ELPI are charged and then passed into a low-pressure cascade impactor with a series of electrically isolated collection stages. Each stage collects progressively smaller particles, enabling the size distribution of the aerosol to be estimated. The electrical current carried by the charged particles onto each impactor stage is measured in real-time by a sensitive multichannel electrometer. Collected particles can be removed from the impactor stages for off-line analysis. Figure 5.6 shows the transfer functions for a 15-stage ELPI.

The Aerodynamic Particle Sizer (APS, e.g., TSI Model 3321) uses time-of-flight aerodynamic sizing to simultaneously count and size particles. The APS characterizes airborne particles in the (aerodynamic) size range from 0.5 – 15 μm and so is frequently used to measure larger particle size distributions.

Differential mobility analyzers (DMAs), operated in either constant voltage (DMPS, differential mobility particle sizer) or scanned voltage (SMPS, scanning mobility particle sizer) mode, are the most commonly used instruments for measuring particle number size distributions. Devices of this type are capable of measuring aerosol size distributions from approximately 3 – 800 nm, although not over the complete range in a single instrument. Larger particles acquire multiple charges, degrading measurement resolution. The size distribution is expressed in terms of particle mobility diameter.

A DMA system operates by charging particles in an aerosol sample, typically by passing them through a charge conditioner that generates a bipolar ion cloud using a radioactive source. The sample then passes through a well-defined electrostatic field. Charged particles in that sample are induced to migrate across a parallel flow of particle-free air by an applied electrical field. Particles with a specific electrical mobility are sampled from the exit of the electrodes, and counted. By scanning or stepping the voltage between the electrodes, particles with electrical mobilities

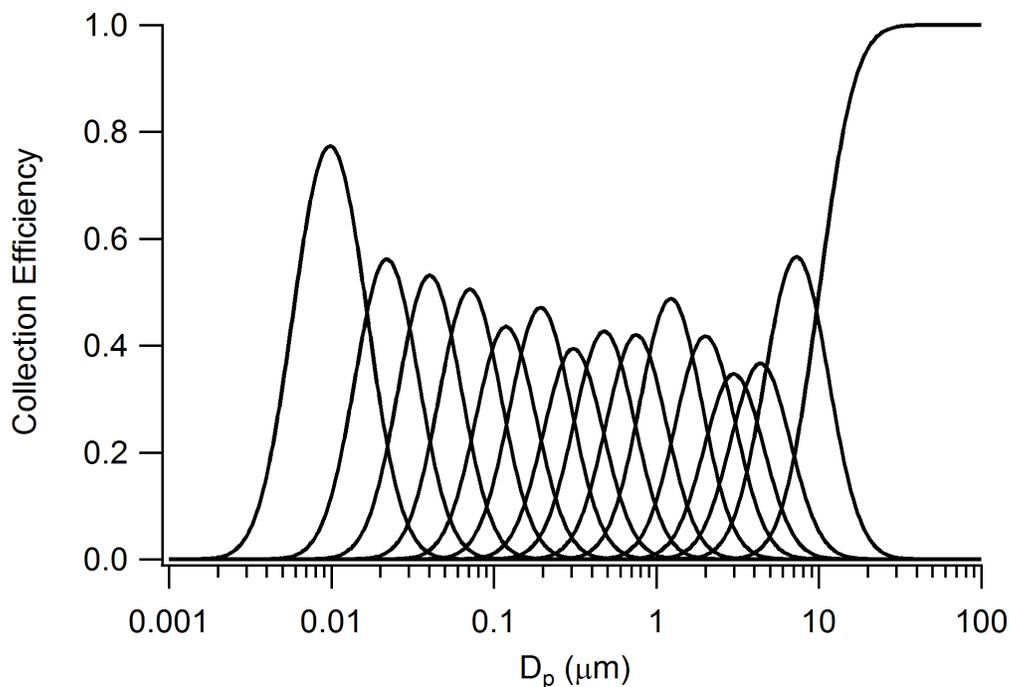


Figure 5.6: Transfer functions for a 15-stage ELPI, with cut-points ranging from 6 nm to 10 μm . The individual peaks cover the range in particle diameter sizes collected on a given stage.

corresponding to a range of particle diameters can be counted sequentially (typically using a CPC), allowing the aerosol size distribution to be determined. Figure 5.7 shows transfer functions for a simulated stepping-mode DMA system.

For the size-discretized measurements considered here, the relationship between the measured responses and the sampled aerosol size spectrum is given by

$$R_i = \int_0^{\infty} n_N(\ln D_p) \Psi(i, D_p) d \ln D_p, \quad i = 1, 2, \dots, I \quad (5.3)$$

where R_i is the i th datum collected, I is the total number of measurements made, $n_N = dN/d \ln(D_p)$ [particles/ m^3] is the particle number size distribution, and $\Psi(i, D_p)$ is the response of the i th measurement channel to a particle of diameter D_p for the device under consideration (i.e., the transfer functions illustrated in Figs. 5.6 and 5.7).

The relationship between measured responses and the aerosol size spectrum, as defined in Eq. (5.3), results in a set of Fredholm integral equations of the first kind. With the response values R_i and response function $\Psi(i, D_p)$ known, solving

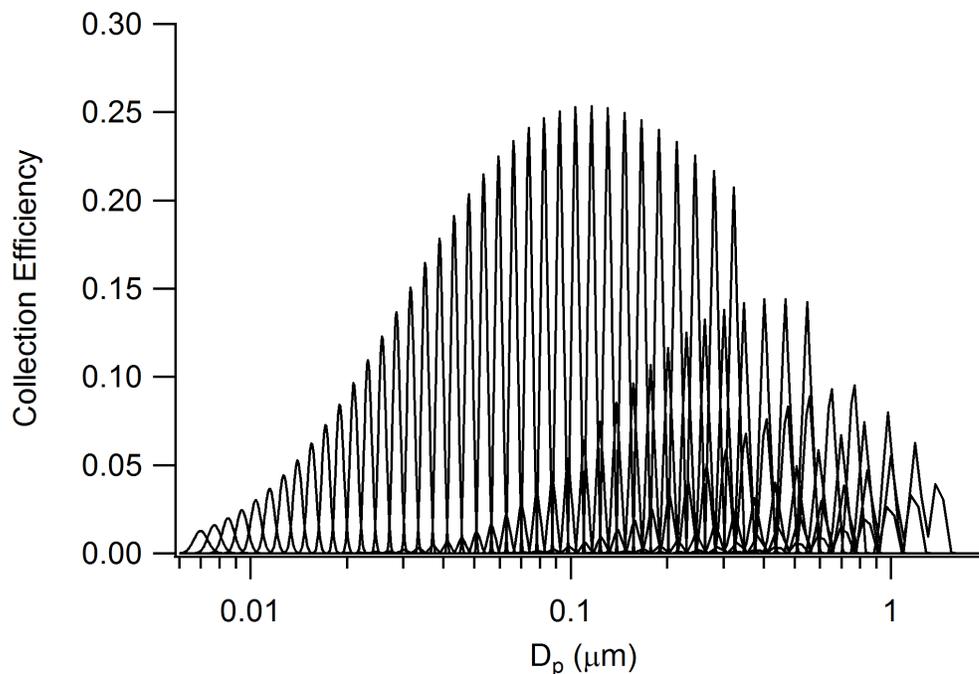


Figure 5.7: Transfer functions for a simulated DMPS system operating with 36 measurement channels. Peaks indicate the range of aerosol sizes that contribute to the measured responses for a given channel. The distinct waves of peaks demonstrate the effect of multiple charging on the response function. Multiple charging is not significant at the smaller sizes, but does become important at the larger sizes.

this set of equations for the unknown sampled size distribution n_N is one important example of the many inverse problems encountered in atmospheric remote sensing. The governing equations and methods of inversion related to DMA measurements have been explored in detail in Chapter 2. We employ the routine developed and described there to solve any inverse problems here in this study.

To address measurement uncertainty for this class of instruments, we impose Poisson counting statistics on the simulated instrument particle counts to produce signals representative of what would be obtained using real instruments. To estimate the uncertainties in inverted size distribution results, we apply a Monte Carlo method. For this, we draw sets of instrument responses as random samples from the calculated instrument responses, assuming a Poisson distribution, and use these as input for the inversion algorithm. We repeat this procedure two hundred times, storing each inversion result, and then statistically evaluate the stored inversion results to obtain a mean result and its uncertainty.

These instruments that provide information on both particle size and concentration

with respect to a given metric are, not surprisingly, larger, more complex, and more expensive than those measuring concentrations alone. The size distribution information can be invaluable, however, in permitting estimation of tissue dose through a set of relatively simple calculations, using Eq. (5.2). By combining size distribution information with state-of-the-art lung deposition models, the data can be analyzed and reanalyzed with parameters tailored to the population of interest to provide the most helpful estimates of lung dose for particle health studies. Mobility analyzers in particular are capable of recovering sufficient information to calculate good estimates of multiple physical metrics of deposition dose, i.e., mass, surface area or number (Oberdörster et al. 2005).

In the present study, we evaluate measurements provided by typical application of mobility analysis technology, and we also explore potential future applications for developing the realistic approximation of the ideal universal aerosol monitor. We provide more details on differential mobility analysis from the context of achieving ideal requirements for health measurements in the next section.

Ideal Measurement Requirements

Ideally, measurements intended to assess human exposures to, and deposited dose of, potentially hazardous airborne particles should be obtained using accurate and precise methods throughout the periods of exposure. Two approaches might be taken: (i) the use of a personal monitor for each subject being studied, or (ii) the creation of a dense network of sensors that maps the concentrations and properties of the aerosol throughout the community with sufficient spatial and temporal resolution to enable dose response profiles to be probed. The former approach requires instruments that are small, lightweight, quiet, and unobtrusive, but that provide sufficiently detailed data that quantitative dose estimates can be made. Present personal monitors for exposure to particulate matter provide too limited information for such assessments, as discussed above. Instruments are available for the latter, community monitoring approach, but they are far too expensive and complex to produce sustainable, wide-reaching networks. To develop the comprehensive exposure/dose data that are needed, these limitations to data collection and trade-offs between spatiotemporal coverage of exposures and the accuracy and resolution with which they can be characterized must be overcome.

The transformation of high resolution measurement techniques into ones that are suitable for personal monitoring or the creation of dense and extensive networks

of sensors involves compromises in instrument design. In this study, we propose that full characterization of particle size distributions need not be time and resource intensive, and should be pursued for the great benefits this information would provide to health studies.

We focus on the technology employed by DMA systems for our discussion of how to develop approaches suitable for personal monitoring or dense networks. A number of researchers have demonstrated the value of the information on ultrafine particles that is provided to health effects research by state-of-the-art aerosol mobility analyzers (Kozawa et al. 2009; Zhu et al. 2002, 2004, among others). Because ultrafine particles deposit primarily by Brownian diffusion (Rissler et al. 2012), and the electrical mobility of an aerosol particle, Z_p , with charge q is proportional to its diffusivity, D , i.e., $Z_p = qD/kT$, where k and T are the Boltzmann constant and temperature, respectively, mobility analysis measures the property that determines where and at what rate particles will deposit in the airways. A number of investigators have, therefore, proposed and developed variants of these instruments that show promise for making the data they can provide more accessible to the health effects research community (Flagan 2004; Mui et al. 2013; Qi et al. 2008; Qi and Kulkarni 2012; Ranjan and Dhaniyala 2008).

The question remains, however: How good must the measurement be to adequately characterize deposited dose? We propose relaxing the requirements on instrument performance to enable design of simpler instruments still capable of recovering sufficient size information to infer lung dose by number, surface area, or mass metrics at the ‘human receptor’ (Nazaroff 2008). To demonstrate the validity of this approach, we define a model instrument and quantitatively evaluate how the information recovered from this system is influenced by relaxing design constraints on its operating performance.

Our simulated measurements are based on the theory of mobility analysis, and we relax the key parameter that characterizes a differential mobility analyzer, its resolution, R . The value of R describes the sizing ability of the instrument and is the ratio of the mobility of the particle that is transmitted through the classification region with the highest efficiency to the range of mobilities that is transmitted with half that peak efficiency, i.e., $R = \frac{Z_p^*}{\Delta Z_{fwhm}}$. For an ideal DMA, the resolution is determined by the ratio of the sheath flow rate to that of the incoming aerosol sample, i.e., $R_{ideal} = \frac{Q_{sh}}{Q_a}$. This is achieved for large particles for which the classification

voltage is high. For small, highly mobile particles, Brownian diffusion degrades the resolution of the DMA. This variation of the response is described by a semi-analytical transfer function that was derived by Stolzenburg (1988) (see also Flagan 1999), along with the charging probability (Fuchs 1963; Hoppel and Frick 1986; López-Yglesias and Flagan 2013; Wiedensohler 1988) and empirical, instrument specific transmission and counting efficiency curves.

Using well-established models of the instrument performance, we simulate measurements using the DMA system according to the principles described by Wiedensohler et al. (2012). Geometry parameters for our modeled instrument are defined according to established instrument design principles (Flagan 1999), and the sampled aerosol flow rate is fixed as required by commonly used CPC detectors, but we allow the number and spacing of sample channels to vary with an assigned flow rate ratio, which also defines the operating resolution. For this, we define the number and spacing of sample channels such that the channels are equally spaced on a logarithmic scale for particle mobility, with overlap in the ideal triangular transfer functions to sample all mobilities within the measurement range. We express this as

$$n_{channels} = \left\lceil \frac{\ln\left(\frac{Z_{min}}{Z_{max}}\right)}{\ln\left(\frac{1-\beta}{1+\beta}\right)} \right\rceil + 1 \quad (5.4)$$

where Z_{min} and Z_{max} are the minimum and maximum particle mobilities sampled, respectively, and $\beta = \frac{1}{R_{ideal}} = \frac{Q_a}{Q_{sh}}$. The mobility centroid, $Z_{p,i}^*$, of the i th channel for a cylindrical DMA column is related to other instrument parameters by (Knutson and Whitby 1975)

$$Z_{p,i}^* = \frac{Q_{sh} + Q_e}{4\pi L V_i} \ln\left(\frac{R_2}{R_1}\right) \quad (5.5)$$

where Q_{sh} is the volumetric sheath flow, Q_e is the volumetric excess flow, L is the length of the DMA column, V_i is the negative potential applied to the inner cylinder, and R_1 and R_2 are the radial location of the aerosol exit and entrance, respectively. The maximum mobility (smallest size) that can be classified with reasonable resolution is that transmitted at a classification voltage of about 10 V. The minimum mobility (largest size) is that transmitted at a classification voltage of about 10 kV, before electrostatic breakdown. Thus, by defining the operating voltages V_i to range from 10 V to 10 kV and setting the instrument geometry but allowing the ratio of the flow rates to vary, the targeted mobilities $Z_{p,i}^*$ will vary, and the number of sample channels will be adjusted accordingly.

The instrument simulated is the long column DMA (e.g., TSI Model 3081, TSI Inc., USA), which is widely used in the form of the so-called scanning mobility particle sizer (SMPS). We do not simulate scanning operation, however, as a number of additional challenges are introduced to the analysis of scanning DMA data by the slow response time of many commonly used CPCs. Without an accurate description of the nonunique relationship between the time a particle exits a scanning-mode DMA and the time it is ultimately detected, inverted distributions will be distorted. The simulations presented here consider stepping the DMA voltage instead, which avoids the additional distortions. Some CPCs do indeed have fast enough response times that SMPS measurements closely approximate stepping-mode DMA operation.

We examine both existing and proposed applications of particle size distribution measurements from which regional deposition can be calculated. We begin by considering the present state-of-the-art in which the DMA is operated with resolution $R_{ideal} = 10$, typical of present-day ambient measurements with this technique. We perform additional simulations with $1.1 < R_{ideal} < 10$, to span a broad range of reduced sizing resolution. We address the measurement uncertainties and inversion analysis as described above for the measures of distributed properties. Additional details on the instrument simulation are presented in Appendix A.

5.3 Evaluation of Measurement Systems

The evaluation methodology illustrated in Figure 5.1 was introduced in Chapter 4. So far in the present study, we have addressed the measurement simulation portion of this procedure in depth. Now, we briefly reintroduce the portions dedicated to defining input size distributions and making the final quantitative comparisons.

For the unbiased descriptions of input aerosol, we use the simulated time evolution of an aerosol particle population during a new particle formation event that was introduced in Chapter 4 (see Figure 4.4). This provides a realistic, transient exposure scenario that includes particles in the low nanometer size regime and extends throughout the submicron size regime.

From our unbiased descriptions of input aerosol, we employ the ICRP model of lung deposition to calculate total and regional deposited dose, in terms of mass, surface area, and number of particles. These values are the “true dose.” Details on the parameters used in the ICRP model are provided in Appendix B.

For the measurements providing integral measures of particle exposure (e.g., $PM_{2.5}$) we compare measured responses (e.g., values calculated from 5.1) directly to the

“true dose” to uncover any possible correlation relationships. It is possible to reveal that a measurement can be used as proxy for the deposited dose.

For measurements providing direct estimates of dose (e.g., RDS), as well as for measurements providing size-distributed information from which dose estimates can be calculated using Eq. (5.2), we calculate the bias in the estimated dose values as the percent difference from the “true dose” value.

$$\% \text{ Bias} = \frac{\text{estimated dose} - \text{"true dose"}}{\text{"truedose"}} \times 100 \quad (5.6)$$

5.4 Results and Discussion

We calculate ideal estimates of ten-minute- and hourly-average doses to the extrathoracic, bronchial, bronchiolar, and alveolar/interstitial regions based on number, surface area, and mass metrics for the 24 hour exposure scenario shown in Figure 4.4. We compare these ideal dose estimates with simulated measurements made by different techniques. Figure 5.8 compares various integral measures to ideal estimates of mass, surface area, and number of particles deposited in the alveolar region. The first column shows correlation plots for hourly average $\text{PM}_{2.5}$ mass concentration exposures. The second and third columns show ten-minute-average concentrations of number and surface area concentrations, respectively. These represent measurements that would be collected using a CPC, or using a sampler capable of measuring total particle surface area. Similar comparisons for other regions of the respiratory tract are collected in an additional results section at the end of this chapter.

The true mass dose correlates well with the $\text{PM}_{2.5}$ mass exposure (linear correlation coefficient, $r = 0.99$). Small deviations from the perfect correlation occur when the atmospheric aerosol under consideration is in either the very early or very late hours of the simulation day. During both periods, the aerosol is an aged one, concentrated in the $50 \text{ nm} < D_p < 500 \text{ nm}$ size regime where many of the inhaled particles are exhaled rather than deposited. During the nucleation burst, from about 10:00 through 15:00, the mass deposited in the alveolar region is reasonably well correlated with $\text{PM}_{2.5}$.

The correlation of deposited surface area with $\text{PM}_{2.5}$ is slightly weaker (linear correlation coefficient, $r = 0.94$). Over the first eight hours of the simulation day, particles grow, but the number concentration and total surface area decrease due to coagulation, leading to trends in number and surface area dose that are orthogonal to that of $\text{PM}_{2.5}$. As the inversion layer lifts between 07:00 and 11:00, the mass and number concentrations diminish, as does the surface area. When

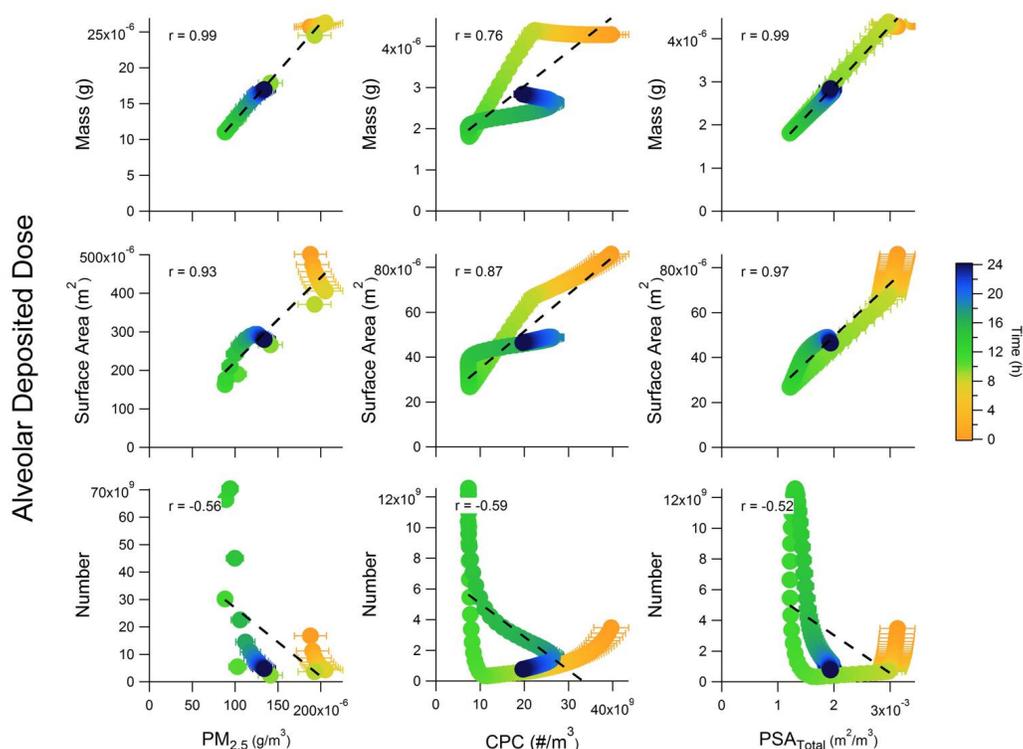


Figure 5.8: Correlation of $\text{PM}_{2.5}$, CPC, and (theoretical) total surface area measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

nucleation commences, the surface area increases more rapidly than does $\text{PM}_{2.5}$. Thus, mass and surface area data follow a rough general trend, but they show systematic differences.

In contrast, the number dose is uncorrelated with $\text{PM}_{2.5}$ (linear correlation coefficient, $r = -0.57$). The aged aerosol has a relatively low number concentration though the mass concentration is high. The deposited number increases dramatically during the nucleation burst.

Table 5.2 summarizes the correlation coefficients for $\text{PM}_{2.5}$ and other integral measures of ambient exposure with various metrics of deposition within in the lungs. As with the $\text{PM}_{2.5}$ sampler, integral measures of number or surface area metrics alone fail to correlate with measures of dose outside their targeted metric, potentially overlooking important links to specific outcomes.

Figure 5.10 compares the hourly average mass, surface area, and number of particles deposited in the alveolar region with hourly dose measurements from the regional

Table 5.2: Correlation coefficients (Pearson’s r) for $PM_{2.5}$ and other integral measures of ambient exposure with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	$PM_{2.5}$	CPC	PSA_{total}
Mass Dose to AI	0.997	0.768	0.997
Surface Area Dose to AI	0.937	0.878	0.974
Number Dose to AI	-0.561	-0.596	-0.529
Mass Dose to TB	0.996	0.770	0.998
Surface Area Dose to TB	0.933	0.868	0.972
Number Dose to TB	-0.576	-0.682	-0.568
Mass Dose to Head	0.998	0.699	0.983
Surface Area Dose to Head	0.998	0.765	0.994
Number Dose to Head	-0.575	-0.689	-0.570
Mass Dose Total	0.999	0.735	0.992
Surface Area Dose Total	0.967	0.847	0.991
Number Dose Total	-0.583	-0.679	-0.571

deposition sampler (RDS). Similar comparisons for other regions of the respiratory tract are shown in the additional results section. The RDS provides mass measurements that are almost perfectly correlated with the true alveolar mass dose. This is not surprising given that the RDS sampler has been designed to reproduce the alveolar mass deposition efficiency. However, like the $PM_{2.5}$ sampler, the RDS mass dose is poorly correlated with the alveolar dose of surface area or number of particles. As suggested by Bartley and Vincent (2011), additional dose estimates could be obtained from such a sampler if surface area or particle count were also measured from the collected aerosol, but then multiple measurements would be required to capture the different metrics of interest.

Table 5.4 summarizes the correlation coefficients for three different theoretical samplers created from the ideal application of the RDS sampling efficiency combined with detection of the various particle metrics. Each of these three ideal regional lung deposition samplers recovers its intended metric well, but cannot reveal other

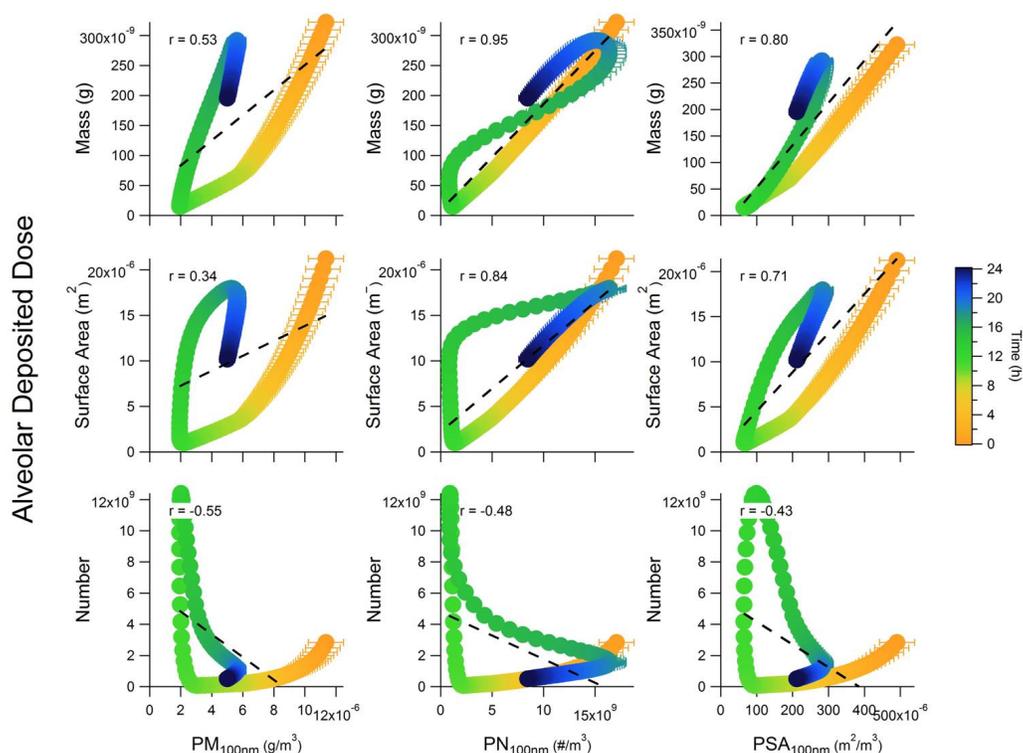


Figure 5.9: Correlation of (theoretical) $PM_{100\text{ nm}}$, $PN_{100\text{ nm}}$, and $PSA_{100\text{ nm}}$ measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

metrics of dose.

Our results so far show that single-metric samplers can accurately reflect the dose of the targeted quantity, but fail to reflect other measures of dose. Efforts to establish statistical links between measures of airborne particulate matter and health impacts may suffer similar deviations if the wrong metric is targeted. An alternate approach is to use a measurement method that characterizes the aerosol in greater detail. The differential mobility analyzer can provide such data. We first consider the information that can be recovered from a DMA operated with $R_{ideal} = 10$, typical of present-day applications of this technique. We obtain the measured dose estimates shown in Figure 5.11 by simulating the measurements of a TSI long column DMA, inverting the data as discussed in detail in Chapter 2, and applying Eq. (5.2) to the recovered particle number size distributions. Error bars are derived from repeating the data simulation and inversion several hundred times while including simulated statistical noise in the measurements due to small count number. When all instrument efficiencies are appropriately accounted for, the number of particles transmitted

Table 5.3: Correlation coefficients (Pearson’s r) for theoretical integral measures of ambient exposure with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	PM _{100 nm}	PN _{100 nm}	PSA _{100 nm}
Mass Dose to AI	0.538	0.954	0.803
Surface Area Dose to AI	0.357	0.841	0.716
Number Dose to AI	-0.554	-0.480	-0.433
Mass Dose to TB	0.539	0.952	0.802
Surface Area Dose to TB	0.297	0.903	0.677
Number Dose to TB	-0.607	-0.603	-0.602
Mass Dose to Head	0.545	0.951	0.803
Surface Area Dose to Head	0.284	0.795	0.664
Number Dose to Head	-0.612	-0.615	-0.623
Mass Dose Total	0.539	0.953	0.803
Surface Area Dose Total	0.331	0.830	0.704
Number Dose Total	-0.609	-0.591	-0.583

through the DMA in any given measurement channel may be small, giving rise to large relative uncertainties in measured counts. We quantify these measurement uncertainties so that we can consider them separately from the systematic errors arising from the overall measurement process as dictated by instrument performance.

Figure 5.11 shows that the DMA can capture the different measures of dose accurately when operated under the high resolution of $R_{ideal} = 10$. Table 5.5 summarizes the correlation coefficients for estimates of dose from DMA measurements with calculated true values of dose within in the lungs. That high resolution ($R_{ideal} = 10$) DMA measurements accurately capture dose estimates to all regions in all metrics is not surprising since particle size distribution recovery from DMA measurements at this resolution is highly accurate. Yet the design and fabrication of mobility analyzers can be made much simpler and less costly if R_{ideal} is reduced.

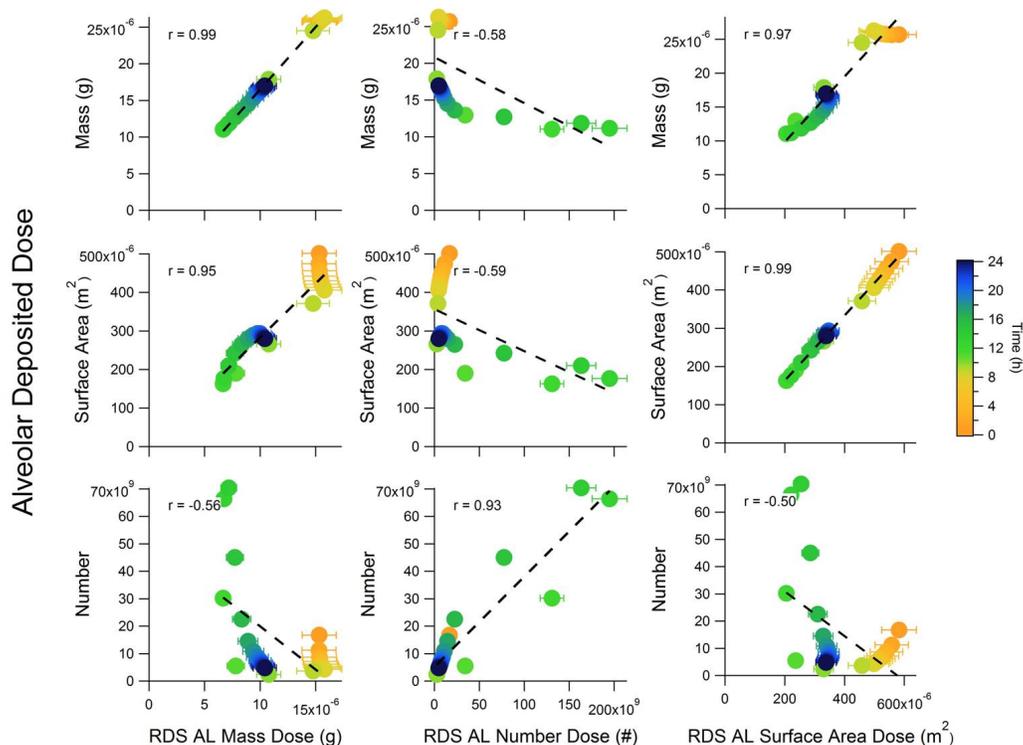


Figure 5.10: Correlation of dose values collected with regional deposition samplers (RDS, ideal sampling efficiency and detection of specific particle metrics) to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

Relaxing DMA Resolution

Figure 5.12 maps the deviations in hourly average dose estimates as a function of R_{ideal} throughout the same New Delhi exposure scenario. For resolving power below 2, the deviations can be substantial, particularly during the nucleation burst. Moreover, deviations may be either positive or negative, depending upon which portion of the size range dominates the size distribution. Once the limiting resolution exceeds 3, deviations are appreciable only at the peak of the nucleation burst, around noon when there is a pronounced peak in the number concentration of ultrafine particles.

Figure 5.13 presents the 24-hour average deviation, collapsing the three-dimensional bias map onto two dimensions to reveal a clear overarching trend for DMA measurements gathered under the range of conditions considered here. The deviations increase as resolving power decreases, but the error builds slowly for a resolution greater than about three. As shown in Figure 5.14, the estimates of mass, surface area, and number dose obtained from a mobility analyzer operated at $R_{ideal} = 3$ are

Alveolar Deposited Dose
DMA Estimate ($R=10$) vs True Dose

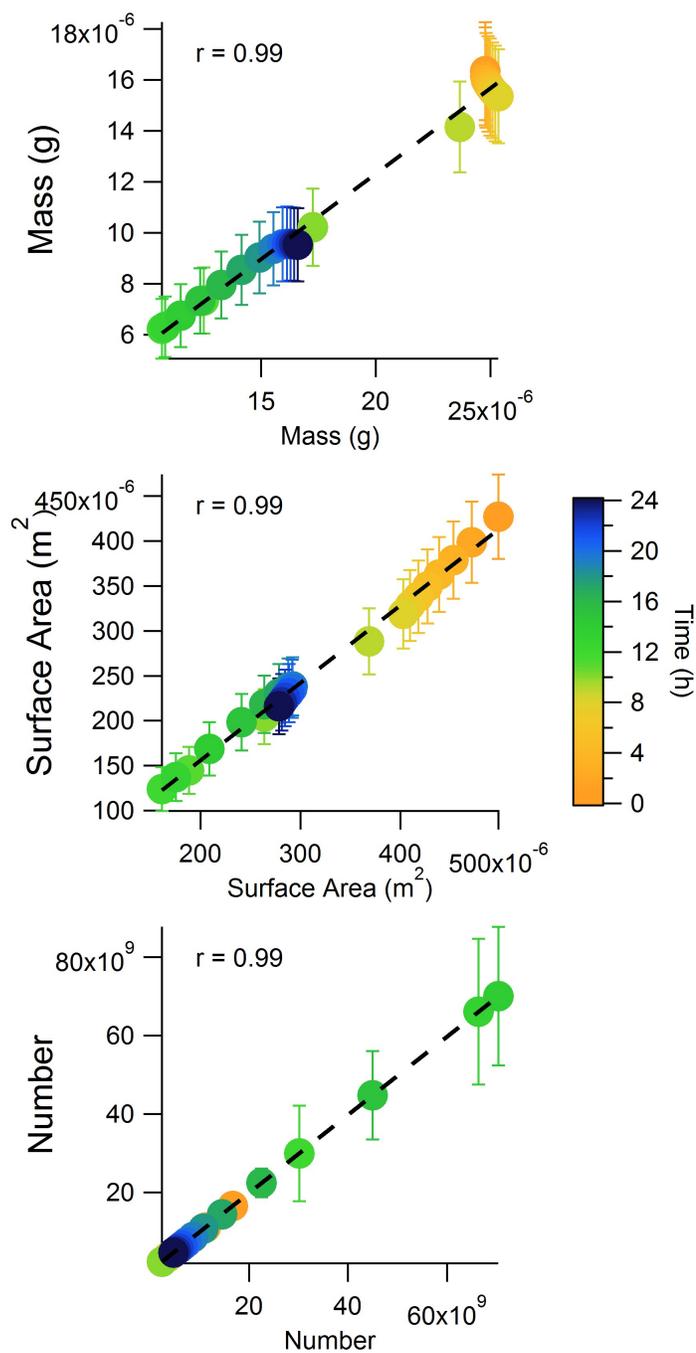


Figure 5.11: Correlation of dose values estimated from high resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 10$) to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

Table 5.4: Correlation coefficients (Pearson’s r) for dose measures from regional deposition samplers (RDS, ideal sampling efficiency and detection of specific particle metrics) with values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	RDS _M	RDS _{SA}	RDS _N
Mass Dose to AI	0.999	0.974	-0.582
Surface Area Dose to AI	0.957	0.997	-0.590
Number Dose to AI	-0.573	-0.503	0.937
Mass Dose to TB	0.998	0.976	-0.573
Surface Area Dose to TB	0.969	0.996	-0.553
Number Dose to TB	-0.558	-0.591	0.997
Mass Dose to Head	0.999	0.945	-0.560
Surface Area Dose to Head	0.992	0.972	-0.593
Number Dose to Head	-0.540	-0.599	0.991
Mass Dose Total	0.999	0.961	-0.571
Surface Area Dose Total	0.973	0.999	-0.593
Number Dose Total	-0.562	-0.586	0.999

highly correlated with true dose (linear correlation coefficient, $r = 0.99$).

5.5 Conclusions

While numerous studies suggest that airborne particulate matter is responsible for a range of adverse health effects, the question of what measure of exposure or dose is most closely associated with different outcomes remains open. The present, *in silico*, study of a range of measurement methods reveals that single-metric instruments cannot provide the data to answer the question of which parameter to use. Moreover, different measures may be appropriate to different aerosol exposure scenarios. The PM_{2.5} metric was developed to satisfy a regulatory need, but it misrepresents dose for ultrafine particles. Instruments that mimic the PM_{2.5} sampler by light scattering measures may perform comparably to mass measurements for aged aerosols, but exacerbate the deficiencies of this metric when ultrafine particles are abundant since such particles scatter disproportionately little light.

Table 5.5: Correlation coefficients (Pearson's r) for estimates of dose from DMA measurements with the values calculated as "true" mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

	DMA, $R = 10$	DMA, $R = 3$
Mass Dose to AI	0.995	0.999
Surface Area Dose to AI	0.997	0.999
Number Dose to AI	0.999	0.999
Mass Dose to TB	0.997	0.999
Surface Area Dose to TB	0.998	0.999
Number Dose to TB	0.999	0.993
Mass Dose to Head	0.997	0.992
Surface Area Dose to Head	0.989	0.997
Number Dose to Head	0.999	0.991
Mass Dose Total	0.994	0.996
Surface Area Dose Total	0.992	0.998
Number Dose Total	0.999	0.989

To fully resolve the role of ultrafine particles in the adverse health impacts of particulate air pollution in the community at large and in the vicinity of localized sources, workplace exposures to engineered nanoparticles, and in other situations, a new paradigm for aerosol measurement is needed. The aerosol research community has made substantial advances in particle measurements, with much of the effort being focused on improving the ability to resolve fine detail about the nature of the particles in the air. These developments have enabled major progress in understanding the fundamental chemistry and physics of the atmospheric aerosol, but the resulting instruments are too large, complex, and expensive to meet the needs of the health research community.

Instead, small, simple, and low-cost instruments that provide data on dose of particulate matter to different regions of the respiratory tract are needed. This paper has compared and contrasted a number of steps in that direction. Though single-metric samplers cannot address the appropriateness of different measures, a suite of samplers or instruments, each designed to probe a single metric can, provided the

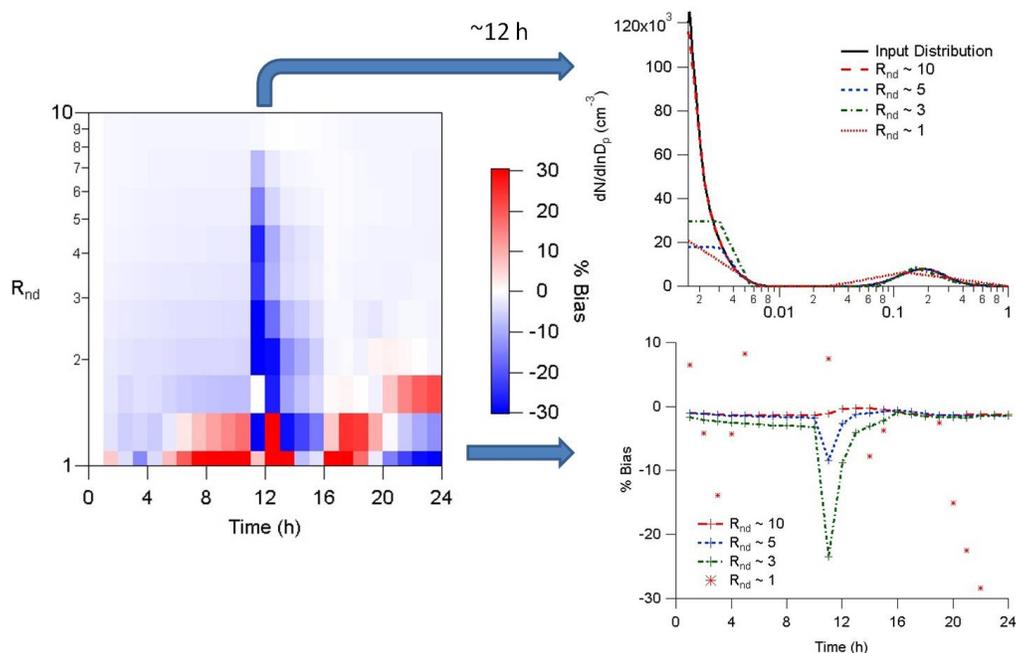


Figure 5.12: Percent bias in dose values estimated from DMA measurements collected under a range of operating conditions compared to those calculated as “true” hourly deposited number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

sampler is designed to measure dose rather than exposure. An alternate approach to capturing various measures of dose is to measure the full particle size distribution. Measurements made with current DMAs operating as usually used today can readily capture the different exposure or dose measures. However, instruments operating at that resolution fall under the category of being too big, complex, and costly for use as personal monitors, or for dense networks. Fortunately, the present simulations demonstrate that much simpler, lower resolution mobility analyzers can provide data with sufficient resolution to serve the needs of health effects researchers.

Small, simplified versions of the DMA have been built (Zhang and Wexler 2006), as have a number of simpler mobility analyzers that do not fully separate particles into differential fractions of particles within a narrow range of mobilities (Chua et al. 2009; Li et al. 2009). The opposed migration aerosol classifier (OMAC; Downard et al. 2011; Flagan 2004; Mui et al. 2013) affords the same functionality of the DMA with a design that is better suited to simplification and miniaturization.

The present study has assessed the suitability of a number of different measurement methods for health effects research. We employed a well-studied type of air pollution

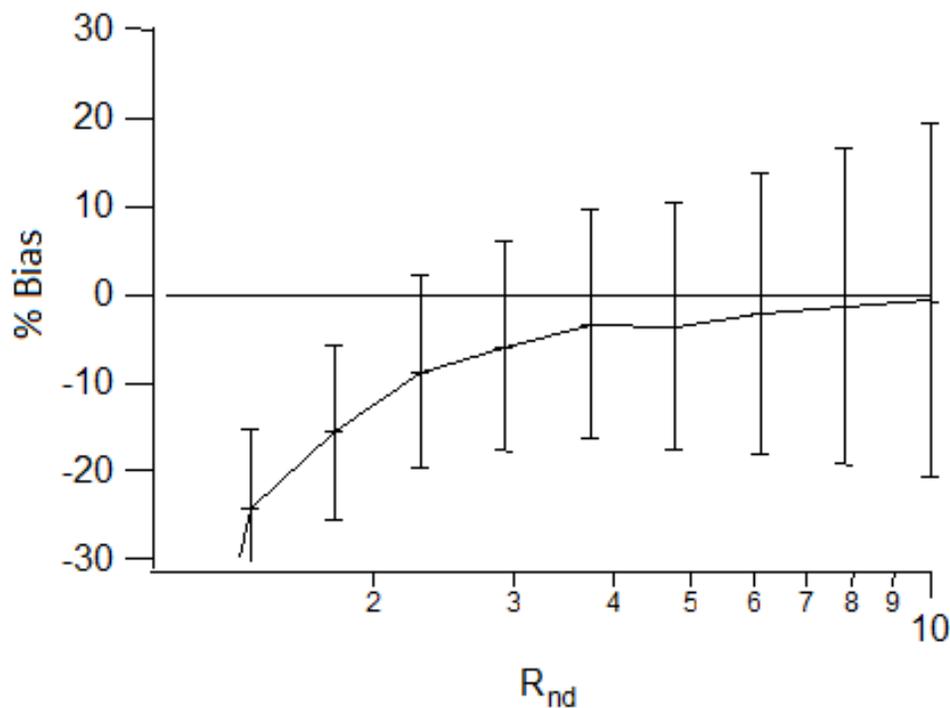


Figure 5.13: Percent bias in dose values estimated from DMA measurements collected under a range of operating conditions compared to those calculated as “true” 24-hour deposited number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

event that spans a wide range of exposures to probe the uncertainties introduced by different measurement scenarios. Provided the performance characteristics of the measurement devices are well understood, this scenario enables quantitative estimations of the measurement errors or uncertainties, as well as subsequent estimation of lung deposited dose values.

Aerosol deposition is highly dependent on the individual (Lippmann 1977) and not trivial to replicate in a sampling device. While a suite of specially designed, single-metric samplers could provide good dose estimates for a specific subpopulation, e.g., healthy young men, measurements that enable dose estimation according to different respiratory tract deposition profiles broaden the applicability of the data to represent different sensitive groups, such as children, the elderly, or those with compromised health. Estimations of dose can also vary with assumptions about activity level throughout the exposure period. A measurement approach that collects

Alveolar Deposited Dose
DMA Estimate ($R=3$) vs True Dose

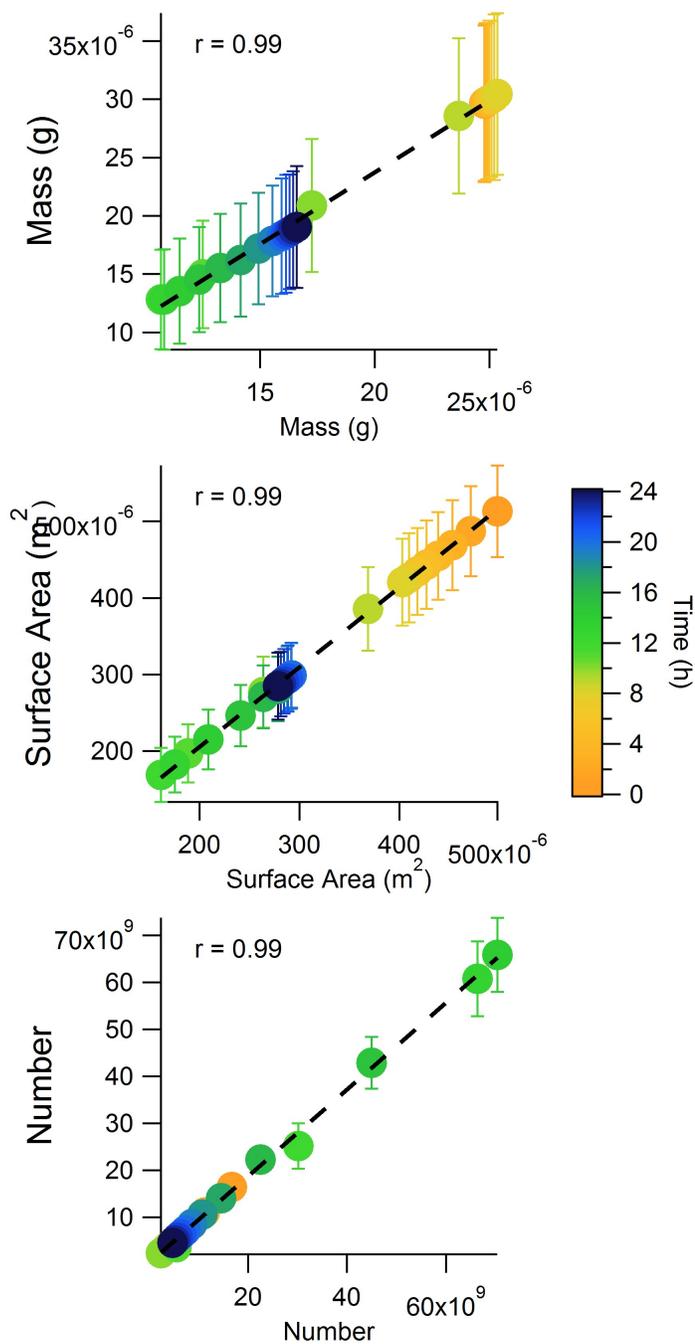


Figure 5.14: Correlation of dose values estimated from low resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 3$) to those calculated as “true” mass, surface area, and number dose delivered to the alveolar/interstitial region for the nucleation simulation time-series of size distributions.

the appropriate quality of information, as described in this study, such that the data can be analyzed and reanalyzed with tailored parameters would provide the most helpful estimates of lung dose for particle health studies.

Any limitations or biases in these measurements will affect the conclusions drawn in PM health effect studies. Various metrics of PM exposure and dose (i.e. number, mass, or surface area concentrations) have potential importance for health effects. Thus, health studies should seek PM measurements that provide enough detail on particle characteristics to investigate different physical metrics of total and regional particle deposition in the human lungs.

In this study, we propose that full characterization of particle size distributions need not be time and resource intensive, and should be pursued for the great benefits this information would provide to health studies. Our analysis contrasts the nature of the health-related information that existing technology can provide with the performance criteria that future technology should target. This demonstration reveals the need for measurement systems with “middle ground” capabilities. We propose relaxing the requirements on DMA instrument performance to enable design of simpler instruments still capable of recovering sufficient size information to infer lung dose. Together with records of individual breathing profiles, size distribution information would permit accurate estimation of tissue doses through a set of relatively simple calculations, and would allow a single data set to be applied to different populations by applying the appropriate respiratory tract deposition profiles.

5.6 Additional Measurement Evaluation Results

The respiratory tract is often considered to consist of three anatomically and functionally distinct units: (a) the head/throat region, from the mouth and nose to the larynx; (b) the tracheobronchial region from the larynx through the conducting airways; and (c) the pulmonary/alveolar region, or the gas exchange zone. Lung regions defined in this way (combining the bronchial and bronchiolar regions of the original ICRP model into one tracheobronchial region) are used for the ICRP grand average deposition model based on Hinds' (1999) parameterization, on which the regional deposition sampler (RDS) is based. Additional dose comparisons for these regions of the respiratory tract are shown in Figures 5.15 through 5.22.

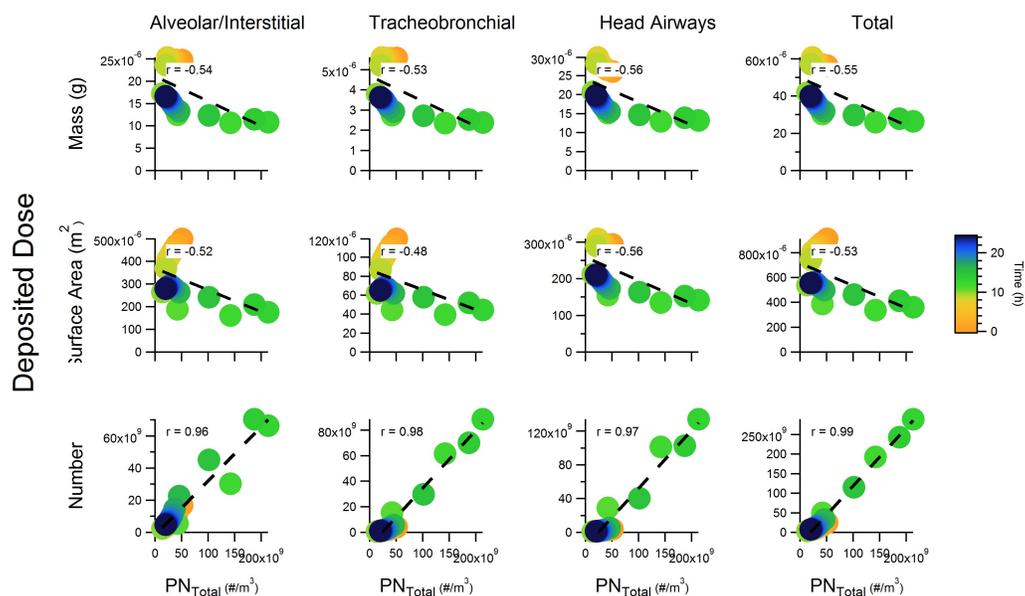


Figure 5.15: Correlation of PN_{total} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

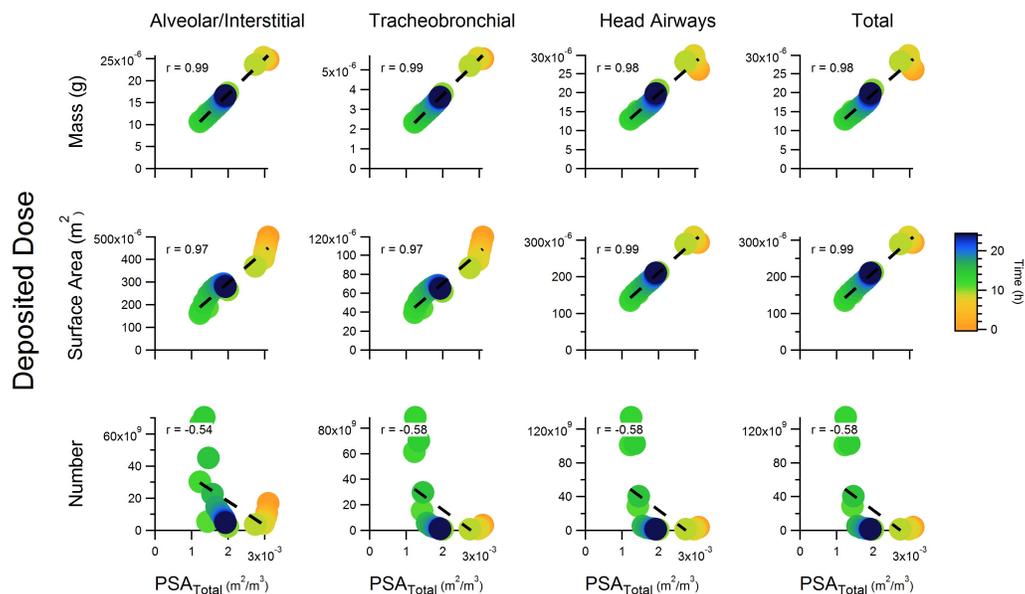


Figure 5.16: Correlation of PSA_{total} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

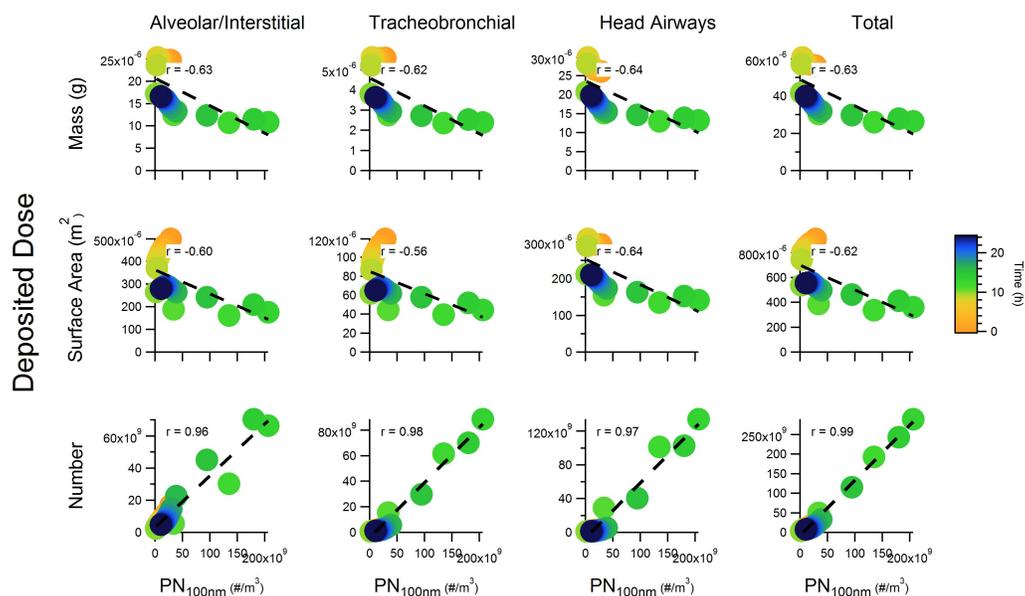


Figure 5.17: Correlation of PN_{100nm} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

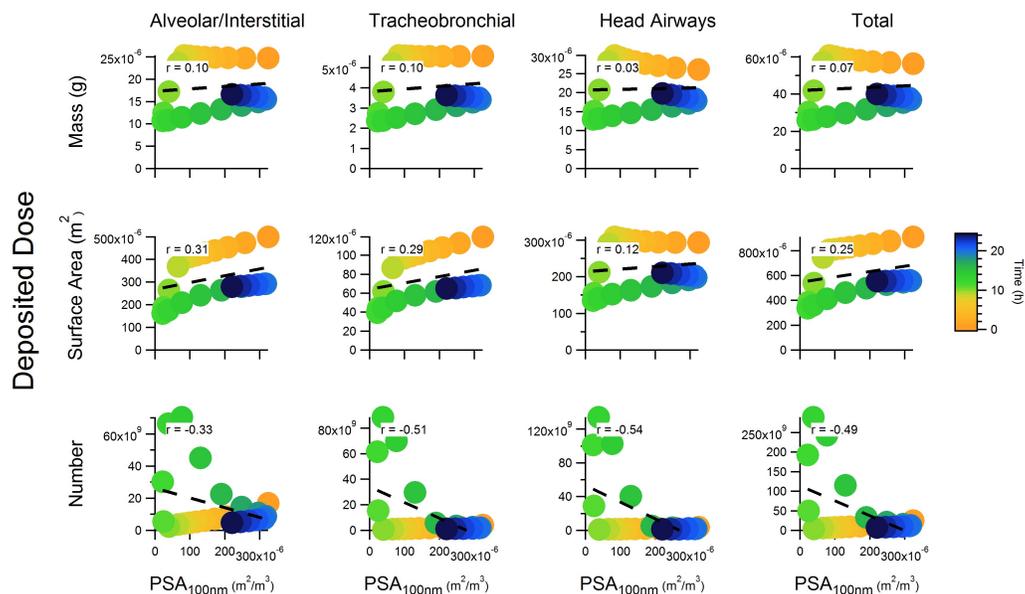


Figure 5.18: Correlation of PSA_{100nm} measurements to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

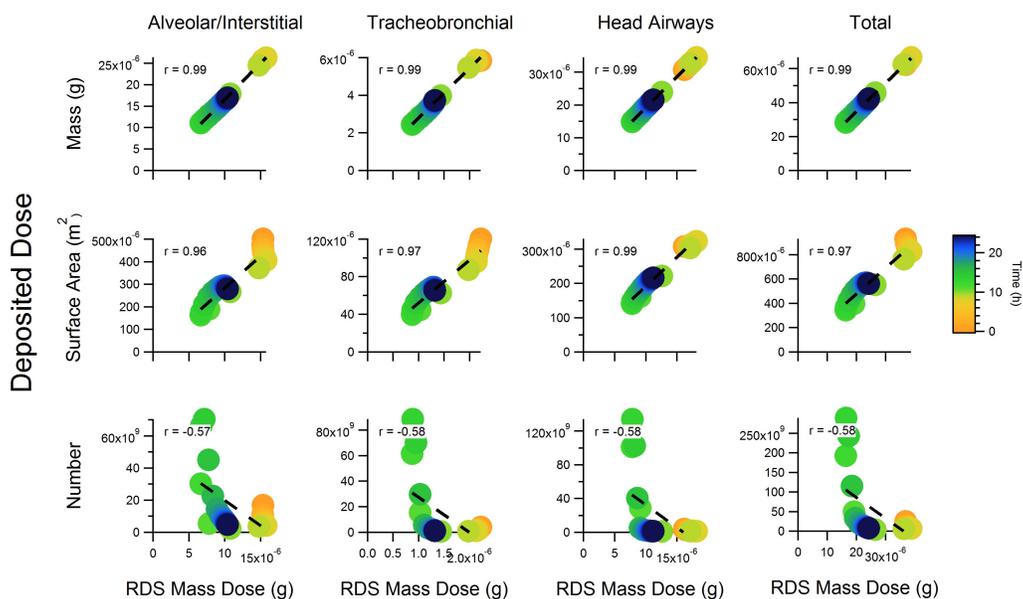


Figure 5.19: Correlation of mass dose collected with the regional deposition sampler (RDS) to values calculated as “true dose” for the nucleation simulation time-series of size distributions.

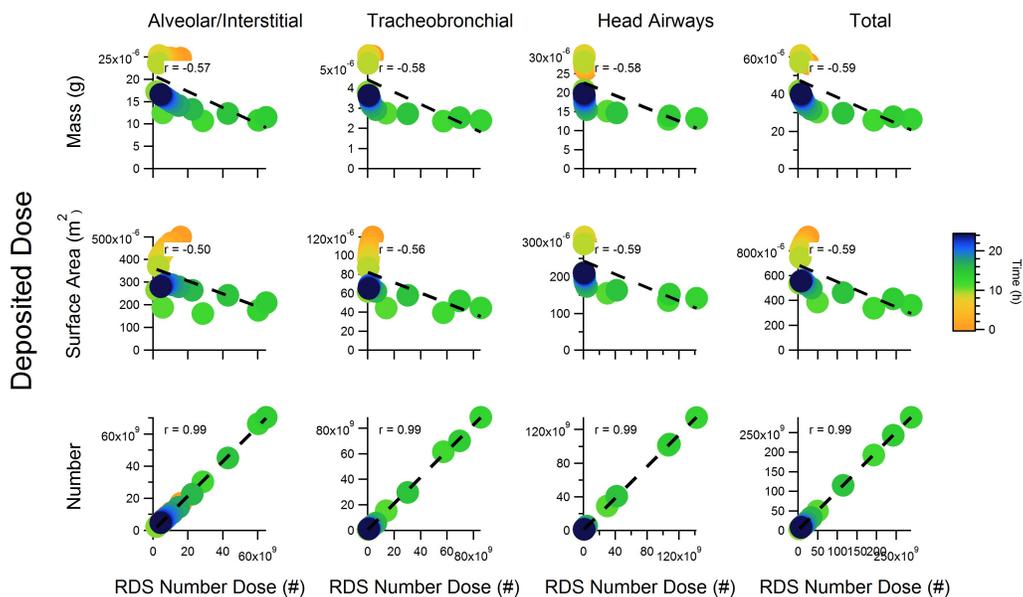


Figure 5.20: Correlation of number dose collected with a theoretical RDS to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

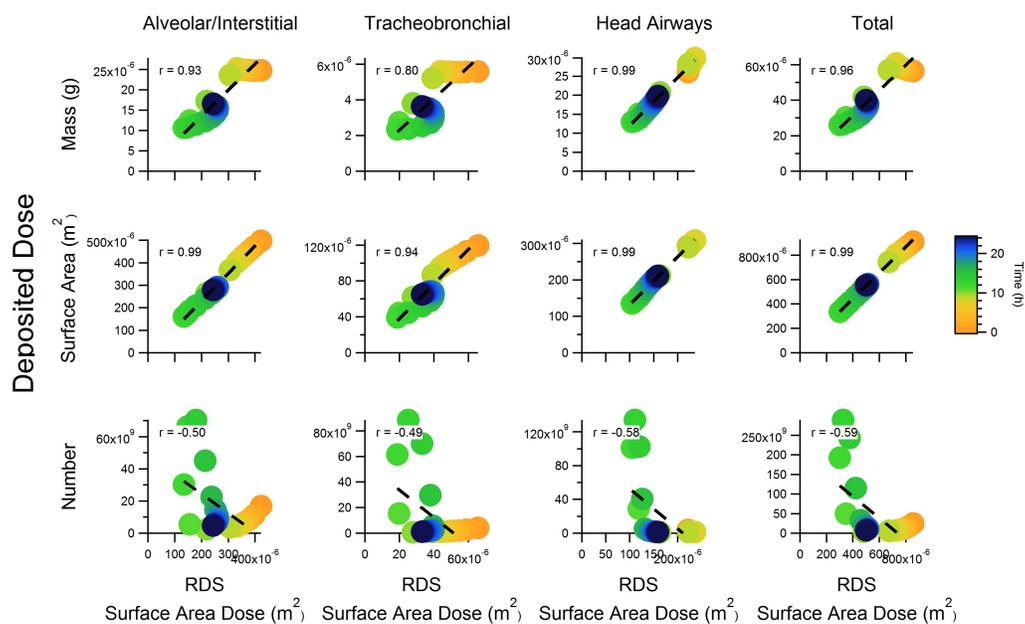


Figure 5.21: Correlation of surface area dose collected with a theoretical RDS to values calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

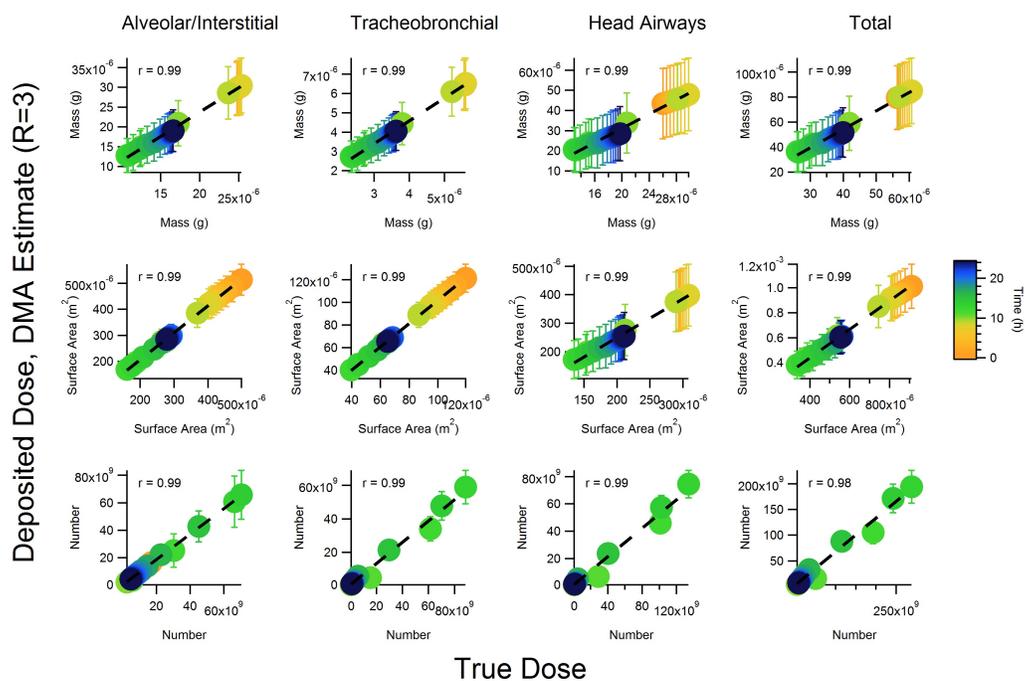


Figure 5.22: Correlation of dose values estimated from low resolution DMA measurements ($R = \frac{Q_{sh}}{Q_a} = 3$) to those calculated as “true” mass, surface area, and number dose to the alveolar/interstitial, tracheobronchial, head airways, and total combined regions of the lungs for the nucleation simulation time-series of size distributions.

Chapter 6

CONCLUSION

6.1 Summary

This thesis work, in the broadest sense, contributes to advancing our knowledge surrounding aerosol particle measurements. More specifically, the investigations presented here are geared towards optimizing the collection and analysis of these measurements for health effects research. Firstly, readers will have found an accessible discussion of the aerosol data inversion problem, which is an important data analysis task in almost any aerosol measurement campaign. Secondly, readers will have found a fresh perspective on defining the measurement system performance necessary to collect health-relevant information.

The aerosol data inversion solution method developed and discussed in Part I of this work is different from other approaches in a number of key ways: it incorporates B-splines to represent the size distribution function as a cubic spline within a discrete linear model of the inverse problem; it performs fine integration for kernel matrix elements; it is able to handle any prescription of number, spacing, and range of discrete solution nodes; and it can interface with three established numerical methods for inverse solution computation (NNLS, TNNLS, regularization). This approach shows improved particle size distribution recovery compared to other common approaches described in the literature.

The treatment and discussion in this work of all things inversion is meant to provide some clarity to what is an important, yet often incomprehensible, aspect of any measurement campaign. Beyond this contribution to aerosol research in general, the inversion routine developed here also serves to lay the groundwork for the exploration of lung tissue dose that becomes possible with access to aerosol particle size measurements.

As discussed in depth in Part II of this work, particle size distribution information can inform accurate estimation of lung tissue doses through a set of relatively simple calculations. The work presented there also reveals that full characterization of particle size distributions need not be time and resource intensive, and so should be pursued for the great benefits this information would provide to PM health-effect studies. Quantitative analyses reveal criteria for the resolution in size information

necessary to infer various metrics of respiratory dose. This is the target criteria for future developments in health-related particle measurement design.

6.2 Future Work

While the measurement evaluation procedure of Chapters 4 and 5 provides direction for the design of future health-related aerosol particle measurement systems, actually creating these samplers is the next step. Air pollution regulation and mitigation efforts need access to small, simple, and low-cost instruments that provide data on dose of particulate matter to different regions of the respiratory tract. Perhaps this can be achieved with a measurement system based on differential electrical mobility classification technology. For such a system, the size resolution requirements can be relaxed enough to simplify the system design and operation, while sufficient size information is still recovered to infer lung dose by number, surface area, or mass metrics.

Design details for this future measurement system can be guided by the measurement evaluation methodology presented in this work. As an important component of this evaluation procedure, the cubic spline inverse solution method developed here also has potential for future applications.

The inversion analysis discussed in Chapters 2 and 3 can in principle be applied to other particle sizing instruments, such as cascade impactors, optical counters, new developments in mobility analysis technology, etc., provided the performance characteristics of the measurement devices are well understood. In addition, the treatment of B-splines, and technique of integrating these basis functions into the aerosol data inversion problem may translate to other data analysis applications that deal with size distribution representations, like following particle growth and coagulation.

BIBLIOGRAPHY

- Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 10 of *Applied Mathematics Series*. US Department of Commerce, Washington, D.C.
- Ahlm, L., Liu, S., Day, D. A., Russell, L. M., Weber, R., Gentner, D. R., Goldstein, A. H., DiGangi, J. P., Henry, S. B., Keutsch, F. N., VandenBoer, T. C., Markovic, M. Z., Murphy, J. G., Ren, X., and Scheller, S. (2012). Formation and growth of ultrafine particles from secondary sources in bakersfield, california. *Journal of Geophysical Research: Atmospheres*, 117(D21):n/a–n/a.
- Alföldy, B., Giechaskiel, B., Hofmann, W., and Drossinos, Y. (2009). Size-distribution dependent lung deposition of diesel exhaust particles. *Journal of Aerosol Science*, 40(8):652–663.
- Alofs, D. J. and Balakumar, P. (1982). Inversion to obtain aerosol size distributions from measurements with a differential mobility analyzer. *Journal of Aerosol Science*, 13(6):513–27.
- Bartley, D. L. and Vincent, J. H. (2011). Sampling conventions for estimating ultrafine and fine aerosol particle deposition in the human respiratory tract. *Ann Occup Hyg*, 55(7):696–709. Bartley, David L Vincent, James H eng Research Support, U.S. Gov't, P.H.S. England 2011/07/13 06:00 Ann Occup Hyg. 2011 Aug;55(7):696-709. doi: 10.1093/annhyg/mer037. Epub 2011 Jul 11.
- Bashurova, V. S., Koutzenogil, K. P., Pusep, A. Y., and Shokhirev, N. V. (1991). Determination of atmospheric aerosol size distribution functions from screen diffusion battery data: Mathematical aspects. *Journal of Aerosol Science*, 22(3):373–388.
- Brouwer, D., van Duuren-Stuurman, B., Berges, M., Jankowska, E., Bard, D., and Mark, D. (2009). From workplace air measurement results toward estimates of exposure? development of a strategy to assess exposure to manufactured nano-objects. *Journal of Nanoparticle Research*, 11(8):1867–1881.
- Brunekreef, B. and Forsberg, B. (2005). Epidemiological evidence of effects of coarse airborne particles on health. *Eur Respir J*, 26(2):309–18. Brunekreef, B Forsberg, B eng Research Support, Non-U.S. Gov't Review Denmark 2005/08/02 09:00 Eur Respir J. 2005 Aug;26(2):309-18.
- Chua, B., Wexler, A. S., Tien, N. C., Niemeier, D. A., and Holmen, B. A. (2009). Electrical mobility separation of airborne particles using integrated microfabricated corona ionizer and separator electrodes. *Journal of Microelectromechanical Systems*, 18(1):13–20.

- Collins, D. R., Flagan, R. C., and Seinfeld, J. H. (2002). Improved inversion of scanning dma data. *Aerosol Science and Technology*, 36(1):1–9.
- Council, N. R. (2004). Research priorities for airborne particulate matter: Iv. continuing research progress. Technical report.
- Crump, J. G. and Seinfeld, J. H. (1982). A new algorithm for inversion of aerosol size distribution data. *Aerosol Science and Technology*, 1(1):15–34.
- Dockery, D. W., Pope, C. Arden, r., Xu, X., Spengler, J. D., Ware, J. H., Fay, M. E., Ferris, B. G., and Speizer, F. E. (1993). An association between air pollution and mortality in six u.s. cities. *New England Journal of Medicine*, 329:1753–1759.
- Downard, A. J., Dama, J. F., and Flagan, R. C. (2011). An asymptotic analysis of differential electrical mobility classifiers. *Aerosol Science and Technology*, 45(6):727–739.
- Elihn, K., Berg, P., and Lidén, G. (2011). Correlation between airborne particle concentrations in seven industrial plants and estimated respiratory tract deposition by number, mass and elemental composition. *Journal of Aerosol Science*, 42(2):127–141.
- Esmen, N. A., Johnson, D. L., and Agron, G. M. (2002). The variability of delivered dose of aerosols with the same respirable concentration but different size distributions. *Annals of Occupational Hygiene*, 46(4):401–407.
- Fissan, H., Neumann, S., Trampe, A., Pui, D. Y. H., and Shin, W. G. (2006). Rationale and principle of an instrument measuring lung deposited nanoparticle surface area. *Journal of Nanoparticle Research*, 9(1):53–59.
- Flagan, R. C. (1999). On differential mobility analyzer resolution. *Aerosol Science and Technology*, 30:556–70.
- Flagan, R. C. (2004). Opposed migration aerosol classifier (omac). *Aerosol Science and Technology*, 38(9):890–899.
- Flagan, R. C. (2008). Differential mobility analysis of aerosols: A tutorial. *KONA Powder and Particle Journal*, 26:254–68.
- Fuchs, N. A. (1963). On the stastation charge distribution on aerosol particles in a bipolar ionic atmosphere. *Geofisica pura e applicata*, 56(1):185–193.
- Gorner, P., Wrobel, R., Micka, V., Skoda, V., Denis, J., and Fabries, J. F. (2001). Study of fifteen respirable aerosol samplers used in occupational hygiene. *Ann Occup Hyg*, 45(1):43–54.
- Hagen, D. E. and Alofs, D. J. (1983). Linear inversion method to obtain aerosol size distributions from measurements with a differential mobility analyzer. *Aerosol Science and Technology*, 2(4):465–475.

- Hinds, W. C. (1999). Respiratory deposition. In *Aerosol Technology: Properties, Behavior, and Measurement of Airborne Particles*, chapter 11. Wiley.
- Hoppel, W. A. (1978). Determination of the aerosol size distribution from the mobility distribution of the charged fraction of aerosols. *Journal of Aerosol Science*, 9:41–54.
- Hoppel, W. A. and Frick, G. M. (1986). Ion-aerosol attachment coefficients and the steady-state charge distribution on aerosols in a bipolar ion environment. *Aerosol Science and Technology*, 5(1):1–21.
- Hussein, T., Löndahl, J., Paasonen, P., Koivisto, A. J., Petaja, T., Hameri, K., and Kulmala, M. (2013). Modeling regional deposited dose of submicron aerosol particles. *Sci Total Environ*, 458-460:140–9. Hussein, Tareq, Löndahl, Jakob, Paasonen, Pauli, Koivisto, Antti, Joonas, Petaja, Tuukka, Hameri, Kaarle, Kulmala, Markku. eng Research Support, Non-U.S. Gov't Netherlands 2013/05/07 06:00 Sci Total Environ. 2013 Aug 1;458-460:140-9. doi: 10.1016/j.scitotenv.2013.04.022. Epub 2013 May 3.
- ICRP (1994). *Publication 66: human respiratory tract model for radiological protection*. Annals of the International Commission on Radiological Protection. Pergamon.
- Ilavsky, J. and Jemian, P. R. (2009). Irena: tool suite for modeling and analysis of small-angle scattering. *Journal of Applied Crystallography*, 42(2):347–353.
- Kandlikar, M. and Ramachandran, G. (1999). Inverse methods for analysing aerosol spectrometer measurements: A critical review. *Journal of Aerosol Science*, 30(4):413–37.
- Knutson, E. O. and Whitby, K. T. (1975). Aerosol classification by electrical mobility: Apparatus, theory, and applications. *Journal of Aerosol Science*, 6(6):443–51.
- Koehler, K. A. and Volckens, J. (2013). Development of a sampler to estimate regional deposition of aerosol in the human respiratory tract. *Ann Occup Hyg*, 57(9):1138–47. Koehler, Kirsten A, Volckens, John. eng R03OH009248/OH/NIOSH CDC HHS/ T42OH009229/OH/NIOSH CDC HHS/ Research Support, U.S. Gov't, P.H.S. England 2013/09/26 06:00 Ann Occup Hyg. 2013 Nov;57(9):1138-47. doi: 10.1093/annhyg/met041. Epub 2013 Sep 23.
- Kousaka, Y., Okuyama, K., Adachi, M., and Mimura, T. (1986). Effect of brownian diffusion on electrical classification of ultrafine aerosol particles in differential mobility analyzer. *J Chem Eng Japan*, 19(5):401–407.
- Kozawa, K. H., Fruin, S. A., and Winer, A. M. (2009). Near-road air pollution impacts of goods movement in communities adjacent to the ports of los angeles and long beach. *Atmospheric Environment*, 43(18):2960–2970.

- Kristensson, A., Rissler, J., Löndahl, J., Johansson, C., and Swietlicki, E. (2013). Size-resolved respiratory tract deposition of sub-micrometer aerosol particles in a residential area with wintertime wood combustion. *Aerosol and Air Quality Research*, 13:24–35.
- Kuo, Y.-M., Huang, S.-H., Shih, T.-S., Chen, C.-C., Weng, Y.-M., and Lin, W.-Y. (2005). Development of a size-selective inlet-simulating icrp lung deposition fraction. *Aerosol Science and Technology*, 39(5):437–443.
- Lawson, C. L. and Hanson, R. J. (1987). Linear least squares with linear inequality constraints. In *Solving Least Squares Problems*, pages 158–73. Prentice-Hall.
- Leppä, J., Kerminen, V. M., Laasko, L., Korhonen, H., Lehtinen, K. E. J., Gagne, S., Manninen, H. E., Nieminen, T., and Kulmala, M. (2009). Ion-uhma: a model for simulating the dynamics of neutral and charged aerosol particles. *Boreal Environment Research*, 14:559–75.
- Li, L., Chen, D.-R., Qi, C., and Kulkarni, P. S. (2009). A miniature disk electrostatic aerosol classifier (mini-disk eac) for personal nanoparticle sizers. *Journal of Aerosol Science*, 40(11):982–992.
- Lioy, P. J. (1990). Assessing total human exposure to contaminants. *Environmental Science & Technology*, 24(7):938–975.
- López-Yglesias, X. and Flagan, R. C. (2013). Ion-aerosol flux coefficients and the steady-state charge distribution of aerosols in a bipolar ion environment. *Aerosol Science and Technology*, 47(6):688–704.
- Maynard, A. and Pui, D. Y. H., editors (2007). *Nanoparticles and Occupational Health*. Journal of Nanoparticle Research.
- Maynard, A. D. and Aitken, R. J. (2009). Assessing exposure to airborne nanomaterials: Current abilities and future requirements. *Nanotoxicology*, 1(1):26–41.
- McMurry, P. H. (2000). A review of atmospheric aerosol measurements. *Atmospheric Environment*, 34:1959–1999.
- Merritt, M. and Zhang, Y. (2005). Interior-point gradient method for large-scale totally nonnegative least squares problems. *J Optim Theory Appl*, 126(1):191–202.
- Mönkkönen, P., Koponen, K. I., Lehtinen, K. E. J., Hameri, K., Uma, R., and Kulmala, M. (2005). Measurements in a highly polluted asian mega city: observations of aerosol number size distribution, modal parameters and nucleation events. *Atmospheric Chemistry and Physics*, 5:57–66.
- Mui, W., Thomas, D. A., Downard, A. J., Beauchamp, J. L., Seinfeld, J. H., and Flagan, R. C. (2013). Ion mobility-mass spectrometry with a radial opposed migration ion and aerosol classifier (romiac). *Anal Chem*, 85(13):6319–26. Mui,

- Wilton Thomas, Daniel A Downard, Andrew J Beauchamp, Jesse L Seinfeld, John H Flagan, Richard C eng Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. 2013/06/05 06:00 Anal Chem. 2013 Jul 2;85(13):6319-26. doi: 10.1021/ac400580u. Epub 2013 Jun 19.
- Nazaroff, W. W. (2008). New directions: It's time to put the human receptor into air pollution control policy. *Atmospheric Environment*, pages 6565–6566.
- Oberdörster, G. (1995). Lung particle overload: Implications for occupational exposures to particles. *Regulatory Toxicology and Pharmacology*, 27:123–135.
- Oberdörster, G., Maynard, A., Donaldson, K., Castranova, V., Fitzpatrick, J., Ausman, K., Carter, J., Karn, B., Kreyling, W., Lai, D., Olin, S., Monteiro-Riviere, N., Warheit, D., Yang, H., and Group, I. R. F. S. I. N. T. S. W. (2005). Principles for characterizing the potential human health effects from exposure to nanomaterials: elements of a screening strategy. *Part Fibre Toxicol*, 2:8. Oberdorster, Gunter Maynard, Andrew Donaldson, Ken Castranova, Vincent Fitzpatrick, Julie Ausman, Kevin Carter, Janet Karn, Barbara Kreyling, Wolfgang Lai, David Olin, Stephen Monteiro-Riviere, Nancy Warheit, David Yang, Hong eng England 2005/10/08 09:00 Part Fibre Toxicol. 2005 Oct 6;2:8.
- Oberdörster, G., Stone, V., and Donaldson, K. (2007). Toxicology of nanoparticles: A historical perspective. *Nanotoxicology*, 1(1):2–25.
- Peters, A., Dockery, D. W., Heinrich, J., and Wichmann, H. E. (1997). Short-term effects of particulate air pollution on respiratory morbidity in asthmatic children. *Eur Respir J*, 10:872–879.
- Peters, A., Hampel, R., Cyrus, J., Breitner, S., Geruschkat, U., Kraus, U., Zareba, W., and Schneider, A. (2015). Elevated particle number concentrations induce immediate changes in heart rate variability: a panel study in individuals with impaired glucose metabolism or diabetes. *Part Fibre Toxicol*, 12:7. Peters, Annette Hampel, Regina Cyrus, Josef Breitner, Susanne Geruschkat, Uta Kraus, Ute Zareba, Wojciech Schneider, Alexandra eng Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. England 2015/04/19 06:00 Part Fibre Toxicol. 2015 Mar 30;12:7. doi: 10.1186/s12989-015-0083-7.
- Pope and Dockery (2006). Health effects of fine particulate air pollution: Lines that connect. *J Air & Waste Management*, pages 709–742.
- Qi, C., Chen, D.-R., and Greenberg, P. (2008). Performance study of a unipolar aerosol mini-charger for a personal nanoparticle sizer. *Journal of Aerosol Science*, 39(5):450–459.
- Qi, C. and Kulkarni, P. (2012). Unipolar charging based, hand-held mobility spectrometer for aerosol size distribution measurement. *Journal of Aerosol Science*, 49:32–47.

- Ramachandran, G. and Leith, D. (1992). Extraction of aerosol-size distributions from multispectral light extinction data. *Aerosol Science and Technology*, 17(4):303–325.
- Ranjan, M. and Dhaniyala, S. (2008). A new miniature electrical aerosol spectrometer (meas): Experimental characterization. *Journal of Aerosol Science*, 39(8):710–722.
- Rissler, J., Swietlicki, E., Bengtsson, A., Boman, C., Pagels, J., Sandström, T., Blomberg, A., and Löndahl, J. (2012). Experimental determination of deposition of diesel exhaust particles in the human respiratory tract. *Journal of Aerosol Science*, 48:18–33.
- Russell, A. G. and Brunekreef, B. (2009). A focus on particulate matter and health. *Environmental Science & Technology*, 43(19):4620–4625.
- Russell, L. M., Flagan, R. C., and Seinfeld, J. H. (1995). Asymmetric instrument response resulting from mixing effects in accelerated dma-cpc measurements. *Aerosol Science and Technology*, 23(4):491–509.
- Sexton, K., Selevan, S. G., Wagener, D. K., and Lybarger, J. A. (1992). Estimating human exposures to environmental pollutants: Availability and utility of existing databases. *Archives of Environmental Health*, 47(6):398–407.
- Stolzenburg, M. R. (1988). *An Ultrafine Aerosol Size Distribution Measuring System*. PhD thesis, University of Minnesota, MN.
- Stolzenburg, M. R. and McMurry, P. H. (1991). An ultrafine aerosol condensation nucleus counter. *Aerosol Science and Technology*, 14(1):48–65.
- Stolzenburg, M. R. and McMurry, P. H. (2008). Equations governing single and tandem dma configurations and a new lognormal approximation to the transfer function. *Aerosol Science and Technology*, 42(6):421–432.
- Stratmann, F., Kauffeldt, T., Hummes, D., and Fissan, H. (1997). Differential electrical mobility analysis: A theoretical study. *Aerosol Science and Technology*, 26(4):368–383.
- Talukdar, S. S. and Swihart, M. T. (2003). An improved data inversion program for obtaining aerosol size distributions from scanning differential mobility analyzer data. *Aerosol Science and Technology*, 37(2):145–161.
- Vedal, S. (1997). Ambient particles and health: Lines that divide. *Journal of the Air & Waste Management Association*, 47(5):551–581.
- Vincent, J. H. (2005). Health-related aerosol measurement: a review of existing sampling criteria and proposals for new ones. *J Environ Monit*, 7(11):1037–53. Vincent, James H eng Review England 2005/10/28 09:00 J Environ Monit. 2005 Nov;7(11):1037-53. Epub 2005 Sep 29.

- Wiedensohler, A. (1988). An approximation of the bipolar charge distribution for particles in the submicron size range. *Journal of Aerosol Science*, 19(3):387–389.
- Wiedensohler, A., Birmili, W., Nowak, A., Sonntag, A., Weinhold, K., Merkel, M., Wehner, B., Tuch, T., Pfeifer, S., Fiebig, M., Fjåraa, A. M., Asmi, E., Sellegri, K., Depuy, R., Venzac, H., Villani, P., Laj, P., Aalto, P., Ogren, J. A., Swietlicki, E., Williams, P., Roldin, P., Quincey, P., Hüglin, C., Fierz-Schmidhauser, R., Gysel, M., Weingartner, E., Riccobono, F., Santos, S., Gruning, C., Faloon, K., Beddows, D., Harrison, R., Monahan, C., Jennings, S. G., O'Dowd, C. D., Marinoni, A., Horn, H. G., Keck, L., Jiang, J., Scheckman, J., McMurry, P. H., Deng, Z., Zhao, C. S., Moerman, M., Henzing, B., de Leeuw, G., Löschau, G., and Bastian, S. (2012). Mobility particle size spectrometers: harmonization of technical standards and data structure to facilitate high quality long-term observations of atmospheric particle number size distributions. *Atmospheric Measurement Techniques*, 5(3):657–685.
- Wiedensohler, A., Orsini, D., Covert, D. S., Coffmann, D., Cantrell, W., Havlicek, M., Brechtel, F. J., Russell, L. M., Weber, R. J., Gras, J., Hudson, J. G., and Litchy, M. (1997). Intercomparison study of the size-dependent counting efficiency of 26 condensation particle counters. *Aerosol Science and Technology*, 27(2):224–242.
- Wolfenbarger, J. K. and Seinfeld, J. H. (1990). Inversion of aerosol size distribution data. *Journal of Aerosol Science*, 21(2):227–247.
- Zhang, M. and Wexler, A. S. (2006). Cross flow ion mobility spectrometry: Theory and initial prototype testing. *International Journal of Mass Spectrometry*, 258(1-3):13–20.
- Zhu, Y., Hinds, W. C., Kim, S., Shen, S., and Sioutas, C. (2002). Study of ultra-fine particles near a major highway with heavy-duty diesel traffic. *Atmospheric Environment*, 36:4323–35.
- Zhu, Y., Hinds, W. C., Shen, S., and Sioutas, C. (2004). Seasonal trends of concentration and size distribution of ultrafine particles near major highways in los angeles special issue of aerosol science and technology on findings from the fine particulate matter supersites program. *Aerosol Science and Technology*, 38(sup1):5–13.

Appendix A

MEASUREMENT SIMULATIONS

DMA

The differential mobility analyzer (DMA) classifies particles according to their electrical mobility. The history of mobility analyzers can be traced back to measurements of atmospheric ions in the early 20th century (Flagan 2008). Since the mid 1970s, DMAs have been used routinely to produce aerosol standards of known size, concentration, and composition, to measure aerosol size distributions, and to measure aerosol properties with tandem measurement instruments systems (Park et al. 2008).

System Setup and Operating Parameters

Measurements from a DMA system were modeled according to principles described by Wiedensohler et al. (2012). The system was considered to consist of a sequential setup of a bipolar diffusion charge conditioner, a long column DMA ($L = 444$ mm, inner radius $R_1 = 9.38$ mm, outer radius $R_2 = 19.58$ mm), and a condensation particle counter (CPC). All flows were considered balanced and strictly laminar.

Within the DMA classification region, charged particles in a sample are introduced near one electrode and induced to migrate across a parallel flow of particle-free air by an applied electrical field. Those that cross from that portion of the flow near the entrance electrode that corresponds to the incoming sample flow to the counter electrode as they are carried to a downstream classified sample outlet port are counted. The DMA selects particles that migrate within a narrow range of velocities within an applied electric field. The voltage is scanned or stepped to sweep through size space to determine the size distribution.

While scanning-mode DMA operation is more common today, the simulations presented here consider the stepping-mode operation for which a semi-analytical instrument response function exists. No such closed-form model for scanning-mode operation is currently available. Without an accurate description of the nonunique relationship between the time a particle exits a scanning-mode DMA and the time it is ultimately detected, inverted distributions will be distorted. To avoid introducing these additional distortions to the analyses of the present work, DMA simulations

here consider the voltage to be changed in discrete steps and held at constant voltage long enough to attain a steady state signal (with noise). This models the differential mobility particle sizer (DMPS).

The maximum mobility (smallest size) that can be classified with reasonable resolution is that transmitted at a classification voltage of about 10 V. The minimum mobility (largest size) is that transmitted at a classification voltage of about 10 kV, before electrostatic breakdown. Geometry of the DMA column and flow rates within the system determine what range of particle mobilities can be classified with given voltage settings. The mobility centroid, $Z_{p,i}^*$, of the i th channel for a cylindrical DMA column is (Knutson and Whitby 1975)

$$Z_{p,i}^* = \frac{Q_{sh} + Q_e}{4\pi L V_i} \ln \left(\frac{R_2}{R_1} \right) \quad (\text{A.1})$$

where Q_{sh} is the volumetric sheath flow, Q_e is the volumetric excess flow, L is the length of the DMA column, V_i is the negative potential applied to the inner cylinder, and R_1 and R_2 are the radial location of the aerosol exit and entrance, respectively. For the particles to be classified, particle mobility, Z_p , is then related to particle diameter, D_p , via

$$Z_p = \frac{\nu e C_c (D_p)}{3\pi \mu D_p} \quad (\text{A.2})$$

where e is the elementary charge, ν is the number of elementary charges on the particle, C_c is the Cunningham slip correction factor, and μ is the dynamic viscosity of air. Thus, defining the operating voltages V_i , flow rates Q_{sh} and Q_{ex} , and geometry parameters defines the targeted mobilities $Z_{p,i}^*$ and corresponding measurement size range.

If diffusion is unimportant, particles included in the classified aerosol flow will span a range of mobilities of $Z_p^* - \Delta Z \leq Z_p \leq Z_p^* + \Delta Z$, where $\Delta Z = Z_p^* \beta$ and β is a dimensionless flow parameter, relating the aerosol inlet Q_a and sample outlet Q_s flows to the particle-free sheath Q_{sh} and particle-laden excess Q_e flows through $\beta = \frac{Q_a + Q_c}{Q_{sh} + Q_e}$. This describes the non-diffusing transfer function illustrated in Figure A.1. In the present work, the non-diffusing transfer function is used to define DMA resolution values as the ratio of the sheath flow rate to that of the incoming aerosol sample for an ideal system with balanced flows, i.e., $R_{ideal} = \frac{1}{\beta} = \frac{Q_{sh}}{Q_a}$.

The non-diffusing transfer function is also used to define the number and spacing of sample channels in DMA simulations as follows. First, the minimum and maximum

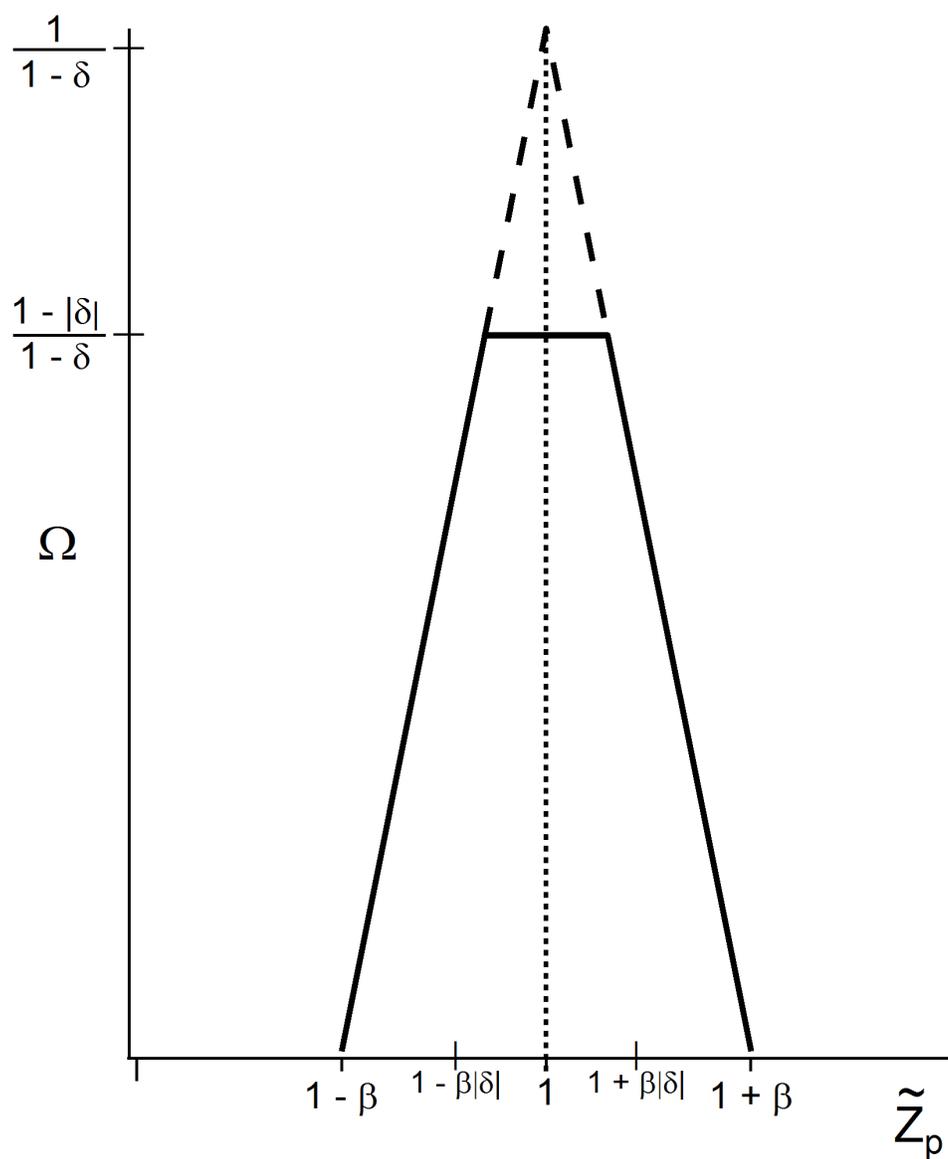


Figure A.1: Non-diffusing transfer function, Ω_{nd} , showing the probability of transiting the DMA for a particle with electrical mobility $Z_p = \tilde{Z}_p \cdot Z_p^*$ where Z_p^* is the centroid of the transfer function.

voltage settings are defined to be $V_{min} = 10 \text{ V}$ and $V_{max} = 10 \text{ kV}$, respectively, from which the corresponding maximum and minimum sampled particle mobilities, Z_{max} and Z_{min} are calculated using Eq. (A.1). The number of voltage steps is then determined such that the channels are equally spaced on a logarithmic scale (for particle mobility and corresponding voltage), and the ideal triangular transfer functions sample all mobilities within the measurement range. This can be expressed as

$$n_{channels} = \left\lceil \frac{\ln \left(\frac{Z_{min}}{Z_{max}} \right)}{\ln \left(\frac{1-\beta}{1+\beta} \right)} \right\rceil + 1 \quad (\text{A.3})$$

With this analysis, for a flow ratio of $\beta = 0.1$, the model system will step through 36 voltage settings equally spaced on a logarithmic scale between 10 V and 10 kV, covering a measured particle size range of approximately $7 \text{ nm} < D_p < 330 \text{ nm}$.

Kernel Function Definition

For the DMA-CPC system, the kernel function $\Psi(i, D_p)$ relating a sampled size distribution to the measured response values is given by

$$\Psi(i, D_p) = Q_a \sum_{\nu=1}^{\infty} W_{\nu} \eta(D_p) \phi_{\nu}(D_p) \Omega(\nu, i, D_p) \quad (\text{A.4})$$

where Q_a denotes the aerosol volume flow rate, ν denotes the number of electric charges on the aerosol particle, W_{ν} denotes the factor relating the sensor response to the flux [particles/sec] of particles carrying ν charges, $\eta(D_p)$ denotes the collection of empirical, instrument specific transmission and counting efficiency terms, $\phi_{\nu}(D_p)$ denotes the fraction of particles of diameter D_p carrying ν charges, and $\Omega(\nu, i, D_p)$ denotes the transfer function of the DMA, i.e., the probability that a particle of size D_p carrying ν charges will pass through the DMA when it is set at voltage V_i . The measured response from a DMA system is then defined as

$$R_i = Q_a \sum_{\nu=1}^{\infty} W_{\nu} \int_0^{\infty} n_N(\ln D_p) \eta(D_p) \phi_{\nu}(D_p) \Omega(\nu, i, D_p) d \ln D_p \quad (\text{A.5})$$

Functional forms for these parameters are key to describing measurement theory for data analysis. Figure A.2 illustrates typical shapes of these individual functions, and the next few sections describe them in more detail.

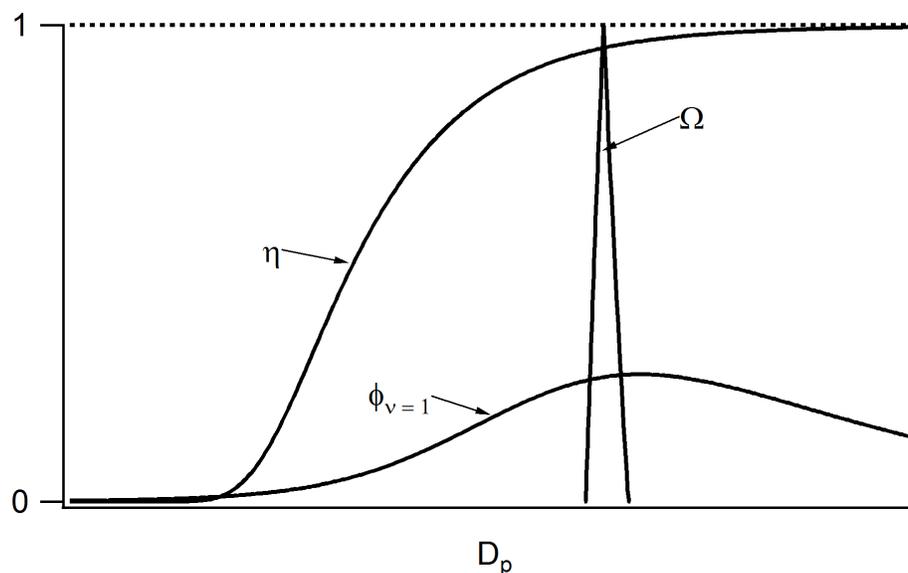


Figure A.2: Typical shapes of the functions contributing to the definition of the integration kernel Ψ_i for measurement channel i of a DMA.

Transfer Function

When planning how to collect information on the particle size distribution with measured data, the relevant property describing the instrument is its transfer function. The DMA transfer function, Ω , is defined as the probability that a particle of a given size, starting at the aerosol entrance of the classification zone, will reach the aerosol exit. Losses in the inlet and outlet zones outside the classification zone are not included in Ω . The DMA transfer function depends on the sampled particle mobility, Z_p , the rod voltage, V , geometry of the DMA column, and flow rates within the system. Particle mobility, Z_p , is related to particle diameter, D_p , via Eq. (A.2), and the mobility centroid, $Z_{p,i}^*$, of the i th channel for a cylindrical DMA column is described by Eq. (A.1).

The transfer function may be assumed to be as simple as the ideal triangular form derived by Knutson and Whitby (1975), or slightly more complex as given by the semi-analytical expression for the diffusive transfer function derived by Stolzenburg (1988), or as complex as the real scanning mobility particle sizer (SMPS) obtained from numerical simulations or empirical data-fitting (Mui et al., 2018ab).

The non-dimensional form of the ideal, non-diffusing transfer function can be written

as (Stolzenburg and McMurry 2008)

$$\Omega_{nd}(\tilde{Z}_p, \beta, \delta) = \frac{1}{2\beta(1-\delta)} \left[|\tilde{Z}_p - (1 + \beta)| + |\tilde{Z}_p - (1 - \beta)| \right. \\ \left. - |\tilde{Z}_p - (1 + \beta\delta)| - |\tilde{Z}_p - (1 - \beta\delta)| \right] \quad (\text{A.6})$$

where the dimensionless mobility is defined as

$$\tilde{Z}_p = \frac{Z_p}{Z_p^*} \quad (\text{A.7})$$

along with dimensionless flow parameters

$$\beta = \frac{Q_a + Q_s}{Q_{sh} + Q_{ex}} \quad (\text{A.8})$$

$$\delta = \frac{Q_a - Q_s}{Q_a + Q_s} \quad (\text{A.9})$$

At the centroid mobility of this triangular transfer function, $\Omega(\tilde{Z}_p) = 1$ and at the full-width-half-max, $\Omega(\tilde{Z}_p - 0.5\Delta Z_p) = \Omega(\tilde{Z}_p + 0.5\Delta Z_p) = 0.5$. No transmission occurs beyond the boundaries of $Z_p^* \pm \Delta Z_p$, i.e., $\Omega(\tilde{Z}_p - \Delta Z_p) = \Omega(\tilde{Z}_p + \Delta Z_p) = 0$. The base half width of the ideal, non-broadened triangular transfer function is

$$\Delta Z_p = \frac{Q_a + Q_s}{Q_{sh} + Q_{ex}} Z_p^* = \beta \cdot Z_p^* \quad (\text{A.10})$$

Recall that this relationship is used to determine the number of measurement channels in the present DMA simulations through Eq. (A.3).

Deviations from the triangular transfer function may occur due to well-known sources of broadening such as particle diffusion, nonuniform distribution of particles in the inlet, imperfections in the electrode geometry or imperfections of the flow field (Flagan 1999). Other functional forms for $\Omega(Z_p^*)$ can be substituted for this ideal form.

To a good approximation, diffusing particles spread in an ever-expanding Gaussian cross-stream profile about the corresponding non-diffusing particle streamline (Stolzenburg 1988). At the aerosol exit the final standard deviation, σ , of this profile in non-dimensional form is

$$\sigma^2 = \frac{G}{Pe_{mig}} \frac{Z_p}{Z_p^*} \quad (\text{A.11})$$

where the Peclet number for particle migration across the gap between the electrodes is

$$Pe_{mig} = \frac{qV}{kT} f(\text{geometry}) \quad (\text{A.12})$$

and G is a parameter that depends weakly on the DMA geometry and the flow rate ratio β . In this definition, q is the particle charge, k is the Boltzmann constant, and T is the temperature. The factor $f(\text{geometry})$ accounts for nonuniformities in the electric field along the migration path. For the cylindrical DMA, $f_{CDMA} = (R_2 - R_1) / (R_2 \ln R_2 / R_1)$. The dimensionless DMA geometry and flow factor, G , is defined as derived by Flagan (1999).

The diffusing transfer function can be approximated as

$$\Omega_{d,approx} = \frac{\beta(1+\delta)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\tilde{Z}_p - 1)^2}{2\sigma^2}\right) \quad (\text{A.13})$$

Stolzenburg (1988) derived a more rigorous empirical diffusing transfer function as

$$\Omega_d(\tilde{Z}_p, \beta, \delta, \sigma) = \frac{\sigma}{\sqrt{2}\beta(1-\delta)} \left[\varepsilon\left(\frac{\tilde{Z}_p - (1+\beta)}{\sqrt{2}\sigma}\right) + \varepsilon\left(\frac{\tilde{Z}_p - (1-\beta)}{\sqrt{2}\sigma}\right) - \varepsilon\left(\frac{\tilde{Z}_p - (1+\beta\delta)}{\sqrt{2}\sigma}\right) - \varepsilon\left(\frac{\tilde{Z}_p - (1-\beta\delta)}{\sqrt{2}\sigma}\right) \right] \quad (\text{A.14})$$

where $\varepsilon(x) = x \cdot \text{erf}(x) + \frac{1}{\pi} e^{-x^2}$ and $\text{erf}(x)$ is the error function. As σ goes to zero, this reduces to $\Omega_{nd}(\tilde{Z}_p, \beta, \delta)$.

Figure A.3 demonstrates the different forms of the non-diffusing and diffusing transfer functions. The non-diffusing transfer function is the simplest to employ for data analysis calculations, but errors arise when it is used outside the range of appropriate conditions (i.e., when it is used to describe conditions where diffusion is important).

Charging Probability

The bipolar diffusion charge conditioner brings the aerosol particles to a bipolar charge equilibrium before entering the DMA. The bipolar charge equilibrium can be theoretically described by the work of Fuchs (1963), Hoppel and Frick (1986), Wiedensohler (1988), and López-Yglesias and Flagan (2013).

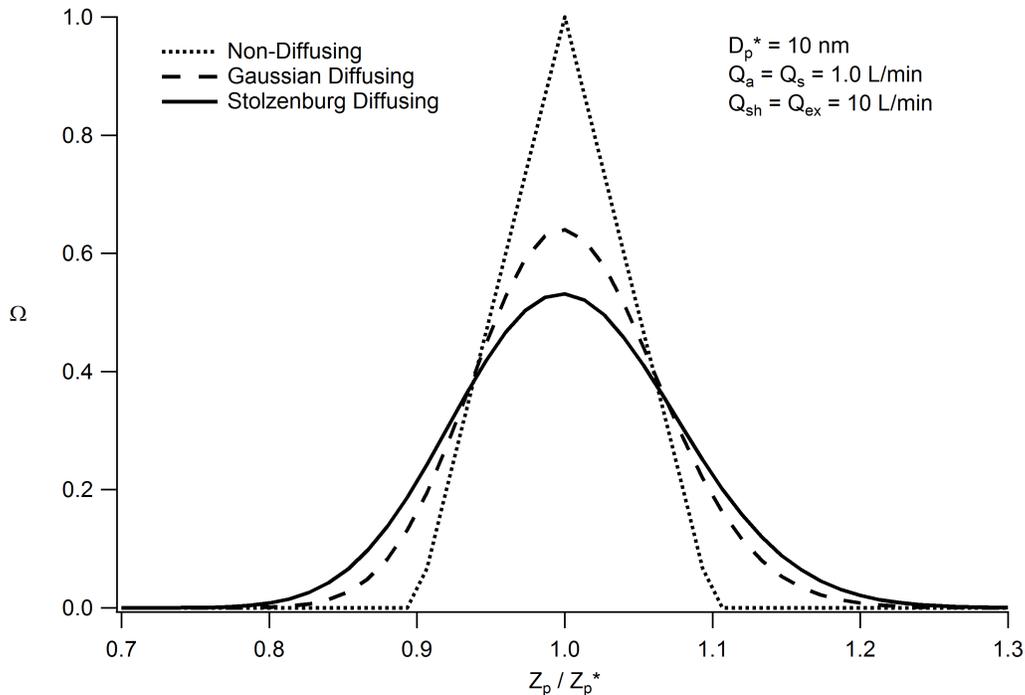


Figure A.3: Transfer functions, Ω , as a function of dimensionless particle mobility, \tilde{Z}_p , for a flow ratio of $\beta = \frac{1}{10}$ ($\delta = 0$) at DMA centroid diameter $D_p^* = 10$ nm.

In the present work, the fraction of particles carrying ν charges at charge equilibrium, $\phi_\nu(D_p)$, is calculated according to the model of López-Yglesias and Flagan (2013)

$$\phi_\nu(D_p) = 10^{\sum_{i=0}^{11} B_i(\nu)(\log_{10}(D_p/2))^i} \quad (\text{A.15})$$

where $B_i(\nu)$ are fit coefficients determined by a least square regression analysis, and the particle diameter D_p has units of meters. The fit coefficients used here are summarized in Table A.1. Equation (A.15) defines the charging probability ϕ_ν used to simulate DMA response values from Eq. (A.5) and also to define the integration kernel Ψ_i of Eq. (A.4) for inversion analysis.

Other Efficiency Terms

Other important elements of the measurement kernel include the response and counting efficiency of the measurement system's particle detector, as well as descriptions of particle losses throughout the system, e.g., inside of pipes, the DMA, aerosol dryer and bipolar charge conditioner.

The performance of a condensation particle counter (CPC) was modeled for the simulated DMA system's particle detector. The response of a CPC has units of

number concentration [cm^{-3}] of particles. For an ideal CPC, the response to a single particle is independent of the number of charges on it. In the basic equation for response, Eq. (A.5), the factor relating detector response to the flux [particles/sec] of particles for the CPC is

$$W_v = \frac{1}{Q_s} \quad (\text{A.16})$$

Expressions relating detector efficiencies to particle diameter are available in Wiedensohler et al. (1997) for a number of different types of CPCs. However, the size-dependent counting efficiency of an individual CPC may depend on geometry or actual supersaturation profile inside the condenser. Since correct measurement of the particle number size distribution for the smallest particle sizes depends critically on the size dependent particle detection efficiency of the CPC, for the greatest accuracy, the CPC used for a given application should be calibrated individually against a reference instrument. Present simulations model a TSI Model 3025A UCPC with the counting efficiency curve shown in Figure A.4.

Particle losses inside CPCs are implicitly included in the measured counting efficiency. Diffusion losses also occur throughout the rest of the system, including inside the bipolar diffusion charge conditioner, in the inlet and outlet regions of the DMA, and through various plumbing elements. These losses can be described using equivalent pipe length and analytical formulas derived for the laminar flow regime (Hinds 1999). Equivalent pipe lengths of different device and plumbing elements aligned in sequence can be simply added if they are traversed by the same rate of aerosol flow. The present work uses equivalent lengths reported by Wiedensohler et al. (2012).

The collective efficiency term $\eta(D_p)$ in Eq. (A.4) is the product of the above terms: the penetration efficiency through the tubing, $\eta_{tube}(D_p)$, the penetration efficiency through the entrance and exit regions of the DMA, $\eta_{DMA}(D_p)$, and the counting efficiency of the detector, $\eta_{det}(D_p)$. Assumptions in describing efficiency terms (e.g., DMA effective length) will influence effective measurement range.

Table A.1: Steady State distribution for “air” ions and conductive particles at 101325 Pa and 298.15 K. Fit coefficients are shown for charge states of negatively charged particles with $\nu = 1$ to 6 charges.

$B_i(\nu)$	Charge State					
	-1	-2	-3	-4	-5	-6
B_0	-16256.24303	-1672.867775	-16600.32586	3938.729459	13244.15207	10835.81051
B_1	-18104.59519	-1583.306766	-16059.17399	4356.029288	13692.18289	11262.95006
B_2	-8385.187516	-582.3720642	-5899.297477	1812.628532	5334.877346	4389.080116
B_3	-2013.540813	-91.97359113	-889.3720619	311.5988457	837.6431824	677.6320994
B_4	-238.212957	-1.568748488	-0.351154004	2.606077463	-6.497776109	-10.76878649
B_5	-4.221364439	1.164629175	13.07820351	-5.25627902	-14.86280614	-12.82753139
B_6	2.134763808	0.037886603	0.149462281	-0.11155572	-0.046687802	0.073479529
B_7	0.153529237	-0.01768478	-0.216553824	0.101397126	0.290738425	0.263044642
B_8	-0.016478637	-0.000577868	-0.002794923	0.002203121	0.001663194	-0.000645683
B_9	-0.003055501	0.000299085	0.004054957	-0.002185346	-0.006490684	-0.006218968
B_{10}	-0.000175132	3.33234E-05	0.000420076	-0.000252875	-0.000702372	-0.000661849
B_{11}	-3.64394E-06	1.06988E-06	1.31482E-05	-8.69076E-06	-2.33259E-05	-2.19138E-05
$D_{p,min}$	0.4 nm	10.6 nm	31.6 nm	68.0 nm	105.3 nm	163.1 nm
$D_{p,max}$	20 μm					

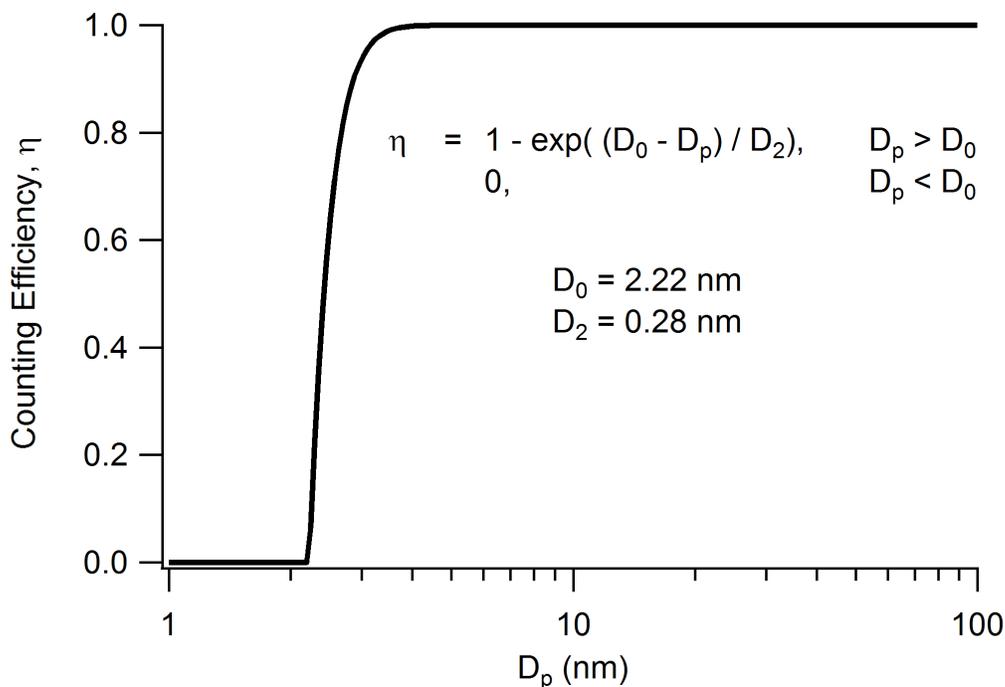


Figure A.4: Counting efficiency curve for the TSI Model 3025A UCPC, determined from exponential fit of experimental data presented by Wiedensohler et al. (1997).

Lung Deposition-Based Samplers

To model the RDS collection of regional deposition fractions, the following expressions are used. These simplified equations were defined by Hinds (1999) to approximate the ICRP model for monodisperse spheres of standard density at standard conditions, averaged for males and females at three exercise levels. The deposition fraction for the head airways DF_{HA} is

$$DF_{HA} = IF \left(\frac{1}{1 + \exp(6.84 + 1.183 \ln d_p)} + \frac{1}{1 + \exp(0.924 - 1.885 \ln d_p)} \right) \quad (\text{A.17})$$

where IF is the inhalable fraction as used by the ICRP, which is given by

$$IF = 1 - 0.5 \left(1 - \frac{1}{1 + 0.00076 d_p^{2.8}} \right) \quad (\text{A.18})$$

The deposition fraction for the tracheobronchial region DF_{TB} is

$$DF_{TB} = \left(\frac{0.00352}{d_p} \right) \left[\exp \left(-0.234 (\ln d_p + 3.40)^2 \right) + 63.9 \exp \left(-0.819 (\ln d_p - 1.61)^2 \right) \right] \quad (\text{A.19})$$

The deposition fraction for the alveolar region DF_{AL} is

$$DF_{AL} = \left(\frac{0.0155}{d_p} \right) \left[\exp \left(-0.416 (\ln d_p + 2.84)^2 \right) + 19.11 \exp \left(-0.482 (\ln d_p - 13.62)^2 \right) \right] \quad (\text{A.20})$$

The total deposition DF is the sum of the regional depositions, or

$$DF = IF \left(0.0587 + \frac{0.911}{1 + \exp(4.77 + 1.485 \ln d_p)} + \frac{0.943}{1 + \exp(0.508 - 2.58 \ln d_p)} \right) \quad (\text{A.21})$$

Integral Measures of Particle Exposure

To evaluate $PM_{2.5}$ measurements, a sampler is modeled that follows the particle size-selective criterion described by Vincent (2005) for $PM_{2.5}$ in which

$$\eta_{PM_{2.5}} = 1 - F(x) \quad (\text{A.22})$$

where $F(x)$ is the cumulative probability function of a standard normal variable (x) given by

$$x = \frac{\ln \left(\frac{d_{ae}}{\Gamma_x} \right)}{\ln(\Sigma_x)} \quad (\text{A.23})$$

in which $\Gamma_x = 2.5 \mu\text{m}$ and $\Sigma_x = 1.50$. The result is that the function reaches 0.5 at $d_{ae} = 2.5 \mu\text{m}$.

To evaluate respirable sampler measurements, first an empirical function is defined for the *inhalability*, which is equivalent to the aspiration efficiency of the human head. This is given by

$$I(d_{ae}) = 0.5 \{1 + \exp(-0.06d_{ae})\} + f(d_{ae}, U) \quad (\text{A.24})$$

where $f(d_{ae}, U)$ is a term that takes account of the observed effects at higher wind speeds. This is another empirical function, expressed as

$$f(d_{ae}, U) \equiv pU^q (\exp(rd_{ae})) \quad (\text{A.25})$$

in which d_{ae} is in μm and the wind speed U is in ms^{-1} , with $p = 10^{-5}$, $q = 2.75$, and $r = 0.055$ for the range $1 \leq U \leq 9 \text{ms}^{-1}$.

The *respirable fraction*, $R(d_{ae})$ is then defined by a curve that follows

$$R(d_{ae}) = I(d_{ae})[1 - F(x)] \quad (\text{A.26})$$

where $F(x)$ is again the cumulative probability function of a standardized normal variable x but now with $\Gamma_x = 4.25 \mu\text{m}$ and $\Sigma_x = 1.50$. The net result is that $R(d_{ae})$ reaches 0.5 at $d_{ae} = 4.0 \mu\text{m}$.

Appendix B

LUNG DEPOSITION MODEL

The “reference worker” defined to receive calculated dose values in the present work is described by the parameters shown in Table B.1. These parameters match those established by Phalen (1999). The ICRP model is then used with these parameters to evaluate the deposition fraction of aerosol particles in the respiratory system, η_{dep} [–].

Table B.1: Parameters defining the reference worker used for lung deposition calculations.

Physiological Parameters	
Subject	Adult male
Functional Residual Capacity	2200 cc
Extrathoracic Dead Space	50 cc
Bronchial Dead Space	49 cc
Bronchiolar Dead Space	47 cc
Height	175 cm
Tracheal Diameter	1.65 cm
First Bronchial Diameter	0.165 cm
Activity Related Parameters	
Activity level	Light exercise
Activity Type	Nose breathing only
Ventilation Rate	1.5 m ³ /hr
Respiratory Frequency	20 breaths/minute
Tidal Volume	1250 cc
Volumetric Flow Rate	725 cc/sec
Fraction Breathed through Nose	1.0
Aerosol Parameters	
Density	1.4 g/cc
Shape Factor	1.0

Appendix C

SPLINE INTERPOLATION

Here we examine the mathematical details behind spline interpolation. Our concern here is obtaining the correct mathematical expressions for interpolating between discrete values of a continuous model such that we represent the continuous model as a piecewise polynomial, or *spline* function. This sets the foundation for subsequent discrete calculations involving the continuous model, such as those as we carry out in our aerosol inversion algorithm. Our discussion here will prove essential to understanding the setup of our discrete inverse calculations, in which we represent the continuous particle size distribution function as a piecewise spline function.

A spline function consists of polynomial pieces joined together at a certain set of points, called the control points, that correspond to locations t_j called knots. We define a knot vector $\mathbf{t} = (t_j)$ to contain these knot points, and we specify simply that this vector describe a nondecreasing sequence of real numbers with no duplicate values. For now, while we examine the mathematics of spline interpolation in general, we leave the precise number and spacing of the vector elements open. We will address these specifications later, when we describe how we use spline interpolation to approximate a continuous particle size distribution function. For now, we assume that we have a bi-infinite set of distinct knot points to work with, prescribed such that

$$\begin{cases} \dots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \dots \\ \lim_{j \rightarrow \infty} t_j = \infty = -\lim_{j \rightarrow \infty} t_{-j} \end{cases} \quad (\text{C.1})$$

Let $S_{\mathbf{t}}^d(x)$ represent a spline function of degree d with knots defined in the vector \mathbf{t} . We define this spline function piece by piece over a closed interval of knot points $[t_1, t_n]$ such that

$$S_{\mathbf{t}}^d(x) = \begin{cases} S_1^d(x) & x \in [t_1, t_2] \\ S_2^d(x) & x \in [t_2, t_3] \\ \vdots & \\ S_{n-1}^d(x) & x \in [t_{n-1}, t_n] \end{cases} \quad (\text{C.2})$$

where each interpolant piece $S_j^d(x)|_{x \in [t_j, t_{j+1}]}$ is a polynomial of degree d , and the points $(t_j)_{j=1}^n = (t_1, t_2, \dots, t_n)$ are the knots at which these piecewise polynomials join together with certain continuity conditions. The continuity conditions depend on the degree d of the polynomial pieces. For example, the spline function of degree $d = 1$ (i.e., linear spline) is continuous in value but discontinuous in first derivative at the knot points (i.e., its slope may change abruptly at the boundary between two adjacent intervals). In contrast, the spline function of degree $d = 3$ (i.e., cubic spline) is continuous through to the second derivative. It is the application of these two cases (i.e., linear and cubic spline functions) that we discuss in depth here.

To help us derive the mathematical expressions we will use to incorporate linear and cubic spline interpolations within our inversion analyses, we first introduce special spline functions called B-splines. B-splines are so named because they are the *basis* functions from which an overall spline function approximation is constructed. By definition, a spline function of degree d with knot sequence \mathbf{t} , such as $S_{\mathbf{t}}^d(x)$ introduced above, is a linear combination of the B-splines associated with that knot sequence. This is expressed as

$$S_{\mathbf{t}}^d(x) = \sum_j c_j B_{j,d}(x) \quad (\text{C.3})$$

where the $B_{j,d}$ represent the constituent B-splines, and the c_j are real numbers called the spline coefficients. We will return to this definition later, within the context of defining our spline function representation of a continuous particle size distribution function. At that point, we will specify the upper and lower bounds on j and discuss the nature of the spline coefficients c_j . Next, we derive the functional forms for linear and cubic B-splines.

B-Splines (Basis Functions)

B-splines are special spline functions that have minimal support with respect to a given knot vector. For our discussion here, we will continue to work with the bi-infinite set of distinct knot points $\mathbf{t} = (t_j)$ described above. Having minimal support means that a given B-spline is nonzero only within a predefined, finite interval of the knot sequence. In fact, B-splines of a given degree d are defined to depend only on the $d + 2$ knots $(t_j, t_{j+1}, \dots, t_{j+d+1})$ so that the finite interval $[t_j, t_{j+d+1})$ constitutes the support of the j th B-spline $B_{j,d}$.

We use the j index to reference a specific B-spline because usually we form a set of B-splines $\{B_{j,d}\}$ of degree d associated with a knot vector $\mathbf{t} = (t_j)$. We can

construct a unique B-spline for each distinct subset of $d + 2$ knots $(t_k)_{k=j}^{j+d+1}$ in a given knot vector. This means that if we specify the bounds on the knot vector such that $\mathbf{t} = (t_j)_{j=1}^{n+d+1}$ for some positive integer n , we can form n B-splines $\{B_{j,d}\}_{j=1}^n$ of degree d associated with this knot vector.

We derive the functional form for B-splines of degree d using the Cox-de Boor recursion formula

$$B_{j,d}(x) = \frac{x - t_j}{t_{j+d} - t_j} B_{j,d-1}(x) + \frac{t_{j+d+1} - x}{t_{j+d+1} - t_{j+1}} B_{j+1,d-1}(x) \quad (\text{C.4})$$

with

$$B_{j,0}(x) = \begin{cases} 1 & \text{if } t_j \leq x < t_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.5})$$

From this, we see that the j th B-spline of degree 0, $B_{j,0}(x)$, is piecewise constant one or zero indicating which knot span x is in. This is illustrated in Figure C.1(a). The support of $B_{j,0}(x)$ is the half-open interval $[t_j, t_{j+1})$.

One application of the recurrence relation describes B-splines of degree 1 (i.e., linear B-splines) and gives

$$B_{j,1}(x) = \frac{x - t_j}{t_{j+1} - t_j} B_{j,0}(x) + \frac{t_{j+2} - x}{t_{j+2} - t_{j+1}} B_{j+1,0}(x) \quad (\text{C.6})$$

so that

$$B_{j,1}(x) = \begin{cases} \frac{x-t_j}{t_{j+1}-t_j} & \text{if } t_j \leq x < t_{j+1} \\ \frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} & \text{if } t_{j+1} \leq x < t_{j+2} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.7})$$

That is, the j th B-spline of degree 1, $B_{j,1}(x)$, is piecewise linear with support over the interval $[t_j, t_{j+2})$. Plotting this linear B-spline reveals a triangular function that is zero below $x = t_j$, ramps to one at $x = t_{j+1}$ and back to zero at and beyond $x = t_{j+2}$, as illustrated in Figure C.1(b). Note that we can define multiple linear B-splines over an extended knot sequence, and that we use the j index to associate a given B-spline with the first knot point of its individual range of support. Note also how each linear B-spline is continuous in value but discontinuous in first derivative at its knot points.

A second application of the recurrence relation describes B-splines of degree 2 (i.e., quadratic splines). We do not explore the use of quadratic splines within our

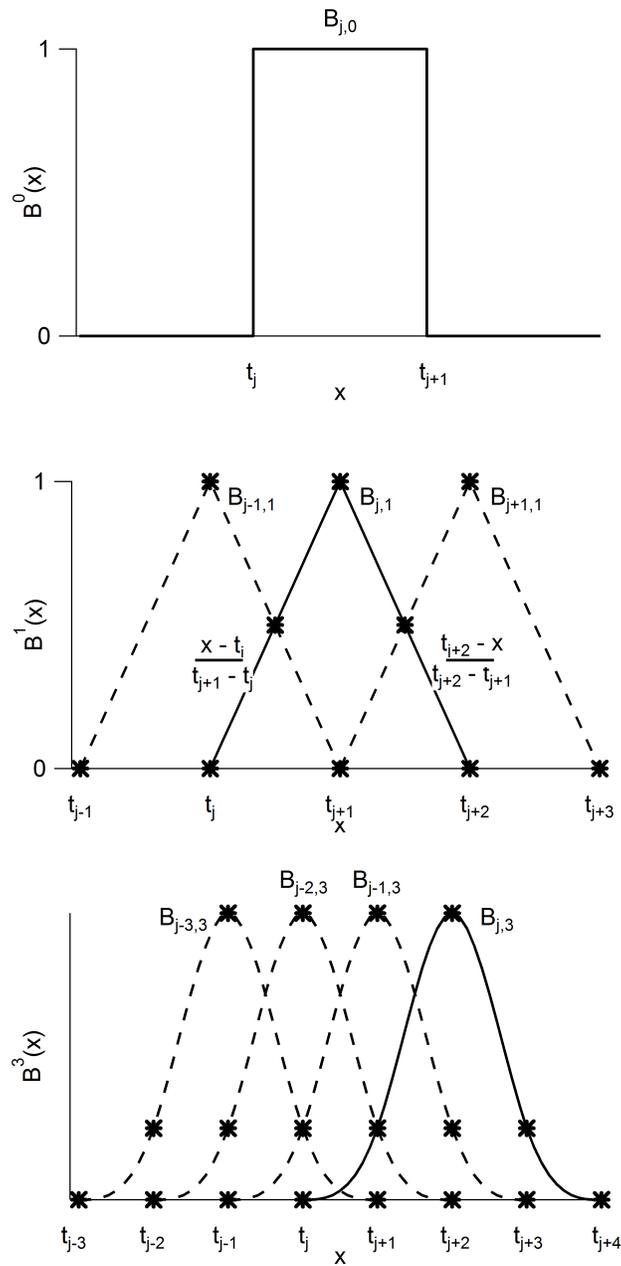


Figure C.1: Basis functions (B-splines) $B_{j,d}$ of degree $d = 0$ (top), $d = 1$ (middle), $d = 3$ (bottom).

inversion algorithm, and in general, quadratic splines are not used in applications as often as cubic or even linear splines. Nevertheless, we include the derivation of their functional form for continuity and clarity in applying the Cox-de Boor recursion formula. We obtain the following explicit expression for a generic quadratic B-spline

$$\begin{aligned}
B_{j,2}(x) &= \frac{x-t_j}{t_{j+2}-t_j} B_{j,1}(x) + \frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} B_{j+1,1}(x) \\
&= \frac{x-t_j}{t_{j+2}-t_j} \left[\frac{x-t_j}{t_{j+1}-t_j} B_{j,0}(x) + \frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} B_{j+1,0}(x) \right] \\
&\quad + \frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \left[\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} B_{j+1,0}(x) + \frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} B_{j+2,0}(x) \right]
\end{aligned} \tag{C.8}$$

and collecting like terms gives

$$\begin{aligned}
B_{j,2}(x) &= \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{x-t_j}{t_{j+1}-t_j} \right) B_{j,0}(x) \\
&\quad + \left[\left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} \right) + \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) \right] B_{j+1,0}(x) \\
&\quad + \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) B_{j+2,0}(x)
\end{aligned} \tag{C.9}$$

so that

$$B_{j,2}(x) = \begin{cases} \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{x-t_j}{t_{j+1}-t_j} \right) & \text{if } t_j \leq x < t_{j+1} \\ \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} \right) + \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) & \text{if } t_{j+1} \leq x < t_{j+2} \\ \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) & \text{if } t_{j+2} \leq x < t_{j+3} \\ 0 & \text{otherwise} \end{cases} \tag{C.10}$$

Moving on with a third application of the recurrence relation, we obtain the following

explicit expression for B-splines of degree 3 (i.e., cubic splines)

$$\begin{aligned}
B_{j,3}(x) &= \frac{x-t_j}{t_{j+3}-t_j} B_{j,2}(x) + \frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} B_{j+1,2}(x) & (C.11) \\
&= \frac{x-t_j}{t_{j+3}-t_j} \left\{ \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{x-t_j}{t_{j+1}-t_j} \right) B_{j,0}(x) \right. \\
&\quad + \left[\left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} \right) + \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) \right] B_{j+1,0}(x) \\
&\quad + \left. \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) B_{j+2,0}(x) \right\} \\
&+ \frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \left\{ \left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) B_{j+1,0}(x) \right. \\
&\quad + \left[\left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) + \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{x-t_{j+2}}{t_{j+3}-t_{j+2}} \right) \right] B_{j+2,0}(x) \\
&\quad + \left. \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+3}} \right) B_{j+3,0}(x) \right\}
\end{aligned}$$

and collecting like terms gives

$$\begin{aligned}
B_{j,3}(x) &= \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{x-t_j}{t_{j+1}-t_j} \right) B_{j,0}(x) & (C.12) \\
&+ \left[\left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} \right) \right. \\
&\quad + \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) \\
&\quad + \left. \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) \right] B_{j+1,0}(x) \\
&+ \left[\left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) \right. \\
&\quad + \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) \\
&\quad + \left. \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{x-t_{j+2}}{t_{j+3}-t_{j+2}} \right) \right] B_{j+2,0}(x) \\
&+ \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+3}} \right) B_{j+3,0}(x)
\end{aligned}$$

so that

$$B_{j,3}(x) = \begin{cases} \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{x-t_j}{t_{j+1}-t_j} \right) & \text{if } t_j \leq x < t_{j+1} \\ \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{x-t_j}{t_{j+2}-t_j} \right) \left(\frac{t_{j+2}-x}{t_{j+2}-t_{j+1}} \right) \\ \quad + \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) & \text{if } t_{j+1} \leq x < t_{j+2} \\ \quad + \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+2}-t_{j+1}} \right) \\ \left(\frac{x-t_j}{t_{j+3}-t_j} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) & \text{if } t_{j+2} \leq x < t_{j+3} \\ \quad + \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{x-t_{j+1}}{t_{j+3}-t_{j+1}} \right) \left(\frac{t_{j+3}-x}{t_{j+3}-t_{j+2}} \right) \\ \quad + \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{x-t_{j+2}}{t_{j+3}-t_{j+2}} \right) & \text{if } t_{j+3} \leq x < t_{j+4} \\ \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+1}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+2}} \right) \left(\frac{t_{j+4}-x}{t_{j+4}-t_{j+3}} \right) & \text{if } t_{j+3} \leq x < t_{j+4} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.13})$$

From this, the j th B-spline of degree 3, $B_{j,3}(x)$, is piecewise cubic with support over the interval $[t_j, t_{j+4})$. Plotting this cubic B-spline reveals a smooth function that is continuous through to second derivative for all real numbers x , including the knot points. We illustrate a series of these cubic B-splines in Figure C.1(c).

Our discussion here reveals some characteristic features of individual B-splines that we summarize as follows:

1. Local knots. The j th B-spline $B_{j,d}$ depends only on the $d+2$ knots $t_j, t_{j+1}, \dots, t_{j+d+1}$.
2. Local support. If x is outside the interval $[t_j, t_{j+d+1})$ then $B_{j,d}(x) = 0$.
3. Positivity. If $x \in (t_j, t_{j+d+1})$ then $B_{j,d}(x) > 0$.
4. Continuity Conditions. A B-spline $B_{j,d}$ is a continuous function at its defining knots $t_j, t_{j+1}, \dots, t_{j+d+1}$. When all knots belonging to $B_{j,d}$ are distinct, derivatives of the B-spline are also continuous up to the derivative of order $d - 1$.
5. Smoothness. The smoothness of $B_{j,d}$ increases with the degree d .
6. Local support. If x lies in the interval $[t_j, t_{j+1})$ then only the $d + 1$ B-splines $B_{j-d,d}, \dots, B_{j,d}$ are nonzero.
7. Unity partition. The sum of all B-splines of degree d is unity, i.e., $\sum_j B_{j,d}(x) = 1$, for all x .

Properties 1 through 4 follow by induction from the Cox-de Boor recurrence relation. With respect to property 4, we note that the continuity of derivative order is reduced by one for each coincident (nondistinct) knot point. We do not consider this situation further, but continue to assume all values within our knot vector $\mathbf{t} = (t_j)$ are distinct.

Properties 5 and 6 are additional properties observed when considering not just one B-spline but the linear span of all B-splines of a given degree d for a given knot sequence \mathbf{t} . We will draw upon all of these properties, along with the B-spline functional forms derived above, to describe the specific details of defining spline function representations of our continuous particle size distribution function.

Spline Function Approximations

In general, any continuous model can be approximated as a spline function of degree d with a given set of control points. Here, finally, we consider the details of our specific example: We want to represent a continuous particle size distribution function as a piecewise spline function by defining spline interpolants to connect a set of discrete control points.

As introduced above, the control points of a spline function must correspond to knot points contained within a knot vector $\mathbf{t} = (t_j)$. So far, we have been working with a bi-infinite set of distinct, nondecreasing knot points, with no constraints on the precise number and spacing of the vector elements. We will continue to assume that we have a sequence of distinct, nondecreasing knot points for our size distribution application, but we will address the specifications on number and spacing now.

A spline associated with knots that are equidistant from each other is called a *cardinal* spline. Because of the uniformity of such a knot sequence, the mathematical formulas involving cardinal B-splines are much simpler than the formulas corresponding to general B-splines, such as those we derived above. Nevertheless, we will continue to leave the spacing of our knot points unconstrained, as this provides flexibility within our inversion algorithm.

With respect to the length of our knot sequence, \mathbf{t} is necessarily a *finite* sequence in any practical situation. It is also necessary to have a minimum number $n \geq d + 1$ of control points to construct a degree d spline curve $S_{\mathbf{t}}^d(x)$. Thus, we set our n control points to correspond to the n knot points $(t_j)_{j=1}^n$ which define the locations at which our piecewise polynomials join together with certain continuity conditions.

At this point, we also need to distinguish between internal knots and end points. Internal knots cover the x -domain we are interested in, i.e., the closed interval $[t_1, t_n]$.

Yet we will ultimately be expressing our spline function as a linear combination of B-splines, and we have shown that a single B-spline extends over $d + 2$ knots. It follows then that the internal knots need to be extended with d end points on each side, to give full support to the first and last B-spline which affect the internal knot intervals.

We can now completely specify the knot vector we will need to obtain all of our spline interpolants. For n control points $s_j = S_{\mathbf{t}}^d(t_j)$, $j = 1, \dots, n$ we need $n + 2d$ knot points $\mathbf{t} = (t_j)_{j=1-d}^{n+d}$ to define $n + d - 1$ B-splines $\{B_{j,d}\}_{j=1-d}^{n-1}$ of degree d , and each B-spline will have a corresponding coefficient value $\mathbf{c} = (c_j)_{j=1-d}^{n-1}$. With this, we can now also specify the upper and lower bounds on the j index in our overall spline function definition:

$$S_{\mathbf{t}}^d(x) = \sum_{j=1-d}^{n-1} c_j B_{j,d}(x) \quad (\text{C.14})$$

In the linear case, the unknown spline coefficients c_j are uniquely defined by the interpolation conditions, and so their values correspond to the magnitude of the interpolated function. For the higher degree cubic splines, there is some arbitrariness in choosing the coefficient values. We will expand on this when we discuss our inversion kernel matrix setup.

We can simplify the spline function definition above. We have seen that on any nontrivial interval $[t_j, t_{j+1})$ between adjacent knot points, at most $d + 1$ of the $B_{j,d}$ are nonzero, viz. $B_{j-d,d}, \dots, B_{j,d}$. This means that any piece $S_j^d(x)|_{x \in [t_j, t_{j+1}]}$ of the spline function $S_{\mathbf{t}}^d(x)$ introduced above is a linear combination of $d + 1$ B-splines. This is expressed as

$$S_j^d(x) = \sum_{k=j-d}^j c_k B_{k,d}(x) \quad (\text{C.15})$$

This linear combination of B-splines satisfies the continuity conditions at the boundaries between adjacent intervals of the spline function $S_{\mathbf{t}}^d(x)$. We will use this expression to define our linear and cubic spline function approximations piece by piece.

Linear Spline

We use the definition above for first degree, or linear, B-splines with shorthand notation to emphasize the dependence of a B-spline on particular individual knots.

$$B^1(x|t_j, t_{j+1}, t_{j+2}) = \begin{cases} B1^1(x|t_j, t_{j+1}, t_{j+2}) & \text{if } t_j \leq x < t_{j+1} \\ B2^1(x|t_j, t_{j+1}, t_{j+2}) & \text{if } t_{j+1} \leq x < t_{j+2} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.16})$$

We then describe any segment $S_j^1(x)|_{x \in [t_j, t_{j+1}]}$ of our size distribution function by

$$\begin{aligned} S_j^1(x) &= \sum_{k=j-1}^j c_k B_{k,1}(x) \\ &= c_{j-1} B_{j-1,1}(x) + c_j B_{j,1}(x) \\ &= c_{j-1} B2^1(x|t_{j-1}, t_j, t_{j+1}) + c_j B1^1(x|t_j, t_{j+1}, t_{j+2}) \end{aligned} \quad (\text{C.17})$$

This should produce the familiar linear interpolation formula

$$S_j^1(x) = c_{j-1} \left(\frac{t_{j+1} - x}{t_{j+1} - t_j} \right) + c_j \left(\frac{x - t_j}{t_{j+1} - t_j} \right) \quad (\text{C.18})$$

Cubic Spline

We use the definition above for third degree, or cubic, B-splines with shorthand notation to emphasize the dependence of a B-spline on particular individual knots.

$$B^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) = \begin{cases} B1^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) & \text{if } t_j \leq x < t_{j+1} \\ B2^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) & \text{if } t_{j+1} \leq x < t_{j+2} \\ B3^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) & \text{if } t_{j+2} \leq x < t_{j+3} \\ B4^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) & \text{if } t_{j+3} \leq x < t_{j+4} \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.19})$$

We then describe any segment $S_j^3(x)|_{x \in [t_j, t_{j+1}]}$ of our size distribution function by

$$\begin{aligned} S_j^3(x) &= \sum_{k=j-3}^j c_k B_{k,3}(x) \\ &= c_{j-3} B_{j-3,3}(x) + c_{j-2} B_{j-2,3}(x) + c_{j-1} B_{j-1,3}(x) + c_j B_{j,3}(x) \\ &= c_{j-3} B4^3(x|t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) + c_{j-2} B3^3(x|t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) \\ &\quad + c_{j-1} B2^3(x|t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3}) + c_j B1^3(x|t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) \end{aligned} \quad (\text{C.20})$$

Appendix D

KERNEL MATRIX SETUP

The indices used here to discuss and demonstrate the setup of inversion kernel matrices can be understood as follows:

1. $i = 1 \dots m$ indexes the observed instrument channel, here specifically the raw voltage bins and instrument responses R_i
2. $j = 1 \dots n$ indexes the physical size bins
3. $j = -d \dots n$ indexes the coefficient values for a spline of degree d
4. $j = -d \dots (n + d + 1)$ indexes the knot points needed to construct the spline curve
5. $p = 0 \dots n_{sub}$ indexes the secondary integration points, or subdivision points, used to evaluate the subinterval integrals for kernel matrix elements

Signals in each of m channels may be written

$$g_i = \int_a^b K_i(x) f(x) dx + e_i, \quad i = 1, \dots, m \quad (\text{D.1})$$

We break the integral over $[a, b]$ into n subintervals centered around n output nodes $(x_j)_{j=1}^n$ so that the $n + 1$ “primary integration points” are defined to be the geometric mean values between successive output nodes,

$$x_{j \pm \frac{1}{2}} = \sqrt{x_j x_{j \pm 1}} \quad (\text{D.2})$$

At the endpoints, we use $x_0 = 2x_1 - x_2$ and $x_{n+1} = 2x_n - x_{n-1}$ to evaluate $x_{-\frac{1}{2}}$ and $x_{n+\frac{1}{2}}$. This gives

$$g_i = \sum_{j=1}^n \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) f(x) dx \quad (\text{D.3})$$

If we represent the size distribution function as a histogram, we then place a discrete value of $f(x)$ outside of the integral such that

$$g_i = \sum_{j=1}^n f(x_j) \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx \quad (\text{D.4})$$

The discrete value $f(x_j)$ becomes the unknown that is to be recovered from the inversion analysis. If x_j corresponds to the point in $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ where $f(x)$ achieves its mean value for that subinterval (i.e., $f(x_j) = f(c_j)$), then this is a simple application of the first mean value theorem for definite integrals. However, the location of this c_j cannot be precisely known a priori, so it is often taken to represent some midpoint value for the j th subinterval. Here, we simply use the discrete output nodes $(x_j)_{j=1}^n$ around which we have defined our size bin intervals.

If the kernel function $K_i(x)$ is also represented by only a single value evaluated at the point x_j within the j th subinterval, then

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx = K_i(x_j) \Delta x_j \quad (\text{D.5})$$

where $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. Ideally, $K_i(x_j)$ represents the true bin-average value for the given subinterval so that this is another application of the first mean value theorem for definite integrals (i.e., $K(x_j) = K(c'_j)$). However, $K_i(x)$ is often simply evaluated at the same midpoint value x_j chosen for the interval as defined above. This provides the simplest approximation which we call “simple quadrature.”

$$g_i = \sum_{j=1}^n f(x_j) K_i(x_j) \Delta x_j \quad (\text{D.6})$$

$$g_i = \sum_{j=1}^n K_{ij} f_j \quad (\text{D.7})$$

$$K_{ij} = K_i(x_j) \Delta x_j \quad (\text{D.8})$$

$$f_j = f(x_j) \quad (\text{D.9})$$

However, if the kernel function can be defined with finer resolution than the width of the size bin intervals, then we can evaluate the integral of the kernel function over

the size bin interval using the trapezoidal rule

$$\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) dx = \sum_{p=0}^{n_{sub}} \beta_p K_i(x_{j,p}) \Delta x_{j,p} \quad (\text{D.10})$$

where n_{sub} is the number of subintervals to use in evaluating the integral over the j th size bin, $x_{j,p}$ represents the secondary integration point at which to evaluate the kernel function within the j th bin, $\Delta x_{j,p}$ is the width of the n_{sub} subinterval bins, and $\beta_p = 1/2$ at the endpoints of the summation and 1 elsewhere. The set of $n_{sub} + 1$ points $(x_{j,p})_{p=0}^{n_{sub}}$ that define the secondary integration points (i.e., subinterval boundaries) are evaluated as

$$x_{j,p} = x_{j-\frac{1}{2}} + \Delta x_{j,p} \cdot p, \quad p = 0, \dots, n_{sub} \quad (\text{D.11})$$

with

$$\Delta x_{j,p} = \frac{\Delta x_j}{n_{sub}} \quad (\text{D.12})$$

$$\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}} \quad (\text{D.13})$$

This provides a slightly improved approximation which we call ‘‘histogram quadrature’’ since the size distribution function is still represented as a histogram.

$$g_i = \sum_{j=1}^n f(x_j) \left[\sum_{p=0}^{n_{sub}} \beta_p K_i(x_{j,p}) \Delta x_{j,p} \right] \quad (\text{D.14})$$

$$g_i = \sum_{j=1}^n K_{ij} f_j \quad (\text{D.15})$$

$$K_{ij} = \sum_{p=0}^{n_{sub}} \beta_p K_i(x_{j,p}) \Delta x_{j,p} \quad (\text{D.16})$$

$$f_j = f(x_j) \quad (\text{D.17})$$

Relying on a single value for the size distribution function $f(x)$ or for the kernel function $K_i(x)$ over the size bin interval corresponding to x_j can be a major source of error in inverse calculations. More generally, the size distribution function $f(x)$ in the response integral can be approximated as a spline of degree d with knots at a

certain set of quadrature points. Then

$$\begin{aligned}
 g_i &= \sum_{j=1}^n \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} K_i(x) f(x) dx \\
 &= \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) S_{j-1}^d(x) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) S_j^d(x) dx \right\} \quad (D.18)
 \end{aligned}$$

We have defined a nondecreasing sequence of real numbers called the knot vector, given by $\mathbf{t} = (t_j)_{j=-d}^{n+d+1}$ such that the knots exactly correspond to the n output nodes of the size distribution, with some extrapolation beyond the upper and lower bounds (i.e., $(t_j)_{j=1}^n = (x_j)_{j=1}^n$).

Linear Spline Quadrature

With the definition above for describing any segment $S_j^1(x)$ of our size distribution function using first degree, or linear, B-splines:

$$S_j^1(x) = c_{j-1} B2^1(x | t_{j-1}, t_j, t_{j+1}) + c_j B1^1(x | t_j, t_{j+1}, t_{j+2})$$

we express our response integral as

$$\begin{aligned}
 g_i &= \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) [c_{j-2} B2^1(x | t_{j-2}, t_{j-1}, t_j) + c_{j-1} B1^1(x | t_{j-1}, t_j, t_{j+1})] dx \right. \\
 &\quad \left. + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) [c_{j-1} B2^1(x | t_{j-1}, t_j, t_{j+1}) + c_j B1^1(x | t_j, t_{j+1}, t_{j+2})] dx \right\} \quad (D.19)
 \end{aligned}$$

and collecting like terms gives

$$\begin{aligned}
 g_i = \sum_{j=1}^n & \left\{ c_{j-2} \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B2^1(x | t_{j-2}, t_{j-1}, t_j) dx \right. \\
 & + c_{j-1} \left[\int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B1^1(x | t_{j-1}, t_j, t_{j+1}) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B2^1(x | t_{j-1}, t_j, t_{j+1}) dx \right] \\
 & \left. + c_j \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B1^1(x | t_j, t_{j+1}, t_{j+2}) dx \right\} \tag{D.20}
 \end{aligned}$$

We evaluate this expression for a simple three knot function, to demonstrate endpoint

behavior. Expanding the sum for $n = 3$ gives

$$\begin{aligned}
 g_i = & c_{-1} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^1(x | t_{-1}, t_0, t_1) dx \\
 & + c_0 \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^1(x | t_0, t_1, t_2) dx + \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B2^1(x | t_0, t_1, t_2) dx \right] \\
 & + c_1 \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B1^1(x | t_1, t_2, t_3) dx \\
 & + c_0 \int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B2^1(x | t_0, t_1, t_2) dx \\
 & + c_1 \left[\int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B1^1(x | t_1, t_2, t_3) dx + \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B2^1(x | t_1, t_2, t_3) dx \right] \\
 & + c_2 \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B1^1(x | t_2, t_3, t_4) dx \\
 & + c_1 \int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B2^1(x | t_1, t_2, t_3) dx \\
 & + c_2 \left[\int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B1^1(x | t_2, t_3, t_4) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B2^1(x | t_2, t_3, t_4) dx \right] \\
 & + c_3 \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B1^1(x | t_3, t_4, t_5) dx
 \end{aligned} \tag{D.21}$$

and collecting like terms gives

$$\begin{aligned}
 g_i = & c_{-1} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^1(x | t_{-1}, t_0, t_1) dx \\
 & + c_0 \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^1(x | t_0, t_1, t_2) dx + \int_{x_1}^{x_2} K_i(x) B2^1(x | t_0, t_1, t_2) dx \right] \\
 & + c_1 \left[\int_{x_1}^{x_2} K_i(x) B1^1(x | t_1, t_2, t_3) dx + \int_{x_2}^{x_3} K_i(x) B2^1(x | t_1, t_2, t_3) dx \right] \\
 & + c_2 \left[\int_{x_2}^{x_3} K_i(x) B1^1(x | t_2, t_3, t_4) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B2^1(x | t_2, t_3, t_4) dx \right] \\
 & + c_3 \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B1^1(x | t_3, t_4, t_5) dx
 \end{aligned} \tag{D.22}$$

Written in matrix form

$$g_i = \begin{bmatrix} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^1(x | t_{-1}, t_0, t_1) dx \\ \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^1(x | t_0, t_1, t_2) dx + \int_{x_1}^{x_2} K_i(x) B2^1(x | t_0, t_1, t_2) dx \right] \\ \left[\int_{x_1}^{x_2} K_i(x) B1^1(x | t_1, t_2, t_3) dx + \int_{x_2}^{x_3} K_i(x) B2^1(x | t_1, t_2, t_3) dx \right] \\ \left[\int_{x_2}^{x_3} K_i(x) B1^1(x | t_2, t_3, t_4) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B2^1(x | t_2, t_3, t_4) dx \right] \\ \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B1^1(x | t_3, t_4, t_5) dx \end{bmatrix}^T \times \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \tag{D.23}$$

Cubic Spline Quadrature

With the definition above for describing any segment $S_j^3(x)$ of our size distribution function using third degree, or cubic, B-splines:

$$\begin{aligned}
 S_j^3(x) &= c_{j-3} B4^3(x | t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) + c_{j-2} B3^3(x | t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) \\
 &= + c_{j-1} B2^3(x | t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3}) + c_j B1^3(x | t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4})
 \end{aligned}$$

we express our response integral as

$$\begin{aligned}
g_i = \sum_{j=1}^n \left\{ \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) [c_{j-4}B4^3(x | t_{j-4}, t_{j-3}, t_{j-2}, t_{j-1}, t_j) + c_{j-3}B3^3(x | t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) \right. \\
+ c_{j-2}B2^3(x | t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) + c_{j-1}B1^3(x | t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3})] dx \\
+ \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) [c_{j-3}B4^3(x | t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) + c_{j-2}B3^3(x | t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) \\
+ c_{j-1}B2^3(x | t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3}) + c_j B1^3(x | t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4})] dx \left. \right\} \quad (D.24)
\end{aligned}$$

and collecting like terms gives

$$\begin{aligned}
g_i = \sum_{j=1}^n \left\{ c_{j-4} \int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B4^3(x | t_{j-4}, t_{j-3}, t_{j-2}, t_{j-1}, t_j) dx \right. \\
+ c_{j-3} \left[\int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B3^3(x | t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B4^3(x | t_{j-3}, t_{j-2}, t_{j-1}, t_j, t_{j+1}) dx \right] \\
+ c_{j-2} \left[\int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B2^3(x | t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B3^3(x | t_{j-2}, t_{j-1}, t_j, t_{j+1}, t_{j+2}) dx \right] \\
+ c_{j-1} \left[\int_{x_{j-\frac{1}{2}}}^{x_j} K_i(x) B1^3(x | t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3}) dx + \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B2^3(x | t_{j-1}, t_j, t_{j+1}, t_{j+2}, t_{j+3}) dx \right] \\
+ c_j \int_{x_j}^{x_{j+\frac{1}{2}}} K_i(x) B1^3(x | t_j, t_{j+1}, t_{j+2}, t_{j+3}, t_{j+4}) dx \left. \right\} \quad (D.25)
\end{aligned}$$

We evaluate this expression for a simple five knot function, to demonstrate endpoint

behavior. Expanding the sum for $n = 5$ gives

$$\begin{aligned}
g_i = & c_{-3} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B4^3(x | t_{-3}, t_{-2}, t_{-1}, t_0, t_1) dx \\
& + c_{-2} \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B3^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx + \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B4^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx \right] \\
& + c_{-1} \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx + \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B3^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \right] \\
& + c_0 \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B2^3(x | t_0, t_1, t_2, t_3, t_4) dx \right] \\
& + c_1 \int_{x_1}^{x_{\frac{3}{2}}} K_i(x) B1^3(x | t_1, t_2, t_3, t_4, t_5) dx \\
& + c_{-2} \int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B4^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx \\
& + c_{-1} \left[\int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B3^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx + \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B4^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \right] \\
& + c_0 \left[\int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B2^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B3^3(x | t_0, t_1, t_2, t_3, t_4) dx \right] \\
& + c_1 \left[\int_{x_{\frac{3}{2}}}^{x_2} K_i(x) B1^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B2^3(x | t_1, t_2, t_3, t_4, t_5) dx \right] \\
& + c_2 \int_{x_2}^{x_{\frac{5}{2}}} K_i(x) B1^3(x | t_2, t_3, t_4, t_5, t_6) dx
\end{aligned} \tag{D.26}$$

cont'd:

$$\begin{aligned}
g_i = & + c_{-1} \int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B4^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \\
& + c_0 \left[\int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B3^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B4^3(x | t_0, t_1, t_2, t_3, t_4) dx \right] \\
& + c_1 \left[\int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B2^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B3^3(x | t_1, t_2, t_3, t_4, t_5) dx \right] \\
& + c_2 \left[\int_{x_{\frac{5}{2}}}^{x_3} K_i(x) B1^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B2^3(x | t_2, t_3, t_4, t_5, t_6) dx \right] \\
& + c_3 \int_{x_3}^{x_{\frac{7}{2}}} K_i(x) B1^3(x | t_3, t_4, t_5, t_6, t_7) dx \\
& + c_0 \int_{x_{\frac{7}{2}}}^{x_4} K_i(x) B4^3(x | t_0, t_1, t_2, t_3, t_4) dx \\
& + c_1 \left[\int_{x_{\frac{7}{2}}}^{x_4} K_i(x) B3^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_4}^{x_{\frac{9}{2}}} K_i(x) B4^3(x | t_1, t_2, t_3, t_4, t_5) dx \right] \\
& + c_2 \left[\int_{x_{\frac{7}{2}}}^{x_4} K_i(x) B2^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_4}^{x_{\frac{9}{2}}} K_i(x) B3^3(x | t_2, t_3, t_4, t_5, t_6) dx \right] \\
& + c_3 \left[\int_{x_{\frac{7}{2}}}^{x_4} K_i(x) B1^3(x | t_3, t_4, t_5, t_6, t_7) dx + \int_{x_4}^{x_{\frac{9}{2}}} K_i(x) B2^3(x | t_3, t_4, t_5, t_6, t_7) dx \right] \\
& + c_4 \int_{x_4}^{x_{\frac{9}{2}}} K_i(x) B1^3(x | t_4, t_5, t_6, t_7, t_8) dx
\end{aligned}$$

cont'd:

$$\begin{aligned}
g_i = & + c_1 \int_{\frac{x_9}{2}}^{x_5} K_i(x) B4^3(x | t_1, t_2, t_3, t_4, t_5) dx \\
& + c_2 \left[\int_{\frac{x_9}{2}}^{x_5} K_i(x) B3^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B4^3(x | t_2, t_3, t_4, t_5, t_6) dx \right] \\
& + c_3 \left[\int_{\frac{x_9}{2}}^{x_5} K_i(x) B2^3(x | t_3, t_4, t_5, t_6, t_7) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B3^3(x | t_3, t_4, t_5, t_6, t_7) dx \right] \\
& + c_4 \left[\int_{\frac{x_9}{2}}^{x_5} K_i(x) B1^3(x | t_4, t_5, t_6, t_7, t_8) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B2^3(x | t_4, t_5, t_6, t_7, t_8) dx \right] \\
& + c_5 \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B1^3(x | t_5, t_6, t_7, t_8, t_9) dx
\end{aligned}$$

and collecting like terms gives

$$\begin{aligned}
g_i = & c_{-3} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B4^3(x | t_{-3}, t_{-2}, t_{-1}, t_0, t_1) dx \\
& + c_{-2} \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B3^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx + \int_{x_1}^{x_2} K_i(x) B4^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx \right] \\
& + c_{-1} \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx + \int_{x_1}^{x_2} K_i(x) B3^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \right. \\
& \quad \left. + \int_{x_2}^{x_3} K_i(x) B4^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \right] \\
& + c_0 \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_1}^{x_2} K_i(x) B2^3(x | t_0, t_1, t_2, t_3, t_4) dx \right. \\
& \quad \left. + \int_{x_2}^{x_3} K_i(x) B3^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_3}^{x_4} K_i(x) B4^3(x | t_0, t_1, t_2, t_3, t_4) dx \right] \\
& + c_1 \left[\int_{x_1}^{x_2} K_i(x) B1^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_2}^{x_3} K_i(x) B2^3(x | t_1, t_2, t_3, t_4, t_5) dx \right. \\
& \quad \left. + \int_{x_3}^{x_4} K_i(x) B3^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_4}^{x_5} K_i(x) B4^3(x | t_1, t_2, t_3, t_4, t_5) dx \right] \\
& + c_2 \left[\int_{x_2}^{x_3} K_i(x) B1^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_3}^{x_4} K_i(x) B2^3(x | t_2, t_3, t_4, t_5, t_6) dx \right. \\
& \quad \left. + \int_{x_4}^{x_5} K_i(x) B3^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B4^3(x | t_2, t_3, t_4, t_5, t_6) dx \right] \\
& + c_3 \left[\int_{x_3}^{x_4} K_i(x) B1^3(x | t_3, t_4, t_5, t_6, t_7) dx + \int_{x_4}^{x_5} K_i(x) B2^3(x | t_3, t_4, t_5, t_6, t_7) dx \right. \\
& \quad \left. + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B3^3(x | t_3, t_4, t_5, t_6, t_7) dx \right] \\
& + c_4 \left[\int_{x_4}^{x_5} K_i(x) B1^3(x | t_4, t_5, t_6, t_7, t_8) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B2^3(x | t_4, t_5, t_6, t_7, t_8) dx \right] \\
& + c_5 \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B1^3(x | t_5, t_6, t_7, t_8, t_9) dx
\end{aligned} \tag{D.27}$$

Written in matrix form

$$\begin{aligned}
 g_i = & \left[\begin{array}{c} \int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B4^3(x | t_{-3}, t_{-2}, t_{-1}, t_0, t_1) dx \\ \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B3^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx + \int_{x_1}^{x_2} K_i(x) B4^3(x | t_{-2}, t_{-1}, t_0, t_1, t_2) dx \right] \\ \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B2^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx + \int_{x_1}^{x_2} K_i(x) B3^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx + \int_{x_2}^{x_3} K_i(x) B4^3(x | t_{-1}, t_0, t_1, t_2, t_3) dx \right] \\ \left[\int_{x_{\frac{1}{2}}}^{x_1} K_i(x) B1^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_1}^{x_2} K_i(x) B2^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_2}^{x_3} K_i(x) B3^3(x | t_0, t_1, t_2, t_3, t_4) dx + \int_{x_3}^{x_4} K_i(x) B4^3(x | t_0, t_1, t_2, t_3, t_4) dx \right] \\ \left[\int_{x_1}^{x_2} K_i(x) B1^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_2}^{x_3} K_i(x) B2^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_3}^{x_4} K_i(x) B3^3(x | t_1, t_2, t_3, t_4, t_5) dx + \int_{x_4}^{x_5} K_i(x) B4^3(x | t_1, t_2, t_3, t_4, t_5) dx \right] \\ \left[\int_{x_2}^{x_3} K_i(x) B1^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_3}^{x_4} K_i(x) B2^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_4}^{x_5} K_i(x) B3^3(x | t_2, t_3, t_4, t_5, t_6) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B4^3(x | t_2, t_3, t_4, t_5, t_6) dx \right] \\ \left[\int_{x_3}^{x_4} K_i(x) B1^3(x | t_3, t_4, t_5, t_6, t_7) dx + \int_{x_4}^{x_5} K_i(x) B2^3(x | t_3, t_4, t_5, t_6, t_7) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B3^3(x | t_3, t_4, t_5, t_6, t_7) dx \right] \\ \left[\int_{x_4}^{x_5} K_i(x) B1^3(x | t_4, t_5, t_6, t_7, t_8) dx + \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B2^3(x | t_4, t_5, t_6, t_7, t_8) dx \right] \\ \int_{x_5}^{x_{\frac{11}{2}}} K_i(x) B1^3(x | t_5, t_6, t_7, t_8, t_9) dx \end{array} \right]^T \times \begin{bmatrix} c_{-3} \\ c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} \quad (D.28)
 \end{aligned}$$

This illustrates the general strategy: successively apply the functional form of the spline function and rearrange terms until we have an expression where the control points (i.e., B-spline coefficients) appear explicitly.

Appendix E

INVERSION CODE

This appendix serves to archive the critical components of the cubic spline inverse solution method discussed in Chapters 2 and 3. All of the following code was written for the *Igor Pro* commercial scientific analysis application, version 6.37 (Wavemetrics, 2015).

Printed here are all functions related to analyzing experimental measurements from a long column DMA system. This includes the calculations involved in evaluating the following:

- All components of the DMA kernel function

$$\Psi(i, D_p) = Q_a \sum_{v=1}^{\infty} W_{v\eta}(D_p) \phi_v(D_p) \Omega(v, i, D_p)$$

- Inversion kernel matrix elements with B-spline quadrature

$$K_{ij} = \int_{\mathcal{I}} K_i(x) B_{j,d}(x) dx$$

- Inverse solution to recover the particle number size distribution

$$n_N(\ln D_p) = \frac{dN}{d \ln D_p} [\text{cm}^{-3}]$$

- Each of the computational solution methods available within this inversion routine
 - NNLS: the non-negative least squares method of Lawson and Hanson (1987);
 - TNNLS: an interior-point gradient method for large-scale totally non-negative least squares problems from the work of Merritt and Zhang (2005);
 - Regularization: a regularization method (Ilavsky and Jemian 2009).

```

// Make matrix to hold recorded data from a given measurement campaign
// values will be read from input data to populate the "DMAmeasParams" wave
function makedatamatrix()
    make/o/n=(1,12) datamatrix
    setdimlabel 1,0,iScan, datamatrix
    setdimlabel 1,1,Qa_lpm, datamatrix
    setdimlabel 1,2,Qm_lpm, datamatrix
    setdimlabel 1,3,Qsh_lpm, datamatrix
    setdimlabel 1,4,Qex_lpm, datamatrix
    setdimlabel 1,5,HVset_V, datamatrix
    setdimlabel 1,6,Counts, datamatrix
    setdimlabel 1,7,tsample_s, datamatrix
    setdimlabel 1,8,Tamb_K, datamatrix
    setdimlabel 1,9,Pamb_Pa, datamatrix
    setdimlabel 1,10,dTdet_C, datamatrix
    setdimlabel 1,11,Qdet_lpm, datamatrix
end

// Make input wave to specify parameters for DMA operation
// input values will be read to the "DMAgeomParams" wave; called by dimension labels
function makeWvDMAparams2specify()
    make/o/n=6 DMAparams2specify
    setdimlabel 0,0,R1dma_m, DMAparams2specify
    setdimlabel 0,1,R2dma_m, DMAparams2specify
    setdimlabel 0,2,Ldma_m, DMAparams2specify
    setdimlabel 0,3,Leq_DMA_m, DMAparams2specify
    setdimlabel 0,4,Leq_other_m, DMAparams2specify
    setdimlabel 0,5,particleDensity_kgm3, DMAparams2specify
end

// Make input wave to specify parameters for kernel matrix calculations
// parameters called from this wave by dimension labels
function makeWvcalresponsematrixParams()
    make/o/n=6 calresponsematrixParams
    setdimlabel 0,0,ndias2nchanRatio, calresponsematrixParams
    setdimlabel 0,1,ZfactorLower, calresponsematrixParams
    setdimlabel 0,2,ZfactorUpper, calresponsematrixParams
    setdimlabel 0,3,nSub4Kernel, calresponsematrixParams
    setdimlabel 0,4,chargemax4Kernel, calresponsematrixParams
    setdimlabel 0,5,extrapMeasChan, calresponsematrixParams // enter 1 for yes; 0 for no

    make/o/t/n=1 CalcTFformula // text wave to call form of transfer function
    make/o/t/n=1 InvMethod // text wave to call solution computation algorithm
end
//=====

```

```

// Make matrix to hold equivalent pipe lengths and flow parameters to account for losses in
// system components/tubing where flow rate may differ from DMA flow rate
// (i.e., sample line splitting to different instruments)
// **Account for losses thru DMA inlet/outlet, and other components with same DMA flow rate,
// in "DMAparams2specify" input wave
// See Wiedensohler 2012 for discussion of method of equivalent pipe length for DMA systems
//
// diffLossMatrix columns:
// 0: effective length of device/component of system where diffusional losses occur [m]
// 1: corresponding flow rate of aerosol through device/component/tubing [lpm]
// Create as many rows as needed
function makeDiffLossMatrix()
    make/o/n=(1,2) diffLossMatrix
    setdimlabel 0,-1,Component, diffLossMatrix
    setdimlabel 1,0,Leq_m, diffLossMatrix
    setdimlabel 1,1,Q_lpm, diffLossMatrix
end

// Make waves to hold parameters from given measurement campaign
// calculations call from these waves to use these parameters; called by dimension labels
function makeWvDMAmeasParams()
    make/o/n=10 DMAmeasParams
    setdimlabel 0,0,Qa_lpm, DMAmeasParams
    setdimlabel 0,1,Qsh_lpm, DMAmeasParams
    setdimlabel 0,2,betaDMA, DMAmeasParams
    setdimlabel 0,3,deltaDMA, DMAmeasParams
    setdimlabel 0,4,tsample_s, DMAmeasParams
    setdimlabel 0,5,dTdet_C, DMAmeasParams
    setdimlabel 0,6,Qdet_lpm, DMAmeasParams
    setdimlabel 0,7,Tamb_K, DMAmeasParams
    setdimlabel 0,8,Pamb_Pa, DMAmeasParams
    setdimlabel 0,9,particleDensity_kgm3, DMAmeasParams
end
function makeWvDMAgeomParams()
    make/o/n=7 DMAgeomParams
    setdimlabel 0,0,R1dma_m, DMAgeomParams
    setdimlabel 0,1,R2dma_m, DMAgeomParams
    setdimlabel 0,2,Ldma_m, DMAgeomParams
    setdimlabel 0,3,f_dma, DMAgeomParams
    setdimlabel 0,4,G_dma, DMAgeomParams
    setdimlabel 0,5,Leq_DMA_m, DMAgeomParams
    setdimlabel 0,6,Leq_other_m, DMAgeomParams
end
//=====

```

```

// Analyze multiple scans of DMA data
// Input options and waves:
// datamatrix - data recorded from measurement campaign
// diffLossMatrix - equivalent pipe lengths and flow parameters for system components
// DMAparams2specify - instrument geometry and Leq; particle density
// calcresponsematrixParams - options for kernel matrix setup
// CalcTFformula - text wave to call form of transfer function
// InvMethod - text wave to call solution computation algorithm
// Kernel_precalc - can input a previously calculated kernel matrix to save computation time;
//                   otherwise, input empty set {}
// Output:
// invsolnMeas      [m x nScans]      dN/dlnDp [cm-3] evaluated over measurement range
// dp_meas          [m x nScans]      Dp [m] corresponding to measurement range
// invsolnCoeffs    [ncoeff x nScans] inverse solution coefficients; use to evaluate
//                   cubic spline solution over size range of interest
function DMAinversion(datamatrix,diffLossMatrix,DMAparams2specify,calcresponsematrixParams,
    wave datamatrix,diffLossMatrix,DMAparams2specify
    wave calcresponsematrixParams
    wave/t CalcTFformula,InvMethod
    wave Kernel_precalc

    matrixop/o scanindxWv = col(datamatrix,0)
    variable nScans = scanindxWv[numpnts(scanindxWv)-1]-scanindxWv[0]+1
    variable ndatapnts = numpnts(scanindxWv)/(nScans)

    variable i=0
    variable scanindx = scanindxWv[0]+i
    extract/indx/o scanindxWv, indexforscan, scanindxWv==scanindx
    duplicate/o/r=[indexforscan[0],indexforscan[numpnts(indexforscan)-1]] datamatrix, scandata

    DMAanalysis(scandata,diffLossMatrix,DMAparams2specify,calcresponsematrixParams,CalcTF
    wave CountData_inv,CountBin_inv,Sigma_countdata_inv
    wave invsoln_meas,dpwave_measdistbin
    wave invsoln_coefs,lnDpbin_inv,BsplineKnotWave_inv
    wave Kernel_inv

```

```

duplicate/o CountData_inv, rawdata_inv; redimension/n=(-1,nScans) rawdata_inv
duplicate/o CountBin_inv, dpmeas_inv; redimension/n=(-1,nScans) dpmeas_inv
duplicate/o Sigma_countdata_inv, sigmadata_inv; redimension/n=(-1,nScans) sigmadata_inv
duplicate/o invsoln_meas, invsolnMeas; redimension/n=(-1,nScans) invsolnMeas
duplicate/o dpwave_measdistbin, dp_meas; redimension/n=(-1,nScans) dp_meas
duplicate/o invsoln_coefs, invsolnCoeffs; redimension/n=(-1,nScans) invsolnCoeffs
duplicate/o lnDpbin_inv, invsolnSplineModelpts
duplicate/o BsplineKnotWave_inv, invsolnSplineKnots

for(i=1;i<nScans;i+=1)
  scanindx = scanindxWv[0]+i
  extract/indx/o scanindxWv, indexforscan, scanindxWv==scanindx
  duplicate/o/r=[indexforscan[0],indexforscan[numpnts(indexforscan)-1]] datamatrix, scandata

  DMAanalysis(scandata,diffLossMatrix,DMAparams2specify,calcreponsematrixParams,Calc
  wave CountData_inv,CountBin_inv,Sigma_countdata_inv
  wave invsoln_meas,dpwave_measdistbin
  wave invsoln_coefs

  rawdata_inv[][i] = CountData_inv[p]
  dpmeas_inv[][i] = CountBin_inv[p]
  sigmadata_inv[][i] = Sigma_countdata_inv[p]
  invsolnMeas[][i] = invsoln_meas[p]
  dp_meas[][i] = dpwave_measdistbin[p]
  invsolnCoeffs[][i] = invsoln_coefs[p]
endfor
end
//=====

```

```

// Analyze a single scan of DMA data
// Input options and waves:
// datamatrix - data recorded from measurement campaign
// diffLossMatrix - equivalent pipe lengths and flow parameters for system components
// DMAparams2specify - instrument geometry and Leq; particle density
// calcresponsematrixParams - options for kernel matrix setup
// CalcTFformula - text wave to call form of transfer function
// InvMethod - text wave to call solution computation algorithm
// Kernel_precalc - can input a previously calculated kernel matrix to save computation time;
//                   otherwise, input empty set {}
// Output:
// dN/dlnDp [cm-3]   size distribution inverse solution(s) in "invsoln_xxx" waves
// Dp [m]           corresponding particle diameters in "dpwave_xxx" waves
function DMAanalysis(datamatrix,diffLossMatrix,DMAparams2specify,calcresponsematrixParams,(
    wave datamatrix,diffLossMatrix,DMAparams2specify
    wave calcresponsematrixParams
    wave/t CalcTFformula,InvMethod
    wave Kernel_precalc

    string TFformula,InvSolnMethod
    TFformula=CalcTFformula[0]
    InvSolnMethod=InvMethod[0]

    // Set measurement/instrument parameters
    DMAparamSet(datamatrix,DMAparams2specify)
    wave DMAmeasParams, DMAgeomParams
    nvar Qae,Qmono,Qsheath,Qex,tsample,Qdet

    // Generate channel parameters
    variable/g numchan=DimSize(datamatrix,0) // number of measured points in scan
    variable/g bbeta=betDMA()
    variable/g ddelta=deltaDMA()

    make/o/n=(numchan) ZstarChan, VstarChan, DstarChan
    VstarChan=datamatrix[p][5] // read channel voltages from datamatrix;
    ZstarChan=ZpCalc_Vdma(VstarChan) // convert to mobility and...
    DstarChan=DpCalc_Vdma(VstarChan) //...diameter targeted by instrument
    variable/g ZmaxDMA=ZstarChan[0]
    variable/g ZminDMA=ZstarChan[numchan-1]

```

```

// Set up wave of raw count data
matrixop/o Counts=col(datamatrix,6) // read counts from datamatrix; raw signal [#]
// extrapolate for inversion if input "extrapYorN" set to 1
duplicate/o ZstarChan, invZstarChan; invZstarChan=1/ZstarChan
duplicate/o ChanZ_inv, invChanZ_inv; invChanZ_inv=1/ChanZ_inv
extrapolateCounts_exp(Counts,invZstarChan,invChanZ_inv,2)
wave ExtrCounts
duplicate/o ExtrCounts, Counts_inv, Sigma_counts_inv
duplicate/o ChanDp_inv, CountBin_inv
// estimate uncertainty using Poisson statistics
Counts_inv = Counts_inv < 1 ? 0 : Counts_inv // if raw count [#] shows <1 particle, set to zero
Sigma_counts_inv=1.96*sqrt(Counts_inv)

// Account for flow rates and sample time in kernel matrix;
// calculate sampled volume for each data point
variable/g sampleQt_m3=Qae*tsample/60000 // should depend on aerosol sample inlet flow
if (Qdet<Qmono) //...unless low detector flow requires diverting part of classified outlet flow
    sampleQt_m3*=Qdet/Qmono //...then account for sampling fraction of particle flux
endif
matrixop/o Kernel_invQt_cc=Kernel_inv*sampleQt_m3*1e6 // [cm3]

// Cubic spline inversion
make/o/n=(mKern) CountData_inv; CountData_inv=Counts_inv
make/o/n=(mKern) Sigma_countdata_inv; Sigma_countdata_inv=Sigma_counts_inv
make/o/n=(mKern,nKern) KernelMatrix_inv; KernelMatrix_inv=Kernel_invQt_cc
invsoln_Bspline(KernelMatrix_inv,lnDpbin_inv,lnDpbin_invmeas,BsplineCoeffKnots_inv,Bspline
wave invsoln_coefs // inverse solution: coefficient values to use to evaluate size distribution..
wave invsoln_controlpts,dpwave_controlptdistbin //...at spline control points
wave invsoln_model,dpwave_modeldistbin //...at model knot points
wave invsoln_meas,dpwave_measdistbin //...at target Dp of measurement channels
//...or any other Dp set; see function invsoln_fineEval(...)
variable/g n_invModel=umpnts(lnDpbin_inv)
variable/g mKernel_inv=dimsize(Kernel_inv,0)
variable/g nKernel_inv=dimsize(Kernel_inv,1)
end
//=====

```

```

// Read datamatrix for single scan of DMA data to set measurement/instrument parameters
function DMAparamSet(datamatrix,DMAparams2specify)
    wave datamatrix,DMAparams2specify
    //-----
    // datamatrix columns:
    // 0: scan index
    // 1: Qaerosol [lpm]
    // 2: Qmono [lpm]
    // 3: Qsheath [lpm]
    // 4: Qexcess [lpm]
    // 5: HVset [volts]
    // 6: raw count data [#]
    // 7: tsample [s]
    // 8: Tambient [K]
    // 9: Pambient [Pa]
    // 10:deltaT_detector [degC]
    // 11:Qdetector [lpm]
    //-----
    //-----
    // DMAparams2specify columns:
    // 0: inner radius, R1 [m]
    // 1: outer radius, R2 [m]
    // 2: column length, L [m]
    // 3: equivalent pipe length, DMA inlet/outlet [m]
    // 4: equivalent pipe length, sum of devices/components with DMA sample flow [m]
    // 5: particle density [kg/m3]
    //-----

    // Create global variables from input datamatrix
    variable/g Qae,Qmono,Qsheath,Qex,tsample,Tamb,Pamb,deltaTdet,Qdet
    Qae=datamatrix[0][1]
    Qmono=datamatrix[0][2]
    Qsheath=datamatrix[0][3]
    Qex=datamatrix[0][4]
    tsample=datamatrix[0][7]
    Tamb=datamatrix[0][8]
    Pamb=datamatrix[0][9]
    deltaTdet=datamatrix[0][10]
    Qdet=datamatrix[0][11]

```

```

// Load parameters/conditions into wave "DMAmeasParams"
// calculations call from this wave to use these parameters; all in SI units
makeWvDMAmeasParams()
wave DMAmeasParams
DMAmeasParams[%Qa_lpm]=(Qae+Qmono)/2           // lpm
DMAmeasParams[%Qsh_lpm]=(Qsheath+Qex)/2       // lpm
DMAmeasParams[%betaDMA]=QaDMA()/QshDMA()
DMAmeasParams[%deltaDMA]=(Qae-Qmono)/(Qae+Qmono)
DMAmeasParams[%tsample_s]=tsample            // s
DMAmeasParams[%dTdet_C]=deltaTdet            // degC
DMAmeasParams[%Qdet_lpm]=Qdet                 // lpm
DMAmeasParams[%Tamb_K]=Tamb                   // K
DMAmeasParams[%Pamb_Pa]=Pamb                  // Pa
DMAmeasParams[%particleDensity_kgm3]=DMAparams2specify[%particleDensity_kgm3]

// Load instrument geometry parameters into wave "DMAgeomParams"
// calculations call from this wave to use these parameters; all in SI units
makeWvDMAgeomParams()
wave DMAgeomParams
DMAgeomParams[%R1dma_m]=DMAparams2specify[%R1dma_m]
DMAgeomParams[%R2dma_m]=DMAparams2specify[%R2dma_m]
DMAgeomParams[%Ldma_m]=DMAparams2specify[%Ldma_m]
DMAgeomParams[%f_dma]=fDMA()
DMAgeomParams[%G_dma]=Gdma()
DMAgeomParams[%Leq_DMA_m]=DMAparams2specify[%Leq_DMA_m]
DMAgeomParams[%Leq_other_m]=DMAparams2specify[%Leq_other_m]
end
//=====

```

```

// Extrapolate measured counts using exponential function; see Hagen 1983
function extrapolateCounts_exp(ChanCounts,ChanMobility,ExtrMobility,numFitPts)
  wave ChanCounts,ChanMobility,ExtrMobility
  variable numFitPts

  variable nChanCounts=umpnts(ChanCounts)
  duplicate/o ChanMobility, Zk
  duplicate/o ChanCounts, Yk
  Zk=ChanMobility
  Yk=ChanCounts==0 ? 0 : ln(ChanCounts)
  matrixop/o ZkYk=Zk*Yk
  matrixop/o Zksq=Zk*Zk
  variable gamma_lo,kstart,kend
  kstart=0
  kend=numFitPts-1
  gamma_lo=(numFitPts*sum(ZkYk,kstart,kend)-sum(Zk,kstart,kend)*sum(Yk,kstart,kend))
  gamma_lo*=((sum(Zk,kstart,kend))^2-numFitPts*(sum(Zksq,kstart,kend)))^(-1)
  variable gamma_hi
  kstart=nChanCounts-numFitPts
  kend=nChanCounts-1
  gamma_hi=(numFitPts*sum(ZkYk,kstart,kend)-sum(Zk,kstart,kend)*sum(Yk,kstart,kend))
  gamma_hi*=((sum(Zk,kstart,kend))^2-numFitPts*(sum(Zksq,kstart,kend)))^(-1)

  duplicate/o ChanCounts, ExtrCounts
  extract/o ExtrMobility, ExtrMobility_hi, ExtrMobility/ChanMobility[nChanCounts-1]>1.01
  duplicate/o ExtrMobility_hi, ExtrCounts_hi; ExtrCounts_hi=0
  variable nExtr_hi=umpnts(ExtrCounts_hi)
  if(nExtr_hi>0)
    ExtrCounts_hi=ChanCounts[nChanCounts-1]
    ExtrCounts_hi*=exp(-gamma_hi*(ExtrMobility_hi(p)-ChanMobility[nChanCounts-1]))
    insertpoints nChanCounts,nExtr_hi,ExtrCounts
    ExtrCounts[nChanCounts,(nChanCounts+nExtr_hi-1)]=ExtrCounts_hi(p-nChanCounts)
  endif
  extract/o ExtrMobility, ExtrMobility_lo, ExtrMobility/ChanMobility[0]<0.99
  duplicate/o ExtrMobility_lo, ExtrCounts_lo; ExtrCounts_lo=0
  variable/g nExtr_lo=umpnts(ExtrCounts_lo)
  if(nExtr_lo>0)
    ExtrCounts_lo=ChanCounts[0]*exp(-gamma_lo*(ExtrMobility_lo(p)-ChanMobility[0]))
    insertpoints 0,nExtr_lo,ExtrCounts
    ExtrCounts[0,(nExtr_lo-1)]=ExtrCounts_lo(p)
  endif
end
//=====

```

```

// Evaluate inverse solution(s) over size range of interest
// Input waves and variables:
// SplineCoeffMatrix [ncoeff x nSolns] (i.e., "invsolnCoeffs")
// SplineControlptKnots [nDias x nSolns] (i.e., "invsolnSplineModelpts")
// SplineKnotWave [(ncoeff+d+1) x nSolns] (i.e., "invsolnSplineKnots")
// diam0_m, diamN_m, numdps - size range and number of points to evaluate
// for final solution representation
// Output:
// invsolnFinal [numdps x nSolns] dN/dlnDp [cm-3] evaluated over size range of interest
// dp_final [numdps x nSolns] Dp [m] corresponding to final solution range
function invsoln_fineEval(SplineCoeffMatrix,SplineControlptKnots,SplineKnotWave,diam0_m,diam
wave SplineCoeffMatrix,SplineControlptKnots,SplineKnotWave
variable diam0_m,diamN_m,numdps

variable Indiam0=ln(diam0_m)
variable IndiamN=ln(diamN_m)
variable dlnDpWvSet=(IndiamN-Indiam0)/(numdps-1)
make/o/n=(numdps) finalDistBinning, dpwave_finaldistbin
finalDistBinning=Indiam0+dlnDpWvSet*x
dpwave_finaldistbin=exp(finalDistBinning)

variable i=0
variable nSolns=dimsize(SplineCoeffMatrix,1)
variable ncoeffs=dimsize(SplineCoeffMatrix,0)
variable nptsfinaldist=umpoints(finalDistBinning)
make/o/n=(ncoeffs) SplineCoeffWave_i
make/o/n=(nptsfinaldist) invsoln_final_i
duplicate/o invsoln_final_i, invsolnFinal; redimension/n=(-1,nSolns) invsolnFinal
duplicate/o dpwave_finaldistbin, dp_final; redimension/n=(-1,nSolns) dp_final
dp_final[p][]=dpwave_finaldistbin[p]

for(i=0;i<nSolns;i+=1)
    SplineCoeffWave_i[]=SplineCoeffMatrix[p][i]
    invsoln_final_i=0
    cubicBspline(SplineCoeffWave_i,SplineControlptKnots,SplineKnotWave,finalDistBinning,inv
wave invsoln_final_i
    invsolnFinal[][i]=invsoln_final_i[p]
endfor
end
//=====

```

```

// Calculate kernel matrix
// Input options and waves:
// calcreponsematrixParams - options for kernel matrix setup
// TFformula - string to call form of transfer function
//
// Output is kernel matrix "Kernel_inv" and Bspline knot waves
function calcreponsematrix(calcreponsematrixParams,TFformula)
    wave calcreponsematrixParams
    string TFformula

    // [Read channel settings from DMA setup function]
    wave ZstarChan, DstarChan, VstarChan
    nvar numchan
    nvar ZmaxDMA,ZminDMA
    duplicate/o VstarChan, VchanSet

    duplicate/o ZstarChan, lnZstarChan           // log-linear channel spacing
    lnZstarChan=ln(ZstarChan)
    duplicate/o DstarChan, lnDstarChan
    lnDstarChan=ln(DstarChan)

    // Define mobility grid for kernel integration
    // read input parameters defining kernel setup
    variable numdias2numchan_ratio,Zfactor_lower,Zfactor_upper
    variable nSub4CalcKernel,chargemax4calcKernel,extrapYorN
    numdias2numchan_ratio=calcreponsematrixParams[%ndias2nchanRatio]
    Zfactor_lower=calcreponsematrixParams[%ZfactorLower]
    Zfactor_upper=calcreponsematrixParams[%ZfactorUpper]
    nSub4CalcKernel=calcreponsematrixParams[%nSub4Kernel]
    chargemax4calcKernel=calcreponsematrixParams[%chargemax4Kernel]
    extrapYorN=calcreponsematrixParams[%extrapMeasChan]

    // find upper and lower mobility beyond measurement range
    variable/g numdias=ceil(numchan*numdias2numchan_ratio)
    variable/d/g dlnZstarChan,ZstarL,nZextrL,ZstarU,nZextrU
    dlnZstarChan = abs(lnZstarChan[0]-lnZstarChan[numchan-1])/(numdias-1)
    ZstarL = ZmaxDMA*Zfactor_lower
    nZextrL = Zfactor_lower==1 ? 0 : ceil((ln(ZstarL)-lnZstarChan[0])/dlnZstarChan)
    make/o/n=(nZextrL) lnZstarChanLwv; lnZstarChanLwv=lnZstarChan[0]+dlnZstarChan*(x+1)
    ZstarU = ZminDMA/Zfactor_upper
    nZextrU = Zfactor_upper==1 ? 0 : ceil((lnZstarChan[numchan-1]-ln(ZstarU))/dlnZstarChan)
    make/o/n=(nZextrU) lnZstarChanUwv
    lnZstarChanUwv=lnZstarChan[numchan-1]-dlnZstarChan*(x+1)
    concatenate/np/o {lnZstarChan,lnZstarChanLwv,lnZstarChanUwv}, lnZstarChanExtr
    sort/r lnZstarChanExtr, lnZstarChanExtr

```

```

duplicate/o InZstarChanExtr, ZstarChanExtr
ZstarChanExtr=exp(InZstarChanExtr)

duplicate/o ZstarChanExtr, VstarChanExtr, DstarChanExtr, InDstarChanExtr
VstarChanExtr=VCalc_Zdma(ZstarChanExtr)
DstarChanExtr=DpCalc_Zdma(ZstarChanExtr)
InDstarChanExtr=ln(DstarChanExtr)

variable/g nChan,nDias
nChan = numchan
nDias = numdias+nZextrL+nZextrU

// Set up kernel matrix for inversion:
// define grids for discretizing kernel matrix and evaluating inversion solution
duplicate/o InDstarChanExtr, InDpbin_inv
duplicate/o InDstarChan, InDpbin_invmeas
if (extrapYorN) // extrapolate channel dimension if input "extrapYorN" set to 1
    duplicate/o ZstarChanExtr, ChanZ_inv
    duplicate/o VstarChanExtr, ChanV_inv
    duplicate/o DstarChanExtr, ChanDp_inv
else
    duplicate/o ZstarChan, ChanZ_inv
    duplicate/o VstarChan, ChanV_inv
    duplicate/o DstarChan, ChanDp_inv
endif

variable/g mKern=umpnts(ChanZ_inv), nKern=0
variable nSub=nSub4CalcKernel
nvar bbeta,ddelta

CalcKernel_cubicBspline(ChanZ_inv,ChanV_inv,InDpbin_inv,nSub,bbeta,ddelta,chargemax4c
wave KernelS3
wave BsplineKnotWave, BsplineCoeffKnots_j, BsplineCoeffKnots, BsplineControlptKnots
duplicate/o KernelS3, Kernel_inv

nKern=dimsize(Kernel_inv,1)
duplicate/o BsplineKnotWave, BsplineKnotWave_inv
duplicate/o BsplineCoeffKnots_j, BsplineCoeffKnots_j_inv
duplicate/o BsplineCoeffKnots, BsplineCoeffKnots_inv
duplicate/o BsplineControlptKnots, BsplineControlptKnots_inv
end
//=====

```

```

// Calculate the inversion matrix using cubic B-splines for numerical quadrature of kernel integral
// no assumptions about knot spacing; no added conditions ["natural cubic"]
//
// Input variables and waves [dimensions in brackets]:
// ChanZ [m] vector of mobility values for measurement channels; Z [m2/V/s]
// ChanV [m] vector of voltage values for measurement channels
// Dpbin [nDias] vector of diameter values to define boundaries of size distribution intervals;
// should be input as ln(Dp) for log-linear channel spacing (Dp in meters)
// nSub - number of subdivisions for evaluating integral over size bin intervals
// bbeta, delta - flow parameters
// chargemax - max charges in kernel function
// TFformula - string to call form of transfer function
//
// Output is kernel matrix "KernelS3" and Bspline knot waves
function CalcKernel_cubicBspline(ChanZ,ChanV,Dpbin,nSub,bbeta,delta,chargemax,TFformula)
    wave ChanZ,ChanV,Dpbin
    variable nSub,bbeta,delta,chargemax
    string TFformula

    // Define spline control points and bounds for integrations
    define_binning(Dpbin)
    wave binvals_geom,binvals_plusuplodiff
    duplicate/o binvals_geom, Dpbinbound
    duplicate/o binvals_plusuplodiff, ControlptBinning
    variable/g Dpbinbound_lo=Dpbinbound[0]
    variable/g Dpbinbound_hi=Dpbinbound[(numpts(Dpbinbound)-1)]
    variable/g Dpbin_lo=Dpbin[0]
    variable/g Dpbin_hi=Dpbin[(numpts(Dpbin)-1)]

    // Make the wave of knot values needed to calculate Bsplines
    variable degree=3 // degree of spline interpolation; cubic, d=3
    MakeBsplineKnotWave(ControlptBinning,degree)
    wave BsplineKnotWave, BsplineCoeffKnots_j, BsplineCoeffKnots, BsplineControlptKnots
    duplicate/o BsplineKnotWave, Dpbin_knots

    variable m = numpts(ChanZ)
    variable n = numpts(Dpbin)
    variable ncoeffs = numpts(BsplineCoeffKnots)
    variable dDpInt
    if (nSub>1) // work with even number of subintervals to capture middle Dp of bin
        variable nSub_even = mod(nSub,2)==0 ? nSub : (nSub+1)
        nSub=nSub_even
    endif
    make/o/n= (nSub+1) DpInt // vector of diameter values for integral over size bins
    make/o/n=(m,ncoeffs) KernelS3=0 // initialize kernel matrix

```

```

variable xk,xk1,xk2,xk3,xk4
variable h
variable i,j,k,knotindx
for(i=0; i<m;i+=1)
  for(k=0;k<ncoeffs;k+=1)
    knotindx=k
    xk=Dpbin_knots[knotindx]
    xk1=Dpbin_knots[knotindx+1]
    xk2=Dpbin_knots[knotindx+2]
    xk3=Dpbin_knots[knotindx+3]
    xk4=Dpbin_knots[knotindx+4]
    // Bpline1
    h=(xk1-xk)
    dDpInt=h/nSub
    DpInt=xk+dDpInt*x
    duplicate/d/o DpInt, ChiSub1
    duplicate/d/o ChiSub1, fSub1; fSub1=BsplineC1(ChiSub1,xk,xk1,xk2,xk3,xk4)
    duplicate/d/o ChiSub1, Ksub1
    Ksub1=KernelFcn_DMA(ChiSub1,ChanZ[i],ChanV[i],bbeta,delta,chargemax,TFformula)
    Ksub1*=fsub1
    // Bpline2
    h=(xk2-xk1)
    dDpInt=h/nSub
    DpInt=xk1+dDpInt*x
    duplicate/d/o DpInt, ChiSub2
    duplicate/d/o ChiSub2, fSub2; fSub2=BsplineC2(ChiSub2,xk,xk1,xk2,xk3,xk4)
    duplicate/d/o ChiSub2, Ksub2
    Ksub2=KernelFcn_DMA(ChiSub2,ChanZ[i],ChanV[i],bbeta,delta,chargemax,TFformula)
    Ksub2*=fsub2
    // Bpline3
    h=(xk3-xk2)
    dDpInt=h/nSub
    DpInt=xk2+dDpInt*x
    duplicate/d/o DpInt, ChiSub3
    duplicate/d/o ChiSub3, fSub3; fSub3=BsplineC3(ChiSub3,xk,xk1,xk2,xk3,xk4)
    duplicate/d/o ChiSub3, Ksub3
    Ksub3=KernelFcn_DMA(ChiSub3,ChanZ[i],ChanV[i],bbeta,delta,chargemax,TFformula)
    Ksub3*=fsub3
    // Bpline4
    h=(xk4-xk3)
    dDpInt=h/nSub
    DpInt=xk3+dDpInt*x
    duplicate/d/o DpInt, ChiSub4
    duplicate/d/o ChiSub4, fSub4; fSub4=BsplineC4(ChiSub4,xk,xk1,xk2,xk3,xk4)
    duplicate/d/o ChiSub4, Ksub4
    Ksub4=KernelFcn_DMA(ChiSub4,ChanZ[i],ChanV[i],bbeta,delta,chargemax,TFformula)
    Ksub4*=fsub4

```

```

if (k>3 && k<(ncoeffs-4))
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1)
  KernelS3[i][k]+=AreaXY(ChiSub2,Ksub2)
  KernelS3[i][k]+=AreaXY(ChiSub3,Ksub3)
  KernelS3[i][k]+=AreaXY(ChiSub4,Ksub4)
endif
if (k==0)
  KernelS3[i][k]=AreaXY(ChiSub4,Ksub4,Dpbinbound_lo,Dpbin_lo)
endif
if (k==1)
  KernelS3[i][k]=AreaXY(ChiSub3,Ksub3,Dpbinbound_lo,Dpbin_lo)
  KernelS3[i][k]+=AreaXY(ChiSub4,Ksub4)
endif
if (k==2)
  KernelS3[i][k]=AreaXY(ChiSub2,Ksub2,Dpbinbound_lo,Dpbin_lo)
  KernelS3[i][k]+=AreaXY(ChiSub3,Ksub3)
  KernelS3[i][k]+=AreaXY(ChiSub4,Ksub4)
endif
if (k==3)
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1,Dpbinbound_lo,Dpbin_lo)
  KernelS3[i][k]+=AreaXY(ChiSub2,Ksub2)
  KernelS3[i][k]+=AreaXY(ChiSub3,Ksub3)
  KernelS3[i][k]+=AreaXY(ChiSub4,Ksub4)
endif
if (k==ncoeffs-4)
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1)
  KernelS3[i][k]+=AreaXY(ChiSub2,Ksub2)
  KernelS3[i][k]+=AreaXY(ChiSub3,Ksub3)
  KernelS3[i][k]+=AreaXY(ChiSub4,Ksub4,Dpbin_hi,Dpbinbound_hi)
endif
if (k==ncoeffs-3)
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1)
  KernelS3[i][k]+=AreaXY(ChiSub2,Ksub2)
  KernelS3[i][k]+=AreaXY(ChiSub3,Ksub3,Dpbin_hi,Dpbinbound_hi)
endif
if (k==ncoeffs-2)
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1)
  KernelS3[i][k]+=AreaXY(ChiSub2,Ksub2,Dpbin_hi,Dpbinbound_hi)
endif
if (k==ncoeffs-1)
  KernelS3[i][k]=AreaXY(ChiSub1,Ksub1,Dpbin_hi,Dpbinbound_hi)
endif
endfor
end
//=====

```

```

// B-spline functional forms
function BsplineC1(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable B1=((xi-xk)^3)/((xk3-xk)*(xk2-xk)*(xk1-xk))
    return B1
end

function BsplineC2(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable B2=(xi-xk)/(xk3-xk)*(xi-xk)/(xk2-xk)*(xk2-xi)/(xk2-xk1)
    B2+=(xi-xk)/(xk3-xk)*(xk3-xi)/(xk3-xk1)*(xi-xk1)/(xk2-xk1)
    B2+=(xk4-xi)/(xk4-xk1)*(xi-xk1)/(xk3-xk1)*(xi-xk1)/(xk2-xk1)
    return B2
end

function BsplineC3(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable B3=(xi-xk)/(xk3-xk)*(xk3-xi)/(xk3-xk1)*(xk3-xi)/(xk3-xk2)
    B3+=(xk4-xi)/(xk4-xk1)*(xi-xk1)/(xk3-xk1)*(xk3-xi)/(xk3-xk2)
    B3+=(xk4-xi)/(xk4-xk1)*(xk4-xi)/(xk4-xk2)*(xi-xk2)/(xk3-xk2)
    return B3
end

function BsplineC4(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable B4=((xk4-xi)^3)/((xk4-xk1)*(xk4-xk2)*(xk4-xk3))
    return B4
end
//=====

```

```

// Make the wave of knot values needed to calculate B-splines; specify degree
function MakeBsplineKnotWave(ControlptBin,degree)
    wave ControlptBin
    variable degree

    variable deg=degree
    variable nControlpts=numpts(ControlptBin)
    variable ntotalKnots=nControlpts+2*deg
    variable nBsplCoeffs=nControlpts+deg-1

    duplicate/o ControlptBin, BsplineControlptKnots, BsplineKnotWave
    variable hEven=(ControlptBin[nControlpts-1]-ControlptBin[0])/(nControlpts-1)
    variable nEndKnots=deg // knot points needed at upper and lower bounds of ControlptBin
    variable i,insertindx
    for(i=1;i<=nEndKnots;i+=1) // add knot points to upper bound
        insertindx=(nControlpts-1)+i
        insertpoints (insertindx),1,BsplineKnotWave
        BsplineKnotWave[insertindx]=ControlptBin[nControlpts-1]+hEven*i
    endfor
    for(i=1;i<=nEndKnots;i+=1) // add knot points to lower bound
        insertpoints 0,1,BsplineKnotWave
        BsplineKnotWave[0]=ControlptBin[0]-hEven*i
    endfor

    duplicate/o/r=(0,(nBsplCoeffs-1)) BsplineKnotWave, BsplineCoeffKnots_j
    duplicate/o/r=((deg+1)/2),(nBsplCoeffs-1)+(deg+1)/2) BsplineKnotWave, BsplineCoeffKnots
end
//=====

```

```

// Define spline control points and bounds for integrations
function define_binning(wavetobin)
    wave wavetobin

    variable numwvpts=umpnts(wavetobin)
    duplicate/o wavetobin, binvals_updiff, binvals_lodiff
    make/o/n=(numwvpts+1) binvals_geom
    make/o/n=(numwvpts) deltavals_updiff, deltavals_lodiff, deltavals_geom

    variable/g deltaval_even=(wavetobin[numwvpts-1]-wavetobin[0])/(numwvpts-1)
    insertpoints numwvpts,1,binvals_updiff
    binvals_updiff[numwvpts]=wavetobin[numwvpts-1]+deltaval_even
    insertpoints 0,1,binvals_lodiff
    binvals_lodiff[0]=wavetobin[0]-deltaval_even

    duplicate/o binvals_updiff, binvals_plusuplodiff
    insertpoints 0,1,binvals_plusuplodiff; binvals_plusuplodiff[0]=binvals_lodiff[0]

    binvals_geom[1,(numwvpts-1)]=sqrt(wavetobin(p-1)*wavetobin(p))*sign(wavetobin(p))
    binvals_geom[0]=sqrt(binvals_lodiff[0]*wavetobin[0])*sign(wavetobin[0])
    binvals_geom[numwvpts]=sqrt(wavetobin[numwvpts-1]*binvals_updiff[numwvpts])
    binvals_geom[numwvpts]*=sign(wavetobin[numwvpts-1])

    deltavals_updiff=binvals_updiff(p+1)-binvals_updiff(p)
    deltavals_lodiff=binvals_lodiff(p+1)-binvals_lodiff(p)
    deltavals_geom=binvals_geom(p+1)-binvals_geom(p)

    make/o/n=(2*numwvpts+1) binmidandgeom
    concatenate/np/o {wavetobin,binvals_geom}, binmidandgeom
    sort binmidandgeom, binmidandgeom
end
//=====

```

```

// Solve inverse problem with cubic B-spline setup
// Input options and waves [dimensions in brackets]:
// KernelMatrix      [m x ncoeff]   kernel matrix, describing measurement collection
// ModelDataBinning  [nDias]        binning for size distribution solution, model
// MeasDataBinning   [m_raw]        binning for measured signals (raw data)
// SplineCoeffKnots  [ncoeff=n+d-1] binning for coefficients of size distribution solution
// SplineControlptKnots [n=nDias+2] binning for control points of size distribution solution
// SplineKnotWave     [n+2d]        wave of knot values needed to calculate Bsplines
// precalcBdata      [m]            vector of [extrapolated] measured signals (raw data)
// Buncertainties     [m]            vector of uncertainty/error values associated with
//                                     [extrapolated] measured signals (raw data)
//
// InvSolnMethod - string to call solution computation algorithm
//
// Output:
// dN/dlnDp [cm-3]      size distribution inverse solution(s) in "invsoln_XXX" waves
// Dp [m]               corresponding particle diameters in "dpwave_XXX" waves
function invsoln_Bspline(Kernelmatrix,ModelDataBinning,MeasDataBinning,SplineCoeffKnots,Spli
    wave Kernelmatrix,ModelDataBinning,MeasDataBinning
    wave SplineCoeffKnots,SplineControlptKnots,SplineKnotWave
    wave precalcBdata,Buncertainties
    string InvSolnMethod

    variable ncoeffs=dimsize(Kernelmatrix,1)
    variable nptsControl=numpts(SplineControlptKnots)
    variable nptsModel=numpts(ModelDataBinning)
    variable nptsmeasrange=numpts(MeasDataBinning)
    make/o/n=(ncoeffs) invsoln_coefs
    make/o/n=(nptsControl) invsoln_controlpts
    make/o/n=(nptsModel) invsoln_model
    make/o/n=(nptsmeasrange) invsoln_meas

    cSplineInv(Kernelmatrix,SplineCoeffKnots,SplineControlptKnots,SplineKnotWave,invsoln_cont
    wave invsoln_controlpts
    wave M_x=root:Packages:cSplineInv:M_x
    duplicate/o M_x, invsoln_coefs
    cubicBspline(M_x,SplineControlptKnots,SplineKnotWave,ModelDataBinning,invsoln_model)
    wave invsoln_model
    cubicBspline(M_x,SplineControlptKnots,SplineKnotWave,MeasDataBinning,invsoln_meas)
    wave invsoln_meas

    duplicate/o SplineControlptKnots, dpwave_controlptdistbin
    dpwave_controlptdistbin=exp(SplineControlptKnots)
    duplicate/o ModelDataBinning, dpwave_modeldistbin
    dpwave_modeldistbin=exp(ModelDataBinning)
    duplicate/o MeasDataBinning, dpwave_measdistbin
    dpwave_measdistbin=exp(MeasDataBinning)
end
//=====

```

```

// Solve inverse problem with cubic B-spline setup
// Input options and waves [dimensions in brackets]:
// AmatrixInput      [m x ncoeff]   kernel matrix, describing measurement collection
// SplineCoeffKnots  [ncoeff=n+d-1] binning for coefficients of size distribution solution;
//                                     number of points must match y-dimension of Kmatrix
// SplineControlptKnots [n]         binning for control points of size distribution solution
// SplineKnotWave     [n+2d]        wave of knot values needed to calculate Bsplines
// ModelWaveOutput    [any]         wave to hold size distribution solution; any number/spacing
// ModelDataBinning   [any]         binning for size distribution solution;
//                                     number of points must match ModelWaveOutput
// Bvector            [m]           vector of measured signals (raw data)
// Bvector_uncertainties [m]        vector of uncertainty/error values associated with
//                                     measured signals (raw data)
// SolnMethod - string to call solution computation algorithm
//
// Output is solution in "ModelWaveOutput"
function cSplineInv(AmatrixInput,SplineCoeffKnots,SplineControlptKnots,SplineKnotWave,ModelM
    wave AmatrixInput,SplineCoeffKnots,SplineControlptKnots,SplineKnotWave
    wave ModelWaveOutput,ModelDataBinning,BvectorInput,Bvector_uncertainties
    string SolnMethod

// create working space
string OldDf=GetDataFolder(1)
NewDataFolder/O/S root:Packages
NewDataFolder/O/S root:Packages:cSplineInv
// create local copies of input waves
duplicate/o AmatrixInput, Kmatrix
duplicate/o SplineCoeffKnots, SplineCoeffBin
duplicate/o SplineControlptKnots, SplineControlptBin
duplicate/o SplineKnotWave, SplineKnotBin
duplicate/o ModelDataBinning, ModelDataBin
duplicate/o BvectorInput, Bvector
duplicate/o Bvector_uncertainties, Uncertainties
redimension/d Kmatrix,SplineCoeffKnots,SplineControlptKnots,SplineKnotWave
redimension/d ModelDataBin,Bvector,Uncertainties

```

```
variable n = dimsize(Kmatrix,1)
make/o/n=(n) M_x; M_x=0

strswitch(SolnMethod)
  case "NNLS":
    NNLS(Kmatrix,Bvector,M_x)
    wave M_x
    break
  case "TNNLS":
    IPG_TNNLS(Kmatrix,M_x,Bvector,Uncertainties,0.6,100)
    wave M_x
    break
  case "reg":
    Regularization(Kmatrix,M_x,Bvector,Uncertainties)
    wave M_x
    break
endswitch

cubicBspline(M_x,SplineControlptBin,SplineKnotBin,ModelDataBin,ModelWaveOutput)
wave ModelWaveOutput
SetDataFolder OldDf
end
//=====
```

```

// Evaluate cubic spline interpolation, given coefficient values from inverse solution
// Input waves [dimensions in brackets]:
// SplineCoeffWave [ncoeff] inverse solution coefficients
// SplineControlptKnots [nDias] binning for control points of size distribution solution
// SplineKnotWave [(ncoeff+d+1)] wave of knot values needed to calculate Bsplines
// SplineOutputBin [any] binning for final size distribution solution; can be very fine
// cSplineOutputWv [any] wave to hold size distribution solution;
// number of points must match SplineOutputBin
// Output is solution in "cSplineOutputWv"
function cubicBspline(SplineCoeffWave,SplineControlptKnots,SplineKnotWave,SplineOutputBin,cSplineOutputWv)
    wave SplineCoeffWave,SplineControlptKnots,SplineKnotWave
    wave SplineOutputBin,cSplineOutputWv

    FindKnotPt(SplineKnotWave,SplineControlptKnots,SplineOutputBin)
    wave KnotPt, KnotPt_indx

    variable fpnts=umpnts(SplineOutputBin)
    variable controlpnts=umpnts(SplineControlptKnots)
    variable x0=SplineControlptKnots[0], xN=SplineControlptKnots[controlpnts-1]

    variable xi, xk,xk1,xk2,xk3,xk4
    variable xk_3,xk_2, xk_1
    variable cj_3,cj_2,cj_1,cj
    variable Bj_3,Bj_2,Bj_1,Bj
    variable dBj_3,dBj_2,dBj_1,dBj
    variable x0N,Cspline0N,dCspline0N
    variable Cspline_i
    variable knotindx
    variable i
    for(i=0;i<fpnts;i+=1)
        xi=SplineOutputBin[i]
        knotindx=KnotPt_indx[i]

        cj_3=SplineCoeffWave[knotindx-3]
        cj_2=SplineCoeffWave[knotindx-2]
        cj_1=SplineCoeffWave[knotindx-1]
        cj=SplineCoeffWave[knotindx]

        xk_3=SplineKnotWave[knotindx-3]
        xk_2=SplineKnotWave[knotindx-2]
        xk_1=SplineKnotWave[knotindx-1]
        xk=SplineKnotWave[knotindx]
        xk1=SplineKnotWave[knotindx+1]
        xk2=SplineKnotWave[knotindx+2]
        xk3=SplineKnotWave[knotindx+3]
        xk4=SplineKnotWave[knotindx+4]

```

```

if (xi<x0 || xi>=xN)
  if (xi<x0)
    x0N=x0
  elseif (xi>=xN)
    x0N=xN
  endif

  Bj_3=BsplineC4(x0N,xk_3,xk_2,xk_1,xk,xk1)
  Bj_2=BsplineC3(x0N,xk_2,xk_1,xk,xk1,xk2)
  Bj_1=BsplineC2(x0N,xk_1,xk,xk1,xk2,xk3)
  Bj=BsplineC1(x0N,xk,xk1,xk2,xk3,xk4)
  Cspline0N=cj_3*Bj_3+cj_2*Bj_2+cj_1*Bj_1+cj*Bj

  dBj_3=dBsplineC4(x0N,xk_3,xk_2,xk_1,xk,xk1)
  dBj_2=dBsplineC3(x0N,xk_2,xk_1,xk,xk1,xk2)
  dBj_1=dBsplineC2(x0N,xk_1,xk,xk1,xk2,xk3)
  dBj=dBsplineC1(x0N,xk,xk1,xk2,xk3,xk4)
  dCspline0N=cj_3*dBj_3+cj_2*dBj_2+cj_1*dBj_1+cj*dBj

  Cspline_i=max((Cspline0N+dCspline0N*(xi-x0N)),0)
else
  Bj_3=BsplineC4(xi,xk_3,xk_2,xk_1,xk,xk1)
  Bj_2=BsplineC3(xi,xk_2,xk_1,xk,xk1,xk2)
  Bj_1=BsplineC2(xi,xk_1,xk,xk1,xk2,xk3)
  Bj=BsplineC1(xi,xk,xk1,xk2,xk3,xk4)
  Cspline_i=cj_3*Bj_3+cj_2*Bj_2+cj_1*Bj_1+cj*Bj
endif

cSplineOutputWv[i]=max(Cspline_i,0)
Cspline_i=0
endfor
end
//=====

```

```

// Find primary knot point to correspond to secondary (fine) knot point
function FindKnotPt(KnotPtWave,ControlPtWave,InterpFinePtWave)
    wave KnotPtWave,ControlPtWave,InterpFinePtWave

    duplicate/o InterpFinePtWave, KnotPt, KnotPt_indx
    variable p=numpts(InterpFinePtWave)
    variable k=numpts(KnotPtWave)

    variable c0knot=ControlPtWave[0]
    variable cNknot=ControlPtWave[numpts(ControlPtWave)-1]
    findvalue/v=(c0knot) KnotPtWave
    variable c0knotindx=V_value
    findvalue/v=(cNknot) KnotPtWave
    variable cNknotindx=V_value

    variable j,i
    for(i=c0knotindx;i<cNknotindx;i+=1)
        for(j=0;j<p;j+=1)
            if (InterpFinePtWave[j]<KnotPtWave[c0knotindx])
                KnotPt[j]=KnotPtWave[c0knotindx]
                KnotPt_indx[j]=c0knotindx
            elseif (InterpFinePtWave[j]>=KnotPtWave[cNknotindx-1])
                KnotPt[j]=KnotPtWave[cNknotindx-1]
                KnotPt_indx[j]=cNknotindx-1
            elseif (InterpFinePtWave[j]>=KnotPtWave[i] && InterpFinePtWave[j]<KnotPtWave[i+1])
                KnotPt[j]=KnotPtWave[i]
                KnotPt_indx[j]=i
            endif
        endfor
    endfor
end
//=====

```

```

// B-spline functional forms; first derivatives
function dBsplineC1(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable dB1=(3*(xi-xk)^2)/((xk3-xk)*(xk2-xk)*(xk1-xk))
    return dB1
end

function dBsplineC2(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable dB2=(-xk^2-2*xk*xk2+4*xk*xi+2*xk2*xi-3*xi^2)/((xk3-xk)*(xk2-xk)*(xk2-xk1))
    dB2+=(-xk*xk1-(xk+xk1)*xk3+2*(xk+xk1)*xi+2*xk3*xi-3*xi^2)/((xk3-xk)*(xk3-xk1)*(xk2-xk1))
    dB2+=(-xk1^2-2*xk1*xk4+4*xk1*xi+2*xk4*xi-3*xi^2)/((xk4-xk1)*(xk3-xk1)*(xk2-xk1))
    return dB2
end

function dBsplineC3(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable dB3=(xk3^2+2*xk3*xk-4*xk3*xi-2*xk*xi+3*xi^2)/((xk3-xk)*(xk3-xk1)*(xk3-xk2))
    dB3+=(xk3*xk4+(xk3+xk4)*xk1-2*(xk3+xk4)*xi-2*xk1*xi+3*xi^2)/((xk4-xk1)*(xk3-xk1)*(xk3-xk2))
    dB3+=(xk4^2+2*xk2*xk4-4*xk4*xi-2*xk2*xi+3*xi^2)/((xk4-xk1)*(xk4-xk2)*(xk3-xk2))
    return dB3
end

function dBsplineC4(xi,xk,xk1,xk2,xk3,xk4)
    variable xi,xk,xk1,xk2,xk3,xk4

    variable dB4=(3*(xk4-xi)^2)/((xk4-xk1)*(xk4-xk2)*(xk4-xk3))
    return dB4
end
//=====

```

```

// Evaluate transmission probability for particle through DMA
function KernelFcn_DMA(Dpart,ChanZ,ChanV,bbeta,delta,chargemax,TFformula)
    variable Dpart,ChanZ,ChanV,bbeta,delta,chargemax
    string TFformula
    // Dpart is some measure of the particle diameter; need to make sure it is expressed as Dp [m]
    // before passing to other functions (in case input is ln(Dp))
    // ChanZ is the channel-targeted mobility of the instrument; Z [m2/V/s]
    variable Dp
    if(Dpart<0)
        Dp=exp(Dpart)
    else
        Dp=Dpart
    endif

    variable Zp=ZpCalc_DpDMA(Dp)
    variable Zp_i=Zp
    variable phi=0 // charging probability
    variable psi=0 // transfer function
    variable Ghipsi=0 // overall kernel value
    variable i=0
    variable echarges=0
    for (i=1;i<=chargemax;i+=1) // set max number of charges to consider
        Zp_i=abs(Zp*i)
        echarges=-i
        phi=chargeprob(Dp,echarges)
        psi=TransferFcn_DMA(Zp_i,i,ChanZ,ChanV,bbeta,delta,TFformula)
        Ghipsi+=phi*psi
    endfor

    wave diffLossMatrix
    variable Leff,Q,eta; eta=1
    for (i=0;i<dimsz(diffLossMatrix,0);i+=1)
        Leff=diffLossMatrix[i][0]
        Q=diffLossMatrix[i][1]/60000
        eta*=eta_pen(Dp,Leff,Q)
    endfor
    eta*=eta_pen(Dp,Leff_total(),QaDMA())
    eta*=counteff_CPC3025(Dp)
    variable F=1*eta // factor for penetration and other efficiencies

    variable kernel=Ghipsi*F // *efficiencies
    if (kernel<1e-9) // limit for kernel matrix; lessens burden on inversion algorithm
        kernel=0
    endif

    return kernel
end
//=====

```

```

// Calculate value of transfer function for particle mobility Zp
// DMA set to voltage ChanV, with associated mobility ChanZ
function TransferFcn_DMA(Zp,ncharges,ChanZ,ChanV,bbeta,delta,TFformula)
    variable Zp,ncharges,ChanZ,ChanV,bbeta,delta
    string TFformula

    variable Ztil,Pe,sigmaStolz
    Ztil=Zp/ChanZ // dimensionless mobility
    Pe=ncharges*echg()*ChanV*fDMA()/(kB()*TK()) // Peclet number for particle migration...
    sigmaStolz=sqrt((Gdma()*Ztil)/Pe) // diffusion parameter //...across electrode gap

    variable omega=omegaDMAcalc(sigmaStolz,Ztil,bbeta,delta,TFformula)
    if (omega<1e-5) // limit for computational efficiency
        omega = 0
    endif

    return omega
end

// Call one of the "omegaDMA_xxx" functions to calculate omega value
function omegaDMAcalc(sigmap,zzz,bbeta,delta,TFformula)
    variable sigmap,zzz,bbeta,delta
    string TFformula

    variable om
    strswitch(TFformula)
        case "stolz":
            om=omegaDMA_stolz(sigmap,zzz,bbeta,delta)
            break
        case "nd":
            om=omegaDMA_nd(zzz,bbeta,delta)
            break
        case "diff":
            om=omegaDMA_diff(sigmap,zzz,bbeta,delta)
            break
    endswitch

    return om
end
//=====

```

```

// DMA diffusive transfer function developed by Stolzenburg
// (1988 thesis, Eq 2.69; see also Stolzenburg and McMurry 2008, AST, Eq [13])
function omegaDMA_stolz(sigmoid,zzz,bbeta,delta)
    variable sigmoid,zzz,bbeta,delta

    variable e1,e2,e3,e4,dummy,om
    dummy=(zzz-(1-bbeta))/(2^0.5*sigmoid)
    e1=dummy*erf(dummy)+exp(-dummy^2)/pi^0.5
    dummy=(zzz-(1+bbeta))/(2^0.5*sigmoid)
    e2=dummy*erf(dummy)+exp(-dummy^2)/pi^0.5
    dummy=(zzz-(1-delta*bbeta))/(2^0.5*sigmoid)
    e3=dummy*erf(dummy)+exp(-dummy^2)/pi^0.5
    dummy=(zzz-(1+delta*bbeta))/(2^0.5*sigmoid)
    e4=dummy*erf(dummy)+exp(-dummy^2)/pi^0.5
    om=sigmoid/(2^0.5*bbeta*(1-delta))*(e1+e2-e3-e4)

    return om
end

// Non-diffusive transfer function; Stolzenburg and McMurry 2008
function omegaDMA_nd(zzz,bbeta,delta)
    variable zzz,bbeta,delta

    variable om=(abs(zzz-(1+bbeta))+abs(zzz-(1-bbeta)))
    om+=(-abs(zzz-(1+bbeta*delta))-abs(zzz-(1-bbeta*delta)))
    om*=1/(2*bbeta*(1-delta))

    return om
end

// Diffusive transfer function; Gaussian approximation
// (with mean zzz=1 and standard deviation = sigmoid)
function omegaDMA_diff(sigmoid,zzz,bbeta,delta)
    variable sigmoid,zzz,bbeta,delta

    variable om=bbeta*(1+delta)/sigmoid/sqrt(2*pi)*exp(-((zzz-1)^2)/(2*sigmoid^2))

    return om
end
//=====

```

```

// Probability of particle diameter dp [m] carrying n charges
// using Xcode fit coefficients for ss charge distribution for air ions and conductive particles at STP
function chargeprob(dp_m,echarges)
    variable dp_m, echarges

    wave XcodeS3
    variable ap = dp_m/2 // radius
    variable b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,apmin,apmax,phi_charge

    matrixop/o echargesWv=col(XcodeS3,0)
    extract/indx/o echargesWv, echargesIndx, echargesWv==echarges
    variable i=echargesIndx[0]
    b0=XcodeS3[i][1]
    b1=XcodeS3[i][2]
    b2=XcodeS3[i][3]
    b3=XcodeS3[i][4]
    b4=XcodeS3[i][5]
    b5=XcodeS3[i][6]
    b6=XcodeS3[i][7]
    b7=XcodeS3[i][8]
    b8=XcodeS3[i][9]
    b9=XcodeS3[i][10]
    b10=XcodeS3[i][11]
    b11=XcodeS3[i][12]
    apmin=XcodeS3[i][15]
    apmax=XcodeS3[i][16]

    variable sumFitCoeffs=0
    if (ap<apmin || ap>apmax)
        phi_charge=0
    else
        sumFitCoeffs=(b0*(log(ap))^0+b1*(log(ap))^1+b2*(log(ap))^2+b3*(log(ap))^3)
        sumFitCoeffs+=(b4*(log(ap))^4+b5*(log(ap))^5+b6*(log(ap))^6+b7*(log(ap))^7)
        sumFitCoeffs+=(b8*(log(ap))^8+b9*(log(ap))^9+b10*(log(ap))^10+b11*(log(ap))^11)
        phi_charge=10^(sumFitCoeffs)
    endif

    return max(phi_charge,0)
end
//=====

```

// Diffusional losses through DMA inlet/outlet; inside pipes/tubes, bipolar charger, dryer
 // use equivalent pipe length in diffusional deposition formula for laminar flow (Baron and Willeke)

```
function eta_pen(dp_m,Leff,Q)
    variable dp_m,Leff,Q           // [m], [m], [m3/s]

    variable eta_pen,mu
    mu=pi*diffusion_coeff(dp_m)*Leff/Q
    mu=numtype(mu)==0 ? mu : 0
    if (mu>0.02)
        eta_pen=0.81905*exp(-3.6568*mu)+0.09753*exp(-22.305*mu)
        eta_pen+=0.0325*exp(-56.961*mu)+0.01544*exp(-107.62*mu)
    else
        eta_pen=1.0-2.5638*mu^(2/3)+1.2*mu+0.1767*mu^(4/3)
    endif

    return max(0,eta_pen)
end
function diffusion_coeff(dp)
    variable dp

    variable D           // diffusion coefficient
    if (dp > 0.3e-9) // mobility diameter of a particle cannot be below 0.3 nm;
        // ...mass diameter would then be negative
        // based on mobility from Stokes-Millikan equation:
        variable Z = mobility(dp,1) // mobility [m2/V/s]; assumed singly-charged
        D = Z*kB()*TK()/echg()
    else
        // free molecular regime: // Li 2003, and Flagan note on nRDMA data analysis
        // D=kT/f: f=(2/3) rho dp^2 sqrt(2 pi kB TK / m) ( 1 + pi alpha / 8)
        variable alpha = 0.9
        variable f = (2/3)*rhogas()*dp^2*sqrt(2*pi*Rgas()*TK()/MW()*(1+pi*alpha/8))
        D = kB()*TK()/f
    endif

    return D
end

// Mobility calculation, according to Stokes-Millikan equation (Ehn et al. 2011, AST)
function mobility(dp,q)
    variable dp,q           // mobility diameter [m], number of charges in particle
    variable Kn = 2*lambda()/dp // Knudsen number
    variable mass = pi/6*(dp-0.3e-9)^3*rhop() // particle mass [kg]
    variable m_g = MW()*1.660538921e-27*1e3 // mass of gas molecule [kg]
    variable mobil = q*echg()/(3*pi*etagas()*(1+Kn*(1.257+0.4*exp(-1.1/Kn)))/dp
    mobil *= (1+m_g/mass)^(-1/2) // mass correction term
    return mobil           // particle mobility [m2/V/s]
end
//=====
```

```

// Conversions between mobility and diameter
function ZpCalc_DpDMA(dp)
    variable dp

    // Stokes regime:
    // D=kTB: B=Cc / 3 pi eta dp
    // Zp = n e Cc / (3 pi eta dp)
    variable gam1 = 2.492/2
    variable gam2 = 0.84/2
    variable gam3 = 0.43*2
    variable Kn = 2*lambda()/dp
    variable Cc = 1+ Kn*(gam1+gam2*exp(-gam3/Kn))
    variable Zp = echg()*Cc/(3*pi*etagas()*dp)

    return Zp // mobility [m2/V/s]; assumed singly-charged
end
function DpCalc_Zdma(Zp)
    variable Zp

    // Stokes regime:
    // D=kTB: B=Cc / 3 pi eta dp
    // Zp = n e Cc / (3 pi eta dp)
    variable gam1 = 2.492/2
    variable gam2 = 0.84/2
    variable gam3 = 0.43*2
    variable Kn, Cc, Dslip, comparison
    variable PAR = echg()/(3*pi*etagas()*Zp)
    variable Dinitial = (PAR+sqrt(PAR*(PAR+(8*lambda()*gam1))))/2
    do
        Kn = 2*lambda()/Dinitial
        Cc = 1+ Kn*(gam1+gam2*exp(-gam3/Kn))
        Dslip = Cc*PAR
        comparison = abs((Dinitial-Dslip)/Dslip)
        if (comparison>1e-14)
            Dinitial = (Dinitial+Dslip)/2
        endif
    while(comparison>1e-14)
    variable Dp = Dinitial

    return Dp
end
//=====

```

```

// Interconversions between diameter, mobility, and voltage
function DpCalc_Vdma(V)
    variable V

    variable Zp = ZpCalc_Vdma(V)

    return DpCalc_Zdma(Zp)
end
function VCalc_DpDMA(dp)
    variable dp

    variable Zp = ZpCalc_DpDMA(dp)

    return VCalc_Zdma(Zp)
end
function VCalc_Zdma(Zp)
    variable Zp

    return QshDMA()*ln(R2dma()/R1dma())/(2*pi*Ldma()*Zp)
end
function ZpCalc_Vdma(V)
    variable V

    return QshDMA()*ln(R2dma()/R1dma())/(2*pi*V*Ldma())
end
//=====

// CPC counting efficiencies; from Wiedensohler 1997 exponential fit of counting efficiency curves
function counteff_CPC3025(dp_m)
    variable dp_m

    variable Dp_nm=dp_m*1e9
    variable Dp0_nm=2.22 // 0% activation efficiency
    variable Dp50_nm=2.42 // 50% activation efficiency
    variable Dp2_nm=0.28 // slope

    variable cpceff=1
    if (Dp_nm<=Dp0_nm)
        cpceff=0
    else
        cpceff=1-exp((Dp0_nm-Dp_nm)/(Dp2_nm))
    endif

    return cpceff
end
//=====

```

```

// Functions to call parameters and constants
//
// Flow rates
function QaDMA()
    wave DMAmeasParams
    return DMAmeasParams[%Qa_lpm]/60000    // m^3/s
end
function QshDMA()
    wave DMAmeasParams
    return DMAmeasParams[%Qsh_lpm]/60000    // m^3/s
end
function betDMA()
    variable bet=QaDMA()/QshDMA()    // flow ratio
    wave DMAmeasParams; DMAmeasParams[%betaDMA]=bet
    return bet
end
function deltaDMA()
    wave DMAmeasParams
    return DMAmeasParams[%deltaDMA]    // flow balance
end
//
// Instrument geometry
function R1dma()
    wave DMAgeomParams
    return DMAgeomParams[%R1dma_m]    // m
end
function R2dma()
    wave DMAgeomParams
    return DMAgeomParams[%R2dma_m]    // m
end
function Ldma()
    wave DMAgeomParams
    return DMAgeomParams[%Ldma_m]    // m
end
//
// Geometry parameters
function fDMA()
    variable r1=R1dma(), r2=R2dma()
    variable f=(r2-r1)/(r2*ln(r2/r1))
    wave DMAgeomParams; DMAgeomParams[%f_dma]=f
    return f
end
function Gdma()
    variable r1=R1dma(), r2=R2dma(), L=Ldma(), bet=betDMA(), delt=deltaDMA()
    variable G=Gc(r1,r2,L,bet,delt)
    wave DMAgeomParams; DMAgeomParams[%G_dma]=G
    return G
end

```

```

// Equivalent length values for calculating diffusional losses
function LeffDMA()
    wave DMAgeomParams
    return DMAgeomParams[%Leq_DMA_m]           // m
end
function Leff_other()
    wave DMAgeomParams
    return DMAgeomParams[%Leq_other_m]       // m
end
function Leff_tot()
    variable Leff_tot=LeffDMA()+Leff_other()
    return Leff_tot                           // m
end
//
// Sampling parameters
function tmeasDMA()
    wave DMAmeasParams
    return DMAmeasParams[%tsample_s]         // s
end
function dTdetDMA()
    wave DMAmeasParams
    return DMAmeasParams[%dTdet_C]          // degC
end
function QdetDMA()
    wave DMAmeasParams
    return DMAmeasParams[%Qdet_lpm]/60000    // m^3/s
end
//
// Ambient temperature and pressure conditions
function TK()
    wave DMAmeasParams
    return DMAmeasParams[%Tamb_K]           // K
end
function PPa()
    wave DMAmeasParams
    return DMAmeasParams[%Pamb_Pa]         // Pa
end

```

```

// Particle density
function rhop()
    wave DMAmeasParams
    return DMAmeasParams[%particleDensity_kgm3] // kg/m^3
end
//
// Air properties and other constants
function MW() // molecular weight of air
    return 0.028 // kg/mol
end
function lambda() // air mean free path; Seinfeld & Pandis p.399
    return 2*etagas()/(PPa()*(8*MW()*1.660538921e-27*1e3/(pi*TK()*kB()))^0.5) // m
end
function etagas()
    variable visc = 18.27e-6 // viscosity of air in reference temperature, t0v, [kg/m/s]
    variable suthc = 120 // Sutherland constant, value 120 for air (111 for N2 & 127 for O2)
    variable t0v = 291.5 // reference temperature for viscosity [K]
    visc = visc*(t0v+suthc)/(TK()+suthc)*(TK()/t0v)^1.5
    return visc // viscosity of air [kg/m/s]
end
function rhogas()
    return PPa()/(TK()*287.058) // density of air [kg/m3]
end
function nugas()
    return etagas()/rhogas() // kinematic viscosity of air [m2/s]
end
function Rgas()
    return 8.3144621 // gas constant [J/mol/K]
end
function kB()
    return 1.3806488e-23 // Boltzmann constant [J/K]
end
function echg()
    return 1.602176565e-19 // elementary charge [C]
end
function ep0()
    return 8.854e-12 // dielectric constant [C^2/(N m^2)]
end
//=====

```

```

// Functions to find G parameter; assuming laminar flow
function Gc(r1,r2,L,beta,delta)
    variable r1,r2,L,beta,delta

    variable gamma=(r1/r2)^2
    variable kappa=(L*r2)/(r2^2-r1^2)
    variable fcin=(beta*(1-delta))/(2*(1+beta))
    variable fcout=(2+beta-beta*delta)/(2*(1+beta))
    variable win=wfracc(fcin,gamma) // find w values corresponding to...
    variable wout=wfracc(fcout,gamma) // ...characteristic streamline at inlet & outlet

    variable Gc=4*(1+beta)^2*(1-gamma^0.5)/((1-gamma)^2)
    Gc*=lc(wout,gamma)-lc(win,gamma)+(win-wout)/kappa^2
    return Gc
end
function fracc(coefs,w)
    wave coefs
    variable w

    variable FCfd=(w*ln(w)+(1-w)+(1/2)*(1-w)^2*ln(coefs[1])/(1-coefs[1]))
    FCfd*=((1/2)*(1+coefs[1])*ln(coefs[1])+(1-coefs[1])^(-1))
    return coefs[0]-FCfd
end
function wfracc(fc,gam)
    variable fc,gam

    make/d/o coefs={fc,gam}
    FindRoots/L=(gam)/H=1/q fracc, coefs
    return V_Root
end
function lc(w,gam)
    variable w,gam

    variable A=(-(1/2)*(1+gam)*ln(gam)-(1-gam))^(-1)
    variable ICfd=(-(1/2)*w^2*((1-gam)*ln(w)-(1-w)*ln(gam))^2)
    ICfd+=((1/2)*w^2*(1-gam)+(1/3)*w^3*ln(gam))*((1-gam)*ln(w)-(1-w)*ln(gam))
    ICfd+=(1/4)*(1-w^2)*(1-gam)^2+(5/18)*(1-w^3)*(1-gam)*ln(gam)+(1/12)*(1-w^4)*(ln(gam))^2
    ICfd*=(A)^2
    return ICfd
end
//=====

```

```

// Non-negative least squares method to solve problems described as:
// B = A x Model, where B is measured data, A is m x n matrix, and Model is solution in question
// input: matrixA (mxn), vectorb (mx1), as well as wave to hold solution vectorx (nx1)
// output: vectorx>0 such that vectorx = arg min ||Ax - b||^2
// index sets P & R defined and modified in execution of algorithm
// initial solution vector x set to (nx1) zero vector
// n-vectors w & s provide working space
//
// Input waves [dimensions in brackets]:
// AmatrixInput      [m x n]  kernel matrix, describing measurement collection
// BvectorInput      [m]      vector of measured signals (raw data)
// ModelWaveOutput [n]      wave to hold size distribution solution;
//                      number of points must match y-dimension of AmatrixInput
// Output is solution in "ModelWaveOutput"
function NNLS(AmatrixInput,BvectorInput,ModelWaveOutput)
    wave AmatrixInput,BvectorInput,ModelWaveOutput

    // create working space
    string OldDf=GetDataFolder(1)
    NewDataFolder/O/S root:Packages
    NewDataFolder/O/S root:Packages:NNLS
    // create local copies of input waves
    duplicate/o AmatrixInput, matrixA
    duplicate/o BvectorInput, vectorb
    redimension/d matrixA, vectorb

    variable/g m,n
    m = dimsize(matrixA,0)
    n = dimsize(matrixA,1)

    // Initialization: P = null, R = {1, 2,..., n}, x = 0, w = A^T (b - Ax)
    make/o/n=(n) setP; setP = 0
    make/o/n=(n) setR; setR = p+1
    make/d/o/n=(n) vectorx; vectorx = 0
    matrixop/o vectorw = matrixA^t x (vectorb - matrixA x vectorx)
    duplicate/o vectorx, ModelWaveOutputIter

    variable/g tolerance = 1e-10
    variable/g i_outer = 0 // initialize count of iteration number
    do
        duplicate/o vectorw, vectorwR
        variable R_index
        for(R_index=0;(R_index+1)<=n;R_index+=1)
            if (setR[R_index]==0)
                vectorwR[R_index] = -inf
            endif
        endfor
    enddo

```

```

matrixop/o wRvtols = sum(greater(tolerance,vectorwR)-1)
matrixop/o anyRs = sum(equal(setR,0)-1)
variable wRvtol = wRvtols[0]
variable anyR = anyRs[0]
if (wRvtol==0 || anyR==0) // proceed if R not empty and [max(wi) > tolerance]
    break
endif
i_outer+=1
if (i_outer>=(3*n))
    break
endif

make/d/o/n=(n) vector_s; vector_s = 0
wavestats/q vectorwR
variable t_index = V_maxloc // t = arg max(wi)
setP[t_index] = setR[t_index] // include the index t in P and remove it from R
setR[t_index] = 0

// Make matrixAp associated with only the variables currently in the passive set P
make/d/o/n=(m,n) matrixAp; matrixAp=0
variable jj = 0
variable j
for(j=0;(j+1)<=n;j+=1)
    if (setP[j]>0)
        matrixAp[[jj]] = matrixA[p][j]
        jj+=1
    endif
endfor
redimension/n=(m,jj) matrixAp
matrixSVD/b matrixAp
wave M_U,W_W,M_V
duplicate/o W_W, W_Winv; W_Winv = 1/W_W
W_Winv = numtype(W_Winv)==1 ? 0 : W_Winv
matrixop/o W_WinvMatrix = DiagRC(W_Winv,jj,jj)
matrixop/o M_x = M_V x W_WinvMatrix x M_U^t x vectorb
make/d/o/n=(jj) vector_sP; vector_sP = M_x[p]
make/d/o/n=(jj) vectorxP
variable kk = 0
variable k
for(k=0;(k+1)<=n;k+=1)
    if (setP[k]==0)
        vector_s[k] = 0
    else
        vector_s[k] = vector_sP[kk]
        vectorxP[kk] = vectorx[k]
        kk+=1
    endif
endfor

```

```

variable/g i_inner = 0
do // sP = [(Ap)^T x Ap]^-1 x (Ap)^T x b
  matrixop/o sPvtols = sum(greater(vector_sP,tolerance)-1)
  variable sPvtol = sPvtols[0]
  if (sPvtol==0) // proceed if min(sP) <= 0
    break
  endif

  i_inner+=1
  if (i_inner>=3*n)
    break
  endif

  make/d/o/n=(jj) alphawave
  variable sPpnts = numpnts(vector_sP)
  variable qq = 0
  variable q
  for(q=0;(q+1)<=sPpnts;q+=1)
    if (vector_sP[q]<=tolerance)
      alphawave[qq] = vectorxP(q)/(vectorxP(q)-vector_sP(q))
      qq+=1
    endif
  endfor
  variable qpnts = qq
  redimension/n=(qpnts) alphawave
  wavestats/q alphawave
  variable/g alpha = V_min // alpha = -min[xi/(xi - si)]
  matrixop/o vectorx = vectorx+alpha*(vector_s-vectorx) // x := x + alpha(s - x)

  // Update R and P: move from setP to setR all indices j in P for which xj = 0
  variable j_index
  for(j_index=0;(j_index+1)<=n;j_index+=1)
    if (setP[j_index]>0 && abs(vectorx[j_index])<tolerance)
      setR[j_index] = setP[j_index]
      setP[j_index] = 0
    endif
  endfor

  // sP = [(Ap)^T x Ap]^-1 x (Ap)^T x b
  // sR = 0
  make/d/o/n=(n) vector_s; vector_s = 0

```

```

// Make matrixAp associated with only the variables currently in the passive set P
make/d/o/n=(m,n) matrixAp; matrixAp=0
variable jj_inner = 0
Variable j_inner
for(j_inner=0;(j_inner+1)<=n;j_inner+=1)
  if (setP[j_inner]>0)
    matrixAp[][j_inner] = matrixA[p][j_inner]
    jj_inner+=1
  endif
endfor
redimension/n=(m,jj_inner) matrixAp
matrixSVD/b matrixAp
wave M_U,W_W,M_V
duplicate/o W_W, W_Winv; W_Winv = 1/W_W
W_Winv = numtype(W_Winv)==1 ? 0 : W_Winv
matrixop/o W_WinvMatrix = DiagRC(W_Winv,jj_inner,jj_inner)
matrixop/o M_x = M_V x W_WinvMatrix x M_U^t x vectorb
make/d/o/n=(jj_inner) vector_sP; vector_sP = M_x[p]
make/d/o/n=(jj_inner) vectorxP
variable kk_inner = 0
variable k_inner
for(k_inner=0;(k_inner+1)<=n;k_inner+=1)
  if (setP[k_inner]==0)
    vector_s[k_inner] = 0
  else
    vector_s[k_inner] = vector_sP[kk_inner]
    vectorxP[kk_inner] = vectorx[k_inner]
    kk_inner+=1
  endif
endfor
while (1)
vectorx = vector_s(p) // x = s
matrixop/o vectorw = matrixA^t x (vectorb - matrixA x vectorx) // w = A^T x (b - Ax)

redimension/n=(-1,(1+i_outer)) ModelWaveOutputIter
ModelWaveOutputIter[][i_outer]=vectorx[p]
while(1)

ModelWaveOutput = vectorx
SetDataFolder OldDf
end
//=====

```

```

// Totally non-negative least squares method to solve problems which can be described as:
// B = A x Model, where B is measured data, A is m x n matrix, and Model is solution in question
// reference: Michael Merritt and Yin Zhang, J Optim Theory Appl 126,1 (2005), pp. 191-202
//
// Input variables and waves [dimensions in brackets]:
// AmatrixInput      [m x n]  kernel matrix, describing measurement collection
// ModelWaveOutput   [n]      wave to hold size distribution solution;
//                               number of points must match y-dimension of AmatrixInput
// BvectorInput      [m]      vector of measured signals (raw data)
// Bvector_uncertainties [m]    vector of uncertainty/error values associated with
//                               measured signals (raw data)
// ApproachParameter - "step" - needs to be smaller than 1, usually 0.6 is good;
//                               reasonable range seems to be 0.3 - 0.99
// MaxNumIterations - limit to sensible number...depends on complexity of problem...
//
// Output is solution in "ModelWaveOutput"
function IPG_TNNLS(AmatrixInput,ModelWaveOutput,BvectorInput,Bvector_uncertainties,Approac
    wave AmatrixInput,ModelWaveOutput,BvectorInput,Bvector_uncertainties
    variable ApproachParameter,MaxNumIterations

    // create working space
    string OldDf=GetDataFolder(1)
    NewDataFolder/O/S root:Packages
    NewDataFolder/O/S root:Packages:TNNLS
    // create local copies of input waves
    duplicate/o AmatrixInput, AmatrixInputE, AmatrixOrig
    duplicate/o BvectorInput, BvectorInputE, BvectorOrig
    duplicate/o Bvector_uncertainties, Uncertainties
    Uncertainties = Bvector_uncertainties==0 ? 1 : Bvector_uncertainties
    // note this code defaults to use uncertainties;
    // if uncertainties not available, set following parameter to 0 and modify the code
    variable useUncertaintiesInput=1
    if (useUncertaintiesInput)
        AmatrixInputE = AmatrixOrig[p][q] / Uncertainties[p]
        matrixop/o BvectorInputE = BvectorOrig / Uncertainties
    endif
    // create local meaningful waves to work with which account for the errors used
    duplicate/o AmatrixInputE, Amatrix
    duplicate/o ModelWaveOutput, ModelWave
    duplicate/o BvectorInputE, Bvector
    redimension/d Amatrix, ModelWave, Bvector, Uncertainties
    // create some parameters for use after run to find out what happened
    variable/g NumberIterations, Chisquare
    // first we will need some variables and waves, etc. to work with
    variable i,j
    matrixop/o AmatrixT=Amatrix^t
    // start with reliably small positive number; assume 1e-32 is such, but may fail in some cases...
    ModelWave = 1e-32

```

```

// working waves & variables
variable numIter=0
variable err=0
variable alphaStar, temp1, temp2
// here the iterations start...
make/o/n=1 IterCount,errorVals
duplicate/o ModelWave, ModelWaveIter
matrixop/o CurrentResultB = AmatrixOrig x ModelWave // calculated data from our model
duplicate/o BvectorOrig, tempWv, NormalizedResidual
tempWv = (BvectorOrig - CurrentResultB) / Uncertainties
NormalizedResidual = tempWv
tempWv = tempWv^2
Chisquare = sum(tempWv)
err=Chisquare/numpts(BvectorOrig)
errorVals[0]=err
do
// start of NNLS interior point gradient method itself; step designations relate to original paper
// step 1
matrixop/o Qk = AmatrixT x Amatrix x ModelWave - AmatrixT x Bvector
matrixop/o Dk = ModelWave / (AmatrixT x Amatrix x ModelWave)
matrixop/o Pk = - Dk * Qk
for(j=0;j<numpts(Pk);j+=1)
  if (Qk[j]==0)
    Pk[j] = 0
  endif
endifor
// step 2
matrixop/o AkSTAR= (Pk^t x AmatrixT x Amatrix x Pk)
temp1 = AkSTAR[0]
matrixop/o AkSTAR= - (Pk^t x Qk)
temp2 = AkSTAR[0]
alphaStar = temp2 / temp1
redimension/n=(numpts(ModelWave)) AkSTAR
AkSTAR = numtype(alphaStar)==0 ? alphaStar : 0
// above is ideal step to make; below is limiting the step so we do not get negative values...
matrixop/o AlphaWv = - ModelWave/Pk // max alpha, which we can make, if Pk is neg
AlphaWv = numtype(AlphaWv)==0 ? AlphaWv : 0
for(i=0;i<numpts(Pk);i+=1)
  if (Pk[i]<0) // if Pk negative, may have to limit the step to smaller of the two values
    AkSTAR[i]=min(ApproachParameter*AlphaWv[i],AkSTAR[i])
  endif
endifor
// step 3; new model
matrixop/o ModelWave = ModelWave + (AkSTAR * Pk) // loop back after calcs below
// end of NNLS interior point gradient method itself

```

```

// figure out Chisquare so we know if we should bail out...
matrixop/o CurrentResultB = AmatrixOrig x ModelWave // calculated data from our model
duplicate/o BvectorOrig, tempWv, NormalizedResidual
tempWv = (BvectorOrig - CurrentResultB) / Uncertainties
NormalizedResidual = tempWv
tempWv = tempWv^2
Chisquare = sum(tempWv)
err=Chisquare/numpts(BvectorOrig)
numlter+=1
NumberIterations=numlter

insertpoints numlter, 1, lterCount,errorVals
lterCount[numlter]=numlter
errorVals[numlter]=err
redimension/n=(-1,(1+numlter)) ModelWavelter
ModelWavelter[][numlter]=ModelWave[p]

if (numlter<10)
    continue
endif
while((err>1 && numlter<MaxNumIterations))
// this bails out from TNNLS if:
//1. Chisquare/number of points is less than 1 (fit within uncertainties)
//2. Number of iterations is above user input value

ModelWaveOutput = ModelWave<=1e-10 ? 0 : ModelWave
SetDataFolder OldDf
end
//=====

```

```

// Regularization method to solve problems which can be described as:
// B = A x Model, where B is measured data, A is m x n matrix, and Model is solution in question
// reference: Ilavsky and Jemian, J Appl Chryst Appl 42,2 (2009), pp. 347-353
//
// Input waves [dimensions in brackets]:
// AmatrixInput      [m x n]  kernel matrix, describing measurement collection
// ModelWaveOutput   [n]     wave to hold size distribution solution;
//                               number of points must match y-dimension of AmatrixInput
// BvectorInput      [m]     vector of measured signals (raw data)
// Bvector_uncertainties [m]   vector of uncertainty/error values associated with
//                               measured signals (raw data)
//
// Internally calculated variables and waves [dimensions in brackets]:
// H_matrix   [n x n]  constraint matrix; here done for second derivative: MakeHmatrix()
// B_vector   [n]     calculated from Kmatrix, Bvector and errors: CalculateBVector()
// D_matrix   [n x n]  calculated from Kmatrix and errors: CalculateDMatrix()
// A_matrix   [n x n]  calculated from A[][] = D[][] + a * H[][] CalculateAmatrix()
// Avalue - for the fitting itself; call with precision (e~0.1 or so): FindOptimumAvalue(Evalue)
// Evalue - internal precision parameter, for now hardwired to 0.1; range 0 to 0.5
//           lower value requires resulting chi^2 to be closer to target
// Chisquared - chi squared sum of the difference value between the fit and measured data
//
// Output is solution in "ModelWaveOutput"
//
// Internal parameter of Regularization:
// Regularization is forced to have at least this * max of SD in each bin to avoid negative values
constant RegularizationMinRatio = 1e-4

function Regularization(AmatrixInput,ModelWaveOutput,BvectorInput,Bvector_uncertainties)
    wave AmatrixInput,ModelWaveOutput,BvectorInput,Bvector_uncertainties

    // create working space
    string OldDf=GetDataFolder(1)
    NewDataFolder/O/S root:Packages
    NewDataFolder/O/S root:Packages:Regularization
    // create local copies of input waves
    duplicate/o AmatrixInput, Kmatrix
    duplicate/o BvectorInput, Bvector
    duplicate/o Bvector_uncertainties, Uncertainties
    Uncertainties = Bvector_uncertainties==0 ? 1 : Bvector_uncertainties
    redimension/d Kmatrix, Bvector, Uncertainties

```

```

MakeHmatrix()      // create H matrix
CalculateBVector() // create B vector
CalculateDMatrix() // create D matrix

variable/g Evaluate=0.1 // may not be needed in the future
variable/g NumberIterations
NumberIterations=FindOptimumAvalue(Evaluate) // do the fitting for given e value

wave CurrentResultSizeDistribution=root:Packages:Regularization:CurrentResultSizeDistributic
wave NormalizedResidual=root:Packages:Regularization:NormalizedResidual
nvar Chisquare=root:Packages:Regularization:Chisquare

if ((numtype(NumberIterations)!=0)||((numberIterations>100)) // no solution found
    CurrentResultSizeDistribution = 0
    NormalizedResidual = 0
    Chisquare = 0 // new Chisquared
endif

ModelWaveOutput=CurrentResultSizeDistribution
SetDataFolder OldDf
end
//=====

```

```

// Make the H_matrix
function MakeHmatrix()

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave Kmatrix=root:Packages:Regularization:Kmatrix
    variable numOfPoints=dimsize(Kmatrix,1), i=0, j=0

    make/d/o/n=(numOfPoints,numOfPoints) H_matrix; H_matrix=0 // make and zero the matrix
    for(i=2;i<numOfPoints-2;i+=1) // fill most of matrix with 1 -4 6 -4 1
        for(j=0;j<numOfPoints;j+=1)
            if (j==i-2)
                H_matrix[i][j]=1
            endif
            if (j==i-1)
                H_matrix[i][j]=-4
            endif
            if (j==i)
                H_matrix[i][j]=6
            endif
            if (j==i+1)
                H_matrix[i][j]=-4
            endif
            if (j==i+2)
                H_matrix[i][j]=1
            endif
        endfor
    endfor
    H_matrix[0][0]=1 // fill in beginning of the H_matrix
    H_matrix[0][1]=-2
    H_matrix[0][2]=1
    H_matrix[1][0]=-2
    H_matrix[1][1]=5
    H_matrix[1][2]=-4
    H_matrix[1][3]=1
    H_matrix[numOfPoints-2][numOfPoints-4]=1 // fill in end of the H_matrix
    H_matrix[numOfPoints-2][numOfPoints-3]=-4
    H_matrix[numOfPoints-2][numOfPoints-2]=5
    H_matrix[numOfPoints-2][numOfPoints-1]=-2
    H_matrix[numOfPoints-1][numOfPoints-3]=1
    H_matrix[numOfPoints-1][numOfPoints-2]=-2
    H_matrix[numOfPoints-1][numOfPoints-1]=1

    SetDataFolder OldDf
end
//=====

```

```

// Make new B_vector, calculated from Kmatrix, Bvector and Uncertainties
function CalculateBVector()

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave Kmatrix =root:Packages:Regularization:Kmatrix
    wave Bvector =root:Packages:Regularization:Bvector
    wave Uncertainties=root:Packages:Regularization:Uncertainties

    variable M=dimsize(Kmatrix,0) // rows, i.e, measured points number
    variable N=dimsize(Kmatrix,1) // columns, i.e., bins in distribution
    variable i=0, j=0
    make/d/o/n=(N) B_vector // points = bins in size dist.
    B_vector=0
    for(i=0;i<N;i+=1)
        for(j=0;j<M;j+=1)
            B_vector[i]+=((Kmatrix[j][i]*Bvector[j])/(Uncertainties[j]*Uncertainties[j]))
        endfor
    endfor

    SetDataFolder OldDf
end
//=====

```

```

// Make new D_matrix, calculated from Kmatrix and Uncertainties
function CalculateDMatrix()

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave Kmatrix=root:Packages:Regularization:Kmatrix
    wave Uncertainties=root:Packages:Regularization:Uncertainties

    variable M=dimsize(Kmatrix,0) // rows, i.e, measured points number
    variable N=dimsize(Kmatrix,1) // columns, i.e., bins in distribution
    variable i=0, j=0, k=0
    make/d/o/n=(N,N) D_matrix; D_matrix=0
    duplicate/o Uncertainties, Uncertainties2
    Uncertainties2=Uncertainties^2

    duplicate/o Kmatrix, Kmatrix_ErrScaled
    for(i=0;i<N;i+=1)
        for(j=0;j<M;j+=1)
            Kmatrix_ErrScaled[j][i]=Kmatrix[j][i]/(Uncertainties2[j])
        endfor
    endfor
    matrixop/o testM = Kmatrix_ErrScaled^t x Kmatrix
    D_matrix = testM

    SetDataFolder OldDf
end
//=====

// Generate A_matrix
function CalculateAmatrix(aValue)
    variable aValue

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave D_matrix=root:Packages:Regularization:D_matrix
    wave H_matrix=root:Packages:Regularization:H_matrix

    duplicate/o D_matrix, A_matrix; A_matrix=0
    A_matrix=D_matrix[p][q]+aValue*H_matrix[p][q]

    SetDataFolder OldDf
end
//=====

```

```

// Do the fitting itself, call with precision (e~0.1 or so)
function FindOptimumAvalue(Evalue)
    variable Evalue

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave Kmatrix=root:Packages:Regularization:Kmatrix
    wave Bvector=root:Packages:Regularization:Bvector
    wave Uncertainties=root:Packages:Regularization:Uncertainties
    variable numOfPoints=dimsize(Kmatrix,1)

    variable ChisquareVal,err
    make/o/n=1 IterCount,errorVals,ChiSqVals, Avals
    make/o/n=(numOfPoints) ModelWaveOutIter; ModelWaveOutIter=0
    MatrixOp/O CurrentResultB = Kmatrix x ModelWaveOutIter // calculated data from our model
    duplicate/o Bvector, tempWv, NormalizedResidual
    tempWv = (Bvector - CurrentResultB) / Uncertainties
    NormalizedResidual = tempWv
    tempWv = tempWv^2
    ChisquareVal = sum(tempWv)
    err=ChisquareVal/numpts(Bvector)
    errorVals[0]=err
    ChiSqVals[0]=ChisquareVal
    Avals[0]=1

    variable LogAmax=100, LogAmin=-100, M=numpts(Bvector)
    variable Chisquared, MidPoint, Avalue, i=0, logAval, LogChisquarerDivN, Smoothness
    do
        MidPoint=(LogAmax+LogAmin)/2
        Avalue=10^MidPoint // calculate a
        CalculateAmatrix(Avalue)
        wave A_matrix
        variable n = dimsize(A_matrix,1)
        make/o/n=(n) M_x; M_x=0
        wave B_vector
        NNLS(A_matrix,B_vector,M_x)
        wave M_x
        duplicate/o M_x, CurrentResultSizeDistribution // put the data into the wave

        Chisquared=CalculateChisquared() // calculate C C=|| I - G M_x ||
        i+=1
        if (i>100) // no solution found
            print i
            return NaN
        endif

```

```

// calculate data from model and track results through iterations
matrixop/o CurrentResultB = Kmatrix x CurrentResultSizeDistribution
duplicate/o Bvector, tempWv, NormalizedResidual
tempWv = (Bvector - CurrentResultB) / Uncertainties
NormalizedResidual = tempWv
tempWv = tempWv^2
ChisquareVal = sum(tempWv)
err=ChisquareVal/numpts(Bvector)
insertpoints i,1, IterCount,errorVals,ChiSqVals,Avals
IterCount[i]=i
errorVals[i]=err
ChiSqVals[i]=Chisquared
Avals[i]=Avalue
redimension/n=(-1,(1+i)) ModelWaveOutIter
ModelWaveOutIter[[i]]=CurrentResultSizeDistribution[p]

variable tolerance=1e-10
if (sum(Uncertainties)==M)
  if (Chisquared>(M/10))
    LogAMax=MidPoint
  else
    LogAmin=MidPoint
  endif
else
  if (Chisquared>(M/100))
    LogAMax=MidPoint
  else
    LogAmin=MidPoint
  endif
endif

if(i<50)
  continue
endif
while((err>1 && i<101))

variable/g Chisquare=Chisquared
return i

SetDataFolder OldDf
end
//=====

```

```

// Calculate Chisquared difference between data in Bvector and result calculated from x_vector
function CalculateChisquared()

    string OldDf
    OldDf=GetDataFolder(1)
    SetDataFolder root:Packages:Regularization

    wave Kmatrix=root:Packages:Regularization:Kmatrix
    wave Bvector=root:Packages:Regularization:Bvector
    wave Uncertainties=root:Packages:Regularization:Uncertainties
    wave M_x=root:Packages:Regularization:M_x

    duplicate/o Bvector, NormalizedResidual, ChisquaredWave // waves for data

    MatrixMultiply Kmatrix, M_x // generate data from current result
    wave M_product
    redimension/d/n=(-1,0) M_product

    NormalizedResidual=(Bvector-M_product)/Uncertainties
    ChisquaredWave=NormalizedResidual^2 // wave with Chisquared
    return (sum(ChisquaredWave,-inf,inf)) // return sum of Chisquared

    SetDataFolder OldDf
end
//=====

```