Advanced Error Modeling and Bayesian Uncertainty Quantification in Mechanistic Liquid Chromatography Modeling

William Heymann1,2, Juliane Glaser3, Fabrice Schlegel4, Will Johnson4, Pablo Rolandi4, Eric von Lieres1,\*

1 Institute of Bio- and Geosciences (IBG-1), Forschungszentrum Jülich, Wilhelm-Johnen-Str., 52428 Jülich, Germany

2 RWTH Aachen University, 52062 Aachen, Germany

3 Digital Integration and Predictive Technologies (DIPT), Amgen Research Munich, Staffelseestr. 2, 81477, München, Germany

4 Digital Integration and Predictive Technologies (DIPT), Amgen, 360 Binney St, Cambridge, MA 02142

\* corresponding author: e.von.lieres@fz-juelich.de

Abstract

Current mechanistic chromatography process modeling methods lack the ability to account for the impact of experimental errors beyond detector noise (e.g. pump delays and variable feed composition) on the uncertainty in calibrated model parameters and the resulting model-predicted chromatograms. This paper presents an uncertainty quantification method that addresses this limitation by determining the probability distribution of parameters in calibrated models, taking into consideration multiple realistic sources of experimental error. The method, which is based on Bayes’ theorem and utilizes Markov chain Monte Carlo with an ensemble sampler, is demonstrated to be robust and extensible using synthetic and industrial data. The corresponding software is freely available as open-source code at https://github.com/modsim/CADET-Match.

1. Introduction

Current mechanistic chromatography process modeling methods lack the ability to account for the impact of experimental errors beyond detector noise (e.g. pump delays and variable feed composition) on the uncertainty in calibrated model parameters and the resulting model-predicted chromatograms [1,2]. It is often infeasible to prevent or remove these experimental errors from the data. We therefore present a method here to account for these errors in quantifying parameter uncertainties. In chromatography, it is crucial to determine the accuracy of predicted chromatograms or derived metrics such as yield and purity. The method presented here determines the probability distribution of each parameter and the impact on resulting chromatograms. A confidence tube is visualized for chromatograms by overlaying random predictions, based on the parameter distributions, and coloring them according to their probability. This tube can then be compared with experimental data to see if the predictions are reasonable.

The parameters are associated with a multivariate probability distribution based on systemic and random errors in the experiments and errors in the model. No model perfectly explains the data and no experiment is performed without any errors. Both types of errors influence the parameter probability distributions. For example, let us assume that the goal is to determine column porosity and axial dispersion by running a non-pore-penetrating and non-interacting pulse through a chromatography column. Bayes’ theorem and Markov chain Monte Carlo are utilized to identify the probability distribution of these model parameters, given errors in loading concentration, pump flow rates, and pump delays. This conditional probability can be written as P (column porosity, axial dispersion | loading concentration, pump flow rate, pump delay).

The uncertainty of parameters is deceptively simple. An obvious approach would be to run a single experiment with multiple replicates, fit the model to each replicate, and then determine the mean and standard deviation of each estimated parameter. However, in chromatography it is common to use a stage-wise procedure [3,4] where several groups of parameters are estimated from different experiments to improve their identifiability and reduce correlations. For example, many models include external tubing and mixing valves along with a column containing porous resin. The column can be disconnected to estimate the volume and dispersion for the external tubing and mixing valves. The next stage would then be to connect the column but use a non-pore-penetrating tracer to estimate the column porosity and axial dispersion in the column.

Instead of simply using the best value from the previous stage, the Bayesian approach allows for the incorporation of previously determined parameters distributions to be incorporated. Furthermore, manufacturers often provide information about the accuracy of pumps, sensors, tolerances on tubing etc. The Bayesian approach allows for such additional information to be incorporated into the estimation.

1. Chromatography Modelling

This paper addresses preparative-scale packed bed liquid chromatography[3]. The general rate model (GRM) (eq. ‑) with boundary conditions (eq. ‑) and linear binding (eq. ) is used here. It describes convection and dispersion in the interstitial column volume and diffusion in the porous particles. Binding inside the porous particles can be described using more elaborate mechanisms, such as the Langmuir or steric mass action models. The linear binding model is applicable in cases where protein concentrations are low enough to exclude competitive effects. Since it has only two parameters, the model is chosen here to demonstrate the Bayesian approach and corresponding open source software for the uncertainty quantification.

The concentrations with refer to the interstitial column liquid (), particle pores (), and stationary phase () of the system. In eq. ‑ the column porosity, , particle porosity, , axial dispersion, , film diffusion, , pore diffusion, , adsorption rate, , and desorption rate, , are typically estimated from the measured chromatograms. The column length, , the particle radius, , and the initial concentrations, , are predefined. The interstitial velocity, , and the inlet concentration profiles, , are controlled during the process.

The tubing is modeled using a dispersive plug flow reactor (DPFR), eq. 8 [3]. The tubing model is solved with the same inlet and outlet boundary conditions as the column model (eq. ‑, with the tubing length, , in place of the column length, .

In the tubing, denotes the concentration, velocity, and axial dispersion. A continuously stirred tank reactor (CSTR) is used to model mixers and mixing effects.

The combined model for column and tubing is solved on the time interval using the finite volume method in the space domain and a backward difference formula (BDF) method in the time domain. The applied CADET solver is published as open-source code and is comprehensively documented at https://cadet.github.io.

* 1. Bayes’ Theorem

Mathematically, the incremental refinement approach is based on Bayes’ theorem (eq. 9). For events A and B, there is a probability, P(A), of A occurring and an independent probability, P(B), of B occurring. There is also the conditional probability of A occurring if B is true, P(A|B), and the probability of B occurring if A is true, P(B|A). For this application, A represents our parameters and B represents the data. P(A) is referred to as the prior and it is our initial probability distribution for A. In the case where we have no prior knowledge of A, a bounded uniform distribution is typically used. P(A|B) is the posterior, and it represents the belief in A given B. P(B) is referred to as the evidence and represents the probability distribution of the data and, importantly, it is a constant. In the context of the uncertainty quantification, P(B|A) is the error model and represents the probability of the data (B) being observed given the parameters (A). The posterior from one stage can be used as the prior in the next stage.

With Bayes’ theorem, the stage-wise example can be directly implemented. The posterior of one stage is used as a prior in the next stage. The first stage can then be written more precisely in determining the posterior P (volume, tubing dispersion | chromatogram). For the next stage, where the column is connected and a non-pore-penetrating tracer is used, the posterior is P (volume, tubing dispersion, axial dispersion, column porosity | chromatogram) with the previous stage’s posterior P (volume, tubing dispersion) used as a prior. At each stage, the number of variables increases but the previous variable distributions are included. For example, uncertainty in the column porosity will also result in increased uncertainty in the particle porosity, as it has similar impacts on the chromatogram. This makes Bayes’ theorem extremely powerful, since it allows stages to be chained together. If a parameter or set of parameters shows a uniform distribution or too wide of a distribution, additional stages or experiments within a stage can be added in an attempt to improve the distribution of the parameter(s).

* 1. Markov Chain Monte Carlo

In general, there is no analytical solution for eq. 9 and Markov chain Monte Carlo (MCMC) is used instead. MCMC allows for the creation of a chain of dependent samples whose distribution approximates the posterior, P(A|B). A simple MCMC algorithm takes a single starting point and proposes a step to a new point and then accepts or rejects it based on the relative probability of the two points. In this way, it walks around, and the elements of the resulting chain approximate the posterior distribution that is sought-after. There are various algorithms for proposing the next point in the chain. The better the algorithm is at proposing steps, the fewer the steps that are needed in the chain to converge to the posterior distribution. In this paper, we use emcee [5], which is a pure Python implementation of an affine-invariant MCMC ensemble sampler [6]. It uses an ensemble of walkers working together to sample more efficiently. This also allows the walkers to be run in parallel, which is a great advantage on modern computers. These parallel walkers create parallel chains. Emcee uses a log probability function, as shown in eq. 10. Only the relative probability of two sets of parameters matters in MCMC. Since P(B) is a constant, it can be omitted, which results in eq. 11. This leaves the error model and the prior as the only required terms.

With MCMC, there are many ways that the algorithm can propose new points for the walkers to evaluate. This is referred to as a move. There are many types of moves available [5–8], with newer ones designed to take advantage of ensembles of walkers. It is important to remember that these are proposals for new points. Acceptance or rejection is controlled by eq. 11. The purpose of more advanced moves is to choose points that are accepted more frequently and that result in less correlation in the chain. The simplest moves are the walk and stretch moves [5,6]. The walk move allows each walker to move randomly in the area around their current point. The walk move is the simplest move but also the least efficient. The stretch move is slightly more complicated and involves walking in the direction of the current position of the other walkers in the ensemble, thus allowing the entire ensemble to walk more efficiently. The differential evolution move [7] proposes a new location by selecting two walkers at random and making a step in the same direction as the vector connecting the current location of the two walkers. The differential evolution move has a parameter (gamma), which controls the size of the step taken of This was left at the default value. A further adaptation of the differential algorithm added a snooker update [8], which increases walker diversity and is designed to be used with the differential evolution move. The same paper proposes a mixture of moves that is also used here, with 81% differential evolution, 9% differential evolution with gamma set to 1.0, and 10% differential evolution snooker moves. This mixture was found to improve sampling efficiency compared to using stretch or walk moves in testing, which reduces the number of steps that must be taken before the chain converges. Other MCMC methods were investigated early on but most of the advanced methods required analytical derivatives of the error model or implicit assumptions to be made about the impact of errors on the output data. The method chosen here was based on parallelization and the ability to integrate custom error models.

A major goal of this project is for the entire error modeling process to be robust and automatic, and to have consistent posterior distributions from run to run as well as to mitigate problems associated with MCMC in high dimensions. To understand the latter, it is important to look at the curse of dimensionality. Let us consider a unit N-sphere and the containing N-cube and look at the ratio of the volume of the sphere to the volume of the cube (eq. 12). As can be seen in Figure 1, in 3 dimensions the ratio is about 0.5. However, as dimensionality increases, the sphere makes up an ever-decreasing percentage of the volume, and by 20 dimensions the ratio is down to 2.5e-8. In this example, the N-cube represents the boundaries of the MCMC search space, and the N-sphere represents the high-probability region. In practice, the constraints in each dimension are often far larger than the high-probability region and the problem is correspondingly worse. Consequently, any walkers starting in the low-probability space tend to stay trapped in that space and reduce the performance of the MCMC process. These issues can be mitigated by narrowing the boundaries automatically and by adjusting the starting point of walkers, as is explained later. Another major problem of MCMC is gathering a sufficient number of samples. To make the system robust and automatic, the chains are monitored for convergence and sampling stops when enough samples are taken (as explained in section 1.2).

* 1. Sources of Error

There are many sources of error in an experiment. Some sources of error have a larger impact than others. This paper considers pump delays, pump flow rate variations, loading concentration, and UV signal noise as sources of error. They were chosen by working with experimentalists. The method demonstrated in this paper can be extended to other sources of error. All sources of error are applied directly to the simulation either as a pre- or post-processing step and without normalization.

Pump delays are caused by the delay between when the pump is instructed to start and when it actually starts. Pump delays represent a period of time where diffusion and binding still occur, but there is no convective flow. If this is not taken into account, it can cause significant problems during scale-up. These delays take place every time the pump is started. In a typical experiment with a load, wash, and an elution cycle, this is three or more times. The main impact of a pump delay on a chromatogram is to shift the chromatogram, and it is not random, independent, and normally distributed. Based on the experience of experimentalists hearing the pump turn on and having seen the impact in the data, we know that there is a delay and that it is significant. A pump delay is accounted for by adding an additional period to the base model before a pump starts where no flow occurs. The length of the period for a pump delay is modeled (eq. 13) with a random uniform distribution. More elaborate distributions can be applied as more knowledge becomes available.

Pump flow rates are extremely stable but tend to differ slightly from the target flow rates based on experimental data. Pump flow rate errors have an asymmetric impact on the chromatogram with higher flow rates leading to peaks that appear sooner and sharper while lower flow rates cause peaks to appear later and more diffuse. While the pump flow rate is random, independent, and normally distributed, the impact on the chromatogram is not. Pump flow rate variations (eq. 14) are implemented by multiplying the volumetric flow rate of the base model by a factor. The factor is drawn from a normal distribution. Most pump manufacturers list their pump accuracy, and this can be directly used in the model.

Loading concentration variability (eq. 15) is implemented by multiplying the concentration of all inlet units by a factor drawn from a normal distribution. A loading concentration error influences the entire chromatogram and – depending on the type of binding – can result in changes in peak height, peak width, and peak shape. As with the pump flow rate, the impact on the chromatogram is not random, independent, and normally distributed.

UV signal noise is the easiest error to deal with, since it is random, independent, and normally distributed. It can be broken down into a detection limit noise and a proportional factor noise. This noise typically has a much higher frequency than the chromatogram and can be almost – but not entirely – automatically filtered out by CADET-Match. In most experiments, this source of error can be completely disregarded. For experiments that have very low concentrations or very short pulses, this source of error can be important. UV signal noise error is applied directly to the resulting chromatogram(s) by multiplying the concentration at each time point in the base chromatogram by a normally distributed factor and adding a normally distributed noise. In this paper, only the proportional factor is used (eq. 16) and was included as an example. Removing it has almost no impact on the results, since UV signal noise is easily removed.

* 1. Error Model

The default error model used in MCMC assumes that the impact of all errors is random, independent, and normally distributed. However, this model may not be appropriate for chromatography model parameters. For the sources of error considered in this paper, only one fulfills the assumption of the default error model. While it is possible to model these sources of error as additional parameters, this increases the computational burden. To reduce computational requirements, this study incorporates experimental sources of error directly into the error model to capture their impact on the chromatogram.

A calibrated model is a prerequisite for the error model. Calibrating a model typically requires parameter estimation. Parameter estimation in CADET-Match has been covered in a previous paper [3], but a few important details and equations will be mentioned here that are relevant to error modeling.

CADET-Match uses a variety of methods for parameter estimation with gradient-based and gradient-free search methods. The goal for CADET-Match can have single or multiple objectives. In this paper, only the multi-objective formulation is used. A multi-objective goal extracts more information from the difference between a simulated and a measured chromatogram. In this paper, the UNSGA3 [9,10] genetic algorithm is used to propose parameters and to find a global minimum of the objectives.

A variety of metrics are briefly covered here. These are scalar values that compare simulated vs measured chromatograms. Shape similarity is determined by adjusting the time-offset, , to maximize eq. 17. Once is determined, eq. 18 and 19 are used to determine the offset (in seconds) between the simulated and the measured chromatograms. The peak height difference is compared using eq. 20. These equations are then combined into a set of metrics on the chromatogram (eq. 21) and the derivative of the chromatogram (eq. 22). Equation 23 is used for all protein case studies and eq. 24 is used for all dextran case studies. Dextran has non-ideal interactions with the column and tubing and eq. 24 is used to exclusively capture the front of the peak.

By using this multi-objective approach, more information is retained that would otherwise be lost in a single-objective formulation. As discussed in section 2.3, the impact of the error sources on the chromatogram are not random-independent and normally distributed. While a peak that occurs early and a peak that occurs late – but with the same height and shape – may result in the same sum of squared errors, they do not have the same probability of occurring.

There are two important features of this parameter estimation process for error modeling. The first is that one or more experiments can be reduced to a vector comparing aspects of the chromatogram(s) to the data. This reduces the dimensionality of the goal compared to the full chromatogram(s) while keeping more information than a single-objective goal. The second major feature is that a multi-objective goal results in a Pareto front instead of a single value where there are conflicts between metrics. The metrics used are not inherently in conflict and only come into conflict when the data is inconsistent or the model is incapable of explaining the data. This Pareto front not only helps in identifying potential issues in experimental data, but can also be used to create an initial set of starting points for MCMC. The calibrated model for MCMC is chosen from the Pareto front. If there are no conflicts between the metrics, the Pareto front will have one parameter set and that parameter set is used. If there is more than one entry on the Pareto front, the parameter set with the lowest geometric mean is used. The selected parameter set is also used to create the error model, with the details explained in section 2.1. The important part here is that the error model is created using many simulations. As explained above, the multi-objective goal allows us to create a vector comparing the simulation to the data. The result is a matrix of metrics, which then needs to be converted to a probability density function to be used as an error model. Kernel density estimation (KDE) is used to convert the metrics from a matrix to a probability density function.

* 1. Kernel Density Estimation

KDE can be thought of as a continuous version of a histogram and – unlike a histogram – the probability of any parameter set can also be evaluated. KDE is used to create a probability density function of one or more continuous random variables. Once the probability density function is created, the probability of a vector of metrics can be calculated and this function is the error model P(B|A). KernelDensity is used from the Python sklearn [11] library. KDE is a machine learning technique and works best when all dimensions have approximately the same feature scale. RobustScaler is used from the sklearn library for rescaling. It scales each dimension by subtracting the median value for each feature and dividing by the interquartile range (25th quantile to 75th quantile). It also involves no hyper parameters. KDE is based on the summation of kernel functions, such as a Gaussian kernel. KDE only has a single scalar hyper parameter (bandwidth), which adjusts the width of these kernels. If the bandwidth is too large, it causes over-smoothing. Conversely, if the bandwidth is too small, it causes under-smoothing. As more samples are added to the system the bandwidth narrows. This causes an issue whereby a kernel density estimator is smooth when there are only a few samples, but as samples are added the KDE can become rougher until it starts becoming smoother again. As the number of samples increases, the computational and memory requirements of the KDE also increase substantially. There is no analytical way to determine the best bandwidth. CADET-Match follows best practices and uses a grid search with n-fold leave-one-out cross-validation to determine the best bandwidth. The data set is split into n equal pieces and n-1 of the pieces are used for training, with the remaining piece used for testing. This is repeated for all combinations of pieces. CADET-Match uses a 20-fold cross-validation. The typical bandwidth range is from 1e-2 to 1e-1 after scaling. A grid search is performed in 60 steps in log-space from 1e-3 to 1. The number of steps and range are chosen based on testing to ensure the robustness of the process. The lowest mean value is determined and all cross-correlation points are selected from one step lower to one step higher in the grid search. A second-degree polynomial is fit to these points and the minimum selected as the optimal bandwidth. This gives a deterministic approach to bandwidth selection and provides run-to-run reproducibility.

* 1. Convergence Analysis

An immediate problem that arises with MCMC is how many steps the MCMC algorithm requires. There is no simple answer to this question. The steps are correlated, and measuring this correlation is difficult and problem-dependent. CADET-Match uses the integrated autocorrelation time (IAT), as recommended [6] for affine invariant MCMC samplers, to measure the correlation between samples. The IAT works as a direct measure of how many steps are taken on average before a new independent sample is generated. The IAT is parameter-specific and can be used as a stopping condition. A typical approach is to run the algorithm until all parameters have a sufficient number of samples. The slowest converging parameter thus determines when the algorithm is stopped. There is one small problem with this approach, which is that estimating the IAT itself requires many samples. The same is true for other metrics. With emcee, around 50 times the IAT is required to reliably estimate the IAT [12].

While IAT measures the correlation between samples, it does not directly measure how accurate the posterior distribution is. A direct way of measuring the accuracy of the posterior distribution is the Monte Carlo standard error (MCSE). The MCSE is the uncertainty in the posterior caused by the sampling error due to the use of a chain of dependent samples. An example should make this clearer. While it is simple to find the mean of the posterior distribution, the MCSE measures the error of that mean value, i.e. how certain we are that it is the correct mean, given it was calculated from a chain of dependent samples. MCSE can also be applied to a single point (e.g. 5%, 95%, etc.) of the posterior and return a point estimate for the error. If the posterior is broken up into 1% intervals, for example, and the MCSE calculated at each point, then a maximum bound on the error in the posterior can be calculated. MCSE is calculated using the Arviz library. Arviz also implements arbitrary point estimates for MCSE beyond the standard application of these metrics to the mean value based on a recent paper [13].

* 1. Posterior Metrics

The first question that arises once the MCMC has been run and the posterior obtained is how to analyze it. It is important to note that the chains approximate the true posterior, and this approximation is not of equal quality along the entire posterior. The tails of the posterior converge more slowly than the rest of the posterior, which is partially due to there being fewer samples in the tails. Once the posterior has been obtained, a credible interval is normally calculated to provide an easy metric for the width of the bulk of the posterior distribution. Typically, an equal-tailed interval (ETI) is determined, for example selecting from 5% to 95% of the posterior distribution. An ETI works well if the posterior is approximately symmetric. If the distribution is asymmetric, a situation can occur where areas outside the selected interval have a higher probability than areas included in the interval. An alternative selection metric is the highest density interval (HDI) [14]. All points in the HDI have a higher probability than points outside the HDI. The HDI can contain a user-defined portion of the posterior distribution. The equivalent of a 5% to 95% ETI interval would be a 90% HDI. For symmetric distributions, the HDI is the same as ETI. HDI is more difficult to calculate than an ETI and we use the Arviz library [15] for this.

1. Error Modelling and Uncertainty Quantification Algorithm

In this study, we present an extensible error model and demonstrate its use on synthetic and industrial data. Error simulations are created from the calibrated model by using the model as a template and adding pump delays, pump flow rate, and loading concentration changes. These error simulations chromatograms are then converted to a matrix of metrics, as outlined in section 1.3. The matrix of metrics is then converted to a probability density function, as described 1.4. This probability density function is the error model, P(B|A), which is then used to assess the probability of any given chromatogram from the included error sources. This method of adding errors directly to the calibrated model template is very flexible and allows for the inclusion of further sources of error as required. For the uncertainty quantification, new simulations are created from the base model with varied parameters of interest such as axial dispersion, column porosity etc. The resulting chromatograms are then compared against the error model. MCMC is used to determine the uncertainty for each parameter of interest on the basis of the errors included in the error model. As outlined in section 1.1, the MCMC algorithm proposes parameters, which are accepted or rejected based on the error model. The chains formed by the acceptance and rejection of parameters form the posterior distribution.

This entire error modeling and uncertainty quantification process is implemented in CADET-Match and is freely available on GitHub. It is implemented as a unified multi-step process with a calibrated model as a prerequisite. By default, it also calibrates a starting model from data.

1. Calibrate base model
2. Add error sources to calibrated model
3. Create error model
4. Set MCMC starting point(s) to high-probability region(s)
5. Adjust MCMC bounds around high-probability region
6. Burn in and run MCMC until convergence
7. Post-processing
   1. Advanced Error Modelling

As previously discussed, the calibrated model is used as a template. A Sobol sequence, a quasi-random low-discrepancy sequence designed to uniformly span a unit hypercube, is used for sampling. This unit hypercube is used to create simulations based on the template using equations 13-15. The simulations are then simulated and equation 16 is used as a post-processing step. Based on experimentation, 1000 simulations proved to be sufficient for creating the error model. While more simulations can be run, they provided strongly diminishing returns while substantially increasing CPU and RAM usage.

The chromatograms are then reduced to vectors [3] that describe their important characteristics, as outlined in section 1.3. Equations 23 and 24 are re-used from the parameter estimation for MCMC with two additional metrics. Re-using the goal function from parameter estimation simplifies the process of error modeling and allows it to be applied to any problem that parameter estimation works for. The absolute height (eq. 25) and position (eq. 26) of the highest peak are added to the other metrics in the goal function (eq. 27 and 28). The parameter estimation metrics for height and position quantify relative differences that are minimized. Flow rate and pump delay errors have an asymmetric impact on the peak, which must be captured by the error model.

The absolute time of the peak max cannot be used, since it also quantizes the problem. The time of the peak max is always a time on the simulated time grid. Quantizing the problem results in a non-smooth distribution of time-offsets, which adversely impacts the KDE and thus also the MCMC. The performance of the KDE is diminished, as it requires a small bandwidth to capture the non-smooth distribution of time-offsets, which then creates many local minima in the KDE. These local minima reduce the performance of MCMC by making it less likely that a small step will be accepted. This in turn adversely affects the entire process. A spline can be fit to the data to provide a smooth interpolation. Multiple spline libraries were tested and found to place a knot at the peak max, which results in the time-offset still being quantized. After testing, using the time-offset for the first point to reach 90% of the peak max resulted in a smooth time-offset, independent of the simulated grid time. The 90% figure is mostly arbitrary. An alternative is to run the simulation on a finer time grid. While this would work in theory, it is not feasible practically. Most experimental systems sample at about 1Hz. To obtain the same accuracy as the spline method, the sample rate had to exceed 1Mhz and would increase memory requirements by a factor of a million as a result.

The next step is to process the error matrix into a probability map, i.e. a function that – given a new reduced vector – can calculate the probability of that vector being explained by the sources of error in the error matrix. KDE is used to turn the error matrix into a probability map, as described in section 2.4. The probability map is P(B|A) and can then be used for MCMC.

* 1. Bayesian Uncertainty Quantification

Once the error model has been created, the next steps involve running MCMC on the error model. As mentioned previously, there is a dimensionality problem in creating the initial population, which can be solved by ensuring that all of the walkers start in high-probability regions. The Pareto front from the parameter estimation is used as a seed for the initial population of the walkers. If there are more entries on the Pareto front than walkers, then the starting points are chosen equal to the number of walkers by randomly sampling from the Pareto front without replacement. If there are fewer entries on the Pareto front than the number of walkers, then all entries of the Pareto front are used and the entries are randomly augmented with a small amount of noise to fill up the remaining slots. The noise is important because if there are two walkers with numerically identical locations and they are used together in a move, then the distance between them is zero, which causes the code to fail. A small amount of noise with a mean of 1.0 and a standard deviation of 0.02 is multiplied by a randomly chosen entry from the Pareto front and then used. The actual amount of noise is not important, provided it is large enough that it does not cause numerical issues. It should therefore be kept above 1e-8. The MCMC sampler is then run for 100 steps. All entries with less than 10% of the maximum observed probability are removed. A KMeans [16] clustering algorithm is used, with the number of clusters equal to the number of walkers to find new starting points dispersed in the higher probability region. This procedure of running for 100 steps, removing, and clustering is repeated until the relative change in the highest probability found is less than 1%. This ensures that all the walkers start off dispersed in a high-probably space and is based on advice from the developer of emcee [17].

After the walkers have been moved to the high-probability region, the next step is to adjust the boundaries of the MCMC process around the high-probability region. The core problem is that the step size for a walker needs to be suitable for all dimensions. The ensemble of walkers working together can mitigate this to some extent, but not completely and not in all situations, such as when non-linear variable transforms are used. Parameter estimation is typically performed before MCMC and the same boundaries for parameter estimation are used for MCMC. Parameter estimation performs searches with search bounds spanning multiple orders of magnitude and log transforms are used to allow for more efficient searching. Parameters, once found, normally take up a tiny slice of the searched region. A solution to this problem is to adjust the boundaries of the MCMC process so that it contains the high-probability region, and for the high probability region to have approximately the same step size in every dimension. MCMC sampling is performed until the 5% and 95% percentile values in the last 200 steps have a standard deviation of less than 1e-3. The 5% and 95% values were chosen because they are far enough away from the center of the distribution to give a good indication of the final width of the distribution. The boundaries are adjusted for each variable by setting the upper bound of the distribution to three times the difference from the 95% bound to the center, and the lower bound to three times the difference from the 5% bound to the center. This centers the high-probability region inside the new boundaries. The high-probability region takes up approximately one third of the region between the boundaries in each dimension. This fixes the step size problem and provides the space needed to define the posterior more precisely.

After the MCMC boundaries have been adjusted, the process is repeated for locating the walkers in high-probability space. The final state of the walkers from the boundary adjustment step is used as the starting state of the walkers for this step. With the change of the search boundaries, areas that may have been difficult to explore in the original area become more accessible and improved starting positions can be found.

The final MCMC step is a combination of burn-in and running the sampler until convergence. By default, CADET-Match uses 52 times the IAT as a stopping criterion and discards the first 2 IATs as burn-in. This ensures the samples used are independent of the starting point and sufficient as a stopping criterion while also being fast enough to calculate. With 128 walkers, 50 times the IAT results in a minimum of 6400 independent samples per parameter, while the actual chain length and total number of samples vary based on how correlated the samples are. Using the MCSE and finding the max error of the posterior is also possible. However, it takes much longer to calculate, meaning that progress either needs to be checked infrequently or that the determination of progress dominates the calculation time compared to the time it takes to run the simulations.

The final step of the entire process is the post-processing of all the data. The chain is processed with KDE to create a probability density function suitable for use as a prior in another MCMC run following the same procedures that were used to turn the error matrix into a probability density function. The maximum a posteriori (MAP) – the point corresponding to the highest posterior probability – is easy to find once MCMC has been completed. With the log probability of each accepted entry in every chain recorded, the entry with the highest log probability can be found via a simple search. It is important to note that the parameters are not independent of each other. The MAP is not the highest probability point for each parameter but is the parameter set that has the highest overall probability.

1. Materials and Methods

Chromatographic cation exchange runs were conducted on Äkta Avant (GE Healthcare). Blue Dextran 2000 (GE Healthcare) was used for non-pore-penetrating pulse experiments. A volume of 1 mL dextran solution with a concentration of 0.002 mM was injected at a flow rate of 5 mL/min with and without the column attached. These experiments were carried out in duplicates. Dextran chromatogram data was measured at UV 280nm.

Pore penetrating was performed as well as load, wash, and elution steps using a Fractogel  (EMD Millipore) resin in a packed bed column with an inner diameter of 16 cm and a length of 25 cm at a flow rate of 5 mL/min. The column was pre-equilibrated for 3 CV with 200 mM sodium acetate buffer containing 1 M NaCl at pH 5.0. The column was equilibrated for 3 CV with 100 mM sodium acetate buffer at pH 5.

Pulse injections under non-binding conditions were run using a previously purified monomeric antibody as a tracer. The antibody was prepared in a 100 mM sodium acetate buffer with 500 mM NaCl at pH 5 to prevent adsorption.

For elution experiments, the column was loaded with 152 mL or 540 mL of filtered virus inactivated pool (FVIP) of the monoclonal antibody product. The material was obtained from a previous capturing step. The antibody material was produced by CHO cell culture. The FVIP material was conditioned with arginine, targeting 50 mM arginine concentration in the FVIP material. A wash step was conducted with 100 mM sodium acetate buffer at pH 5. The elution step was conducted with a linear 1 mM/CV gradient between two buffers with 100 mM sodium acetate and 100 mM sodium acetate plus 1 mM NaCl, respectively, at pH 5.0. After elution, the column was cleaned using 1 M NaOH solution. Chromatogram data of the load, wash, and elution experiments were measured at UV 300 nm and UV 280 nm. For the 152 mL load volume run, fractionated samples were taken at 8 mL fraction volume starting at 0.1 AU of UV 280 signal.

1. Case Studies

All synthetic and experimental case studies share several fixed parameters (shown in Table S2 and Table S3). All model equations and solver settings are described in a previous publication [3]. All software versions can be found in Table S1. The synthetic and experimental case studies also share some of the same auxiliary models used to describe the impact of tubing and mixing devices. A dispersive plug flow reactor (DPFR) is used to represent tubing. A continuously stirred tank reactor (CSTR) is used to represent mixing valves and other areas in the tubing that generate mixing-like behavior. The general rate model (GRM) is used to represent the column. Synthetic and experimental case studies use the same problem discretization and other solver settings as detailed in the supplement in Table S2. The DPFR, CSTR, and GRM are connected in a single simulation using CADET (<https://cadet.github.io>) [18]. The equations are spatially discretized with a finite volume scheme [19]. The goal is to find the distribution that corresponds to each parameter, ensuring that – as far as possible – unaccounted errors in one stage are not carried over to further stages. For example, a stage that does not accurately account for the error in the tubing dispersion will carry over any residual error into the axial dispersion of the column. This makes it important to design the base model and chosen error sources to minimize unaccounted errors.

Two error models are used for the synthetic and experimental case studies and shown in Table 1 (labeled ES1 and ES2, respectively). A normal distribution is denoted with an N (mean, standard deviation) and a uniform distribution with U (lower bound, upper bound). The ES1 error source is chosen to have a small amount of error, while the ES2 error source is chosen to have an error close to what we have typically observed in industrial data. An HDI of 90% is used for reporting upper and lower bounds and the maximum MCSE is reported for this interval. The supplement contains corner plots, HDI plots, autocorrelation plots, and mixing plots for each study. On the left-hand side of the mixing plots are the parameter distributions for each chain, while the mixing of the chains is depicted on the right-hand side. It is observed that the autocorrelation plots and mixing plots for all data presented in this paper appear normal. A corner plot is a matrix of plots where the diagonal plots contain the distribution for each parameter and below the diagonal is the joint distribution for each combination of two parameters. These plots provide a quick way of observing the correlations between parameters and all the distributions in a single plot.

* 1. Synthetic Case Studies

The synthetic model is designed to be representative of a real system but with known parameters to verify the proper functioning of the error modeling process. The synthetic model uses a DPFR connected to a CSTR connected to a column described by the GRM. All parameters can be found in the supplement.

A stage-wise [3,4] estimation is performed with the posterior from one stage used as a prior in the next stage. Four stages are used in the synthetic model with a single component. The four stages are bypass pulse, non-pore-penetrating pulse, pore-penetrating but non-binding pulse, and binding pulse. For the bypass pulse, the column is replaced with a zero-volume connector and removed from the simulation. This stage is designed to account for extra column effects. In the next stage, a non-pore-penetrating pulse is used to find the column porosity and dispersion. A non-binding but pore-penetrating pulse is then used to find the film diffusion, particle diffusion, and particle porosity. Finally, a binding pulse is used with a linear isotherm to find the adsorption and desorption rates. Instead of a single optimal value for each parameter carried over from one stage to the next, the parameter distribution is carried over. This means that with each stage, there are more parameters to estimate.

* 1. Dextran Pulse with Detached Column

The first stage involves a bypass experiment to determine the posterior distributions for the tubing dispersion, cross-sectional area, and mixing volume. Table 2 shows ES1, with the MAP closely resembling the ground truth of the calibrated base model, and the narrow bounds on all parameters. The "% MAP" column displays the HDI width as a percentage of the MAP, facilitating the comparison of HDI widths for parameters that vary over orders of magnitude. The max(%MCSE) value for all parameters indicates a low average error of approximately 0.06% across the HDI interval. For ES2 in Table 2, the effect of the error source is apparent from the wider bounds shown in % MAP, although the MAP remains near the ground truth. Table 3 shows that both the ES1 and ES2 have a similar performance, with runtimes differing by only a few seconds and almost identical acceptance rates and IAT.

* 1. Non-Pore-Penetrating Pulse

In the second stage, a non-pore-penetrating pulse is employed to determine the posterior distributions of column dispersion and column porosity as well as the variables carried over from the prior. Table 4 shows ES1 with a MAP close to the ground truth and a narrow HDI interval for all parameters. While the MAP for all parameters is also near the ground truth for ES2, the corresponding HDI interval is wider and the max(%MCSE) values are larger. In Table 5, the ES1 and ES2 are similar, with the ES1 having a slightly higher IAT and taking slightly less time to run. ES2 takes about 30% more time to run, mainly due to the slightly longer time required to run the ES2 simulations and other server tasks. Notably, the non-pore-penetrating pulse takes much longer to run than the bypass pulse (approximately 6-8 times slower). This is due to the dimensionality problem, as the non-pore-penetrating pulse involves 5 parameters compared to the 3 parameters for the bypass pulse. As a result, there is a longer total runtime before convergence as well as lower acceptance rates and a larger IAT.

* 1. Non-Binding Protein Pulse

The third stage involves a pore-penetrating, but non-binding, protein pulse to find posterior distributions for film diffusion, pore diffusion, and particle porosity, in addition to the variables from the previous stages. With eight variables, Table 7 shows that the IAT increases, the acceptance rate decreases, and the number of simulations required exceeds one million. The MAP is close to the ground truth for both ES1 and ES2, but the HDI interval is narrower for ES1 than for ES2, as seen in Table 6. The increase in IAT and total runtime is due to the coupling between variables and the addition of more variables. Figure 2 shows the posterior distributions for film diffusion and particle diffusion as well as their joint probability distribution, which reveals a strong correlation between these parameters. The peaks for film diffusion and pore diffusion are also flat, indicating that small changes near the optimum have a minimal impact on the simulated chromatogram.

* 1. Binding Protein Pulse

In the final stage of the estimation process, a pore-penetrating binding protein pulse is used to determine the posterior distributions for the adsorption and desorption rates of a linear binding model in addition to the variables already included in the prior.

Table 9 shows that the IAT is 3-6 times higher for the binding case compared to the pore-penetrating, non-binding case. This is due to the increased number of variables and the coupling between them. This increases the time required for the calculation to approximately 6 days for ES2 and 2 days for ES1. Table 8 shows that the same pattern from previous tables can be observed, with ES1 having a narrower HDI interval compared to ES2.

After four stages, the MAP is close to the ground truth for all parameters for both error sources. Figure 3 shows a corner plot for ES1 with joint distribution plots for each variable pair. Any joint distribution plots that are approximately circular indicate that the two parameters are independent, while other shapes indicate some degree of correlation. From the joint distribution plots, we can see that there are weak linear relationships between particle porosity and film diffusion, particle porosity and column porosity, and kA and keq. There is a strong linear relationship between film diffusion and particle diffusion. Although it is evident that most parameters are coupled to varying degrees, the coupling is not strong.

Looking at the same entries in Figure 4 for ES2, we see similar results. kA and keq have a stronger linear correlation, while film diffusion and particle diffusion show a stronger, but more non-linear, correlation. Particle porosity and film diffusion also show a similar change, with the correlation becoming more non-linear.

Finally, it is worth noting that the error models used are very similar to each other, with the main difference being a 1s delay in pump activation. This underlines the importance of accurately accounting for errors in the early stages of estimation and highlights the need for the precise control of experimental errors, as even small errors can substantially increase the uncertainty of parameters.

Figure 5 presents the results of using the posterior of one stage as the prior of the next stage. The top row illustrates the error model, P(B|A), with ES1 and ES2 shown in A and B, respectively. The graphs are shaded according to the relative probability of the chromatogram based on the errors used. Compared to A, B includes a 0-2s pump delay and a slightly larger error on the flow rate, resulting in a wider and more asymmetric probability region. The bottom row displays the chromatograms generated by sampling from the posterior distribution, with ES1 and ES2 shown in C and D, respectively. The black line indicates the ground truth. The posterior is not expected to match the shape or width of the error model. Not all errors generated by pump delays, concentration changes, or flow rate changes can be explained by model parameters, which is what causes the expected difference. The 90% HDI confidence tube on the prediction in C is extremely tight, indicating a negligible parameter impact within the HDI range for ES1 in Table 8. In contrast, the bounds on the prediction in D are wider than for ES1. Despite the wider HDI region for all parameters, the high-probability region is narrow. Diffusion-related parameters have the widest ranges for both error sources. This makes them difficult to fit precisely, as the relatively large changes in their values have a small impact on the resulting chromatogram. The ground truth goes through the high-probability region in both C and D, as can be seen in the insets.

1. Experimental Case Studies

An experimental dataset for two charge variants of a monoclonal antibody was measured at Amgen. It includes two dextran pulses with a detached column, two dextran pulses with an attached column, two protein pulses under non-binding conditions, a gradient elution with fractionation, and a gradient elution with an extended loading phase but without fractionation. The dextran pulses and non-binding protein pulses are used for method development. As in the synthetic example, the posterior from one stage is used as a prior in the next stage. The same two error sources are used from the synthetic case. The experimental setup consists of a DPFR connected to a column implemented with the GRM. The experimental setup presented here does not use a CSTR. More complex models with one or more CSTR and DPFR units were unable to better reproduce the observed behavior but suffered from high parameter correlations and wider HDI ranges (data not shown). The external holdup volumes are therefore lumped, and the DPFR model parameters are not meant to reflect the real dimensions of the tubing.

* 1. Dextran Pulses with Detached Column

The first stage of the experiment aims to identify the tubing dispersion and cross-sectional area by replacing the column with a zero-volume connector and utilizing a dextran pulse. In this case, the tubing length was fixed due to its high correlation with the dispersion.

The results of the experiment are presented in Table 10, where ES1 shows an extremely narrow HDI and a low maximum error in the interval. ES2 shows a similar value for the MAP for both parameters as the narrow error source, but with a wider HDI. As with the synthetic examples, early stages with only a few parameters have a low IAT and higher acceptance rates, as can be seen in Table 11

* 1. Dextran Pulses with Attached Column

In the second stage, a dextran pulse with attached column is used to estimate the posterior distributions for the column porosity and column dispersion in addition to the variables carried over in the prior. It can be seen from the ES1 error source shown in Table 12 that the upper and lower bounds of the HDI are quite narrow with a low max(%MCSE) in the interval. In Table 12, with the ES2 error source, the upper and lower bounds of the HDI are much wider with a higher max(%MCSE) in the interval. The MAPs are quite close to each other with the ES2 error source having a wider interval than the ES1 error source. The ES2 error source took about 37% longer to converge than the ES1 error source, with the IAT increasing compared to the previous stage (Table 13).

The second stage employs a non-pore-penetrating dextran pulse with a connected column to estimate the posterior distributions for the column porosity and dispersion, along with the variables carried over from the previous stage. Table 12 shows that the upper and lower bounds of the HDI for the ES1 error source are narrow, with a low maximum error in the interval. Similarly, the MAPs for ES1 and ES2 are close to each other. ES2 has wider HDI bounds and a higher max(%MCSE) in the interval. The ES2 error source takes about 37% longer to converge, as shown in Table 13, and the IAT increases compared to the previous stage.

* 1. Non-Binding Protein Pulses

In the third stage, a non-binding protein pulse is utilized to estimate the posterior distributions of particle porosity, film diffusion, and pore diffusion, in addition to the variables from the prior stages. ES1, presented in Table 14, maintains a narrow range on the HDI, except for film diffusion. ES2, also shown in Table 14, continues the pattern with a wider HDI range. While the MAPs are still relatively close for ES1 and ES2, they are not as close as in the previous stages. From Table 15, ES1 and ES2 converge at a similar rate. The convergence required approximately two million simulations, highlighting the difficulty in solving these types of problems.

In Figure 6, the top row shows the error model, P(B|A), with the first pulse in cell A and the second pulse in cell B for the ES1 error source. The top row of the figure shows the error model, which is flat and quickly falls off at the edges, indicating that the two experiments created a confined region but did not fully agree with each other. The bottom row shows chromatograms generated by sampling from the posterior distribution with the first pulse in cell C and the second pulse in cell D. The posterior distributions in C and D have a broad high-probability area and a rapid falloff at the edges, with approximately the same width as the error model in A and B. However, the experimental data on the front and back of the peak are outside the posterior region or even the error model, indicating an error in the model. This information is valuable and cannot be obtained through parameter estimation alone, highlighting the need for further work. Figure 7 for ES2 exhibits similar results, with a wider error model in cells A and B, but still exhibits a flat probability behavior that falls off quickly at the edges. The posterior plots in cells C and D capture the experimental data slightly better within the high probability region, but the ES2 error does not explain the front and back of the peak. This suggests that something is missing from the model and that this effect is not related to pump delay errors.

1. Conclusions

A complete process was demonstrated for modeling errors and obtaining parameter distributions along with the uncertainty of the resulting chromatogram(s). The process covers everything from creating the error model and sampling it to how to assess it when the sampling converges. The most commonly used error models assume that all errors are random, independent, and normally distributed, which is often inaccurate for chromatography. A novel approach to constructing error models for chromatography that is capable of handling such error sources was introduced and demonstrated using synthetic and experimental data. The error model presented here covers errors caused by pump delays, loading concentration, pump flow rates, and UV detector noise, and can be easily extended to additional sources of error. This entire process is based on Bayesian uncertainty analysis and MCMC. MCMC provides a way of incrementally refining the probability distribution of the parameters within a single stage until convergence. Bayes’ theorem then allows the distribution to be carried over from one stage to the next. Additional stages can be added as required to refine the parameters further. The posterior chromatograms resulting from the posterior probability distribution of parameters also provide information to help further understand the studied systems. In the synthetic binding example, the posterior chromatograms show narrow confidence tubes, which indicates that despite some of the diffusion parameters having a wide HDI, their effect is minimal over that interval. In the experimental non-binding pulse example, the posterior chromatograms do not capture the very front or tail of the peak, indicating that no combination of parameters estimated captures that behavior. This provides critical information suggesting that the model is lacking some kind of mechanism that is present in the system and not covered by any of the modeled error sources. The posterior probability distribution of parameters can also be used to determine where improvements are needed in the modeling and experimental process. For example, experiments can be added or changed, and the posteriors compared for all parameters. These error models can be used to find out what types of measurements are needed and how precise they need to be to obtain the desired results.

This paper also demonstrates several ways of using MCMC that should be considered best practices. MCMC convergence metrics are not the same across all types of MCMC, but each type does have ways of measuring convergence and appropriate burn-in. MCMC creates a dependent chain between samples, but the quality of the posterior distribution is based on the number of independent samples. The integration autocorrelation time (IAT) then measures how many dependent steps must be taken before a new independent sample is generated. In this study, IAT as low as 7 and as high as 600 were encountered. To generate enough samples, the chain length must be varied based on the IAT and a static chain length is not appropriate. The choice of where to start the walkers is also important due to the problem of dimensionality. As the numbers of parameters increase, it becomes critically important that the walkers start off in high-probability regions. A method for doing so was demonstrated in this paper. Beyond just a few parameters and without good starting points, most of the walkers end up far away from the main distribution and remain effectively stuck in a low-probability space, thus distorting the posterior distribution.

CADET-Match implements the methods in this paper and is freely available on GitHub. It is written in Python using standard libraries and is designed as a monolithic approach for solving parameter estimation and error modeling problems with minimal human intervention. CADET-Match mostly requires CADET models, experimental data, and what errors to use in the error model. It can perform the rest of the steps automatically. Everything from parameter estimation, building the error model, calibrating the walkers, adjusting the boundaries, and running the MCMC until convergence is completely automatic with output written as the system runs to measure progress. The code has been extensively tested on synthetic and experimental data to ensure its robustness. The open code provides full transparency of the applied procedures and allows the software to be adapted and integrated in operational workflows. CADET-Match can be used with all model variants available in the parent project, CADET, and covers a wide range of transport and binding models. The scripts for running all case studies in this publication are freely available and can be easily adjusted to other scenarios.

Looking ahead, one of the clear problems is the amount of time it takes for convergence. As the dimensionality increases, more simulations are needed. To further compound this problem, later stages of estimation often involve simulations that become more complex and require more computing time. The most promising solution to this problem would be to use an artificial neural network (ANN) to approximate the CADET simulator and to run MCMC on the ANN. Initial tests indicate that an ANN is several hundred thousand times faster than CADET.

Bibliography

[1] T. Briskot, F. Stückler, F. Wittkopp, C. Williams, J. Yang, S. Konrad, K. Doninger, J. Griesbach, M. Bennecke, S. Hepbildikler, J. Hubbuch, Prediction uncertainty assessment of chromatography models using Bayesian inference, Journal of Chromatography A. 1587 (2019) 101–110. https://doi.org/10.1016/j.chroma.2018.11.076.

[2] Y. Yamamoto, T. Yajima, Y. Kawajiri, Uncertainty quantification for chromatography model parameters by Bayesian inference using sequential Monte Carlo method, Chemical Engineering Research and Design. 175 (2021) 223–237. https://doi.org/10.1016/j.cherd.2021.09.003.

[3] W. Heymann, J. Glaser, F. Schlegel, W. Johnson, P. Rolandi, E. von Lieres, Advanced score system and automated search strategies for parameter estimation in mechanistic chromatography modeling, Journal of Chromatography A. 1661 (2022) 462693. https://doi.org/10.1016/j.chroma.2021.462693.

[4] U. Altenhöner, M. Meurer, J. Strube, H. Schmidt-Traub, Parameter estimation for the simulation of liquid chromatography, Journal of Chromatography A. 769 (1997) 59–69. https://doi.org/10.1016/s0021-9673(97)00173-8.

[5] D. Foreman-Mackey, D.W. Hogg, D. Lang, J. Goodman, emcee: The MCMC Hammer, Publications of the Astronomical Society of the Pacific. 125 (2013) 306–312. https://doi.org/10.1086/670067.

[6] J. Goodman, J. Weare, Ensemble samplers with affine invariance, Communications in Applied Mathematics and Computational Science. 5 (2010) 65–80. https://doi.org/10.2140/camcos.2010.5.65.

[7] B. Nelson, E.B. Ford, M.J. Payne, RUN DMC: AN EFFICIENT, PARALLEL CODE FOR ANALYZING RADIAL VELOCITY OBSERVATIONS USING N -BODY INTEGRATIONS AND DIFFERENTIAL EVOLUTION MARKOV CHAIN MONTE CARLO, The Astrophysical Journal Supplement Series. 210 (2013) 11. https://doi.org/10.1088/0067-0049/210/1/11.

[8] C.J.F. ter Braak, J.A. Vrugt, Differential Evolution Markov Chain with snooker updater and fewer chains, Statistics and Computing. 18 (2008) 435–446. https://doi.org/10.1007/s11222-008-9104-9.

[9] H. Seada, K. Deb, A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives, IEEE Transactions on Evolutionary Computation. 20 (2016) 358–369. https://doi.org/10.1109/TEVC.2015.2459718.

[10] J. Blank, K. Deb, Pymoo: Multi-Objective Optimization in Python, IEEE Access. 8 (2020) 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567.

[11] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, S. 1. 0 Contributors, Author Correction: SciPy 1.0: fundamental algorithms for scientific computing in Python, Nature Methods. 17 (2020) 352. https://doi.org/10.1038/s41592-020-0772-5.

[12] D. Foreman-Mackey, Autocorrelation analysis & convergence — emcee, Autocorrelation Analysis & Convergence. (n.d.). https://emcee.readthedocs.io/en/stable/tutorials/autocorr/ (accessed August 19, 2021).

[13] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, P.-C. Bürkner, Rank-Normalization, Folding, and Localization: An Improved Rˆ for Assessing Convergence of MCMC, Bayesian Analysis. 1 (2021). https://doi.org/10.1214/20-ba1221.

[14] J.K. Kruschke, JAGS, in: Doing Bayesian Data Analysis, Elsevier, 2015: pp. 193–219. https://doi.org/10.1016/b978-0-12-405888-0.00008-8.

[15] R. Kumar, C. Carroll, A. Hartikainen, O. Martin, ArviZ a unified library for exploratory analysis of Bayesian models in Python, Journal of Open Source Software. 4 (2019) 1143. https://doi.org/10.21105/joss.01143.

[16] J.B. MacQueen, Some Methods for Classification and Analysis of MultiVariate Observations, in: L.M.L. Cam, J. Neyman (Eds.), Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967: pp. 281–297.

[17] walkers stuck in low probability space, Google Groups. (n.d.). https://groups.google.com/g/emcee-users/c/fg7sQNw8YcU?pli=1 (accessed August 19, 2021).

[18] S. Leweke, E. von Lieres, Chromatography Analysis and Design Toolkit (CADET), Computers & Chemical Engineering. 113 (2018) 274–294. https://doi.org/10.1016/j.compchemeng.2018.02.025.

[19] E. von Lieres, J. Andersson, A fast and accurate solver for the general rate model of column liquid chromatography, Computers & Chemical Engineering. 34 (2010) 1180–1191. https://doi.org/10.1016/j.compchemeng.2010.03.008.

Table 1. Error source description for ES1 and ES2 error sources

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ES1 | ES2 | Units |
|  | - |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 2. Synthetic bypass experiment results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | GT | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Tube dispersion | ES1 | 2.70e-06 | 2.51e-06 | 2.71e-06 | 2.88e-06 | 14.0 | 0.13 |  |
| Tube cross-section area | ES1 | 1.90e-05 | 1.89e-05 | 1.90e-05 | 1.91e-05 | 1.3 | 0.01 |  |
| CSTR volume | ES1 | 4.00e-06 | 3.94e-06 | 4.00e-06 | 4.08e-06 | 3.5 | 0.03 |  |
| Tube dispersion | ES2 | 2.70e-06 | 2.30e-06 | 2.70e-06 | 3.17e-06 | 32.1 | 0.34 |  |
| Tube cross-section area | ES2 | 1.90e-05 | 1.82e-05 | 1.90e-05 | 1.97e-05 | 8.0 | 0.06 |  |
| CSTR volume | ES2 | 4.00e-06 | 3.91e-06 | 4.01e-06 | 4.11e-06 | 4.9 | 0.04 |  |

Table 3. Synthetic bypass experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 73,728 | 67,584 |
| Chain length | 576 | 528 |
| Simulation time | 0:26:15 | 0:25:36 |
| max(IAT) | 11 | 10 |
| Acceptance ratio | 0.339 | 0.354 |

Table 4. Synthetic non-pore-penetrating experiment results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | GT | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Column dispersion | ES1 | 3.00e-07 | 2.92e-07 | 3.00e-07 | 3.10e-07 | 6.0 | 0.07 |  |
| Column porosity | ES1 | 0.350 | 0.347 | 0.350 | 0.352 | 1.5 | 0.01 |  |
| Tube dispersion | ES1 | 2.70e-06 | 2.55e-06 | 2.71e-06 | 2.86e-06 | 11.2 | 0.10 |  |
| Tube cross-section area | ES1 | 1.90e-05 | 1.89e-05 | 1.90e-05 | 1.91e-05 | 1.2 | 0.01 |  |
| CSTR volume | ES1 | 4.00e-06 | 3.95e-06 | 4.00e-06 | 4.05e-06 | 2.6 | 0.02 |  |
| Column dispersion | ES2 | 3.00e-07 | 2.87e-07 | 3.02e-07 | 3.16e-07 | 9.6 | 0.08 |  |
| Column porosity | ES2 | 0.350 | 0.346 | 0.350 | 0.355 | 2.5 | 0.02 |  |
| Tube dispersion | ES2 | 2.70e-06 | 2.32e-06 | 2.73e-06 | 3.08e-06 | 27.6 | 0.26 |  |
| Tube cross-section area | ES2 | 1.90e-05 | 1.82e-05 | 1.90e-05 | 1.96e-05 | 7.3 | 0.08 |  |
| CSTR volume | ES2 | 4.00e-06 | 3.93e-06 | 4.01e-06 | 4.08e-06 | 3.8 | 0.03 |  |

Table 5. Synthetic non-pore-penetrating experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 224,640 | 196,864 |
| Chain length | 1755 | 1538 |
| Simulation time | 3:02:39 | 4:13:10 |
| max(IAT) | 34 | 30 |
| Acceptance ratio | 0.197 | 0.205 |

Table 6. Synthetic non-binding experiment results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | GT | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Film diffusion | ES1 | 2.00e-06 | 1.48e-06 | 2.05e-06 | 2.57e-06 | 53.7 | 0.79 |  |
| Particle porosity | ES1 | 0.330 | 0.325 | 0.331 | 0.336 | 3.1 | 0.04 |  |
| Particle diffusion | ES1 | 2.50e-11 | 2.22e-11 | 2.54e-11 | 2.82e-11 | 23.5 | 0.18 |  |
| Column dispersion | ES1 | 3.00e-07 | 2.93e-07 | 3.03e-07 | 3.09e-07 | 5.2 | 0.06 |  |
| Column porosity | ES1 | 0.350 | 0.348 | 0.349 | 0.352 | 1.3 | 0.01 |  |
| Tube dispersion | ES1 | 2.70e-06 | 2.57e-06 | 2.68e-06 | 2.84e-06 | 10.2 | 0.09 |  |
| Tube cross-section area | ES1 | 1.90e-05 | 1.89e-05 | 1.91e-05 | 1.91e-05 | 1.1 | 0.01 |  |
| CSTR volume | ES1 | 4.00e-06 | 3.95e-06 | 4.03e-06 | 4.05e-06 | 2.4 | 0.02 |  |
| Film diffusion | ES2 | 2.00e-06 | 1.30e-06 | 1.98e-06 | 2.67e-06 | 69.5 | 0.70 |  |
| Particle porosity | ES2 | 0.330 | 0.322 | 0.329 | 0.338 | 4.7 | 0.04 |  |
| Particle diffusion | ES2 | 2.50e-11 | 2.20e-11 | 2.52e-11 | 2.97e-11 | 30.5 | 0.26 |  |
| Column dispersion | ES2 | 3.00e-07 | 2.87e-07 | 2.93e-07 | 3.14e-07 | 9.1 | 0.10 |  |
| Column porosity | ES2 | 0.350 | 0.346 | 0.352 | 0.354 | 2.3 | 0.02 |  |
| Tube dispersion | ES2 | 2.70e-06 | 2.38e-06 | 2.75e-06 | 3.04e-06 | 23.7 | 0.29 |  |
| Tube cross-section area | ES2 | 1.90e-05 | 1.83e-05 | 1.88e-05 | 1.96e-05 | 6.7 | 0.09 |  |
| CSTR volume | ES2 | 4.00e-06 | 3.94e-06 | 4.03e-06 | 4.07e-06 | 3.3 | 0.03 |  |

Table 7. Synthetic non-binding experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 1,141,376 | 981,376 |
| Chain length | 8917 | 7667 |
| Simulation time | 16:01:42 | 14:15:51 |
| max(IAT) | 178 | 153 |
| Acceptance ratio | 0.092 | 0.095 |

Table 8. Synthetic linear isotherm binding experiment results ES1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | GT | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| kA | ES1 | 4.00e-04 | 3.78e-04 | 3.97e-04 | 4.24e-04 | 11.6 | 0.11 |  |
| keq | ES1 | 1.00e-01 | 8.94e-02 | 9.42e-02 | 1.10e-01 | 21.9 | 0.18 |  |
| Film diffusion | ES1 | 2.00e-06 | 1.70e-06 | 2.03e-06 | 2.47e-06 | 38.3 | 0.33 |  |
| Particle porosity | ES1 | 0.330 | 0.328 | 0.332 | 0.334 | 1.9 | 0.02 |  |
| Particle diffusion | ES1 | 2.50e-11 | 2.29e-11 | 2.55e-11 | 2.71e-11 | 16.6 | 0.13 |  |
| Column dispersion | ES1 | 3.00e-07 | 2.94e-07 | 3.03e-07 | 3.08e-07 | 4.7 | 0.09 |  |
| Column porosity | ES1 | 0.350 | 0.348 | 0.349 | 0.352 | 1.1 | 0.01 |  |
| Tube dispersion | ES1 | 2.70e-06 | 2.58e-06 | 2.68e-06 | 2.83e-06 | 9.4 | 0.11 |  |
| Tube cross-section area | ES1 | 1.90e-05 | 1.89e-05 | 1.91e-05 | 1.91e-05 | 1.0 | 0.02 |  |
| CSTR volume | ES1 | 4.00e-06 | 3.96e-06 | 4.03e-06 | 4.04e-06 | 2.1 | 0.03 |  |
| kA | ES2 | 4.00e-04 | 3.71e-04 | 4.21e-04 | 4.33e-04 | 14.6 | 0.11 |  |
| keq | ES2 | 1.00e-01 | 8.52e-02 | 1.02e-01 | 1.14e-01 | 28.0 | 0.22 |  |
| Film diffusion | ES2 | 2.00e-06 | 1.60e-06 | 1.75e-06 | 2.92e-06 | 74.9 | 0.38 |  |
| Particle porosity | ES2 | 0.330 | 0.328 | 0.335 | 0.338 | 3.2 | 0.03 |  |
| Particle diffusion | ES2 | 2.50e-11 | 2.20e-11 | 2.79e-11 | 2.78e-11 | 20.7 | 0.14 |  |
| Column dispersion | ES2 | 3.00e-07 | 2.89e-07 | 2.85e-07 | 3.13e-07 | 8.6 | 0.08 |  |
| Column porosity | ES2 | 0.350 | 0.346 | 0.352 | 0.354 | 2.0 | 0.03 |  |
| Tube dispersion | ES2 | 2.70e-06 | 2.41e-06 | 2.82e-06 | 3.02e-06 | 21.6 | 0.34 |  |
| Tube cross-section area | ES2 | 1.90e-05 | 1.84e-05 | 1.91e-05 | 1.95e-05 | 6.0 | 0.09 |  |
| CSTR volume | ES2 | 4.00e-06 | 3.95e-06 | 4.00e-06 | 4.07e-06 | 3.0 | 0.03 |  |

Table 9. Synthetic linear isotherm binding experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 1,550,848 | 3,824,512 |
| Chain length | 12116 | 29879 |
| Simulation time | 1 day, 22:20:25 | 6 days, 9:05:02 |
| max(IAT) | 241 | 597 |
| Acceptance ratio | 0.092 | 0.066 |

Table 10. Bypass experiment results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Tube dispersion | ES1 | 2.71e-06 | 2.77e-06 | 2.80e-06 | 3.3 | 0.02 |  |
| Tube cross-section area | ES1 | 1.86e-05 | 1.87e-05 | 1.89e-05 | 1.2 | 0.02 |  |
| Tube dispersion | ES2 | 2.47e-06 | 2.72e-06 | 3.12e-06 | 23.8 | 0.22 |  |
| Tube cross-section area | ES2 | 1.79e-05 | 1.87e-05 | 1.94e-05 | 8.2 | 0.11 |  |

Table 11. Bypass experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 52,352 | 58,240 |
| Chain length | 409 | 455 |
| Simulation time | 0:31:40 | 0:27:46 |
| max(IAT) | 7 | 9 |
| Acceptance ratio | 0.345 | 0.296 |

Table 12. Dextran experiment results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Column dispersion | ES1 | 5.08e-07 | 5.17e-07 | 5.28e-07 | 3.8 | 0.03 |  |
| Column porosity | ES1 | 0.332 | 0.337 | 0.338 | 1.6 | 0.01 |  |
| Tube dispersion | ES1 | 2.71e-06 | 2.76e-06 | 2.80e-06 | 3.1 | 0.02 |  |
| Tube cross-section area | ES1 | 1.87e-05 | 1.87e-05 | 1.89e-05 | 1.1 | 0.01 |  |
| Column dispersion | ES2 | 5.00e-07 | 5.27e-07 | 5.38e-07 | 7.1 | 0.07 |  |
| Column porosity | ES2 | 0.330 | 0.334 | 0.339 | 2.7 | 0.02 |  |
| Tube dispersion | ES2 | 2.46e-06 | 2.63e-06 | 3.09e-06 | 23.8 | 0.25 |  |
| Tube cross-section area | ES2 | 1.79e-05 | 1.91e-05 | 1.94e-05 | 7.5 | 0.10 |  |

Table 13. Dextran experiment metrics

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 181,376 | 264,448 |
| Chain length | 1417 | 2066 |
| Simulation time | 6:21:52 | 8:45:33 |
| max(IAT) | 28 | 41 |
| Acceptance ratio | 0.171 | 0.129 |

Table 14. Non-binding experiment results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Source | LB HDI | MAP | UB HDI | % MAP | max(%MCSE) | Units |
| Film diffusion | ES1 | 1.76e-06 | 2.19e-06 | 2.52e-06 | 34.9 | 0.37 |  |
| Particle porosity | ES1 | 0.319 | 0.327 | 0.333 | 4.2 | 0.04 |  |
| Particle diffusion | ES1 | 2.15e-11 | 2.21e-11 | 2.33e-11 | 8.3 | 0.10 |  |
| Column dispersion | ES1 | 5.09e-07 | 5.17e-07 | 5.27e-07 | 3.6 | 0.03 |  |
| Column porosity | ES1 | 0.332 | 0.337 | 0.337 | 1.5 | 0.01 |  |
| Tube dispersion | ES1 | 2.71e-06 | 2.76e-06 | 2.79e-06 | 2.9 | 0.02 |  |
| Tube cross-section area | ES1 | 1.87e-05 | 1.87e-05 | 1.88e-05 | 1.0 | 0.01 |  |
| Film diffusion | ES2 | 1.44e-06 | 1.69e-06 | 2.39e-06 | 56.1 | 0.75 |  |
| Particle porosity | ES2 | 0.316 | 0.326 | 0.335 | 5.6 | 0.12 |  |
| Particle diffusion | ES2 | 2.19e-11 | 2.46e-11 | 2.52e-11 | 13.1 | 0.19 |  |
| Column dispersion | ES2 | 5.02e-07 | 5.01e-07 | 5.36e-07 | 6.8 | 0.06 |  |
| Column porosity | ES2 | 0.331 | 0.337 | 0.340 | 2.5 | 0.03 |  |
| Tube dispersion | ES2 | 2.52e-06 | 3.07e-06 | 3.09e-06 | 18.5 | 0.21 |  |
| Tube cross-section area | ES2 | 1.79e-05 | 1.81e-05 | 1.92e-05 | 7.2 | 0.09 |  |

Table 15. Non-binding experimental results

|  |  |  |
| --- | --- | --- |
| Name | ES1 | ES2 |
| Chain simulations | 1,689,216 | 2,046,080 |
| Chain length | 13197 | 15985 |
| Simulation time | 2 days, 7:43:25 | 2 days, 3:48:41 |
| max(IAT) | 263 | 319 |
| Acceptance ratio | 0.058 | 0.048 |

Shape, square

Description automatically generated

Figure 1: Volume ratio of unit N-sphere to containing N-cube on linear (A) and log (B) ordinates.

Logo

Description automatically generated

Figure 2: Film diffusion vs pore diffusion posterior distribution plot for the ES1 error source

Graphical user interface, application

Description automatically generated

Figure 3: ES1 error source binding pulse posterior plot subset

Graphical user interface, application

Description automatically generated

Figure 4: ES2 error source binding pulse posterior plot subset

Logo, company name

Description automatically generated

Figure 5: Plots of the error model and posterior chromatograms colored by relative probability. A: ES1 error model, B: ES2 error model, C: ES1 posterior with ground truth shown in black, D: ES2 posterior with ground truth shown in black

Chart

Description automatically generated with low confidence

Figure 6: Plots of the ES1 error model and posterior chromatograms colored by relative probability. A and B: ES1 error model experiments 1 and 2; C and D: ES1 posterior with experimental data shown in black for experiments 1 and 2

A picture containing icon

Description automatically generated

Figure 7: Plots of the ES2 error model and posterior chromatograms colored by relative probability. A and B: ES2 error model experiments 1 and 2; C and D: ES2 posterior with experimental data shown in black for experiments 1 and 2

[Figure 1: Volume ratio of unit N-sphere to containing N-cube on linear (A) and log (B) ordinates.](#_Toc135392684)

[Figure 2 Film diffusion vs pore diffusion posterior distribution plot for the ES1 error source](#_Toc135392685)

[Figure 3 ES1 error source binding pulse posterior plot subset](#_Toc135392686)

[Figure 4 ES2 error source binding pulse posterior plot subset](#_Toc135392687)

[Figure 5 Plots of the error model and posterior chromatograms colored by relative probability A: ES1 Error Model B: ES2 Error Model C: ES1 Posterior with ground truth shown in black D: ES2 Posterior with ground truth shown in black](#_Toc135392688)

[Figure 6 Plots of the ES1 error model and posterior chromatograms colored by relative probability A and B: ES1 error model experiment 1 and 2 C and D: ES1 posterior with experimental data shown in black for experiment 1 and 2](#_Toc135392689)

[Figure 7 Plots of the ES2 error model and posterior chromatograms colored by relative probability A and B: ES2 error model experiment 1 and 2 C and D: ES2 posterior with experimental data shown in black for experiment 1 and 2](#_Toc135392690)

[Table 1 Error source description for ES1 and ES2 error sources 26](#_Toc134354269)

[Table 2 Synthetic Bypass experiment results 27](#_Toc134354270)

[Table 3 Synthetic Bypass experiment metrics 29](#_Toc134354271)

[Table 4 Synthetic Non-Pore Penetrating experiment results 30](#_Toc134354272)

[Table 5 Synthetic Non-Pore Penetrating experiment metrics 32](#_Toc134354273)

[Table 6 Synthetic Non-Binding experiment results 33](#_Toc134354274)

[Table 7 Synthetic Non-Binding experiment metrics 35](#_Toc134354275)

[Table 8 Synthetic Linear isotherm binding experiment results ES1 36](#_Toc134354276)

[Table 9 Synthetic Linear isotherm binding experiment metrics 38](#_Toc134354277)

[Table 10 Bypass experiment results 39](#_Toc134354278)

[Table 11 Bypass experiment metrics 41](#_Toc134354279)

[Table 12 Dextran experiment results 42](#_Toc134354280)

[Table 13 Dextran experiment metrics 44](#_Toc134354281)

[Table 14 Non-Binding experiment results 45](#_Toc134354282)

[Table 15 Non-Binding experimental results 47](#_Toc134354283)