Response to Reviewer Comments

July 1, 2021

Author Statement

I thank the referee for their time to review this manuscript and their constructive critiques. Below are itemized responses to the referees' comments. In response to the comments, Section 2.3 of the manuscript was significantly expanded. The complete revised section is mentioned in several responses. To avoid repetition, the revised section is reproduced at the end of the document.

Response to Reviewer #1

Overview
This manuscript presents a software package to invert aerosol size distributions from measurements, in particular from scanning mobility particle sizers (SMPSs), using the Tikhonov regularization approach.

This manuscript sits at the intersection of producing open-source, scientific code and presenting new scientific ideas. The reviewer admits this is an awkward position, as existing dissemination methods are not amendable to publishing well-maintained software, which is a critical component to modern, reproducible analysis. That being said, the scientific contributions of the underlying code are not hugely significant. Inversion of aerosol distributions using Tikhonov regularization is well-established. In fact, the author already notes one other instance of open-source software designed - at least in part - for this task (Hansen). (Other codes undoubtedly exist, though, admittedly most of these codes are closed source or not immediately available to the user, with very few exceptions, as the author notes.) Otherwise, this code does little to innovate on existing methods and is somewhat behind in terms of state-of-the-art, such as not presenting any form of uncertainty quantification - see Kandlikar and Ramachandran (1999); Voutilainen et al., (2001); and Voutilainen, Kolehmainen, Stratmann, & Kaipio (2001). The use of a GSVD to speed computation is insightful but is still based on existing literature. The code does also extend existing analysis tools to the Julia programming language, but it is this reviewer's opinion that this contributes little in terms of a novel scientific contribution.

Of note, the author could focus on the less-investigated HTDMA problem and the specific challenges that arise for that application (e.g., present the underlying integral equation for that scenario), which the authors note in the conclusion is one of the more novel aspects of this manuscript.

Altogether, this reviewer thinks the manuscript could be reoriented more towards novel scientific components, including more of a focus on HTDMA. As such, the author SHOULD be given the chance to respond to comments and refine the manuscript. MAJOR REVISIONS are recommended.

[1] *I do not disagree with the referee about the scientific novelty of the regularization code. The manuscript makes no claim in this regard. It also is not the purpose of the paper. I agree*

[1] *Response*

that the most novel part of the work is the new HTDMA inversion. I have changed the title to "Revisiting Matrix-Based Inversion of SMPS and HTDMA Data" to reflect that. Uncertainty quantification described in the papers brought up by the referee is now included in the discussion.

That said, I will defend the work largely "as is" as a significant contribution in Atmospheric Measurement Techniques. Here is why. At issue is neither the mathematical novelty of Tikhonov regularization nor it's application to size distribution data. As mentioned by the referee, and in the draft manuscript, this has been demonstrated in the literature long before. The issue is about accessibility and extensibility of these techniques to the measurement community that is not trained in inverse problem solving. This is still true for size distribution inversion. Quoting from an anonymous referee of the preceding 2018 manuscript:

> In computational work, graduate students and senior scientists tend to "reinvent the wheel." This wastes time and introduces errors. (In contrast, we happily use purchased instruments to make measurements sometimes with "black box" codes that contain errors that are exceedingly difficult to discover.)

To perform size distribution inversion one can either use inflexible closed-source code, somehow be lucky enough to be handed down code from established laboratories and use them to ones own advantage, or write ones own. As stated by the referee quoted (it echos my opinion), the latter option is unrealistic for researchers that do not seek a career in inverse techniques, but simply want to make good measurements or develop instruments. This work reports on critical improvements to the open DifferentialMobilityAnalyzers.jl package that addresses this problem. It significantly improves inversion speed and extending the capabilities to higher order inversions. Those improvements are based on implementing high-performance algorithms. Reporting these improvements in the literature is valuable in its own right.

As discussed by Gysel et al. [2009], HTDMA inversions are complex to develop and not yet applied universally to data. The novelty and purpose of this work is to describe a methodology to tame the complexity developing inversion schemes and to provide a means to apply inversion to data for practitioners. Taming complexity is proposed to be achieved by three new ideas. First, the code systematically classifies regularization input assumptions and creates a simple interface for practitioners for trying out methods. I am not aware that such an interface is available anywhere else. Second, the work introduces a means to create design matrices from arbitrary forward models, although the details were only described in the supplementary material. Third, this work further extends the formalisms first introduced in the Petters 2018 paper, to show how it can be applied to HTDMA inversion. Breaking the problem into three independent parts should help prototyping and adapting future inversion approaches. The HTDMA inversion reported here is scientifically novel by addressing the limitations of oscillatory solutions reported in Cubison et al. [2005] and including multi-charge correction in the matrix. The conceptional and practical framework on how to approach the forward problem is novel.

Specific Comments

[2] The focus on the programming aspects also often distracts from the underlying science. For example, presenting the underlying mathematics in a programming language- and program-specific representation without the more standard mathematical forms (e.g., the underlying integral equations) makes the manuscript harder to follow. It seems that in an attempt to tread a line between a scientific manuscript and code documentation, the manuscript does not accomplish either task particularly well. In this respect, the manuscript may be better structured by clearly presenting the underlying scientific principles in a more standard mathematical notation, moving coding references out of the body of the manuscript. The alternative - presenting the manuscript as a form of documentation for a program - is better structured with specific coding examples in the text. However, this latter route is less amendable to a research article in AMT. In this case, another platform (a technical note in a journal or an article in a computational journal) may be more suitable. As a hybrid, the SI could be formally formed into documentation for the code that refers to the scientific principles in the base article without cluttering the body with code snippets and representations. Regardless, clarifications should be made before further review.

[2] Referee

[3] *I thank the referee for this comment, even though I disagree with it. Before responding in detail, I would like to list the revisions to section 2.3 made in response to the comment.*

[3] *Response*

1. **Included explicit references to the standard integral equations.**
2. **Clarified the purpose of the notation as a formal representation of the problem (they are not code snippets or code documentation; there is a separate detailed code documentation that is a supplement to this work).**
3. **Cleaned up the notation to limit it to standard computational concepts, i.e. eliminated parts that could be interpreted as programming language specific constructs.**
4. **Significantly expanded the text to aid parsing of the expressions.**

Rationale: The expressions for the forward model presented in the manuscript are unconventional and were perceived by the referee as "code snippets". This is incorrect. What they really represent is a domain specific language comprising a set of simple building blocks that can be used to algebraically express the response functions intuitively through a form of pseudo code. The expressions evaluate to a deterministic answer and represent just a different form of mathematics. ***The main advantage of this approach is that the expressions simultaneously encode the theory governing the transfer through the DMA and the algorithm to compute the solution.*** *The resulting expressions are concise. They are easily identified within actual source code. This makes the code easily modifiable by non-experts to change existing terms or add new convolution terms without the need to develop algorithms.*

I want to elaborate on the computational viewpoint I have taken. The expressions evaluate in the same way than mathematical functions. The applied concepts are borrowed from the functional programming community and makes use of broadly understood concepts such as

*lambda functions, generic functions, pure functions, higher order functions, function composition, and domain specific algebras. The expressions are a valid format to represent the mathematics. I have expanded the text to more carefully define each of the building blocks. I also recognize that these concepts are less widely used in the atmospheric community than the standard mathematical form. The expressions themselves can at first glance be more difficult to parse than the seemingly simple and familiar integral equations. Nevertheless, referee Mark Stolzenburg was able to follow the work (and call out two hidden assumptions) from the admittedly not-so-well written initial draft section. The assumptions had a very small effect on the final result, but of course it is important to address them when striving for correctness (which is done in the revised manuscript). This proves my point(s) above. It is trivial to recite the integral equations from one of the many preceding papers. Yet these equations do not fully communicate the model. The assumptions I made that were identified by Mark Stolzenburg would likely have never been detected in review, because the mathematical form is completely detached from the algorithmic solution. However, I understand the value of these equations and I now refer the reader to those work*s.

A disadvantage of the computational approach over the traditional mathematical approach is that algorithmic descriptions lack standardization of notation. This can blur the line between the pseudo code notation and language specific syntax. The reviewer brought to light that I had used some julia language specific constructs which I had introduced in the 2018 work. This is not ideal, because the expressions are really general and programming language independent. I therefore eliminated language specific constructs and only use generic functions that fall into the domain of general computing concepts. This results in more general expressions that are interpretable in most modern programming languages/syntax frameworks.

I firmly believe that the advantages of the computational approach outweigh their drawbacks. This work is in part an experiment on how to conceptionally model DMA transfer in the computational domain. It may in the end remain an obscure approach, and one that is not the preferred one by the referee or the majority of the field, but this is not a justification to hide it in a supplement or a computational journal. The work addresses atmospheric measurement techniques using computational concepts, not computational concepts themselves. Publication of this work is only adding to the list of available approaches; it does not force anyone adopt either the notation or approach.

[4] **Please see revised section 2.3 at the end of this document.** [4] **Revision**

[5] In the abstract, the authors note that the inversion speed is improved by ~200 times down to 2-5 ms. Is the implication to work towards online inversion of the measurements? If not, there is a fair degree of flexibility in terms of inversion speed, such that speed may not be the only or the best metric gauging improvement. Can the authors comment? If the hope is for online inversion, can the authors comment on the interface with the instrument, which would be a substantial component of the overall process. [5] Referee

[6] *Improving performance in terms of speed is desirable as long as the inversion step presents some form of bottle neck for a particular application. Two example applications for inversion discussed in this manuscript involve either inversion of large data sets using different assumptions or inversion in real time during data acquisition. The quoted times are approaching the speed where the inversion bottle-neck disappears, although that will depend on the specific circumstance. As mentioned in the manuscript data acquisition and inversion on inexpensive reduced-instruction-set architecture is now possible. The interface to instrumentation depends on the user. We use julia as language to write all data acquisition software. The inversion is then just a function call to the software package(s) given as a supplement in this manuscript. The author shares the data acquisition software for scanning mobility particle sizers via GitHub (https://github.com/mdpetters/smpsDAQ) that is widely used in our laboratory. However, it is currently not well-documented. We mostly run the software on x86 and we are currently experimenting with running it on ARM v8 systems. Translating the approach to Python or other languages should be fairly easy.*

[6] *Response*

[7] **Since there is no peer-reviewed publication of the SMPS software, and since the response to the referee comment is publicly available, we do not discuss this further in the manuscript.**

[7] **Revision**

[8] Related to the above, this code is 200 times faster compared to what? A previous version of this code? It is worth noting that Tikhonov regularization for these distributions is a relatively straightforward problem, solving a simple linear system. As such, the speed improvements are likely linked to the external libraries that solve the linear system, something which the authors do imply later in the work. However, this does limit the novelty of using those methods for a different problem

[8] Referee

[9] *Yes, 200 times faster compared to a previous non-optimized regularized inverse. The speed improvement is due to the application of factorization techniques and implementing the numerical algorithms described in Section 2.1.2 instead of relying on the naive matrix inverse used in Petters (2018). The implemented algorithms are general. Virtually all languages, including julia, outsource basic linear algebra computations (e.g. the QR factorization) to highly performant external libraries (LAPACK, OpenBLAS, MKL) and the inversion speeds of these libraries are fairly similar. The reason that RegularizationTools.jl is distributed as a separate package is that it can be applied to any inversion problem, not just the DMA examples highlighted here. Examples for the generality of the approach are given in Section 2.1.4.*

[9] *Response*

[10] Line 93: What is the dimension/size of the different quantities defined here? Based on the subsequent discussion, it seems that A is assumed to be square (same reconstruction and measurement discretization). The A matrix is not required to be square, but this reviewer thinks it does make the GSVD simpler to compute (a

[10] Referee

non-square matrix may require special treatment) and should be stated clearly.

[11] *The matrix* **A** *does not need to be square. All algorithms are implemented to allow for non-square problems. An example for a non-square problem is given in the documentation to RegularizationTools.jl, which is a formal supplement to this paper as stated in the "Code and data availability." section. The relevant example for a non-square problem here: https://mdpetters.github.io/RegularizationTools.jl/stable/manual/#Creating-a-Design-Matrix under "Example 2". It is now stated that the A matrix need not to be square. The description for matrix A2 has been updated to describe the discretization, where it also mentions that the matrix does not need to be square. It was given as square due to the specific discretization scheme used to generate the figures in the draft.* [11] *Response*

[12] **..., A is the design matrix (which may or may not be square), x is the true quantity of interest, and ffl is the random error.** [12] **Revision**

[13] **..., the matrix $\mathbf{A}_2$ is understood to be computed for a specific input aerosol size distribution, and ϵ is a vector that denotes the random error that may be superimposed as a result of measurement uncertainties. The size of $\mathbf{A}_2$ is $j \times n$.** [13] **Revision**

[14] Line 108: Consider explicitly noting that automating the L-curve method, while feasible, is often more challenging than other automated methods and can be affected by noise and type of solver (which the authors admittedly imply later when they state that the L-curve algorithm used previously occasionally failed). [14] Referee

[15] *Done.* [15] *Response*

[16] **The optimal λ occurs at the corner of the L-curve, which can be found algorithmically. However, automating the L-curve method can be more challenging than other automated methods, as further discussed below.** [16] **Revision**

[17] Line 112: Clarify "standard form". What is the standard form? How would one compute this standard form? Under what conditions does one not use the standard form? [17] Referee

[18] *Equation (3) is in standard form if* $\mathbf{L} = \mathbf{I}$*. (Stated a few lines above). The text around line 112 has been slightly reworded to make this clear.* [18] *Response*

[19] **If $\mathbf{L} \neq \mathbf{I}$, Eq. (3) is transformed to standard form using the generalized singular value decomposition of A and L as derived by Eldén (1982) and summarized by Hansen (1998).** [19] **Revision**

[20] Line 208: Is Petters (2018) the best reference for this? The underlying equations [20] Referee

for the discrete transfer function of the SMPS have been stated more formally many times before this work. If there is something specific in Petters (2018) about which the authors can be more explicit? There are also multiple ways to discretize the problem, which could be a route to a more specific representation from Petters. Further, why not present this in a more standard form, such as the transfer function given by Stolzenburg (2018), rather than a programming language-specific representation?

[21] Through Section 2.3: Similar to above, why not present all of the physics in terms of its underlying integration equations rather than language-specific concatenation and mapping operators or convolution "*" operators? For the discrete version, why not state these in terms of matrices instead? Interestingly, there are multiple ways to discretize the problem (e.g., finite element bases), which is also not detailed here. The HTDMA problem is based on a double convolution with three components to the underlying integral equation/kernel: 1) the transfer function of the first DMA, 2) a kernel describing the humidification process, and 3) the transfer function of the second DMA. This feature is not clear from the current reading. [21] Referee

[22] *Combined response to 20 and 21. Section 2.3 has been significantly revised based on comments by Mark Stolzenburg and comments above. The method used for matrix generation is discussed in Section 2.2. The method is equivalent to the quadrature method, as discussed in the supplemental documentation (https://mdpetters.github.io/RegularizationTools.jl/stable/manual/#Creating-a-Design-Matrix).* [22] *Response*

[23] **See revised section 2.3 at the end of this document.** [23] **Revision**

[24] Line 268+: The current manuscript structure makes it challenging to ascertain the role of the 30 (or other) bins for the growth factor in the overall procedure. This reviewer would expect that the growth factor would contribute to another matrix that bridges the mobility distribution output by the first DMA to the mobility distribution input to the second DMA. In this respect, since the other components of the problem have a constant number of bins (at least this reviewer gathered as such), would it not make more sense to have a matrix with the same dimension/number of bins as the larger problem? Further, depending on whether one is inferring these quantities or not, this matrix could be combined with one or more of the DMA transfer function kernels and thus be pre-computed, with little effect on the overall computational effort. If one is inferring the growth factor, the structure of the problem deviates somewhat from the more general aerosol inversion problem, a fact that should be clarified. Namely, there will be at least two integrations (over the mobility distributions for each DMA) with an intermediary quantity that is being inferred. Then, there is also the question of the uncertainties in the input size distribution, which is measured independently, also inferred, or assumed. Overall, these definitions could be clarified. [24] Referee

[25]*The discretization/structure of the HTDMA inversion problem is now explained in more detail. Uncertainty in the size distribution will propagate into* $\mathbf{A}_2$*. This uncertainty is now mentioned in the manuscript.* [25] *Response*

[26]**For purposes of the forward model, the mobility grid for DMA 1 is discretized at a resolution of** i **bins. Transmission through DMA is computed for a specified** z^s **(the dry mobility),** g_0 **(the growth factor), and an input size distribution, which results in a vector** i **concentrations along this grid. If the input size distribution does not match the mobility grid the grids are merged through interpolation. The mobility grid for DMA 2 is discretized at a resolution of** j **bins. The transmitted and grown distribution from DMA 1 (**i **bins along the mobility axis of DMA 1) is interpolated onto the mobility grid of DMA 2. The outer integral in Eq. (17) is discretized into** n **bins that models** P_g**. If the output mobility of grid of DMA 2 does not match, the grids are merged through interpolation. The choice of** i**,** j**,** n**, the ranges of mobility grids for DMA 1, DMA 2, and the range of** P_g **is only constrained by computing resources and a physically reasonable representation of the problem domain. Reasonable choices are** $i = 120$**,** $j = n = 30$**. The forward model is used to cast Eq. (17) into matrix form such that the humidified mobility distribution function is given by** [26] **Revision**

$$\mathbb{m}_t^{\delta_2} = \mathbf{A}_2 P_g + \epsilon \tag{18}$$

where the subscript 2 specifies transmission through DMA 2, the matrix $\mathbf{A}_2$ **is understood to be computed for a specific input aerosol size distribution, and** ϵ **is a vector that denotes the random error that may be superimposed as a result of measurement uncertainties. The size of** $\mathbf{A}_2$ **is** $j \times n$**. Uncertainties in the size distribution propagate into** $\mathbf{A}_2$**. The main influence of the error will be the relative fractions of +1, +2, and +3 charged particles. Assuming a random error of** $\pm$**20% in concentration, the overall effect on the** $\mathbb{m}_t^{\delta_2}$ **is expected to small.**

[27]Line 276: The use of Poisson noise could be used to appropriately weight the data. Why was this not considered (i.e., use weighted least-squares instead of naïve least-squares)? One limitation is that measurements that span multiple orders of magnitude will result in numerical instabilities, such that a baseline amount of background noise may be required. Can the authors comment? [27] Referee

[28]*I have not tried weighted least-squares. It's plausible that it helps. However, Poisson noise may not be the only source of error in the measurement. (For examples, false counts from leaks in the line, fluctuations of RH in sample flow, flow rate fluctuations, electronic noise, etc. may all contribute to the error). It's not clear how to estimate the total error from data. Since L2-regularization works well for the problem there is no need to explore this approach.* [28] *Response*

[29]Line 280: How often would this a priori information be known? In the experimental section to follow, there is a short phrase about this being computed using the inverse of the S matrix. Would it be worth noting this here? Also, what is S? This information does not seem to be immediately available. [30]*For the HTDMA problem (line 280), the "a-priori estimate x_0 is taken to be the normalized apparent growth factor distribution, where the normalization ensures that the sum over all bins is unity." This information is derived from the measured data, so it is always available. This is now mentioned in the text. The* **S** *matrix is used to compute the a-priori guess for size distribution inversion (line 355). It is explained there how it is derived at the location where it is first introduced: "..., where S is obtained by summing the rows of* **A** *and placing the results on the diagonal of* **S** *(Talukdar and Swihart, 2003)."*

[29] Referee

[30] *Response*

[31]**The *a-priori* estimate x_0 is taken to be the normalized apparent growth factor distribution derived from the measured response function, where the normalization ensures that the sum over all bins is unity**

[31] **Revision**

[32]Line 280: Continuing from above, do the choices for x_o make sense given the chosen Tikhonov matrix? For example, a first-differences Tikhonov matrix encodes information about the expected slope of the solution. Using (x - x_o) implies regularization of the slope of the residual with respect to an a priori estimate. Can the author comment?

[32] Referee

[33]*With the exception of the of $L_0 D_{1e-3} B_{[0,1]}$ method, which is creates a Tikhonov matrix that is less sensitive to sharp edges, all of the methods worked similarly well when tested against simulated test data. For example the inversion using $L_2 x_0 B_{[0,1]}$ and $L_2 B_{[0,1]}$ produces reconstructions of similar quality (see supplementary information). So empirically, inclusion of this particular a-priori x_0 does not make a difference when smoothing with derivative operators L_1 and L_2. I also experimented with $L_2 x_0 B_{[0,\infty]}$ for size distribution inversion (now discussed in the paper) and found that it works, though without the smoothing benefits. This is because for large regularization parameters, the solution converges toward the initial guess, regardless of the choice of* L. *Letting a-priori information through the filter may thus negate the benefit of smoothing.*

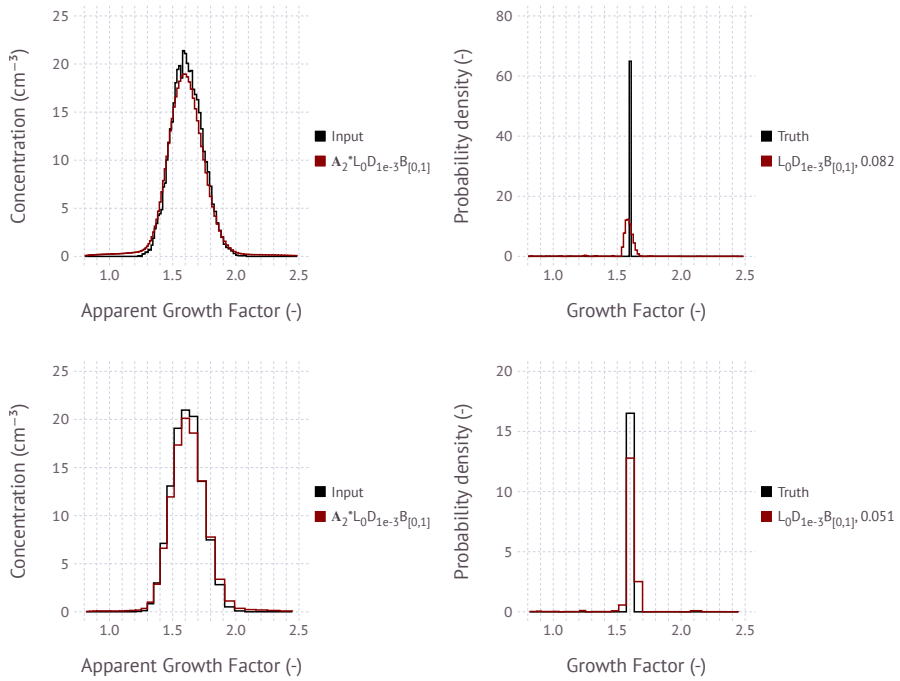
[33] *Response*

[34]**Second order inversion using $L_2 B_{[0,\infty]}$ produces a smooth, denoised solution due to application of the derivative operator in the regularization filter matrix. The solution converges even though no *a-priori* estimate is used, i.e., $x_0 = 0$. Inclusion of an *a-priori* in the form of $L_2 x_0 B_{[0,\infty]}$ is possible. However, noise in the *a-priori* propagates into the solution, thus negating the intended benefit of the second order Tikhonov matrix.**

[34] **Revision**

[35]With respect to, "Higher resolution grids generally lead to poor performance even for method LoD1e−3B[0,1]." This is *slightly* surprising. Given the way Tikhonov prior operates, one may expect the extra grid points to be filled using the prior (a little like interpolating between lower resolution points, but not quite the same). Could this be an indication of limitations in the error metric used (there are most points at which the error is being calculated such that one is not comparing the same quantity)? Alternatively, the regularization parameter would change depending on the reconstruction grid. Was the regularization parameter re-optimized each time? [35] Referee

[36]*The regularization parameter is optimized for each inversion. The effect is not due to the definition of the error metric. The effect of higher-resolution grids leading to poor performance is limited to the case with a single bin/sharp edges and using the two-step data-based regularization. The figure below shows and example of this for 120 bins and 30 bins and the same input distribution.* [36] *Response*



The two-step regularization technique first performs a reconstruction based on L_0 and the uses this to build a revised Tikhonov matrix. More bins generally lead to the same spread in the first reconstruction. Narrowing the solution down further is not possible based on that input. The text now clarifies that this only applies to the discrete resolution cases.

[37]**Higher resolution grids generally lead to poor performance for discrete populations even for method $L_0D_{1e-3}B_{[0,1]}$.** [37] **Revision**

[38]Paragraph around Line 315: Is the unweighted residual really the best metric? How about measurement noise (one may have more confidence in some measure- [38] Referee

ments than others)? Should this be accommodated in terms of calculating this residual?

[39] *Please see a detailed response to referee Mark Stolzenburg for detailed discussion about why the RMSE was selected as residual (the last comment in my response to his comments). Weighing the RMSE by the measurement error is possible, but not desirable. Specifically, that would mean that bins with low or zero counts would effectively be excluded from the error estimation. However, there are cases where the model produces false oscillatory solutions (predicted counts) when measured (or expected counts) are zero. Filtering these in the error metric would bias the results.*

[39] *Response*

[40] Line 355: Small values in the A matrix do not matter as much as where they are located. Small diagonals or nearly all-zero rows/columns are the real issues. Consider clarifying. There is the issue of numerical noise (scattered small values) in the kernel, which does little but slow down the inversion. Was this dealt with?

[40] Referee

[41] *Thank you for pointing this out. The language is revised (see below). No attempt was made to filter numerical noise in the kernel.*

[41] *Response*

[42] **Inclusion of these terms results in a more ill-posed inverse problem due to increasing overlap between the kernels [Kandlikar and Ramachandran, 1999].**

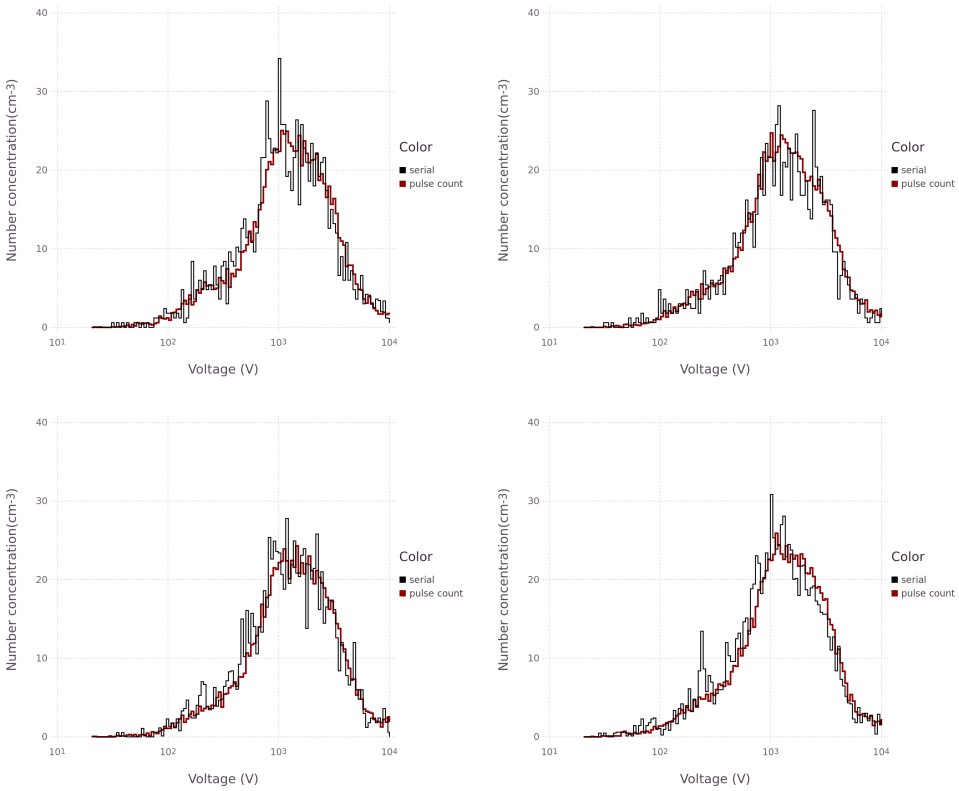
[42] **Revision**

[43] Figure 5: The real-world noise in Fig. 5 does not seem to match noise in other number concentrations reported in the theoretical components of the work. Can the authors comment on this difference and/or update the earlier scenarios to be more representative?

[43] Referee

[44] *This is an excellent observation. There are a couple of differences between the earlier scenarios and this real-world example. The example is for size distribution measurement, while the previous scenarios are for growth factor measurements. However, I verified that it is true that random error in Fig. 5 is larger than what one predict from Poisson counting statistics alone. As mentioned earlier, there are other factors that may increase random noise in the data. In this particular case, the additional noise is related to the internal electronics of the specific CPC model.*

[44] *Response*



The Figure shows a voltage scan from a DMA (TSI long column, 9:1 flow ratio, 120 s voltage scan) acquired with the same CPC model (TSI 3771/3772) as in the paper. Each bin corresponds to 1 second data. The two data streams are the digitized pulse output acquired using an external pulse counter card and the output from the serial port. (For the distributions in the paper, only serial port data were available). The serial port output is much noisier than the pulse count and the issue is present for all units of that particular model series. The pulse counts are more consistent with Poisson statistics. It is not entirely clear to me why the CPC serial output is so poor for this model. It seems to be related to the on-board processing of raw counts, which appears to be too slow. We identified this issue in 2016 and now always acquire both serial port and pulse data when available. Since this issue is related to a specific model and data acquisition mode there is no need to update the hypothetical HTDMA scenarios. Obviously noisier data is more difficult to invert. The observation that the noise exceeds the Poisson noise in the example is now is now mentioned in the text.

[45]**The ragged structure is typically explained by random noise due to Poisson counting statistics. However, in this example the noise level is larger than Poisson statistics alone, which is thought to be due to the processing of raw data internal to the specific CPC model that was used to collect the data.**

[45] **Revision**

[46]For the temporally-evolving measurements, recent work by Ozon et al.

[46] Referee

(https://acp.copernicus.org/preprints/acp-2021-99/) presents an improvement to this existing technique and is closer to state-of-the-art. Can the authors comment and cite appropriately?

[47] *Thank you for the comment. The possibility is now mentioned. It is not clear though how this would work in ambient settings where conditions can change rapidly and unpredictably due to emissions or wind-direction changes. The possibility is mentioned in the revised manuscript.*

[47] *Response*

[48] **In situation where the temporal evolution of the size distribution is predictable, e.g. environmental chamber measurements, Kalman smoothing might be used to predict the in-between states [Ozon et al., 2021b,a].**

[48] **Revision**

[49] Line 460: Code would never involve writing out the Fredholm integral equations, making this statement a bit confusing. Further, scientific manuscripts supporting such code probably should state the underlying Fredholm integral equations. As before, program-specific language makes the scientific components of the article harder to follow.

[49] Referee

[50] *Please see my response to the earlier comment about the motivation for this approach.*

[50] *Response*

Revised Section 2.3

Design Matrices For Differential Mobility Analyzers

Differential mobility analyzers consist of two electrodes held at a constant- or time-varying electric potential. Cylindrical [Knutson and Whitby, 1975] and radial [Zhang et al., 1995, Russell et al., 1996] electrode geometries are the most common. Charged particles in a flow between the electrodes are deflected to an exit slit and measured by a suitable detector, usually a condensation particle counter. The fraction of particles carrying k charges is described by a statistical distribution that is created by the charge conditioner used upstream of the DMA. The functions governing the transfer through bipolar charge conditioners, single DMAs, and tandem DMAs are well understood [Knutson and Whitby, 1975, Rader and McMurry, 1986, Reineking and Porstendörfer, 1986, Wang and Flagan, 1990, Stolzenburg and McMurry, 2008, Jiang et al., 2014].

The traditional mathematical formulation of transfer through the DMA is summarized in Stolzenburg and McMurry [2008] and references therein. Briefly, the integrated response downstream of the DMA operated at voltage V_1 is given by a single integral that includes a summation over all selected charges. The size distribution is measured by varying voltage V_1, which produces the raw response function defined as integrated response downstream of the DMA as a function of upstream voltage. The size distribution is found by inversion. The basic mathematical problem associated with inverting the response function to find the size distribution is summarized by Kandlikar and Ramachandran [1999]. The integral is discretized by quadrature to find the design matrix that maps the size distribution to the response function. L_2 regularization is one of several methods to reconstruct the size distribution from the response function [Voutilainen et al., 2001, Kandlikar and Ramachandran, 1999].

The integrated response downstream of a tandem DMA that is operated at voltages V_1 and V_2 is given by a double integral and the summation of all selected charges. The integrals are over the upstream size distribution and the aerosol conditioner function, which here is the growth factor frequency distribution. Scanning over a range of voltages V_2 results in the raw TDMA response function. The objective is to find find a design matrix that maps the growth factor frequency distribution to the raw TDMA response function.

Petters [2018] introduced a computational approach to model transfer through the DMA. The main idea of the approach is to provide a domain specific language comprising a set of simple building blocks that can be used to algebraically express the response functions intuitively through a form of pseudo code. The main advantage of this approach is that the expressions simultaneously encode the theory governing the transfer through the DMA and the algorithmic solution to compute the response function. The resulting expressions are concise. They are easily identified within actual source code. This makes the code easily modifiable by non-experts to change existing terms or add new convolution terms without the need to develop algorithms.

A disadvantage of the computational approach over the traditional mathematical approach is that computation lacks standardization of notation. This can blur the line between general pseudo code and language specific syntax. Some of the applied computing concepts may be less widely known when compared to standard mathematical approaches. Nevertheless, the author believes that the advantages of the computational approach outweigh the drawbacks. Therefore, this work builds upon the expressions reported in Petters [2018]. Updates and clarifications to the earlier work are noted where appropriate.

The computational language includes a standardized representation of aerosol size distributions, operators to construct expressions, and functions to evaluate the expressions. Size distributions encoded as a *SizeDistribution* composite data type. Composite data types combine multiple arrays into a single symbol for ease of use, facilitating faster experimental design and analysis. *SizeDistribution* consists of vectors of bin edges, bin midpoints, number concentration, log-normalized spectral density, and logarithmic bin widths. *SizeDistributions* are denoted in blackboard bold font (e.g., $\mathbb{n}$, $\mathbb{r}$, etc.). *SizeDistributions* are the building block of composable algebraic expressions through operators that evaluate to transformed *SizeDistributions*. For examples, $\mathbb{n}_1 + \mathbb{n}_2$ is the superposition of two size distributions and $f * \mathbb{n}$ is the uniform scaling of the concentration fields by factor f, $\mathbf{A} * \mathbb{n}$ is matrix multiplication of $\mathbf{A}$ and concentration fields of the size distribution, and $f \cdot \mathbb{n}$ is the uniform scaling of the diameter field of the size distribution by factor f, and $T \cdot \mathbb{n}$ is the elementwise scaling of the diameter field by factor T. (Note that the Petters (2018) used $T. \cdot \mathbb{n}$ is the elementwise scaling. The extra dot has which has been dropped to stay consistent with the current software implementation).

Functions are used to reduce expressions. Generic functions include, $\sum(f, m)$ evaluates the function $f(x)$ for $x = [1, \ldots, m]$ and sums the result. If $f(X)$ evaluates to a vector, the sum is the sum of the vectors. The function $\text{map}(f, x)$ applies $f(x)$ to each element of vector x and returns a vector of results in the same order. The function $\text{reduce}(f, x)$ applies the bivariate function $f(x, y)$ to each element of x and accumulates the result. The function $\text{mapreduce}(f, g, x)$ combines map and reduce. It applies function f to each element in x, and then reduces the result using the bivariate function function $g(x, y)$. The function $\text{vcat}(x, y)$ concatenates arrays x and y along one dimension. Anonymous functions are used as arguments to reducing functions. Anonymous functions are denoted as $x \rightarrow expression$, where x is the argument consumed in the evaluation of the $expression$. These functions are generic and represent widely used computing concepts. They are implemented in most modern programming languages.

DMA geometry, dimensions, and configuration are abstracted into composite types Λ (configuration comprising flow rates, power supply polarity, and thermodynamic state) and δ (DMA domain defined by a mobility/size grid). Each DMA is fully described by a pair Λ,δ. Subscripts and superscripts are used to distinguish between different configurations in chained DMA setups, e.g. δ_1 and δ_2 denoting the first and second DMA, respectively. Application of size distribution expressions to transfer functions constructs a concise model of the transmitted DMA mobility distribution, denoted as the DMA response function. Implementation of the language is distributed through a freely-available and independently documented package *DifferentialMobilityAnalyzers.jl*, written in the Julia language. Expressions in the text are provided in general mathematical form for readability.

Petters [2018] gives a simple expressions that model transfer through the DMA. The function $T_{size}^{\Lambda,\delta}(k, z^s)$ evaluates to a vector representing the fraction of particles carrying k charges that exit $\text{DMA}^{\Lambda,\delta}$ as a function of mobility

$$T_{size}^{\Lambda,\delta}(k, z^s) = \Omega(Z, z^s/k, k). * T_c(k, D_{p,1}). * T_l(Z, k) \tag{10}$$

where z^s is the centroid mobility selected by the DMA (determined by the voltage and DMA geometry), Z is a vector of mobilities, Ω is the diffusing DMA transfer function [Stolzenburg and McMurry, 2008], T_c is the charge frequency distribution [Wiedensohler, 1988], and T_l is the diameter-dependent transmission loss [Reineking and Porstendörfer, 1986]. The functions Ω and

T_l have been updated from Petters (2018). The version in Petters (2018) computed the shape of the transfer function and losses for the mobility diameter corresponding to singly charged particles and then apply the same shape of the transfer function and diffusional loss to the multiply charged particles. Binding the charge state explicitly to Ω and T_l results in proper accounting of diffusional losses and broadening of the transfer function for multiply charged particles in $T_{size}^{\Lambda,\delta}(k, z^s)$.

Petters [2018] also gives an expression that evaluates to the convolution matrix for passage through a single DMA.

$$\mathbf{A} = \text{mapreduce}\{z^s \rightarrow \Sigma[k \rightarrow T_{size}^{\Lambda,\delta}(k, z^s), m]^T, \text{vcat}, Z\} \tag{11}$$

where, m is the upper number of multiply charged particles, ^{T} is the transpose operator, and Z is a vector of centroid mobilities scanned by the DMA. Eq. (11) evaluates to the same as Eq. (8) in Petters (2018), but the notation is revised to be more general by removing the julia specific splatting construct and replacing it with widely used generic functions.

To help with parsing the expression, $T_{size}^{\Lambda,\delta}(k, z^s)$ evaluates to a vector of transmission for k charges and set point centroid mobility z^s as a function of the entire mobility grid (e.g. 120 bins discretized between mobility z_1 and z_2). The function $\Sigma[k \rightarrow T_{size}^{\Lambda,\delta}(k, z^s), m]$ superimposes the vectors for all charges. Mapping $z^s \rightarrow \Sigma[k \rightarrow T_{size}^{\Lambda,\delta}(k, z^s), m]$ over the mobility grid Z produces an array of vectors, each corresponding to the transmission for a single size bin. Transposing the vectors and reducing the collection through concatenation produces the design matrix that links the mobility size distribution to the response function, i.e.

$$\mathbb{r} = \mathbf{A}\mathbb{n} + \epsilon \tag{12}$$

where $\mathbb{r}$ is the response distribution, $\mathbb{n}$ is the true mobility size distribution, and ϵ is a vector denoting the random error that may be superimposed as a result of measurement uncertainties. Note that by design $\mathbb{n}$ and $\mathbb{r}$ are *SizeDistribution* objects, which represented the distribution as a histogram in both spectral density units (dN/dlnD) and concentration per bin units. The latter is the raw response function defined as integrated response downstream of the DMA as a function of upstream voltage (or corresponding z^s or corresponding apparent +1 mobility diameter).

The mobility distribution exiting DMA 2 in the humidified tandem DMA is evaluated using the expressions

$$\mathbb{M}_k^{\delta_1} = \Pi_k \cdot \left\{ g_0 \cdot \left[T_{size}^{\Lambda,\delta}(k, z^s) * \mathbb{n} \right] \right\} \tag{13}$$

In Eq. (13), $\mathbb{M}_k^{\delta_1}$ evaluates to the apparent +1 mobility distribution particles that exit the DMA$^{\Lambda,\delta}$ at the nominal setpoint-diameter defined by mobility z^s (or z-star) in DMA 1 and particle charge k. Subscripts are used to differentiate DMA 1 and 2 which possibly have different geometries, flow rates, and grids, e.g. Λ_1, Λ_2 and δ_1, δ_2. $\Pi_k^{\Lambda,\delta}$ is the projection of particles having physical diameter D and carrying k charges onto the apparent +1 mobility grid. It is a function that converts each diameter/charge pair to mobility and interprets the result as apparent +1 mobility diameter. $g_0 = D_{wet}/D_{dry}$ is the true diameter growth factor, D_{dry} is the selected diameter by DMA 1, D_{wet} is the diameter after the humidifier, $T_{size}^{\Lambda,\delta}(k, z^s)$ is as in Eq. (10), and $\mathbb{n}$ is the mobility size distribution upstream of DMA 1.

To help parse Eq. (13), the product $T_{size}^{\Lambda,\delta}(k, z^s) * \mathbb{n}$ evaluates to the transmitted mobility distributions of particles carrying k charges at the set-point mobility z^s in DMA 1. The size distribution is

grown by the growth factor g_0. The resulting size distribution is shifted to the apparent +1 mobility diameter using $\Pi_k^{\Lambda,\delta}$. Equation (13) differs from that in Petters [2018] where it was assumed that particles of all charges grow by the same amount. This is incorrect. Particles carrying more than a single charge alias at a smaller particle size [Gysel et al., 2009, Shen et al., 2021]. The effect is due to the size dependence of the slip-flow correction factor and captured through the function $\Pi_k^{\Lambda,\delta}$. Equation (13) assumes that g_0 applies to all particle sizes.

The total humidified mobility distribution $\mathbb{m}_t^{\delta_2}$ exiting DMA 2 is given by

$$\mathbb{m}_t^{\delta_2} = \sum_{k=1}^{m} \left(\mathbf{O}_k * \mathbb{M}_k^{\delta_1} \right) \tag{14}$$

where, m is upper number of charges on the multiply charged particles, Z is a vector of centroid mobilities scanned by DMA 2, and

$$\mathbf{O}_k = \text{mapreduce}\{z^s \rightarrow [\Omega^{\Lambda_2,\delta_2}(Z, z^s, k). * T_l^{\Lambda_2,\delta_2}(Z, k)]^T, \text{vcat}, Z\} \tag{15}$$

is the convolution matrix for transport through DMA 2 and particles carrying k charges. Equations (14) and (15) modified from those in Petters (2018) in the following manner. The convolution matrix $\mathbf{O}_k$ is computed individually for each charge. The version in Petters (2018) computed the matrix corresponding to singly charged particles and then apply the same matrix to multiply charged particles. Since $\mathbf{O}_k$ is now charge resolved, it is moved into the summation in Eq. (14). Computation of $\mathbf{O}_k$ through Eq. (15) has been revised to be more general by removing a julia language specific construct. $\mathbf{O}_1$ computed by Eq. (15) produces the same matrix as in Petters (2018).

If the aerosol is externally mixed, the humidified distribution function is given by

$$\mathbb{m}_t^{\delta_2} = \int_0^\infty P_g \left[\sum_{k=1}^{m} \left(\mathbf{O}_k * \mathbb{M}_k^{\delta_1} \right) \right] dg_0 \tag{16}$$

where P_g is the growth factor probability density function and the diameters in $\mathbb{M}_k^{\delta_1}$ are normalized by D_{dry}. $\mathbb{m}_t^{\delta_2}$ in Eq. (16) is the forward model through the tandem DMA. Using the notation in section 2.2,

$$F(\mathrm{x}, \mathrm{c}) = \int_0^\infty P_g \left[\sum_{k=1}^{m} \left(\mathbf{O}_k * \mathbb{M}_k^{\delta_1} \right) \right] dg_0 \tag{17}$$

where x is the true P_g and the vector c of constraining parameters comprises the DMA setup $\Lambda_1, \Lambda_2, \delta_1, \delta_2$ and upstream size distribution $\mathbb{n}$. Computer code that creates a forward model for tandem DMAs has been added to the *DifferentialMobiltyAnalyzers.jl* package and is annotated in the documentation of the package. For purposes of the forward model, the mobility grid for DMA 1 is discretized at a resolution of i bins. Transmission through DMA is computed for a specified z^s (the dry mobility), g_0 (the growth factor), and an input size distribution, which results in a vector i concentrations along this grid. If the input size distribution does not match the mobility grid the grids are merged through interpolation. The mobility grid for DMA 2 is discretized at a resolution of j bins. The transmitted and grown distribution from DMA 1 (i bins along the mobility axis of DMA 1) is interpolated onto the mobility grid of DMA 2. The outer integral in Eq. (17) is discretized into n bins that models P_g. If the output mobility of grid of DMA 2 does not match, the grids are merged through interpolation. The choice of i, j, n, the ranges of mobility grids for DMA 1, DMA 2, and the range of P_g is only constrained by computing resources and a physically

reasonable representation of the problem domain. Reasonable choices are $i = 120$, $j = n = 30$. The forward model is used to cast Eq. (17) into matrix form such that the humidified mobility distribution function is given by

$$\mathrm{m}_t^{\delta_2} = \mathbf{A}_2 P_g + \epsilon \tag{18}$$

where the subscript 2 specifies transmission through DMA 2, the matrix $\mathbf{A}_2$ is understood to be computed for a specific input aerosol size distribution, and ϵ is a vector that denotes the random error that may be superimposed as a result of measurement uncertainties. The size of $\mathbf{A}_2$ is $j \times n$. Uncertainties in the size distribution propagate into $\mathbf{A}_2$. The main influence of the error will be the relative fraction of +1, +2, and +3 charged particles. Assuming a random error of $\pm 20\%$ in concentration, the overall effect on the $\mathrm{m}_t^{\delta_2}$ is expected to small.

References

M.J. Cubison, H. Coe, and M. Gysel. A modified hygroscopic tandem DMA and a data retrieval method based on optimal estimation. *Journal of Aerosol Science*, 36(7):846–865, July 2005. ISSN 0021-8502. DOI: 10.1016/j.jaerosci.2004.11.009.

M. Gysel, G.B. McFiggans, and H. Coe. Inversion of tandem differential mobility analyser (TDMA) measurements. *Journal of Aerosol Science*, 40(2):134–151, February 2009. ISSN 0021-8502. DOI: 10.1016/j.jaerosci.2008.07.013.

Jingkun Jiang, Chungman Kim, Xiaoliang Wang, Mark R. Stolzenburg, Stanley L. Kaufman, Chaolong Qi, Gilmore J. Sem, Hiromu Sakurai, Naoya Hama, and Peter H. McMurry. Aerosol Charge Fractions Downstream of Six Bipolar Chargers: Effects of Ion Source, Source Activity, and Flowrate. *Aerosol Science and Technology*, 48(12):1207–1216, December 2014. ISSN 0278-6826. DOI: 10.1080/02786826.2014.976333.

Milind Kandlikar and Gurumurthy Ramachandran. Inverse Methods for Analysing Aerosol Spectrometer Measurements: A Critical Review. *Journal of Aerosol Science*, 30(4):413–437, 1999. ISSN 0021-8502. DOI: 10.1016/S0021-8502(98)00066-4.

E. O. Knutson and K. T. Whitby. Aerosol classification by electric mobility: Apparatus, theory, and applications. *Journal of Aerosol Science*, 6(6):443–451, 1975. ISSN 0021-8502. DOI: 10.1016/0021-8502(75)90060-9.

M. Ozon, A. Seppänen, J. P. Kaipio, and K. E. J. Lehtinen. Retrieval of process rate parameters in the general dynamic equation for aerosols using Bayesian state estimation: BAYROSOL1.0. *Geosci. Model Dev.*, 14(6):3715–3739, June 2021a. ISSN 1991-9603. DOI: 10.5194/gmd-14-3715-2021.

M. Ozon, D. Stolzenburg, L. Dada, A. Seppänen, and K. E. J. Lehtinen. Aerosol formation and growth rates from chamber experiments using Kalman smoothing. *Atmospheric Chemistry and Physics Discussions*, 2021:1–22, 2021b. DOI: 10.5194/acp-2021-99.

M. D. Petters. A language to simplify computation of differential mobility analyzer response functions. *Aerosol Science and Technology*, 52(12):1437–1451, December 2018. ISSN 0278-6826. DOI: 10.1080/02786826.2018.1530724.

D.J. Rader and P.H. McMurry. Application of the tandem differential mobility analyzer to studies of droplet growth or evaporation. *Journal of Aerosol Science*, 17(5):771–787, January 1986. ISSN 0021-8502. DOI: 10.1016/0021-8502(86)90031-5.

A. Reineking and J. Porstendörfer. Measurements of Particle Loss Functions in a Differential Mobility Analyzer (TSI, Model 3071) for Different Flow Rates. *Aerosol Science and Technology*, 5(4): 483–486, January 1986. ISSN 0278-6826. DOI: 10.1080/02786828608959112.

Lynn M. Russell, Shou-Hua Zhang, Richard C. Flagan, John H. Seinfeld, Mark R. Stolzenburg, and Robert Caldow. Radially Classified Aerosol Detector for Aircraft-Based Submicron Aerosol Measurements. *Journal of Atmospheric and Oceanic Technology*, 13(3):598–609, June 1996. ISSN 0739-0572. DOI: 10.1175/1520-0426(1996)013<0598:RCADFA>2.0.CO;2.

C. Shen, G. Zhao, and C. Zhao. Effects of multi-charge on aerosol hygroscopicity measurement by a HTDMA. *Atmospheric Measurement Techniques*, 14(2):1293–1301, 2021. DOI: 10.5194/amt-14-1293-2021.

Mark R. Stolzenburg and Peter H. McMurry. Equations Governing Single and Tandem DMA Configurations and a New Lognormal Approximation to the Transfer Function. *Aerosol Science and Technology*, 42(6):421–432, April 2008. ISSN 0278-6826. DOI: 10.1080/02786820802157823.

A. Voutilainen, V. Kolehmainen, and J. P. Kaipio. Statistical inversion of aerosol size measurement data. *Inverse Problems in Engineering*, 9(1):67–94, January 2001. ISSN 1068-2767. DOI: 10.1080/1741597010880277753.

Shih Chen Wang and Richard C. Flagan. Scanning Electrical Mobility Spectrometer. *Aerosol Science and Technology*, 13(2):230–240, January 1990. ISSN 0278-6826. DOI: 10.1080/02786829008959441.

A. Wiedensohler. An approximation of the bipolar charge distribution for particles in the submicron size range. *Journal of Aerosol Science*, 19(3):387–389, June 1988. ISSN 0021-8502. DOI: 10.1016/0021-8502(88)90278-9.

Shou-Hua Zhang, Yoshiaki Akutsu, Lynn M. Russell, Richard C. Flagan, and John H. Seinfeld. Radial Differential Mobility Analyzer. *Aerosol Science and Technology*, 23(3):357–372, January 1995. ISSN 0278-6826. DOI: 10.1080/02786829508965320.