

Response to Reviewer Comments

August 3, 2021

Author Statement

The manuscript has been reviewed by an anonymous referee and Mark Stolzenburg. I responded to these via AC₁ and AC₂ in July 1. As part of the response process, section 2.3 was significantly revised and the revised version was appended to AC₁ and AC₂. Dr. Stolzenburg contacted me via email and offered to send additional comments regarding Section 2.3 as written AC₁ and AC₂. He did so on July 10, 2021. Since I had not uploaded a revised version of the manuscript prior to receiving these comments, I responded to them online on August 3 via AC₃.

A once more revised version of section 2.3 is included at the end of AC₃. It supersedes the version presented in AC₁ and AC₂. The revised version submitted to AMT takes into account the comments in AC₁, AC₂, and AC₃. A tracked changes version is submitted alongside the revised manuscript.

The main changes to the manuscript are related to describing the tandem DMA model in Section 2.3, including clarifying the notation and underlying assumptions. The changes had no impact on the conclusions reached in the paper.

I thank the anonymous reviewer for their input. I am especially grateful for the detailed input by Mark Stolzenburg, whose comments were invaluable in clarifying the notation and the tandem DMA forward model. Line-by-line responses in AC₁, AC₂, and AC₃ are appended to this document.

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.

¹I 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 its 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

³I 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 works.

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.

⁴Please 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/smppsDAQ>) 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

⁷Since 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

⁸Related 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

¹⁰Line 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.

¹¹The matrix \mathbf{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 \mathbf{A} matrix need not to be square. The description for matrix \mathbf{A}_2 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..., \mathbf{A} is the design matrix (which may or may not be square), \mathbf{x} is the true quantity of interest, and \mathbf{f} 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 $\boldsymbol{\epsilon}$ 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

¹⁴Line 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

¹⁵Done.

¹⁵ Response

¹⁶The 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

¹⁷Line 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

¹⁸Equation (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

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

¹⁹ Revision

²⁰Line 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?

²¹Through 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://mdpeters.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

²⁴Line 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

²⁵The discretization/structure of the HTDMA inversion problem is now explained in more detail. Uncertainty in the size distribution will propagate into A_2 . This uncertainty is now mentioned in the manuscript.

²⁵ Response

²⁶For 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

$$m_t^{\delta_2} = A_2 P_g + \epsilon \quad (18)$$

where the subscript 2 specifies transmission through DMA 2, the matrix 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 A_2 is $j \times n$. Uncertainties in the size distribution propagate into 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 $m_t^{\delta_2}$ is expected to be small.

²⁷Line 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

²⁸I 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

²⁹Line 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. ³⁰For the HTDMA

²⁹ Referee

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).”

³⁰ Response

³¹The 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

³²Line 280: Continuing from above, do the choices for x_0 make sense given the chosen Tikhonov matrix? For example, a first-differences Tikhonov matrix encodes information about the expected slope of the solution. Using $(x - x_0)$ implies regularization of the slope of the residual with respect to an a priori estimate. Can the author comment?

³² Referee

³³With the exception of the of $L_0D_{1e-3}B_{[0,1]}$ 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 $L_2x_0B_{[0,1]}$ and $L_2B_{[0,1]}$ 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 $L_2x_0B_{[0,\infty]}$ 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

³⁴Second order inversion using $L_2B_{[0,\infty]}$ 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 $L_2x_0B_{[0,\infty]}$ is possible. However, noise in the a-priori propagates into the solution, thus negating the intended benefit of the second order Tikhonov matrix.**

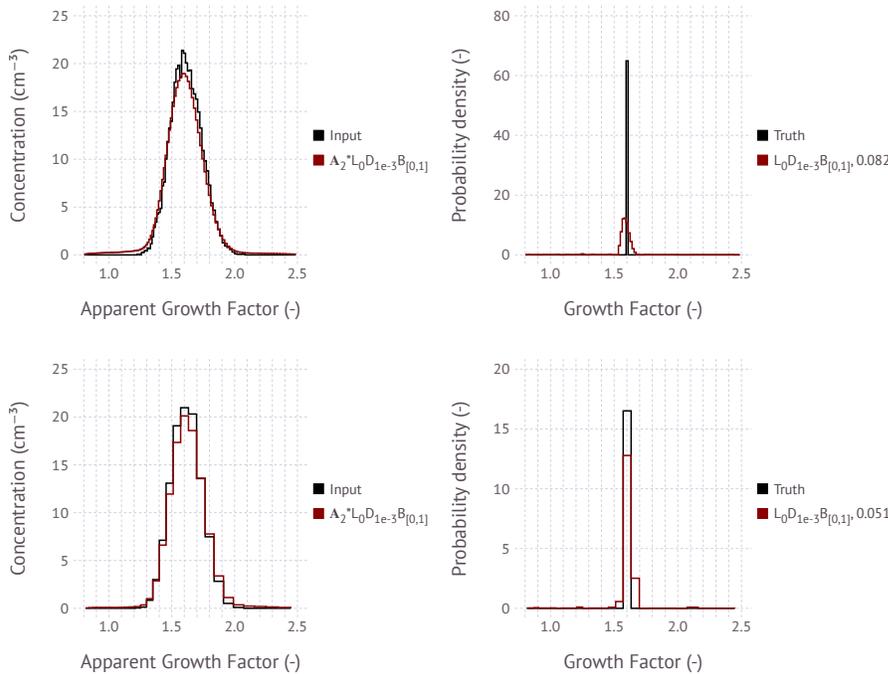
³⁴ Revision

³⁵With respect to, "Higher resolution grids generally lead to poor performance even for method $L_0D_{1e-3}B_{[0,1]}$." 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

³⁶The 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 an 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 then 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

³⁸Paragraph 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

⁴⁰Line 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].**

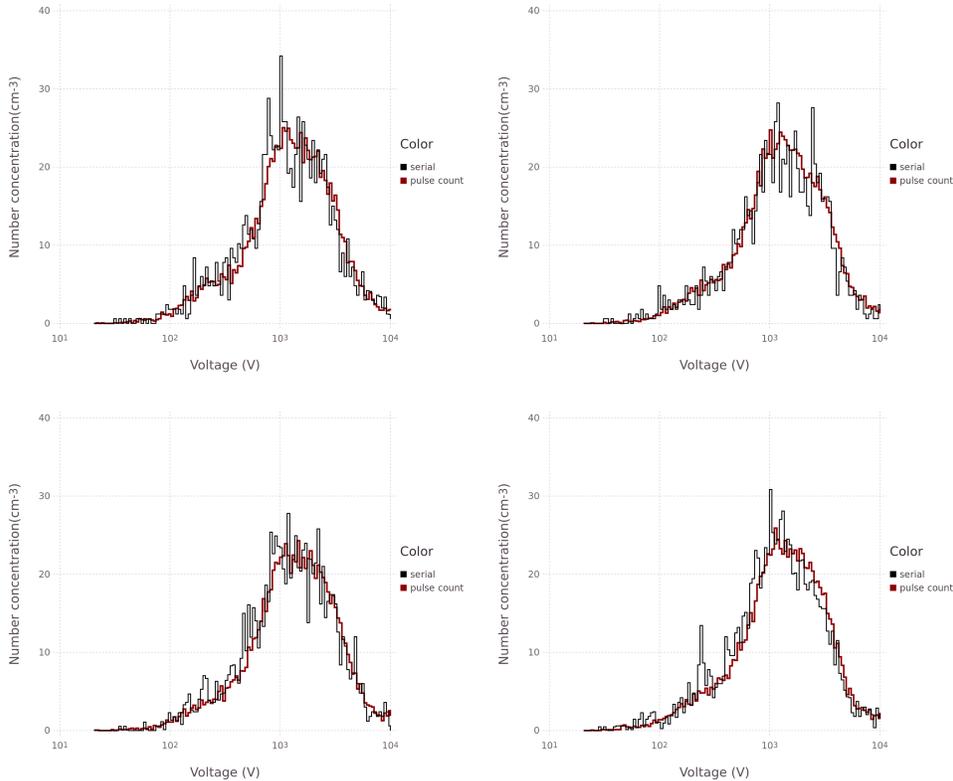
⁴² Revision

⁴³Figure 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 mentioned in the text.

⁴⁵The 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

⁴⁶For 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?

⁴⁷Thank 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

⁴⁸In 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

⁴⁹Line 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

⁵⁰Please 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 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, $\Sigma(f, m)$ evaluates the function $f(x)$ for $x = [1, \dots, m]$ 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 $x \rightarrow \text{expression}$, 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 DMA Λ, δ as a function of mobility

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

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.

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

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.

$$\mathbf{r} = \mathbf{A}\mathbf{n} + \epsilon \quad (12)$$

where \mathbf{r} is the response distribution, \mathbf{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 \mathbf{n} and \mathbf{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

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

In Eq. (13), $\mathbf{M}_k^{\delta_1}$ 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 \mathbf{n} is the mobility size distribution upstream of DMA 1.

To help parse Eq. (13), the product $T_{size}^{\Lambda,\delta}(k, z^s) * \mathbf{n}$ 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 $m_t^{\delta_2}$ exiting DMA 2 is given by

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

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

$$\mathbf{O}_k = \text{mapreduce}\{z^s \rightarrow [\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) * T_1^{\Lambda_2, \delta_2}(Z, k)]^T, \text{vcat}, Z\} \quad (15)$$

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

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

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

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

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 m . Computer code that creates a forward model for tandem DMAs has been added to the *DifferentialMobilityAnalyzers.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

$$m_i^{\delta_2} = \mathbf{A}_2 P_g + \epsilon \quad (18)$$

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 $m_i^{\delta_2}$ is expected to be 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: 10.1080/174159701088027753.
- 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.

Response to Reviewer Comments

July 1, 2021

Author Statement

I thank Dr. Stolzenburg for his time to review this manuscript and his 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 #2 (Mark Stolzenburg)

Overview

¹ Disclaimer: Other than just some broad principles, this reviewer is not familiar with regularization techniques or the Julia syntax and is therefore ill-equipped to properly review the technical nature of that aspect of this work. Attention is generally focused on other aspects of this paper. Also, the lack of full comprehensive documentation of all the notation used in the equations presented here has frequently hampered a thorough understanding of these equations. However, it is still possible to discern the general meaning of most equations. Equation (10) is a good example of this. The definition of the `map()` function and the interpretation of the right arrow (\rightarrow) are not given in the text here. At least the `map()` function is defined in the Petters (2018) reference. It appears the arrow notation is part of notation for a series or sequence.

¹ Referee

² *I apologize for the missing definitions in the draft. These are now included in the revised version.*

² Response

³ This manuscript addresses the important issue of automating the processing of tandem DMA data. The idea of inverting data with regularization is sound. However, there are problems with the forward model of calculating system response from a known input distribution. If these issues can be properly addressed, the resulting software package should prove of great utility.

³ Referee

⁴ *I thank Dr. Stolzenburg for his detailed and helpful review comments below. The issue raised regarding the forward model is addressed via a revision of the text and equations. The comments highlighted two assumptions that had no impact on the result, but were important to revise to be as correct as possible.*

⁴ Response

Major Comments

⁵ There appears to be a problem with proper accounting of diffusional losses and broadening of the transfer function, Ω , for DMA2. Eq. (10) in the form of \mathbf{A} characterizes the transfer through DMA1 while the equation for \mathbf{O} (line 230) characterizes transfer through DMA2. As noted in Petters [2018], these two expressions are analogous except for the inclusion of T_c in the former and the limitation of the summation to $k = 1$ in the latter. In the DMA, a particle is sized according to its apparent mobility diameter whereas diffusional losses as well as broadening of the transfer function are dependent on the true mobility diameter via particle diffusivity. Given one of these diameters, the particle charge is required to calculate the other and ultimately $T_{size}^{\Lambda, \delta}$. Thus, it is important to sum over all charge states individually to calculate the diffusing transfer through a DMA. As this is not done for the second DMA, the given expression cannot be properly accounting for transfer of multiply charged particles.

⁵ Referee

⁶*Thank you for the raising this issue. In fact, the issue affects both the matrix \mathbf{A} and the matrix \mathbf{O} . The version in Petters (2018) and the draft manuscript compute 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. The error that is introduced by this assumption/simplification is generally small since the fraction of multiply charged particles is small for sizes when diffusional broadening becomes important, and because the change in the shape of the transfer function/diffusional loss rate between the sizes is small. Nevertheless, there is no need to make this simplification. The formalism is now updated to properly account for the transfer of multiply charged particles.*

⁶ Response

⁷Section 2.3 of the manuscript has been revised to include this effect. Since this section includes changes in response to multiple other comments, please see the changed section at the end of this document for details.

⁷ Revision

⁸ The interpretation of Eq. (11) and its components would be greatly facilitated by an explicit indication of the independent parameters of distribution for the input size distribution n^{cn} . Also, the precise form of n^{cn} (e.g. dN/dDp , $dN/d\ln Dp$, or $dN/d\log Dp$) is important. The most obvious set of independent particle parameters would be (true) mobility diameter, D_1 , and charge, k . However, it appears that n^{cn} is distributed according to apparent mobility diameter, D_k , and k in order to have the balance of the equation work out. The apparent mobility diameter is then pre-multiplied by the effective, or apparent, growth factor, $gf_k(z^s, gf_0)$, and then by the ratio of true to apparent mobility diameters, D_1/D_k . However, this ratio is being evaluated at the DMA1 centroid mobility, z^s , but applied to the Z grid after growth. Since this ratio is a function of size, this does not work out. Also, this means that the input distribution to the operator characterizing DMA2 transfer is in terms of true mobility diameter, in contrast to the n^{cn} input to DMA1 and \mathbf{A} . All of this switching back and forth between true and apparent mobility diameter

⁸ Referee

seems overly complicated.

⁹The precise form \mathfrak{n}^{cn} is now clarified in the text just above the equation in question. It is a histogram in $dN/d\ln D$ units. The “cn” has been dropped based on a later comment. The distribution \mathfrak{n} is along the actual mobility diameter. The referee is correct that the way it was formulated in the draft was confusing due to multiple switches between true and apparent mobility diameter. The referee is also correct that the ratio was being evaluated at the DMA1 centroid mobility, z^s , but applied to the Z grid after growth. The assumption was that particles within a charge grouping all behave the same. I changed the equation and the code to make it more intuitive and more correct, i.e. when projecting the physically grown diameter back to mobility space, the correction is applied for each point in the Z grid. The impact on the calculation due to the change is almost imperceptibly. Text has been added to help parsing the equation.

⁹ Response

¹⁰

$$\mathfrak{r} = \mathbf{A}\mathfrak{n} + \epsilon \quad (12)$$

¹⁰ Revision

where \mathfrak{r} is the response distribution, \mathfrak{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 \mathfrak{n} and \mathfrak{r} are *SizeDistribution* objects, which represented the distribution as a histogram simultaneously as spectral density units ($dN/d\ln D$) 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

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

In Eq. (13), $\mathbb{M}_k^{\delta_1}$ evaluates to the apparent +1 mobility distribution particles that exit the DMA Λ, δ 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 \mathfrak{n} is the mobility size distribution upstream of DMA 1.

To help parse Eq. (13), the product $T_{size}^{\Lambda, \delta}(k, z^s) * \mathfrak{n}$ 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.

Minor Comments and Corrections

¹¹ line 158: Insert a space between “as” and “x”.

¹¹ Referee

¹²Done.

¹² Response

¹³ line 186: The description of a DMA here is a bit too brief, saying nothing about the flow. Try “Charged particles in a flow between the electrodes are deflected to an exit slit...”

¹³ Referee

¹⁴Thank you for the suggestion. Done.

¹⁴ Response

¹⁵ lines 188-189: “The functions ... and tandem DMAs is are well understood ...”

¹⁵ Referee

¹⁶Done.

¹⁶ Response

¹⁷line 200: “ $T \cdot n$ ” should be “ $T \cdot n$ ” according to Petters (2018). Presumably T is a vector, but this differs from the notation conventions given in lines 87-88.

¹⁷ Referee

¹⁸Yes, “ $T \cdot n$ ” should be “ $T \cdot n$ ” according to Petters (2018). (T is a vector). The original version was developed on julia v0.6 and it allowed me to create the “ $T \cdot n$ ” construct which was desirable to create a consistent treatment of vectors. Once julia updated to 1.x series, it was not longer possible to use this notation and I dropped the extra dot. This difference is now noted in the text.

¹⁸ Response

¹⁹See updated section 2.3

¹⁹ Revision

²⁰ Eq. (10): Here, $T_{size}^{\Lambda, \delta}$ alone characterizes transfer through the DMA. Evidently the balance of this expression puts this into the required form for later matrix manipulation. Some additional explanation of how this matrix is created from $T_{size}^{\Lambda, \delta}$ would be useful here. And though perhaps only parentheses may be used in programming, the readability of this equation would be greatly improved by alternating “()” with “[]” and “{ }”.

²⁰ Referee

²¹The equation(s) have been updated for readability by alternating “()” with “[]” and “{ }”

²¹ Response

}". The equation has also been rewritten for clarity by removing julia language specific constructs and giving much more details about the functions used in the text. As pointed out in Petters (2018) "It may not be immediately obvious why the expression ... evaluates to the convolution matrix (or that it evaluates to a matrix at all). A step-by-step explanation is in Notebook S2." The reference to the supplement of the preceding work is still valid. Nevertheless, some additional explanation has also been added to the text.

²² Please see entire updated section 2.3 for further context and definitions. The specific changes referring to this comment are

²² Revision

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

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

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 language specific splatting construct and replacing it with the widely used higher-order function `mapreduce` defined earlier.

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.

$$\mathbb{r} = \mathbf{A}\mathbb{n} + \epsilon \quad (12)$$

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.

²³ lines 213, 230: Though z^s is defined in lines 223-224, what is z_k^s in the indicated lines? Since z^s is used in Eq. (10), it might be more conveniently defined in line 211 (rather than line 224) along with Z as "... Z is a vector of centroid mobilities, z^s , scanned by the DMA..."

²³ Referee

²⁴The term z_k^s has been rewritten as z^s/k for clarity

²⁴ Response

²⁵ lines 216-217: “The size distribution after passage through the DMA is given by $r = \mathbf{A}n + \epsilon$, where r is the response function... .” The size distribution exiting the DMA and the response of the detector are not the same thing. The former is usually given as $dN/d\log Dp$ while the latter, as in the case of a CPC, is given by N_{CPC} , a simple number concentration. Also, there is the matter of the detector efficiency as well as the transport efficiency between the DMA and the detector, unless the latter has been subsumed into the DMA transport efficiency. As \mathbf{A} is to later serve as the operator corresponding to transfer through DMA₁ in a tandem DMA setup, \mathbf{A} must represent a size distribution, not a response function.

²⁵ Referee

²⁶ *It is both. Please see revisions to preceding comment which spells this out. Currently the detector efficiency is not treated separately, but it can be easily added by adding terms to $T_{size}^{\Lambda, \delta}(k, z^s)$.*

²⁶ Response

²⁷ Eq. (11) and following: The double character notation for growth factor as “gf” is atypical as far as normal mathematical notation is concerned. It is too easily interpreted as g times f, rather than as a single parameter. And in this draft of the manuscript there is actually extra space between the two letters, increasing the likelihood of the wrong interpretation. However, it is seen that this space is eliminated in Petters (2018) so presumably it can and will be eliminated in the final typeset form. If not for this preexisting work and a strong preference to remain consistent with that, it would be better to change this to a single character form such as simply “g”. Also, the reason for the choice of the cn superscript on n^{cn} for the input distribution is quite obscure. Does that stand for something?

²⁷ Referee

²⁸ *The origin for the “cn” superscript was carryover from the previous publication where it was desirable to distinguish the distribution that is measured with a CN counter as detector from a distribution that is measured with a CCN as detector. This way the true activated fraction could be conveniently modeled as $n^{cn} ./ n^{cn}$. The “cn” is not needed here and has been dropped; gf has been changed to g for clarity as suggested.*

²⁸ Response

²⁹line 230: Given the length and complexity of the expression for the operator \mathbf{O} , it would be better placed on a line by itself and numbered.

²⁹ Referee

³⁰ *Done*

³⁰ Response

³¹ **See updated section 2.3**

³¹ Revision

³² Discretization: As noted (lines 461-462), the forward model for the TDMA rep-

³² Referee

resents a triple integral. The parameters of integration may be denoted as D_i and D_o , the mobility diameters before and after growth, and gf_0 , the size-independent growth factor. Though the discretization of these parameters is automated in the software, some discussion of the constraints on this discretization should be included here. For instance, is there a restriction between the number of particle diameter bins and the number of measurement bins? Eq. (15) and the statement (line 248) “The size of \mathbf{A}_2 is n^2 , ...” would imply that the number of gf bins must be equal to the number of measurement bins. Is this a necessary condition and, if so, why?

³³Thank you for raising this. The under-the-hood binning procedure is now explained. The text reflected a choice I made when I wrote the code such that the measurement grid(s) can, but don't have to be interpolated onto any desired grid representation through interpolation (see below). It is worth pointing out here that the generic interface described in section 2.2 is designed such that the user can query the forward model at arbitrary points, which creates non-square matrices that link the grid of P_g and that of DMA 2 measurement representation.

³³ Response

³⁴Using the notation in section 2.2,

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

³⁴ Revision

where \mathbf{x} is the true P_g and the vector \mathbf{c} of constraining parameters comprises the DMA setup $\Lambda_1, \Lambda_2, \delta_1, \delta_2$ and upstream size distribution \mathbf{m} . Computer code that creates a forward model for tandem DMAs has been added to the *DifferentialMobilityAnalyzers.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

$$\mathbf{m}_t^{\delta_2} = \mathbf{A}_2 P_g + \epsilon \quad (18)$$

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 $j \times n$.

³⁵Figs. 1-4: Frequency vs. Growth Factor: Growth factor gf and its frequency distribution P_{gf} are naturally continuous functions, though the former is (artificially) discretized for the purposes of inversion. Just as the size distribution, n^{cn} , is explicitly written as dN/dDp or $dN/d\ln Dp$ with total integral N , the growth factor frequency distribution is also a derivative, dF/dg for $dF/d\ln gf$, with total integral $F = 1$. However, in the indicated plots, the frequency is plotted as for a parameter with truly discrete values such that the sum of the heights, rather than the areas, of the bars is equal to 1. That is, the height of each bar is given by $(dF/d\ln gf) \cdot \ln gf$ where $\ln gf$ is the width of the bar. If the growth factor is discretized such that $\ln gf$ is constant, then what is plotted is simply a uniformly scaled version of the more traditional $dF/d\ln gf$ plot, though this would normally be versus $\ln gf$. As plotted, the area under these curves is not equal to 1.

³⁵ Referee

³⁶Number Concentration vs. Apparent Growth Factor: In these plots, the Apparent Growth Factor is evidently given by

$$gf_{app} = D_1(z_2^s) / D_1(z_1^s).$$

The "Concentration" parameter is apparently the first-order inverted number distribution function given by

$$dN_{app}/d\ln D_{p2} = (dN_{app}/d\ln Z_{p2})(d\ln Z_p/d\ln D_{p2}) = (NCPC/\beta_2)(d\ln Z_p/d\ln D_{p2})$$

where $\beta_2 = Q_{aerosol}/Q_{sheath}$ for DMA 2. This is also seen to be a scaled version of the apparent growth factor frequency distribution as

$$dN_{app}/d\ln D_{p2} = N_{t,2} \cdot (dF_{app}/d\ln gf_{app})$$

where $N_{t,2}$ is the total concentration exiting DMA 2. If this is to be compared to the Frequency vs. Growth Factor plot, this would need to be multiplied by $\ln gf_{app} = \Delta \ln D_1(z_2^s)$. For the two plots to be directly comparable, $\Delta \ln D_1(z_2^s)$ would have to be a constant.

³⁶ Referee

³⁷Combined response to 35 and 36. In the submitted draft, the concentration is the raw number concentration the detector would measure for that bin. Neither the concentration nor the frequency histograms were normalized by the bin width. I changed the revised version to show the probability density functions such that the area under the curve equals to 1. I want to retain the number concentration vs. apparent growth factor plots since the values represent the measurement. I clarify the representation in the text.

³⁷ Response

³⁸Please see revised manuscript.

³⁸ Revision

³⁹line 315-316: "...the residual is high is if the true input is a broad growth factor frequency distribution..."

³⁹ Referee

⁴⁰Fixed

⁴⁰ Response

⁴¹lines 332-333: “Errors from scans with low non-zero concentration at the edge of the size distribution propagate back into the inversion at other dry sizes.” ⁴¹ Referee

⁴²*Fixed* ⁴² Response

⁴³line 345: “...a cylindrical DMA column (TSI 3080).” Model “3080” does not specify the actual DMA column. Assuming it is the TSI long DMA, this should be specified as either “TSI 3080L” for the whole system or “TSI 3081” for just the column. ⁴³ Referee

⁴⁴*Thank you, we have just the column. Corrected.* ⁴⁴ Response

⁴⁵lines 386-387: “...with the timestamp closest to the a scan...” Eliminate “a” ⁴⁵ Referee

⁴⁶*Corrected.* ⁴⁶ Response

⁴⁷line 419: “...a marine inflow event on March 27–28 2015.” Use a date format consistent with the other dates, i.e. 27-28 March 2015. However, this date is beyond the limits of the plot in Fig. 6. ⁴⁷ Referee

⁴⁸*The date formats are now consistent. Also, the text should have been 27-28 **February** 2015, which is on the plot.* ⁴⁸ Response

⁴⁹line 422: “...9 February 2015,...”. Shouldn’t this be 11 February 2015? ⁴⁹ Referee

⁵⁰*Thank you. Corrected.* ⁵⁰ Response

⁵¹Lines 505-513: “The inverted dataset ...closure (Mahish et al., 2018).” This is a very long run-on sentence. It needs to be broken up into several sentences. ⁵¹ Referee

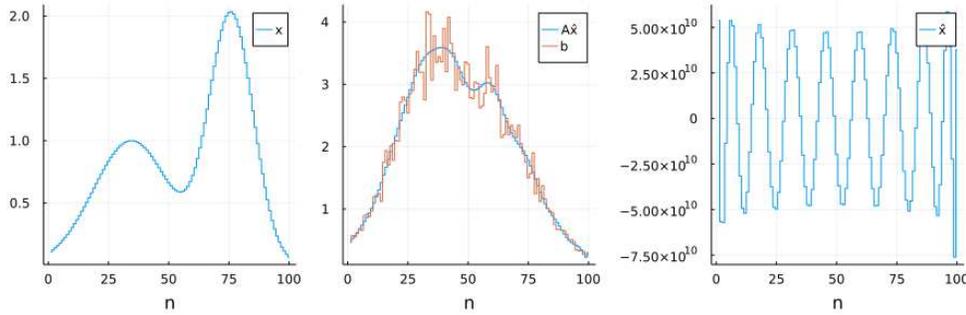
⁵²*Done.* ⁵² Response

⁵³“Best fit” vs “good fit”: Though regularization produces what might be considered a best fit solution to the inversion problem, this does not necessarily imply it is a good fit. It would be best to calculate a fit parameter such as the chi square of the normalized residuals over the degrees of freedom. For a good fit, this should be near 1. That is, the residuals are on the order of what is predicted by Poisson ⁵³ Referee

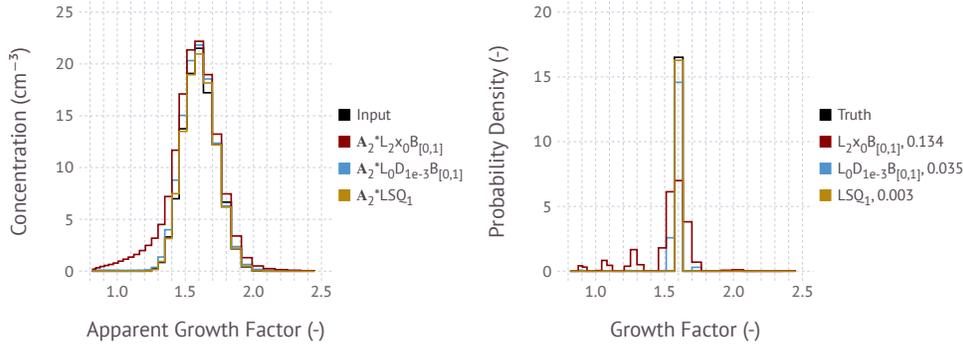
statistics. Values an order of magnitude or more greater than that would suggest some sort of problem either with the dataset or the model.

⁵⁴It is correct that the best-fit is not necessarily a good fit. Worse, even a good fit may be a poor model. Unregularized regression can almost eliminate the residual, but produce estimation parameters that are extremely poor, even if the regression looks good. For example:

⁵⁴ Response



The left panel shows a true input vector x . The middle panel on the left shows a response vector b (red) computed as $b = Ax + \epsilon$, where ϵ is some random error. The right panel shows the estimate $\hat{x} = A^{-1}b$, which is extremely poor. Computing the model response from the estimate, $A\hat{x}$, shows the best fit solution. If we only have the observations, we can only compute error metrics based on some residual between $A\hat{x}$ and b . The comment in question refers to some form of Figure 4



where the left panel corresponds to $A\hat{x}$ (colors) and b (Input) in the example above, and the right panel to the comparison between x (Truth) and \hat{x} (colors) in the example above. The question raised by this referee (and referee #1) is what the best error metric might be to determine the goodness of the fit and by extension and goodness of model reconstruction. I selected the root mean square error (RMSE), $RMSE = \sqrt{\sum \frac{(O_i - E_i)^2}{n}}$, where O_i are the observed and E_i the expected values. The RMSE is zero for a perfect fit and greater than zero for a less-than optimal fit.

Many goodness-of-fit statistical tests involve some form of the chi-square statistic. Computing chi-square as $\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$, where O_i are the observed and E_i the expected values is not valid for many of the cases here, because the expected values can be zero. Any residual in a bin with zero expected value would immediately raise χ^2 to infinity. Excluding bins with zero expected value would be incorrect, as it would not capture "bad" models that

predict output for zero bins. The referee's language "chi square of the normalized residuals over the degrees of freedom" seems to refer to the adjusted goodness-of-fit index, commonly abbreviated as AGFI [e.g. Sun, 2005], which for the same reason is not valid here.

The main desired property of a goodness-of-fit index is that it might provide a clean quantitative measure on when to reject a solution. The AGFI seemingly provides this information, with values near 1 indicating a good fit. Although the AGFI is not applicable here, the RMSE can be used in a similar manner. Values below a certain threshold indicate a good (or good enough) fit. The only difference is that the lower threshold value is not immediately clear.

The way RMSE is used here is in a relative comparison between $L_0D_{1e-3}B_{[0,1]}$, LSQ_1 , and LSQ_2 . LSQ_1 , and LSQ_2 are well-behaved and do not have oscillatory solutions. However, they will fail when true growth factor frequency distribution is broader than can be explained by one or two compounds. Conversely $L_0D_{1e-3}B_{[0,1]}$ will have poor solution (oscillatory solution) when the true input distribution is narrow. Truncation of the negative values is what amplifies the RMSE in this case. Thus RMSE is not quite used to declare that the fit is good or that a model is valid. It is used to determine whether the input distribution is narrow enough to warrant fit to a single component, two component, or multicomponent model.

This still leaves the ultimate question unaddressed. How well can we trust the proposed (regularized) solution? As I argue in the manuscript, the simulations address this point.

Since the true noise-free input growth factor frequency distribution is known, the fidelity of the inversion can be evaluated by computing the root mean square error between the noise-free solution and the regularized solution. The figure shows that both inversion methods produce a root mean square error between 0.02 and 0.03. These values are typical for the of reconstruction (see supporting information). Visual evaluation of the agreement between the reconstruction and the input suggest that either method is suitable for inversion.

Whether this is acceptable remains ultimately up to the user. I am skeptical that a statistical procedure such as AGFI (if it were applicable) would really help here. Tests should be performed to validate the physical plausibility of the solution. For examples, the mode of the apparent growth factor and the mode of the inverted growth factor should be similar. The retrieved growth factors should be physically plausible. The distribution of RMSE can be plotted for a large data set. Visual inspection of fits for large RMSE can be used to derive a threshold above which fits are automatically rejected. The text now mentions these quality assurance examples.

⁵⁵Note, however, that the low residuals between the apparent growth factor distribution and the model do not automatically ensure that the algorithm a good or adequate solution. Additional tests should be performed to validate the physical plausibility of the solution. For example, the retrieved growth factors should be physically plausible at the applied relative humidity. The mode of the apparent growth factor distribution and the mode of the inverted growth factor distribution should be similar. A histogram of the root mean square error between

⁵⁵ Revision

can be plotted for a large data set. Visual inspection of fits for large root mean square error can be used to derive a threshold above which reconstructions are automatically rejected. The integrated probability density function of the reconstructions should be near unity. Deviations from unity may occur due to concentration errors between the size distribution measurement and the growth factor distribution measurement, unaccounted transmission losses, and errors from the inversion. Reconstructions deviating significantly from unity should be flagged and rejected.

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 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, $\Sigma(f, m)$ evaluates the function $f(x)$ for $x = [1, \dots, m]$ 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 $x \rightarrow \text{expression}$, 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 DMA Λ, δ as a function of mobility

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

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.

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

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.

$$\mathbf{r} = \mathbf{A}\mathbf{n} + \epsilon \quad (12)$$

where \mathbf{r} is the response distribution, \mathbf{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 \mathbf{n} and \mathbf{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

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

In Eq. (13), $\mathbf{M}_k^{\delta_1}$ 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 \mathbf{n} is the mobility size distribution upstream of DMA 1.

To help parse Eq. (13), the product $T_{size}^{\Lambda,\delta}(k, z^s) * \mathbf{n}$ 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 $m_t^{\delta_2}$ exiting DMA 2 is given by

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

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

$$\mathbf{O}_k = \text{mapreduce}\{z^s \rightarrow [\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) * T_1^{\Lambda_2, \delta_2}(Z, k)]^T, \text{vcat}, Z\} \quad (15)$$

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

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

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

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

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 m . Computer code that creates a forward model for tandem DMAs has been added to the *DifferentialMobilityAnalyzers.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

$$m_i^{\delta_2} = \mathbf{A}_2 P_g + \epsilon \quad (18)$$

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 $m_i^{\delta_2}$ is expected to be small.

References

- 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. 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.
- Jun Sun. Assessing Goodness of Fit in Confirmatory Factor Analysis. *Measurement and Evaluation in Counseling and Development*, 37(4):240–256, January 2005. ISSN 07481756. DOI: 10.1080/07481756.2005.11909764.
- 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: 10.1080/174159701088027753.
- 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.

Response to Reviewer Comments

August 3, 2021

Author Statement

Dr. Stolzenburg contacted me via email and offered to send additional comments regarding Section 2.3 as written in my response to the original comments. He did so on July 10, 2021. I thank Dr. Stolzenburg for his additional effort (and very appreciated comments) to help me improve the clarity of the manuscript. Since I had not uploaded a revised version of the manuscript prior to receiving these comments, I respond to them here. The revised version of the manuscript that will be submitted to AMT will have taken these comments into account. A new version of section 2.3 is included at the end of this comment. It supersedes the version in AC1 and AC2.

Response to Reviewer #2 (Mark Stolzenburg)

The comments were sent in the form of an annotated pdf document. Below I transcribe the comments in the following form (1) Text passage that the comment refers to, (2) Verbatim annotated comment, (3) response to comment, (4) revisions. If the context is unclear from the text fragment, please see the full text on starting on pg. 13 of AC2 on the discussion site.

Overview

¹ 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.

¹ Text

²The proper verbiage used with integrals is as follows: $\int f(x)dx$ is the integral of $f(x)$ over x . $f(x)$ is called the integrand. The integrals here are actually over the upstream particle size, or some substitute for it, and the grown or downstream particle size, or some substitute for that. The functions you specify represent only part of the integrand in each case given here. Even if you change the "over"s to "of"s, the statement is still very misleading given the many unaccounted for factors missing in the integrands. An integral over "the aerosol conditioner function" makes little sense since it is a function of both upstream and downstream sizes.

² Referee

³ Thank you.

³ Response

⁴ **The integrated response downstream of a tandem DMA that is operated at voltages V_1 and V_2 requires solving integrals of the upstream particle size distribution over size and the grown particle size distribution over size. The integration must be repeated for each charge state.**

⁴ Revision

- ⁵The objective is to find find a design matrix that maps the growth factor frequency distribution to the raw TDMA response function. ⁵ Text
- ⁶ Perhaps it would be clearer to say "For the forward calculation, the objective is ...". Correct "find find". ⁶ Referee
- ⁷ *Thank you.* ⁷ Response
- ⁸ **For the forward calculation, the objective is to find a design matrix that maps the growth factor frequency distribution to the raw TDMA response function.** ⁸ Revision
- ⁹The resulting expressions are concise. They are easily identified within actual source code. ⁹ Text
- ¹⁰ For anyone familiar with the specific language used. ¹⁰ Referee
- ¹¹**The resulting expressions are concise. They are easily identified within actual source code when working through the examples provided with the package documentation.** ¹¹ Revision
- ¹²Size distributions encoded as a *SizeDistribution* composite data type. ¹² Text
- ¹³ This is not a sentence, there is no verb. "Size distribution are encoded .."? ¹³ Referee
- ¹⁴**Size distributions are represented as a histogram and internally stored in the form of the *SizeDistribution* composite data type.** ¹⁴ Revision
- ¹⁵Composite 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. ¹⁵ Text
- ¹⁶"**+1 mobility diameter bin edges, bin midpoints**". Since later many different "size" parameters are introduced and used, it is best to be specific here. As with the DMA transfer function, are corresponding +1 mobilities also part of this data type? ¹⁶ Referee
- ¹⁷*I agree with the suggestion. Yes, the mobility grid is also included. In practice arrays of centroid mobilities and mobility bin edges are created, from which the +1 mobility diameter is computed.* ¹⁷ Response

¹⁸Composite data types combine multiple arrays into a single symbol for ease of use, facilitating faster experimental design and analysis. The size distribution data type *SizeDistribution* includes vectors of the selected mobility bins considered by the DMA, +1 mobility diameter bin edges and +1 mobility diameter bin midpoints computed from the mobility grid, number concentration, log-normalized spectral density, and logarithmic bin widths. ¹⁸ Revision

¹⁹(Note that **the** Petters (2018) used $T \cdot \mathbf{n}$ **is** the elementwise scaling. The extra dot has **which has** been dropped to stay consistent with the current software implementation). ¹⁹ Text

²⁰ [The referee highlighted multiple places of dangling wording introduced during editing and made suggestions for improvements.] ²⁰ Referee

²¹ (Note that Petters (2018) used $T \cdot \mathbf{n}$ **as** the elementwise scaling. The extra dot has been dropped to stay consistent with the current software implementation). ²¹ Revision

²²Functions are used to **reduce** expressions. ²² Text

²³ Not all of the following reduce the dimension of the expression. Or are you talking about a reduction or compactness in notation? If so, use a different word to avoid confusion with later usage. ²³ Referee

²⁴Functions are used to **evaluate** expressions. ²⁴ Revision

²⁵ If $f(X)$ evaluates to a vector, the sum is the sum of the vectors. ²⁵ Text

²⁶Should this be $f(x)$ or are you intentionally using X in indicate a vector? ²⁶ Referee

²⁷ **If $f(x)$ evaluates to a vector, the sum is the sum of the vectors.** ²⁷ Revision

²⁸The function $map(f, x)$ applies $f(x)$ to each element of vector x and returns a vector of results in the same order. ²⁸ Text

²⁹Perhaps this should be (f, X) ? ²⁹ Referee

³⁰ *Based on the above, no change.*

³⁰ *Response*

³¹ The function $reduce(f, x)$ applies the bivariate function $f(x, y)$ to each element of x and accumulates the result.

³¹ *Text*

³² From this and what I have read online, my understanding is that $f(x, y)$ uses the result of its previous application and combines that with the next x value. Nearly every online example I can find only uses functions that treat x and y the same, e.g. f is given as simply "+" or "*" such that interchanging x and y has no effect. Since f can be a user-defined function, it need not be symmetric in x and y , e.g. $f(x, y) = x^2 - y$. The documentation does not make it clear whether such usage is allowed. However, if it is, the order of arguments in $f(x, y)$ matters, that is, which is the result of the previous operation and which is the new x value. From online examples, I have gotten the impression that x is the previous result and y is the new x value. This makes the limited documentation such as given here quite misleading. Given no additional information, the most natural assumption would be that the new x value is associated with the x in $f(x, y)$ and the previous result is associated with y . But this would be just the opposite of what is needed. At minimum, I would suggest writing this as " $f(y, x)$ where y is the result of the preceding operation". Also according to an online example at jhub.com/julia, note that the for the first operation, x_1 is the previous result and x_2 is the new x .

³² *Referee*

³³ *The referee is correct that reduce is not associative. The order of operation matters. The more general version of reduce(f,x) is foldl(f,x) and foldr(f,x), which guarantee left or right associativity. In regular Julia programs, reduce(f,x) = foldl(f,x). However, there is no guarantee made by the language and any applied parallelism could break the expression. Using strictly associative folds is more precise. The change has been made throughout. The definition has been clarified as suggested by the referee.*

³³ *Response*

³⁴ **The function $foldl(f, x)$ applies the bivariate function $f(a, x)$ to each element of x and accumulates the result, where a represents the accumulated value. For the first element in x , a is the neutral value. For example $foldl(-, [1, 2, 3])$ evaluates the function $-(a, x)$ and yields $1 - 2 - 3 = -4$. The function $mapfoldl(f, g, x)$ combines map and $foldl$.**

³⁴ *Revision*

³⁵

$$\mathbf{A} = \text{mapfoldl}\{z^s \rightarrow \Sigma[k \rightarrow T_{size}^{\Lambda, \delta}(k, z^s), m]^T, \text{vcats}, Z_s\} \quad (11)$$

³⁵ *Revision*

³⁶ It applies function f to each element in x , and then reduces the result using the bivariate function function $g(x, y)$.

³⁶ *Text*

³⁷ $g(y, x)$ or even $g(a, x)$ where a represents the accumulated value so far. See preceding note. ³⁷ Referee

³⁸ *Thanks for the suggestion.* ³⁸ Response

³⁹ **It applies function f to each element in x such that $y = f(x)$ and then reduces the result using the bivariate function $g(a, y)$ where a represents the accumulated value. For the first element $f(x)$, a is the neutral value. For example, `mapfoldl(sqrt, -, [4, 16, 64])` evaluates to `foldl(-, [2, 4, 8]) = 2 - 4 - 8 = -10`.** ³⁹ Revision

⁴⁰ The function $vcats(x, y)$ concatenates arrays x and y along **one** dimension. ⁴⁰ Text

⁴¹ For clarity, shouldn't this be "first dimension" according to online documentation? ⁴¹ Referee

⁴² **The function $vcats(x, y)$ concatenates arrays x and y along the first dimension in Julia. However, other programming languages may concatenate along a different dimension as definition of horizontal and vertical is arbitrary.** ⁴² Revision

⁴³ Petters (2018) gives a simple expressions that model transfer through the DMA. ⁴³ Text

⁴⁴ Either "a simple expression that models" or "simple expressions that model" with no preceding "a". ⁴⁴ Referee

⁴⁵ **Petters (2018) gives a simple expression that models transfer through the DMA.** ⁴⁵ Revision

⁴⁶ ⁴⁶ Text

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

⁴⁷ The given dependence of ω does not make sense to me. Let z be an element of Z . For the mobility passed in a basic transfer calculation, it is the value of z/z^s that matters. For the diffusion calculation, it is z/k that matters. z^s/k does not get used directly. Though what is shown may not be technically wrong, it obscures the true dependencies. Simply (Z, z^s, k) would be better, as in Eq. (15) for DMA 2. ⁴⁷ Referee

⁴⁸ *Note that revisions will be indicated after responding to the next few comment about this equation. The referee is correct that implementation of the function is $\Omega(Z, z^s, k)$. The charge state is needed to compute the diffusion coefficient. However, for the $T_{size}^{\Lambda, \delta}(k, z^s)$* ⁴⁸ Response

function z^s/k is the correct argument to produce the actually transmitted mobility size distribution. One way to think about this is to ask the question: what centroid mobility z^s would I have to set the DMA to transmit particles carrying k charges if they had only a single charge? Eq. (10) therefore represents the mobility size distribution transmitted. In Eq. (15) is $\Omega(Z, z^s, k)$, as all particles have the same mobility. This version allows using the same omega function in both cases.

49

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

49 Text

⁵⁰What is $D_{p,1}$? Apparently, it is the true particle mobility (not electrical mobility) diameter, $D_{p,1} = D_p(z, k=1)$, where z is an element of Z . If it is to be used, has it been defined elsewhere? Perhaps it is part of the definition of the size distribution composite data type. If so, it should be defined there. T_l (penetration efficiency, not loss) also depends on this diameter, $D_{p,1}$. Why is it not written that way? Otherwise, simply $T_c(Z, k)$.

50 Referee

⁵¹Are the dots necessary here in "·" or should they be dropped as in Eq. 13?

51 Referee

⁵²In Eq. (10), Z is a vector of mobilities -> "a vector of particle mobilities". As Z is apparently used for both particle mobilities as well as DMA centroid mobilities (see note below), it would be best to be clear to which it applies each time it is used.

52 Referee

⁵³ Loss has been changed to "penetration efficiency". Yes, the dots are still required for the * operator. They were only dropped for the · operator. The interpretation of $D_{p,1} = D_p(z, k = 1)$, where z is an element of Z is correct. T_l was written this way in the original draft and in Petters (2018), but changed during the first round of revisions. The change is reverted here to stay consistent with previous notation. I revised the text as follows to better explain how this equation works. The meaning of Z is also further clarified when discussing the discretization scheme near the end of the document.

53 Response

⁵⁴ Petters [2018] gives a simple expression 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 DMA ^{Λ, δ} as a function of mobility

54 Revision

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

where z^s is the centroid mobility selected by the DMA (determined by the voltage and DMA geometry), Z is a vector of particle 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 penetration efficiency [Reineking and Porstendörfer, 1986]. The diameter $D_{p,1} = D_p(z, k = 1)$, where z is an element of Z . The function Ω has been updated from Petters (2018). The version in Petters (2018) computed the shape of the transfer function corresponding to singly charged particles and then applied the same shape of

the transfer function and diffusional loss to the multiply charged particles. The functional Ω depends on three arguments $\Omega(Z, z^s, k)$ [Stolzenburg and McMurry, 2008]. The charge state is used to compute the diffusion coefficient and thus account for diffusional losses and broadening of the transfer function for multiply charged particles.

The output of $T_{size}^{\Lambda, \delta}(k, z^s)$ is the transmission of particles through the DMA in terms of the true particle mobility diameter. This is achieved by passing z^s/k as argument to Ω , which corresponds to the centroid mobility setting for the DMA to transmit particles with the size of particles with k charges under the assumption that they carry only a single charge. The net result is that $D_{p,1} = D_p(z, k = 1)$, where z is an element of Z becomes equal to the true mobility diameter axis. As a consequence the charge fraction $T_c(k, D_{p,1})$ and penetration efficiency $T_l(D_{p,1})$ are evaluated at the correct diameter. The function $T_{size}^{\Lambda, \delta}(1, z^s)$ evaluates to a vector of the same length as Z . Performing an elementwise sum over all $T_{size}^{\Lambda, \delta}(k, z^s)$ produces the net mobility distribution transmitted by the DMA. Examples for $T_{size}^{\Lambda, \delta}(1, z^s)$, $T_{size}^{\Lambda, \delta}(2, z^s)$, and $T_{size}^{\Lambda, \delta}(3, z^s)$ is shown in Figure 2, right panel in Petters (2018). Note that Eq. (10) can be evaluated using arbitrarily discretized Z vectors.

55 corresponding to singly charged particles and then **apply** the same shape of the transfer function

55 Text

56 "applied"

56 Referee

57 **fixed in the revised paragraph above.**

57 Revision

58 Petters (2018) also gives an expression that evaluates to the convolution matrix for passage through a single DMA.

58 Text

59 Summation over k means the information on particle physical diameter of multiply charged particles is

59 Referee

lost. Eq. (11) will not work for the first DMA in a TDMA setup using diffusing transfer functions or other diffusion effects in the second DMA. Something should be included here to indicate that, or that this expression is only for something like an SMPS system. I feel that "single DMA" is just not sufficient. The reference to "apparent +1 mobility diameter" at the end of this topic is useful in making this point but does not really make the point of "only apparent diameter, not physical" and comes far too late to make the point in question.

60 *Correct. The information is now explicitly included*

60 Response

⁶¹ Petters (2018) also gives an expression that evaluates to the convolution matrix for passage through a single DMA that is valid in the context of size distribution measurement system, e.g. SMPS. Since the expression includes a summation over all charges, the information on particle physical diameter of multiply charged particles is lost.

⁶¹ Revision

⁶²

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

⁶² Text

⁶³I feel it would be cleaner to first define a convolution matrix for transport through a DMA, with no summation over k, as in Eq. 15, perhaps O_{kv} or better yet $O_v(k)$ where v (actually nu) is the DMA. This could then be used for DMA 2 as well below, using an analogous definition of T_{size} . Here, for use as the convolution matrix through a single DMA, $A_v = \text{sum}_k(O_v(k))$. Otherwise, in the current development of the matrices, the similarity of treatment for DMA 1 and DMA 2 is buried. At the very least, though hardly preferable, Eq. 11 should be rearranged to put the summation as the outermost operation on the right side. This would at least provide a little more symmetry of treatment of the two DMAs.

⁶³ Referee

⁶⁴ I do not fully disagree with the referee about the potential elegance of making the expressions more symmetric. However, the matrices \mathbf{A} and \mathbf{O} serve two separate purposes. The former is valid for passage through a single DMA that is valid in the context of size distribution measurement system (which is now clarified), the latter is valid for evaluating the response after passage through the tandem DMA, starting with the transmitted distribution. The expressions for these are well-defined. It should be clear by now that the approach is in principle highly expressive. The version here, or the version in Petters (2018) are two examples how to write \mathbf{A} . More ways certainly exist, perhaps even more elegant ways. In the context of this work, I believe that this approach will lead to further distance the expressions from the original work and in the end result in more confusion rather than additional clarity.

⁶⁴ Response

⁶⁵ ... and Z is a vector of centroid mobilities scanned by the DMA.

⁶⁵ Text

⁶⁶ $Z=Z_s$ (see following note) implies \mathbf{A} is a square matrix. Do you really want to introduce that restriction at this point?

⁶⁶ Referee

⁶⁷Eq. 10 used Z as a vector of particle mobilities. Are these two vectors identical, that is, the DMA centroid mobilities and input particle mobility bin midpoints? If so, that should be explicitly noted to avoid confusion. Better yet would be to change Z here to Z_s noting somewhere previously that $Z=Z_s$.

⁶⁷ Referee

68 *Although I have been using the same Z vectors in Eq. (10) and (11) - and thus square matrices - this is not a necessary restriction. This is also further clarified when discussing the discretization scheme.* 68 Response

69 **... and Z_s is a vector of centroid mobilities scanned by the DMA. The matrix is square if $Z_s = Z$ in Eq. 10. However, this is not a necessary restriction.** 69 Revision

70 *...removing the julia specific splatting...* 70 Text

71 *Julia documentation uses this as capitalized, "Julia". Also, this should be hyphenated as "Julia-specific".* 71 Referee

72 *Changed capitalization here and throughout the work.* 72 Response

73 **...removing the Julia-specific splatting...** 73 Revision

74 *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).* 74 Text

75 *So Tsize is actually a matrix of size $[n \times m]$ where $n=120$ is the number of particle mobility bins. As the above equations are defined it would be useful to clearly indicate the dimensions of the operands and the result.* 75 Referee

76 *"entire particle mobility grid". Though Z serves a dual purpose, in this context it is particle mobility.* 76 Referee

77 *$T_{size}^{\Lambda, \delta}(1, z^s)$ evaluates to a vector of length Z; $T_{size}^{\Lambda, \delta}(2, z^s)$ evaluates to a vector of length Z. The text prior to this paragraph is updated to better explain the origin and dimensionality $T_{size}^{\Lambda, \delta}(k, z^s)$.* 77 Response

78 **See revisions to previous comments.** 78 Revision

79 *Note that by design \mathfrak{n} and \mathfrak{r} are *SizeDistribution* objects, which represented the distribution as a histogram in both spectral density units (dN/dlnD) and concentration per bin units.* 79 Text

80 *The input distribution, (blackboard "n"), is readily defined in terms of the true mobility ($k=1$) diameter, D_{p1} . However, information on the original charge state exciting the charger is lost from the response distribution (blackboard "r") leaving* 80 Referee

only electrical mobility, z , as the size parameter. Thus, the response distribution cannot accurately be converted to $dN/d\ln D$. Any such conversion must neglect the true charge distribution.

⁸¹ *That is correct. Nonetheless the mathematical/computational representation is that of a SizeDistribution object. The limitation is now noted.*

⁸¹ Response

⁸² **By design \mathfrak{n} and \mathfrak{r} are *SizeDistribution* objects, which represent the distribution as a histogram in both spectral density units ($dN/d\ln D$) 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 apparent +1 mobility diameter but not true physical diameter for multiply charged particles). Note, however, that the response function is not a true particle size distribution in the scientific sense since information about multiply charged particles is lost. The representation of \mathfrak{r} as *SizeDistribution* object is to allow response functions to be used in the expression-based framework used here.**

⁸² Revision

⁸³ 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).

⁸³ Text

⁸⁴ The first "corresponding" is sufficient for both, delete second "corresponding". Perhaps one could add here ", but not true physical diameter for multiply charged particles". Just a thought as to how to make this point clear.

⁸⁴ Referee

⁸⁵ *Thank you for the suggestion.*

⁸⁵ Response

⁸⁶ **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 apparent +1 mobility diameter but not true physical diameter for multiply charged particles).**

⁸⁶ Revision

⁸⁷

⁸⁷ Text

$$M_k^{\delta_1} = \Pi_k \cdot \left\{ g_0 \cdot \left[T_{size}^{\Lambda, \delta}(k, z^s) * \mathfrak{n} \right] \right\} \quad (13)$$

In Eq. (13), $M_k^{\delta_1}$ evaluates to the apparent +1 mobility distribution particles that exit the DMA ^{Λ, δ} at the nominal setpoint-diameter defined by mobility z^s (or z -star) in DMA 1 and particle charge k .

⁸⁸ Π_k : Given the appending of superscripts to this function in the following text, it

⁸⁸ Referee

should appear the same here with (capital lambda)₁

⁸⁹ $T_{size}^{\Lambda,\delta}(k, z^s)$: The superscripts should have subscript 1 corresponding to the sub-
script 1 of the left side of the equation.

⁸⁹ Referee

⁹⁰You say that this is for DMA 1 but there should be an index/subscript of 1 on the
superscripts of the first "DMA" in this sentence and of Tsize in the above equation
to indicate this. Otherwise, there is no clear relationship between (delta)₁ on the
left side of the equation and (delta) (no subscript) on the right side. And (capital
lambda)₁ should also have a subscript of 1.

⁹⁰ Referee

⁹¹ DMA 1 and 2 which possibly have different geometries, flow rates, and grids,
e.g. Λ_1, Λ_2 and δ_1, δ_2 .

⁹¹ Text

⁹²You introduce this notation here but then fail to properly apply it in Eq. (13) and
following.

⁹² Referee

⁹³ *Thank you for pointing this out. Superscripts have been added in the appropriate places.*

⁹³ Response

⁹⁴ $\Pi_k^{\Lambda,\delta}$ is the projection of particles having physical diameter D and carrying k
charges onto the apparent +1 mobility grid.

⁹⁴ Text

⁹⁵This function needs to interpolate the diameters of the grown particles onto the
+1 mobility diameter grid corresponding to Z . From your description, it sounds
like multiply charged particles are mis-sized to smaller diameters. If this function
is used with storing the apparent diameter, rather than true diameter, in the distri-
bution, then there would have to be a later function just to undo that before use in
Eq. 14.

⁹⁵ Referee

⁹⁶This function involves both calculation of apparent mobility but also interpolation
of the grown bin sizes from Z back onto Z . This interpolation step is important as
it affects the propagation of the random error of the input distribution. As such, it
should be explicit noted here. In general, any interpolation of a noisy distribution
or decedents there of will tend to reduce the overall noise level, but not, I believe,
in a very predictable way.

⁹⁶ Referee

⁹⁷ $\Pi_k^{\Lambda,\delta}$ calculates the apparent +1 diameter of a particle that carries multiple charges. The
implementation is quite simple. Start with a physical size e.g. 100 nm. Next compute
the mobility of that particle given it's charge state. Next, reinterpret that mobility as if
the particle were to carry only a single charge. Finally divide this by the initial diameter
to get the projection. Obviously $\Pi_1^{\Lambda,\delta} = 1$. Less obviously $\Pi_2^{\Lambda,\delta} < 1$. $\Pi_k^{\Lambda,\delta}$ does not
do any interpolation. In the expression, $\left\{ g_0 \cdot \left[T_{size}^{\Lambda,\delta}(k, z^s) * n \right] \right\}$ evaluates to the grown
transmitted size distribution. $\Pi_k \cdot \left\{ g_0 \cdot \left[T_{size}^{\Lambda,\delta}(k, z^s) * n \right] \right\}$ evaluates to the apparent size

⁹⁷ Response

distribution. The \cdot operator is what shifts the sizes. If a misfit occurs during the shift (which invariably happens), the result is interpolated onto the original size grid of n . The role of interpolation is now discussed together with the discretization scheme (see further below).

⁹⁸ Also, this function does NOT depend on the DMA configuration (capital lambda). It does depend on (delta), including the subscript 1.

⁹⁸ Referee

⁹⁹ It does depend on the DMA configuration, which includes temperature and pressure, because the conversion from mobility to diameter, and hence the projection is temperature and pressure dependent.

⁹⁹ Response

¹⁰⁰ ...humidifier, $T_{size}^{\Lambda, \delta}(k, z^s)$ is as in Eq. (10), and n is the mobility size distribution upstream of DMA 1.

¹⁰⁰ Text

To help parse Eq. (13), the product $T_{size}^{\Lambda, \delta}(k, z^s) * n$ evaluates to...

¹⁰¹ Superscripts should have subscripts 1.

¹⁰¹ Referee

¹⁰² Fixed

¹⁰² Response

¹⁰³ 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.

¹⁰³ Text

¹⁰⁴ This does not make sense unless g_0 is a function of particle size, or of z^s and k . If this is the case, it has not been made clear. Otherwise, it seems that both here and in Petters it is assumed that particles of all charges grow by the same amount, g_0 , a constant.

¹⁰⁴ Referee

¹⁰⁵ Yes, here and in Petters it is assumed that particles of all charges grow by the same amount, g_0 , a constant. It is said explicitly in "Equation (13) assumes that g_0 applies to all particle sizes." The initial wording was rather poor and the text is revised.

¹⁰⁵ Response

¹⁰⁶ Equation (13) differs from that in Petters [2018] where it was assumed **that the apparent growth factor for particles carrying multiple charges is the same as for single charged particles**. 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

¹⁰⁶ Revision

through the function $\Pi_k^{\Lambda, \delta}$. Equation (13) assumes that g_0 applies to all particle sizes.

107

$$\mathbf{O}_k = \text{mapreduce}\{z^s \rightarrow [\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) * T_1^{\Lambda_2, \delta_2}(Z, k)]^T, \text{vcat}, Z\} \quad (15)$$

107 Text

¹⁰⁸ z^s has been used as the centroid mobility for both DMAs. These need to be distinguished as two different parameters. Use subscripts "1" and "2" on these as appropriate to avoid confusion, e.g. here and Eq. 13. There are others in the text as well.

108 Referee

¹⁰⁹Is the range of the DMA 2 scan really the same as the range of the input distribution when the former uses 4 times fewer bins? The same Z is used for both.

109 Referee

¹¹⁰ I clarified Z as Z_s (same as your earlier comment). Subscript has been added as suggested. The range in Z is arbitrary. Clarification about how the various Z and Z_s relate to the discretization has been added to the text (see further below).

110 Response

¹¹¹**Please see revised section.**

111 Revision

¹¹² Equations (14) and (15) modified from those in Petters (2018)

112 Text

¹¹³"are modified" or "have been modified"

113 Referee

¹¹⁴ **Equations (14) and (15) have been modified from those in Petters (2018)**

114 Revision

¹¹⁵ ... matrix corresponding to singly charged particles and then apply the same matrix ...

115 Text

¹¹⁶Referee suggested: "applied"

116 Referee

¹¹⁷ ... matrix corresponding to singly charged particles and then applied the same matrix ...

117 Revision

¹¹⁸ If the aerosol is externally mixed, the humidified distribution function is given by ...

118 Text

¹¹⁹"function exiting DMA 2". Otherwise, it sounds like the distribution entering

119 Referee

DMA 2, right after humidification.

¹²⁰ **If the aerosol is externally mixed, the humidified distribution function exiting DMA 2 is given by ...** ¹²⁰ Revision

¹²¹ ... where P_g is the growth factor probability density function ... ¹²¹ Text

¹²² This still doesn't say anything about the growth factor being a function of dry diameter. So is P_g the same for all dry particle sizes, including those of multiply charged particles? ¹²² Referee

¹²³ Yes, it is. There is explicit discussion later in the text on how to potentially relax this assumption by using 2D inversions. ¹²³ Response

The following few comments are given without direct response. They all relate to the discretization of the grid and the underlying interpolation scheme. In response (see below) the paragraph was revised to clarify. The individual comments referenced here have been taken in to account.

¹²⁴ 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 ¹²⁴ Text

¹²⁵ This is not clear - "DMA 2", "DMA1" or "both DMAs"? From the following text, it would appear that what is needed and being described is the "transmission through DMA1 and subsequent growth". That is, up to the point of entering DMA 2. Note, simply changing "DMA" to "DMA1" does not work as DMA 1 transmission does not depend on g_0 . ¹²⁵ Referee

¹²⁶ If the input size distribution does not match the mobility grid the grids are merged through interpolation. ¹²⁶ Text

¹²⁷ Presumably, the input size distribution bins are interpolated onto diameter bins corresponding to the Z bins. Saying they are "merged" is ambiguous as to which is interpolated onto the other. ¹²⁷ Referee

¹²⁸ 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. ¹²⁸ Text

¹²⁹ It seems to me the DMA 2 mobility grid must be dynamically set according to how much growth there is. This would be good to note here. Wouldn't it be simpler ¹²⁹ Referee

to just have the DMA 2 grid be a subset of the DMA 1 grid? However, if the setups of the two DMAs are such that they do not have the same non-diffusing resolution, Q_a/Q_{sh} , then perhaps it would be better to use grids of different resolutions. As noted before, interpolations tend to smooth the data, thereby confounding the error analysis.

¹³⁰ *First, in this framework, the DMA 2 mobility grid is arbitrary and fixed. Interpolation is used throughout. The effect is factored into the framework through the size distribution operators, specifically the \cdot operator. Potential smoothing effects are factored into the framework through the numerical tests.*

¹³⁰ Response

*The reason for using interpolation throughout is to ensure generality of the approach. For example, the size distribution used in the forward model may come from a separate SMPS system (or even a model) that comes with binning that is not necessarily known ahead of time. The way we configured our TDMA is to set the voltage/size in DMA 1 denoted as D_d and then perform an SMPS scan over the range, for example, $0.7 * D_d$ to $2.5 * D_d$ over 60 s. The flow ratios in DMA 1 and DMA can differ, though we usually keep both 5:1. The bins are constructed as a geometrically stepped mobility grid between the lower and upper range. The only information about DMA is that of the nominal diameter.*

There might be clever ways to select the bins in DMA 2 to be a subset of DMA 1, but this does not quite obviate the need for interpolation in the forward model, unless one also matches the allowable growth factor to the discrete bin values. Even then, the growth for particles that carry multiple charges the corresponding mobility will not match that of the discrete binning. The paragraph on discretization includes now text explaining where interpolation is necessary, and where it can be avoided.

¹³¹ Reasonable choices are $i = 120$, $j = n = 30$.

¹³¹ Text

¹³² The transitions between mobility and diameter with their different natural binning make it difficult to minimize the problems of unpredictable smoothing by interpolations. It seems little can be done about that unless you are willing to set up a universal scale throughout based on either mobility or mobility diameter. Then bin midpoints could be translated to the other parameter and from there to all other scales. Each scale should either match the universal resolution and e midpoints or use an integer multiple of the resolution (e.g. each midpoint in one scale matches every third midpoint in another scale). If this approach were used, it would seem best to use a universal scale with uniform increments in either $\ln(dp)$ or $\ln(z)$ as appropriate. I believe that if the universal scale matched that of P_g and with uniform increments in $\ln(g)$ or $\ln(dp)$, then grown particles from one bin would land exactly into another bin with no interpolation or fractional bin calculation required.

¹³² Referee

¹³³ *Please see response to previous comment and text below. Interpolation is deeply interwoven in into this framework. Since the binning along all dimensions is arbitrary, it is possible to setup a universal (or near universal) grid in which interpolation is minimized, which is*

¹³³ Response

now mentioned. The paragraph describing the discretization is revised as follows.

¹³⁴ For purposes of the forward model, the mobility grid for DMA 1 is discretized at a resolution of i bins by specifying the Z vector in Eq. (10). If the Z vector does not match that of the aerosol size distribution \mathfrak{m} , the size distribution bins are interpolated onto the diameter bins corresponding to the Z bins. Transmission through DMA 1 is computed for a specified z^s (the dry mobility) and g_0 (the growth factor) via Eq. (13). The resulting $\mathbb{M}_k^{\delta_1}$ lie on the same Z grid with i bins. Any mismatches between the apparent growth factor and the underlying Z grid are resolved via interpolation implicit in the \cdot operator. ($f \cdot \mathfrak{m}$ is the uniform scaling of the diameter field of the size distribution by factor f . If the resulting diameters are off the original diameter grid, the result is interpolated onto the grid defined within \mathfrak{m})

¹³⁴ Revision

The mobility grid for DMA 2 is represented by the vector $Z_{s,2}$ in Eq. (15) and discretized at a resolution of j bins over a custom mobility range. If the vector Z inside the square bracket of Eq. (15), $[\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) \cdot T_l^{\Lambda_2, \delta_2}(D_{p,1})]$ equals that of DMA 1, the matrix is non square. The product $\mathbb{O}_k \cdot \mathbb{M}_k^{\delta_1}$ will map the i bins from DMA 1 to the j bins in DMA 2. Alternatively, if the Z vector inside the square bracket of Eq. (15) is taken to be equal to $Z_{s,2}$, the matrices \mathbb{O}_k are square. In that case, 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 prior to evaluating $\mathbb{O}_k \cdot \mathbb{M}_k^{\delta_1}$. The advantage of interpolation is that the matrices \mathbb{O}_k are smaller.

The forward model, defined by Eq. (14) can be evaluated for arbitrary g_0 values. Thus the growth factor probability distribution P_g in Eq. (17) can be discretized into n arbitrary growth factor bins. A natural choice is to accept growth factor values that coincide with the mobility grid of DMA 2, i.e. the bins align with $g = D_{p,1}/D_d$, where D_d is the nominal diameter selected by DMA 1 and $D_{p,1} = D_p(z, k = 1)$ and z is an element of $Z_{s,2}$. However, this is not required for evaluating Eq. (17). Equation (17) is cast into matrix form such that the humidified mobility distribution function is given by

$$\mathfrak{m}_t^{\delta_2} = \mathbf{B}P_g + \epsilon \quad (18)$$

where the matrix \mathbf{B} 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. If the grids for P_g and that of DMA 2 do not align, interpolation is used to map the P_g grid onto the DMA 2 grid. 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 size of \mathbf{B} is $j \times n$. Uncertainties in the size distribution propagate into \mathbf{B} . 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 $\mathfrak{m}_t^{\delta_2}$ is expected to be small.

Note that interpolation is widely used in this framework. Interpolation may affect how errors propagate through the model. Interpolation in Eq. (13) is unavoidable. However, interpolation can be minimized by working with non-square O_k and matching the grid of P_g to that of DMA 2. Informal tests working with different binning schemes suggests that the influence of interpolation choices on the final result is smaller than typical experimental errors.

135

$$m_t^{\delta^2} = \mathbf{A}_2 P_g + \epsilon \quad (18)$$

135 Text

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

¹³⁶The definition of this \mathbf{A} is not analogous to that in Eq. 11; this one subsumes the input distribution $Mk(\delta)_1$. To avoid confusion, use a different letter.

136 Referee

¹³⁷ Done - see above.

137 Response

¹³⁸ is expected to small.

138 Text

¹³⁹ Done - see above.

139 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 requires solving integrals of the upstream particle size distribution over size and the grown particle size distribution over size. The integration must be repeated for each charge state. Scanning over a range of voltages V_2 results in the raw TDMA response function. For the forward calculation, the objective is to find a design matrix that maps the growth factor frequency distribution to the raw TDMA response function. The objective is to 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 when working through the examples provided with the package documentation. 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 are represented as a histogram and internally stored in the form of the *SizeDistribution* composite data type. Composite data types combine multiple arrays into a single symbol for ease of use, thus facilitating faster experimental design and analysis. The size distribution data type *SizeDistribution* includes vectors of the selected mobility bins considered by the DMA, +1 mobility diameter bin edges and +1 mobility diameter bin midpoints computed from the mobility grid, 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 Petters (2018) used $T \cdot \mathbb{n}$ as the elementwise scaling. The extra dot has been dropped to stay consistent with the current software implementation).

Generic functions are used to evaluate expressions. The function $\Sigma(f, m)$ evaluates the function $f(x)$ for $x = [1, \dots, m]$ 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 $\text{foldl}(f, x)$ applies the bivariate function $f(a, x)$ to each element of x and accumulates the result, where a represents the accumulated value. For the first element in x , a is the neutral value. For example $\text{foldl}(-, [1, 2, 3])$ evaluates the function $-(a, x)$ and yields $1 - 2 - 3 = -4$. The function $\text{mapfoldl}(f, g, x)$ combines map and foldl . It applies function f to each element in x such that $y = f(x)$ and then reduces the result using the bivariate function $g(a, y)$ where a represents the accumulated value. For the first element $f(x)$, a is the neutral value. For example, $\text{mapfoldl}(\text{sqrt}, -, [4, 16, 64])$ evaluates to $\text{foldl}(-, [2, 4, 8]) = 2 - 4 - 8 = -10$. The function $\text{vcat}(x, y)$ concatenates arrays x and y along the first dimension in Julia. However, other programming languages may concatenate along a different dimension as definition of horizontal and vertical is arbitrary. Anonymous functions are used as arguments to reducing functions. Anonymous functions are denoted as $x \rightarrow \text{expression}$, 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 expression 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 DMA Λ,δ as a function of mobility

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

where z^s is the centroid mobility selected by the DMA (determined by the voltage and DMA geometry), Z is a vector of particle 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 penetration efficiency [Reineking and Porstendörfer, 1986]. The diameter $D_{p,1} = D_p(z, k = 1)$, where z is an element of Z . The function Ω has been updated from Petters (2018). The version in Petters (2018) computed the shape of the transfer function for the mobility diameter corresponding to singly charged particles and then applied the same shape of the transfer function and diffusional loss to the multiply charged particles. The functional Ω depends on three arguments $\Omega(Z, z^s, k)$ [Stolzenburg and McMurry, 2008]. The charge state is used to compute the diffusion coefficient and thus account for diffusional losses and broadening of the transfer function for multiply charged particles.

The output of $T_{size}^{\Lambda,\delta}(k, z^s)$ is the transmission of particles through the DMA in terms of the true particle mobility diameter. This is achieved by passing z^s/k as argument to Ω , which corresponds to the centroid mobility setting for the DMA to transmit particles with the size of particles with k charges under the assumption that they carry only a single charge. The net result is that $D_{p,1} = D_p(z, k = 1)$, where z is an element of Z becomes equal to the true mobility diameter axis. As a consequence the charge fraction $T_c(k, D_{p,1})$ and penetration efficiency $T_l(D_{p,1})$ are evaluated at the correct diameter. The function $T_{size}^{\Lambda,\delta}(1, z^s)$ evaluates to a vector of the same length as Z . Performing an elementwise sum over all $T_{size}^{\Lambda,\delta}(k, z^s)$ produces the net mobility distribution transmitted by the DMA. Examples for $T_{size}^{\Lambda,\delta}(1, z^s)$, $T_{size}^{\Lambda,\delta}(2, z^s)$, and $T_{size}^{\Lambda,\delta}(3, z^s)$ is shown in Figure 2, right panel in Petters (2018). Note that Eq. (10) can be evaluated using arbitrarily discretized Z vectors.

Petters [2018] also gives an expression that evaluates to the convolution matrix for passage through a single DMA that is valid in the context of size distribution measurement system, e.g. SMPS. Since the expression includes a summation over all charges, the information on particle physical diameter of multiply charged particles is lost.

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

where, m is the upper number of multiply charged particles, T is the transpose operator, and is a vector of centroid mobilities scanned by the DMA. The matrix is square if $Z_s = Z$ in Eq. 10. However, this is not a necessary restriction. 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 more 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 centroid mobility grid Z_s 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.

$$\mathbf{r} = \mathbf{A}\mathbf{n} + \epsilon \quad (12)$$

where \mathbf{r} is the response distribution, \mathbf{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. By design \mathbf{n} and \mathbf{r} are *SizeDistribution* objects, which represent 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 apparent +1 mobility diameter but not true physical diameter for multiply charged particles). Note, however, that the response function is not a true particle size distribution in the scientific sense since information about multiply charged particles is lost. The representation of \mathbf{r} as *SizeDistribution* object is to allow response functions to be used in the expression-based framework used here.

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

$$\mathbb{M}_k^{\delta_1} = \Pi_k^{\Lambda_1, \delta_1} \cdot \left\{ g_0 \cdot \left[T_{size}^{\Lambda_1, \delta_1}(k, z^s) * \mathbf{n} \right] \right\} \quad (13)$$

In Eq. (13), $\mathbb{M}_k^{\delta_1}$ evaluates to the apparent +1 mobility distribution particles that exit the DMA $^{\Lambda_1, \delta_1}$ 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, $T_{size}^{\Lambda_1, \delta_1}(k, z^s)$ is as in Eq. (10), and \mathbf{n} is the mobility size distribution upstream of DMA 1.

To help parse Eq. (13), the product $T_{size}^{\Lambda, \delta}(k, z^s) * \mathbf{n}$ 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 that the apparent growth factor for particles carrying multiple charges is the same as for single charged particles. 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 $\mathbf{m}_i^{\delta_2}$ exiting DMA 2 is given by

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

where, m is upper number of charges on the multiply charged particles and

$$\mathbf{O}_k = \text{mapfoldl}\{z^s \rightarrow [\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) * T_1^{\Lambda_2, \delta_2}(D_{p,1})]^T, \text{vcat}, Z_{s,2}\} \quad (15)$$

is the convolution matrix for transport through DMA 2 and particles carrying k charges. In Eq. (15), $Z_{s,2}$ is a vector of centroid mobilities scanned by DMA 2. Equations (14) and (15) have been 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 applied 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 exiting DMA 2 is given by

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

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

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

where \mathbf{x} is the true P_g and the vector \mathbf{c} of constraining parameters comprises the DMA setup $\Lambda_1, \Lambda_2, \delta_1, \delta_2$ and upstream size distribution \mathfrak{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 by specifying the Z vector in Eq. (10). If the Z vector does not match that of the aerosol size distribution \mathfrak{n} , the size distribution bins are interpolated onto the diameter bins corresponding to the Z bins. Transmission through DMA 1 is computed for a specified z^s (the dry mobility) and g_0 (the growth factor) via Eq. (13). The resulting $\mathbb{M}_k^{\delta_1}$ lie on the same Z grid with i bins. Any mismatches between the apparent growth factor and the underlying Z grid are resolved via interpolation implicit in the \cdot operator. ($f \cdot \mathfrak{n}$ is the uniform scaling of the diameter field of the size distribution by factor f . If the resulting diameters are off the original diameter grid, the result is interpolated onto the grid defined within \mathfrak{n}).

The mobility grid for DMA 2 is represented by the vector $Z_{s,2}$ in Eq. (15) and discretized at a resolution of j bins over a custom mobility range. If the vector Z inside the square bracket of Eq. (15), $[\Omega^{\Lambda_2, \delta_2}(Z, z^s, k) * T_l^{\Lambda_2, \delta_2}(D_{p,1})]$ equals that of DMA 1, the matrix is non square. The product $\mathbf{O}_k * \mathbb{M}_k^{\delta_1}$ will map the i bins from DMA 1 to the j bins in DMA 2. Alternatively, if the Z vector inside the square bracket of Eq. (15) is taken to be equal to $Z_{s,2}$, the matrices \mathbf{O}_k are square. In that case, 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 prior to evaluating $\mathbf{O}_k * \mathbb{M}_k^{\delta_1}$. The advantage of interpolation is that the the matrices \mathbf{O}_k are smaller.

The forward model, defined by Eq. (14) can be evaluated for arbitrary g_0 values. Thus the growth factor probability distribution P_g in Eq. (17) can be discretized into n arbitrary growth factor bins. A natural choice is to accept growth factor values that coincide with the mobility grid of DMA 2, i.e. the bins align with $g = D_{p,1}/D_d$, where D_d is the nominal diameter selected by DMA 1 and $D_{p,1} = D_p(z, k = 1)$ and z is an element of $Z_{s,2}$. However, this is not required for evaluating Eq.

(17). Equation (17) is cast into matrix form such that the humidified mobility distribution function is given by

$$m_t^{\delta^2} = \mathbf{B}P_g + \epsilon \quad (18)$$

where the matrix \mathbf{B} 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. If the grids for P_g and that of DMA 2 do not align, interpolation is used to map the P_g grid onto the DMA 2 grid. 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 size of \mathbf{B} is $j \times n$. Uncertainties in the size distribution propagate into \mathbf{B} . 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 $m_t^{\delta^2}$ is expected to be small.

Note that interpolation is widely used in this framework. Interpolation may affect how errors propagate through the model. Interpolation in Eq. (13) is unavoidable. However, interpolation can be minimized by working with non-square \mathbf{O}_k and matching the grid of P_g to that of DMA 2. Informal tests working with different binning schemes suggests that the influence of interpolation choices on the final result is smaller than typical experimental errors.

References

- 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. 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: 10.1080/174159701088027753.
- 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.