Comparison of Frequentist and Bayesian Confidence Analysis Methods on a Viscoelastic Stenosis Model

Zackary R. Kenz, H.T. Banks, and Ralph C. Smith
Center for Research in Scientific Computation
and
Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8212

April 19, 2013

Abstract
We compare the performance of three methods for quantifying uncertainty in model parameters: asymptotic theory, bootstrapping, and Bayesian estimation. We study these methods on an existing model for one-dimensional wave propagation in a viscoelastic medium, as well as corresponding data from lab experiments using a homogeneous, tissue-mimicking gel phantom. In addition to parameter estimation, we use the results from the three algorithms to quantify complex correlations between our model parameters, which are best seen using the more computationally expensive bootstrapping or Bayesian methods. We also hold constant the parameter causing the most complex correlation, obtaining results from all three methods which are more consistent than when estimating all parameters. Concerns regarding computational time and algorithm complexity are incorporated into discussion on differences between the frequentist and Bayesian perspectives.

Mathematics Subject Classification:

Key words: viscoelastic model; asymptotic theory; bootstrapping; Bayesian; MCMC; DRAM.

1 Introduction
The need to quantify the accuracy of model parameter values determined during model calibration has become increasingly important. In practice, this is done in the frequentist approach by quantifying the sampling distribution (which is an uncertainty statement about the estimator for an assumed “true” parameter vector) or in the Bayesian approach by estimating the parameter posterior density which provides uncertainty information (which is a statement about the estimation of an assumed underlying parameter density). The methods used to understand parameter uncertainty make fundamentally different assumptions, which we will later examine in more detail. Differences between model prediction and measurements result from the fact that not only are models not perfectly descriptive of underlying phenomena (due to modeling error), but data (e.g., from lab experiments) inherently has measurement error. Thus, when working with mathematical models and the model parameters used in attempting to describe real-life phenomenon, we need to consider the methods by which we describe the uncertainty in the parameter estimates which “best” match the data. To this end, we will examine different inverse uncertainty quantification methods in the context of a viscoelastic wave propagation model developed in [3, 4]. This provides us with a nonlinear, partial differential equation model of a complex phenomenon whereby we can examine the performance of the different methods on an active research problem.

In [4], a one-dimensional model for the propagation of waves in a viscoelastic medium was developed. The model incorporated Hooke’s law, an overall damping parameter, and a hysteresis integral composed...
of “internal variables” which describe microscopic responses to stresses and strains to the material under study. Corresponding data was gathered from a novel acoustic phantom lab experiment. This data was used to estimate the material and loading process parameters which best fit the data. In a general framework, errors between the model prediction and data values are due to model discrepancies and to errors when taking measurements. In the case of interest here, we assume that our model describes the system behavior well enough so that the main source of error in the data is measurement error (error which is natural and inherent in any experiment). This assumption is supported by an analysis of the residuals between model predictions and the data, since the residuals appear to be independent and identically distributed (iid). Absolute and relative error models were considered in [4], necessitating different least squares computational algorithms. Confidence analysis using (frequentist) asymptotic error theory was computed for both error models. Throughout, we had good success in matching the model to data in the least squares framework and were able to provide a measure of the quality of our estimation procedure based on estimating the sampling distribution.

In this work, we take the uncertainty analysis another step, comparing three different methods. The previously used asymptotic error theory is a frequentist method, meaning we have assumed that a single true parameter value exists and are trying to estimate that value and then study uncertainty in the estimator. Another frequentist method is bootstrapping, which tests the robustness of the parameter values by taking the residuals between the model predictions and data points, randomly mixing the residuals across the time points to create new “simulated” data sets, then solving the inverse problem on the simulated data sets in order to obtain new parameter estimates. These are compiled, and statistics such as mean and variance are computed which then describe the uncertainty in the parameter estimates. In both frequentist cases, the uncertainty is related to the so-called “sampling distribution,” which is a measure of how well the estimation method performs in quantifying the parameter values for which the corresponding model solution fits the data under consideration.

In a fundamentally different approach, Bayesian methods regard the underlying parameters as random variables with associated densities and attempt to construct these parameter densities directly. A random, Markovian walk is built which steps through the parameter space (often initialized with a least squares or maximum likelihood estimate) which accepts parameter values based on their closeness to the data. This approach, along with the frequentist methods, will be briefly described later in this document.

Though one may be concerned with issues like propagation of parameter uncertainty though the model solution and subsequent model output predictive intervals, we are concerned primarily here with the uncertainty in the parameter values themselves. Thus, when comparing algorithms we will focus on the following considerations:

1. Complexity of the algorithm;
2. Computational time considerations (including parallelization);
3. Insight into correlation between parameters; and
4. Ability to provide a density that can be subsequently propagated through models.

The asymptotic, bootstrapping, and Bayesian methods will be compared and contrasted in this work. The viscoelastic model and data provide an example of where these methods can be successful and also reveal some of the various drawbacks to each of the methods. Since these methods are developed from very different theory and under different basic assumptions about the parameter being estimated, they are hard to compare in the abstract; hence, results from the wave propagation modeling problem will be used to provide an example problem in active research for which we demonstrate the performance of the methods.
1.1 Mathematical and statistical models

The example problem we study here is the inverse problem of estimating \( \hat{q} = (E, E_1, \tau_1, \gamma_1, A, \Upsilon) \) in the pressure wave propagation model (see the technical report version of \([4]\))

\[
\rho \frac{\partial^2}{\partial t^2} u(x, t) - \frac{\partial}{\partial x} \sigma(x, t) = 0
\]

\[
u(0, t) = 0, \quad \sigma(L, t) = \frac{A}{L} \left[ \sum_{j=1}^{N_p} \gamma_j \exp(-t/\tau_j) - \sum_{j=1}^{N_p} \gamma_j \exp(-(t - \Upsilon)/\tau_j) \right]
\]

\[
u(x, 0) = \frac{A}{L} x, \quad \nu_t(x, 0) = 0,
\]

where

\[
\sigma(t) = \left( E + \sum_{j=1}^{N_p} \gamma_j \right) u_x(t) + E_1 u_{xt}(t) - \sum_{j=1}^{N_p} \gamma_j e^j(t),
\]

with the internal variables subject to

\[
\tau_j \frac{d}{dt} e^j(t) + e^j(t) = u_x, \quad e^j(0) = 0,
\]

for \( j = 1, 2, \ldots, N_p \). The solution to (1.1) is then given by \( u(x, t; \hat{q}) \), where \( 0 \leq x \leq L, 0 \leq t \leq T, L = 0.051 \) and \( T \) is some finite final evaluation time. The parameter \( \rho = 1010 \) represents the material density. The macroscopic damping parameter is \( E_1 \), and \( E \) represents an instantaneous relaxation modulus. The \( \gamma_j \) values are weightings on relaxation times \( \tau_j \). Based on previous findings with experimental data \([3, 4]\), we take \( N_p = 1 \). Data was taken at the top position \( (x = L) \) on a phantom composed of tissue-mimicking gel and compared with the model solution (the “inverse problem”) to find parameters which produced a model solution that minimized the least squares distance to the data points.

In all the discussion which follows, we will actually be estimating the log-scaled parameters, since the parameter values are on various orders of magnitude. This means that we will use as our sought-after parameters

\[
q = (\log_{10}(E), \log_{10}(E_1), \log_{10}(\tau_1), \log_{10}(\gamma_1), \log_{10}(-A), \log_{10}(-\Upsilon)).
\]

The notation \( \tilde{q} = 10^q \) signifies that the parameter values enter the model in their original scale and with proper sign so that \( A, \Upsilon < 0 \). We assume that \( q \in Q \), where \( Q \) is some compact set of admissible parameters. We enforce the bounds \(-15 \leq q_i \leq 15 \) for \( i = 1, \ldots, 5 \) and \(-15 \leq q_6 \leq \log_{10}(20) \). As discussed in \([4]\), the upper bound on \( q_6 = \log_{10}(-\Upsilon) \) is due to modeling considerations.

The specific inverse problem method we use is here the ordinary least squares (OLS). Based on an analysis of residual plots, the lab measurements are considered the most significant source of error. We assume these are additive, independent, and identically distributed errors, which leads to the error process \( U_j = u(L, t_j; 10^{q_0}) + \mathcal{E}_j \). This has realizations

\[
u_j = u(L, t_j; 10^{q_0}) + \varepsilon_j,
\]

where \( q_0 \) is some hypothesized “true” parameter value (this common assumption of a true parameter value underlies relevant frequentist theory; see, e.g., \([5, 13]\)). With this error model, we use the ordinary least squares cost function

\[
\mathcal{J}_{ols}(q) = \sum_{j=0}^{n-1} [u_j - u(L, t_j; 10^q)]^2
\]

which yields the estimate

\[
\hat{q}_{ols} = \arg \min_{q \in Q} \mathcal{J}_{ols}(q) = \arg \min_{q \in Q} \sum_{j=0}^{n-1} [u_j - u(L, t_j; 10^q)]^2.
\]

In the frequentist case, we assume that the errors are iid, have mean zero (i.e., \( E[\mathcal{E}_j] = 0 \)), and constant variance \( \text{var}(\mathcal{E}_j) = \sigma^2_0 \), but do not need to assume a particular distributional form. In the Bayesian case, we
will make the stronger assumption of normally distributed errors, each with the same mean and variance so that they are also identically distributed.

The function $J_{\text{ols}}(q)$ minimizes the distance between the data $\bar{u} = \{u_j\}_{j=0}^{n-1}$ and model where all data observations are considered to have equal importance or weight. Since $u(L, t_j; 10^9)$ is a continuous function of $q$, $J_{\text{ols}}(q)$ is also a continuous function of $q$, which means we are minimizing a continuous function of $q$ over a compact set $Q$, and thus this inverse problem has a solution. Other error models exist (e.g., relative error is discussed in [4]) which would require a different formulation for the cost function. Here we use additive error because it was shown [4] to be a reasonable assumption for the pressure data and because it simplifies the resulting algorithms so we can focus more on comparisons across methods rather than being concerned with notation and more complex estimation algorithms. In [4], this inverse problem was solved in a frequentist manner using ordinary least squares (and also generalized least squares when a relative error statistical model was investigated), with confidence intervals then computed using asymptotic error theory.

The issue of independent errors requires some discussion. In practice, measurements may be taken so closely that neighboring data points are related, either through the material response (for example, taking measurements near the peak values of an oscillating spring) or measurement device itself (e.g., hysteresis in the mechanism). This was studied for the current situation in [4], and we found that data frequency had some effect on the independence of the residuals. We will use a data measurement frequency of 1024 Hz in the present work, which should be sufficiently frequent to provide useful information for the inverse problem but sufficiently infrequent that the data points can be assumed independent.

We note that the OLS inverse problem must be initiated with a guess for the parameter values. By examining the literature and previous fittings to the data, we chose the initial values $E = 4.5 \times 10^4$ Pa, $E_1 = 55$ Pa-s, $\gamma_1 = 1.9 \times 10^5$ Pa, $\tau_1 = 0.05$ s, $A = -1.75 \times 10^{-4}$ m, and $T = -0.01$ s. These are all physically reasonable; for more information, see the model development in [4]. The log-scaled values are

$$q_{\text{init}}^{\text{ols}} = (4.6532, 1.7404, 5.2788, -1.3010, -3.7570, -2)^T.$$

2 Methods for studying confidence in parameters

On an earlier, nearly equivalent, version of our viscoelastic model, we studied and compared in [3] the results of the frequentist methods. We found that the asymptotic theory was comparable to the bootstrap method, and thus the asymptotic theory was preferable since it requires significantly less computational time. In addition to updating the model to the form (1.1), we now consider Bayesian methods, since this approach directly provides densities for the parameters rather than for the sampling distribution. This distinction is particularly important if one is concerned with propagating parameter uncertainty through the model in order to provide solution confidence and/or predictive intervals; in that case, the parameter densities obtained from Bayesian methods must be used since propagating uncertainty requires direct knowledge of the parameters rather than just knowledge of the sampling distribution. We will give descriptions of the algorithms for each method below (with some additional theoretical background for the Bayesian methods), referring to specific references which provide more detailed information on the development of each method.

Note that in all cases, we assume that we have already computed the OLS estimate $q_0 = \hat{q}_\text{ols}$ by solving (1.3). We have computed the estimate using the Matlab Optimization Toolbox routine \texttt{lsqnonlin}, which was found to be the most effective when working with the parameters in our problem [3]. This initial OLS estimate is common to all three methods, and as such this initial step will not be considered in later reporting on computational times. We comment that a common, pre-computed initial OLS estimate is not necessary for all Bayesian methods one might consider. In particular, parallel methods such as DREAM [17, 19, 20] are designed to be global optimizers as well as quantify the parameter densities. A different comparison of computational times would be needed when using these methods.

2.1 Asymptotic analysis (frequentist)

Discussion on the development of the asymptotic theory can be found in [5, 13] and the references therein. For our purposes, we need only to describe the algorithm used to compute the asymptotic confidence intervals. This incorporates the sensitivity equations, which are the partial derivatives of the model with respect to
the parameter values; these equations can be found in the appendix of the technical report version of [4]. The algorithm (see, e.g., [2, 3, 4, 5, 13]) is then the following:

**Algorithm 2.1** (Asymptotic analysis).

1. Find $\hat{q} = q_{ols}$ using (1.3).

2. Compute the sensitivity equations to obtain $\frac{\partial u(x, t; 10^j)}{\partial q_k}$ for $k = 1, 2, \ldots, p$ where $p = 6$ is the number of parameters under consideration. Data is only taken at position $x = L$, so the sensitivity matrix is

$$\chi_{j,k}(\hat{q}) = \frac{\partial u(L, t_j; 10^j)}{\partial q_k}, \quad j = 0, 1, \ldots, n - 1, k = 1, 2, \ldots, p.$$

Note that $\chi(\hat{q})$ is then an $n \times p$ matrix. We also compute an approximation for the constant variance as $\hat{\sigma}^2 = \frac{1}{n - p} \sum_{j=0}^{n-1} [u_j - u(L, t; 10^j)]^2$.

3. Asymptotic theory yields that the estimator $Q$ for $q$ is asymptotically (as $n \to \infty$) normal with mean approximated by $\hat{q}$ and covariance matrix approximated by

$$\text{Cov}(Q) \approx \hat{\Sigma} = \hat{\sigma}^2 \chi^T(\hat{q}) \chi(\hat{q})^{-1}.$$

4. The standard errors for each element in the parameter estimator can be approximated by

$$SE(Q_k) = \sqrt{\hat{\Sigma}_{kk}}, \quad k = 1, 2, \ldots, p,$$

where $Q_k$ is the $k$th element of $Q$ and $\hat{\Sigma}_{kk}$ is the $(k, k)^{th}$ entry of the matrix $\hat{\Sigma}$. Hence, the endpoints of the confidence intervals are given by

$$\hat{q}_k \pm t_{1-\alpha/2}^{n-p} SE(Q_k)$$

for $k = 1, 2, \ldots, p$. The value $t_{1-\alpha/2}^{n-p}$ is determined from a table for the Student’s t-distribution based on the number of data points $n$, the number of parameters $p$, and the level of significance $\alpha$. Here we choose the level of significance to be 95%, so that $\alpha = 0.95$.

This algorithm gives 95% confidence intervals for each component of the parameter vector. This is a statement about the estimation procedure: if the experiment were re-run 100 times then the confidence intervals obtained from approximately 95 of those experiments could be expected to contain the true parameter value $q_0$. Note that the theory is asymptotic, so it is only guaranteed to hold for large values of $n$. This may be quite suspect in many real-life applications since $n$ may be somewhat small. Also, in the development of the theory, linearizations are made which then makes asymptotic theory invalid in regions of high nonlinearity. Finally, asymptotic theory constructs confidence intervals as combinations of Gaussians, which is a possible limitation.

### 2.2 Bootstrapping (frequentist)

Here we again only discuss the algorithm for bootstrapping. Some theory is discussed in [2, 7], and examples of computing bootstrap confidence intervals are included in [2, 3]. Though it is usually much more computationally intensive than asymptotic theory (due to the fact that we will solve the inverse problem an additional $M$ times), the attraction of bootstrapping is that there are fewer assumptions like linearity which are likely to be violated in practice. Thus, the extra computational burden can allow for more certainty (in general) that the confidence intervals obtained are reasonable. We repeat here the bootstrapping algorithm from [2, 3]:

---

5
Algorithm 2.2 (Bootstrapping).

1. Find $\hat{q} = q_{ols}$ using (1.3).

2. Define the standardized residuals to be
   \[ \bar{r}_j = \sqrt{\frac{n}{n-p}} (u_j - u(L,t_j;10^\hat{q})) \]
   for $j = 0, 1, \ldots, n-1$. Set $m = 0$.

3. Create a sample of size $n$ by randomly sampling, with replacement, from the set of standardized residuals $\{\bar{r}_j\}_{j=0}^{n-1}$ to form a bootstrap sample $\{r^m_0, \ldots, r^m_{n-1}\}$.

4. Create bootstrap sample points (or, simulated data)
   \[ u^m_j = u(L,t_j;10^\hat{q}) + r^m_j \]
   for $j = 0, 1, \ldots, n-1$.

5. Solve the OLS minimization problem (1.3) with the bootstrap-generated data $\{u^m_j\}_{j=0}^{n-1}$ to obtain a new estimate $\hat{q}^m$, which we store.

6. Set $m = m + 1$ and repeat steps 3-5. This iterative process should be carried out $M$ times where $M$ is large (we used $M = 1000$ as suggested by [2]). This will give $M$ estimates $\{\hat{q}^m\}_{m=1}^M$.

Upon completing all $M$ simulation runs, we can compute the mean, covariance, and standard error for the bootstrap estimator $Q_{boot}$ as:

\[
\hat{q}_{boot} = \frac{1}{M} \sum_{m=1}^{M} \hat{q}^m,
\]

\[
\hat{\Sigma}_{boot} = \frac{1}{M-1} \sum_{m=1}^{M} (\hat{q}^m - \hat{q}_{boot})(\hat{q}^m - \hat{q}_{boot})^T,
\]

\[
(SE_{boot})_k = \sqrt{\left(\hat{\Sigma}_{boot}\right)_{kk}}, \quad k = 1, 2, \ldots, p,
\]

where $\left(\hat{\Sigma}_{boot}\right)_{kk}$ is the $(k,k)^{th}$ entry of the covariance matrix $\hat{\Sigma}_{boot}$. Hence, the endpoints of the confidence intervals are given by

\[
(\hat{q}_{boot})_k \pm t_{1-\alpha/2}(SE_{boot})_k
\]

for $k = 1, 2, \ldots, p$.

Bootstrapping requires solving the inverse problem $M$ times. This is significantly more computationally expensive than solving the asymptotic sensitivity equations. Depending on how easily the inverse problem is solved, we will not know a priori whether bootstrapping or the to-be-discussed Bayesian methods will end up computing more forward solves. Regardless, asymptotic theory will be significantly faster to implement. Methods also exist [7] whereby one can obtain confidence intervals directly from the bootstrap distribution without needing to assume that the form of the sampling distribution is normal; however, here we use the form in (2.1) since it is simple and effective for our case.

2.3 Bayesian parameter estimation and confidence analysis

In a Bayesian framework, one assumes that the parameters are random variables with associated densities. Initially, the parameters are described by a prior density $\pi_0(q)$; in this work, we use a "noninformative" prior (of course, we still implement bounds on parameter values). We then use the data realizations $\vec{u}$ to compute a posterior density $\pi(q|\vec{u})$. The data values are incorporated through a likelihood function $\pi(\vec{u}|q)$. In this work, we will assume the measurement errors are normally distributed, so that the likelihood function becomes the multivariate normal density. Note that this assumes a specific form for the measurement error process density; the frequentist methods only required we specify the first two moments, instead of the entire error distribution. To solve the inverse problem in this framework, we use "Bayes’ theorem for inverse problems" which is given by the following statement:
Definition 2.3 (Bayes’ theorem for inverse problems, referred to as such by [11]). We assume that the parameter vector $q$ is a random variable which has a known prior density $\pi_0(q)$ (possibly noninformative), and corresponding realizations $\vec{u}$ of the random variable $\vec{U}$ associated with the measurement process. The posterior density of $q$, given measurements $\vec{u}$, is then

$$
\pi(q|\vec{u}) = \frac{\pi(\vec{u}|q)\pi_0(q)}{\int_{\mathbb{R}^p} \pi(\vec{u}|q)\pi_0(q) dq}
$$

(2.2)

where we have assumed that the marginal density $\pi(\vec{u}) = \int_{\mathbb{R}^p} \pi(q, \vec{u}) dq = \int_{\mathbb{R}^p} \pi(\vec{u}|q)\pi_0(q) dq \neq 0$ (a normalizing factor) is the integral over all possible joint densities $\pi(q, \vec{u})$. Note here that $\pi(\vec{u}|q)$ is a likelihood function which describes how likely a data set $\vec{u}$ is when given a model solution at the parameter value $q$.

We could solve the inverse problem directly, using cubature or Monte Carlo techniques to compute the integral in the definition. This is computationally viable only for a small number of parameters – as the number of parameters becomes larger, the space over which the integral must be evaluated becomes so large as to be computationally prohibitive. The alternate method, which we discuss here, is to build a Markov chain whose stationary distribution is the posterior density $\pi(q|\vec{u})$ of (2.2). This is a method in which we sample the parameter space, accepting the parameters based on closeness of the model solution at the parameters to the data. It is known (see, e.g., [15]) that a Markov chain defined by the random walk Metropolis algorithm will converge if the chain is run sufficiently long (i.e., if we allow the algorithm to sample the parameter space, accepting the parameters based on closeness of the model solution at the parameters to the data). There are no analytic results to confirm one has run the chain sufficiently long (i.e., if we allow the algorithm to sample the parameter space a large number of times). A chain is a set of parameter values, begun with an initial guess $q_0$, that results from the random walk. There are no analytic results to confirm one has run the chain sufficiently long for convergence, so one must run chains long enough to ensure that they are sampling the posterior density (this is called mixing). Early chain values are considered a burn-in phase, which are not included in the parameter density results. The burn-in and total chain values are highly problem-dependent; we will run $M = 50,000$ chain values in this work with varying lengths of burn-in depending on how quickly the chain seems to visually settle into a random walk. For each individual problem, one must balance between long enough chains to hopefully ensure that the posterior density is adequately sampled while also keeping the runtime reasonable. More discussion on this issue can be found throughout the literature, for example in [1, 14, 15].

We briefly discuss the meaning of the prior term, $\pi_0(q)$. If any information is known about the parameters beforehand, perhaps if they are is known to follow a particular distribution, this information can be incorporated via the prior and result in faster convergence. However, it is known (and demonstrated in, for example, the forthcoming manuscript [14]) that a poor choice for the prior can result in much slower convergence of the posterior. Besides placing bounds on the parameter values, we know nothing else about the parameter densities, so we take a noninformative prior $\pi_0(q) = H(0)$ where $H(0)$ is the Heaviside step function with step at 0. For discussion purposes, though, we will leave $\pi_0$ in the formulations below.

We now turn to a discussion of the Metropolis algorithm which is used to create the Markov chain of parameters sampled from the posterior distribution. We will describe and implement here a delayed rejection adaptive Metropolis (DRAM) algorithm; for more general details on the background and algorithm see [1, 8, 9, 15]. We must begin the chain at some initial parameter value with positive likelihood; this will be $q_0 = q_{OLS}$, the solution to the ordinary least squares pressure inverse problem (1.3). To step through the parameter space, we require a proposal distribution $J(q^*|q^{k-1})$ which provides a new sample parameter $q^*$ that depends only on the previously sampled parameter $q^{k-1}$. We also must form a probability $\alpha(q^*|q^{k-1})$, dependent on the prior density $\pi_0(q)$ and likelihood function $\pi(\vec{u}|q)$, with which we accept or reject the new parameter value.

We will use a normally distributed proposal function $J(q^*|q^{k-1}) = \mathcal{N}(q^{k-1}, V_0)$, where $V_0$ is an estimate for the covariance matrix at $q_0$. In this case we use the standard asymptotic theory estimate for the covariance matrix, namely $V_0 = \sigma_{ols}^2 [\chi^T(q_{ols})\chi(q_{ols})]^{-1}$, with $\sigma_{ols}^2 = \frac{1}{n-p} \sum_{j=0}^{n-1} (u_j - u(L, t; q_{ols}))^2$ and $
 \chi_{jk}(q) = \frac{\partial u(L, t; j; q)}{\partial q_k}$ where $n$ is the number of data points and $p = 6$ is the number of parameters being estimated. This choice for $V_0$ hopefully ensures that the shape of the proposal function distribution matches,
to some extent, the shape of the posterior distribution. Adaptive methods change the proposal covariance $V_k$ (where $V_0$ is the initial covariance and later matrices are denoted $V_k$) in a prescribed manner depending on the previous states. Since a general adaptation breaks the Markov property of the chain (the adaptive $V_k$ will depend on more than just the preceding state), it must be carefully constructed and abide by a condition that requires the adaptation to decrease as the chain progresses in order to retain convergence of the chain to the posterior density. Particular definitions of adaptation are beyond the scope of this document and more information can be found in [1, 8, 9, 12, 15, 18].

As for the acceptance of a candidate $q^*$, we focus first on defining the acceptance probability $\alpha$. We form the ratio of the likelihoods of the new parameter $q^*$ with that of the preceding parameter $q^{k-1}$ as

$$r(q^*|q^{k-1}) = \frac{\pi(\theta|q^*)\pi_0(q^*)}{\pi(\theta|q^{k-1})\pi_0(q^{k-1})}.$$  

We then define $\alpha = \min(1, r)$ and set $q^k = q^*$ with probability $\alpha$; otherwise we reject $q^*$ and set $q^k = q^{k-1}$. Rejections occur when the parameter is less likely (quantified through $r$). This form for $\alpha$ is known to provide the properties necessary for the chain to properly converge. However, we know that the proposed $q^*$ is dependent on the choice of $V$. Even when adaptive methods are considered, it may sometimes be better to delay rejection and construct an alternative candidate. In algorithms using delayed rejection, if $q^*$ is rejected a second stage candidate $q^{*2}$ is found using the proposal function $J(q^{*2}|q^{k-1}, q^*) = \mathcal{N}(q^{k-1}, \gamma V_k)$. Here, $V_k$ is the current adaptive covariance matrix and $\gamma_2 < 1$ ensures that the second stage proposal function is narrower than the original. This can be carried on for as many stages as desired, though we worked with just two stages based on the default in the software we use. Thus, delayed rejection is an open loop mechanism that alters the proposal function in a predetermined manner to improve mixing in the parameter space. Recall that mixing is the ability of the algorithm to properly, randomly sample the parameter space.

At this point, we have described the way in which we construct a (local) random walk using a proposal function $J$ and acceptance probability $\alpha$ to sample the posterior density and thus solve (2.2). We introduced adaptation, which provides feedback to the proposal function based on the chain to that point, and delayed rejection. We now assume that the measurement errors are iid, additive, and normally distributed. This means that the likelihood function is defined as

$$\pi(\theta|q^2) = \frac{1}{(2\pi\sigma^2)^{n/2}}e^{-\sum_{j=0}^{n-1} |u_j - \mu(L, t_j; q)|^2 / 2\sigma^2}.$$  

Note that this likelihood is dependent on $\sigma^2$, for which we have an estimate $\sigma^2_{ols}$. However, this parameter can also be treated in a Bayesian framework, namely by assuming it has a density which we sample through realizations of another Markov chain. The particular likelihood we have defined means $\sigma^2$ is in the inverse-gamma family (the reason is discussed in, e.g., [14, 15]), so that we have

$$\sigma^2|(\theta, q) \sim \text{Inv-gamma} \left( n_s + n, \frac{n_s \sigma^2_{ols} + \sum_{j=0}^{n-1} |u_j - \mu(L, t_j; q)|^2}{2} \right)$$

where $n_s$ represents the number of observations encoded in the prior (which for us means $n_s = 1$ since we have the single estimate for $\sigma^2_{ols}$) and $\sigma^2$ is the mean squared error of the data observations (for which a reasonable choice [14] is just $\sigma^2 = \sigma^2_{ols}$).

The combined algorithm we use is delayed rejection adaptive Metropolis, or DRAM, with the assumption of iid, additive, normally distributed measurement errors. We now summarize the DRAM algorithm (with $SS_q = J_{ols}(q)$ throughout):

**Algorithm 2.4** (Delayed Rejection Adaptive Metropolis).

1. Set design parameters (e.g., the adaptation interval).
2. Determine $q^0 = q_{ols} = \arg\min_q SS_q$ (i.e., using (1.3)).
3. Set the initial sum of squares $SS_{q^0}$. 

8
4. Compute the initial variance estimate \( \sigma^2_0 = \frac{1}{n-p} SS_{q^0} \).

5. Construct the covariance estimate \( V_0 = \sigma^2_0 [\chi^T(q^0) \chi(q^0)]^{-1} \) and compute \( R_0 = \text{chol}(V_0) \), the Cholesky decomposition of \( V_0 \).

6. For \( k = 1, \ldots, M \)
   
   (a) Sample \( z_k \sim \mathcal{N}(0, 1) \).
   
   (b) Construct candidate \( q^* = q^{k-1} + R_k z_k \).
   
   (c) Sample \( u_2 \sim \mathcal{U}(0, 1) \), where \( \mathcal{U}(0, 1) \) is the uniform distribution on (0,1).
   
   (d) Compute \( SS_{q^*} \).
   
   (e) Compute \( \alpha(q^*|q^{k-1}) = \min \left(1, e^{-[SS_{q^*} - SS_{q^{k-1}}]/2\sigma^2_{k-1}}\right) \).
   
   (f) If \( u_2 < \alpha \), set \( q^k = q^* \), \( SS_{q^k} = SS_{q^*} \). Otherwise, enter the delayed rejection procedure ([14, Algorithm 5.10]):
      
      i. Set the design parameter \( \gamma_2 \).
      
      ii. Sample \( z_k \sim \mathcal{N}(0, 1) \).
      
      iii. Construct second-stage candidate \( q^{*2} = q^{k-1} + \gamma_2 R_k z_k \).
      
      iv. Sample \( u_2 \sim \mathcal{U}(0, 1) \).
      
      v. Compute \( SS_{q^{*2}} \).
      
      vi. Compute \( \alpha_2(q^{*2}|q^{k-1}, q^*) \), where \( \alpha_2 \) is a modified acceptance criterion described in Sec. 5.6.2 of [14].
      
      vii. If \( u_2 < \alpha_2 \), set \( q^k = q^{*2} \), \( SS_{q^k} = SS_{q^{*2}} \). Otherwise, set \( q^k = q^{k-1} \), \( SS_{q^k} = SS_{q^{k-1}} \).
   
   (g) Update \( \sigma^2_k \sim \text{Inv-gamma} \left((n_s + n)/2, (n_s\sigma^2_{k-1} + SS_{q^k})/2\right) \).
   
   (h) If at an adaptation interval (for us, every 10 chain values), compute adaptive update for proposal covariance \( V_k \), then set \( R_k = \text{chol}(V_k) \); the adaptive update is computed recursively and using the preceding chain values.

We will later depict the MCMC chains (the sequence of parameter values accepted by the algorithm) which result from Algorithm 2.4, along with plots of the (posterior) parameter density. We used the DRAM options of the MCMC toolbox for Matlab, available from Marko Laine at [10] The density plots are created using kernel density estimation (KDE), which constructs a density without a specified structure by weighted sums of a particular defined density. This is not unlike a finite element or polynomial approximation to a function. KDE software for Matlab is available from the Mathworks File Exchange at [6].

2.4 General comparison between methods

We first discuss the complexity of the algorithms. The asymptotic algorithm is by far the least complex. At its core, the asymptotic theory linearizes about the true parameter value \( q^0 \) to obtain an estimate for the covariance matrix. The estimate \( \hat{q} \) for \( q^0 \) obtained by (1.3) and a corresponding approximation \( \hat{\sigma}^2 \) for \( \sigma^2 \) are then used, along with the sensitivity equations, to provide an estimate for the covariance matrix for \( \hat{q} \) (with specifics described previously in Algorithm 2.1). One important point of consideration is that the asymptotic theory is only able to construct confidence intervals that are combinations of Gaussian densities. Thus, if the density is multimodal or non-Gaussian, asymptotic theory will not be successful; there is no indication from asymptotic results that the form is anything other than Gaussian. The simplicity and speed of the asymptotic theory comes with the price of possibly reduced accuracy.

Bootstrapping does not make the linearization assumption. There, we resample from the error distribution (as approximated by the standardized residuals), generate simulated data, and obtain the \( M \) bootstrap estimates for \( \hat{q} \). Nowhere in this process do we linearize; hence, compared with asymptotic theory, bootstrapping is better able to accurately incorporate correlation between parameters, particularly nonlinear correlation.
The MCMC DRAM algorithm also does not make linearization assumptions, using the likelihood function in order to accept or reject parameter candidates and construct the posterior parameter density. With DRAM, we can create pairwise plots of the components of the accepted parameter estimates to directly examine parameter correlations. Thus, we obtain the most information about the parameter estimate when using the Bayesian method since it constructs a density of the parameter itself rather than a sampling distribution. If one wishes to remain in a frequentist context, bootstrapping gives more accurate information about the confidence intervals than asymptotic theory. However, as one can see from the preceding algorithm descriptions, asymptotic theory is less complex than the other two methods. Thus, if one needs a quick estimate for the confidence in the parameter identification procedure, asymptotic theory may be the best choice. If one needs more or better information about the confidence in parameters, bootstrapping or Bayesian methods may be superior.

In terms of global optimization (finding the smallest possible cost function value), the Bayesian algorithm is naturally a (crude) global optimizer. The OLS estimation procedure we described earlier can possibly (and in practice is likely to) pick out a local minimum in the cost function which may not be the global minimum. Methods exist to make this process more global, including the obvious notion of starting the OLS procedure from different parameter values and selecting the result which has the lowest cost value. Though this is not necessarily our concern in this work, we mention this global property of the Bayesian algorithm as a reminder that the more complex algorithm may be able to reduce some of the work one may have otherwise done manually in order to find a good initial parameter guess to compute the OLS estimate.

In terms of computational time, asymptotic theory is significantly faster than the other two methods. One must be sure to have an accurate and reasonably fast way of solving for the sensitivity equations when using asymptotic theory, but that is the case in most problems of interest. Though we will later report that bootstrapping is faster than the Bayesian results for our model, this is not necessarily the case in general. First, bootstrapping can easily be parallelized by creating the $M$ bootstrap samples and then splitting up the $M$ inverse problems across multiple processor cores. Thus, if given enough processors, bootstrapping could be as fast as the asymptotic theory (of course, with significantly higher hardware requirements than the asymptotic theory). As for the Bayesian method, DRAM (Algorithm 2.4) is inherently serial since Markov chains in general are a serial process. However, successful methods for parallel Bayesian estimation have been developed and implemented including parallel DRAM [16] and Differential Evolution Adaptive Metropolis (DREAM) [17, 19, 20]. The DREAM algorithm employs multiple MCMC chains and combines them using differential evolution to improve the overall estimation procedure. Thus, though we will later report times for the serial bootstrapping and Bayesian estimation procedures, these times will be different when using more sophisticated (and complex) implementations.

Finally, regarding correlation between parameters, asymptotic theory performs acceptably well as long as there is minimal or linear correlation between parameters which can be represented by a multivariate normal. Due to the linearizations made when developing the theory, any more complicated correlation may result in spurious conclusions. We will see this effect later; $E$ and $\gamma_1$ in our model are known to be correlated and together influence the frequency of the oscillations in the wave. These parameters turn out to be correlated in a nonlinear way, which the asymptotic theory is unable to handle. Bootstrapping and Bayesian methods, on the other hand, are able to properly estimate the confidence in parameters even with some nonlinear correlations. Also, the Bayesian method allows us examine the correlations directly through pairwise plots of parameter components. As we will later see, the DRAM results clearly show nonlinear correlation between $E$ and $\gamma_1$.

Overall, the asymptotic analysis, though simple to implement, has its drawbacks in terms of being less able to properly incorporate complex relationships between model parameters. Booststrapping and Bayesian methods, though more intensive to implement, are able to better describe more complex behavior. If one wishes to know the most about the model parameters, the Bayesian approach may be the most successful as it directly estimates the posterior parameter density (and is in fact the only method to do so, as frequentist methods only estimate the sampling distribution). The parameter density can then be used to make the most accurate predictions of model solution behavior. The sampling distribution can only be used to predict the behavior of the model solution if one is convinced that the sampling distribution and the parameter distribution are the same for a particular problem. In general, there is no one superior method for all problems; rather, one must choose the method which best fits the situation at hand and the goals of a particular problem.
3 Computing parameter estimates and comparing methods

We now turn to using the methods discussed in the previous section on the stenosis model with experimental data. At first, we estimate all six parameters in \( q \). \( E \) and \( \gamma_1 \) are known to be correlated from the development of the model. We will examine how this manifests itself, and which methods handle the correlation properly. We will then fix \( E \) and estimate the remaining five parameters, in order to examine how removing the highly correlated parameter from consideration changes the results.

3.1 OLS inverse problem results

We begin by solving the inverse problem (1.3) from the initial condition \( q_{\text{ols}}^{\text{init}} \) specified in Section 1.1. The results are shown in the left column of Table 1. Note that we have refined the OLS results from what was presented in [4], though the values here differ from the previously reported results only slightly. We show the model solution, as well as residual plots, in Figure 1. The residuals appear random, which indicates the validity of our original assumption of iid errors due primarily to measurements. Our next step will be to run an MCMC chain using the OLS results as a starting place. We will also compute the asymptotic and bootstrap confidence analysis around this new OLS parameter value.

![Figure 1: Pressure data fit using OLS, with initial guess set to be the mean of the MCMC parameter chains, with data sampled at 1024 Hz, \( N_p = 1 \). (a) OLS model fit to data. (b) Model vs absolute residuals. (c) Time versus absolute residuals.](image)

3.2 Comparison between methods

The results of computing the asymptotic analysis using the OLS parameter values are reported in Table 1. We also compute the bootstrap confidence intervals around that OLS parameter estimate (using \( M = 1000 \) bootstrap samples). The bootstrapping results are shown in Table 2. Finally, we run the Bayesian DRAM algorithm (with \( M = 50,000 \)), reporting the chains in Figure 3. A comparison of the resulting densities is shown in Figure 2, and a comparison of the parameter estimate (OLS) and parameter means (bootstrapping and Bayesian) is shown in Table 3 along with computational times.

We see that the densities for all three methods are largely centered at the same location. The asymptotic theory gives the widest confidence band, which is actually quite unexpected. Since the asymptotic theory holds only for large numbers of data points (we only use \( n = 251 \) data points here) and incorporates a linearization, the use of asymptotic theory can result in too-small confidence intervals. The wider confidence intervals we found may be due to complex nonlinearities between parameters, which we will soon examine further. For the damping parameter \( E_1 \), bootstrapping and Bayesian methods give largely comparable results, with the asymptotic theory being wider. For the parameter \( A \), the results are roughly the same between methods – this is expected, again since that is the parameter most readily verified by directly from the data. This parameter is also largely uncorrelated with the remaining parameters, providing an indication that correlations may be causing problems with the asymptotic theory. For the remaining four parameters, the bootstrapping confidence bands were the narrowest, though more comparable to the Bayesian results than the asymptotic analysis. We also point out the ability of the DRAM algorithm to clearly show the nonlinear correlations, particularly between \( E \) and \( \gamma_1 \), is a significant achievement. It indicates that the DRAM algorithm is effective on this problem. Also, the chains in Figure 3 are mixing reasonably well after the burn-in phase, an indication that we are likely sampling adequately from the posterior density.
In this case of estimating all six parameters, in order to gain a sense of the uncertainty we would need to use either bootstrapping or Bayesian methods due to the parameter correlation which asymptotic theory is unable to accommodate. Even though the asymptotic results are much faster to compute, this situation necessitates the use of the more complex algorithms to obtain correct results. Whether one should use bootstrapping or Bayesian methods in practice on this problem depends on situational considerations (e.g., if the parameter uncertainty needs to be propagated through the model or whether we are just concerned with uncertainty in the estimation procedure). It is also of interest to see how these results change when the nonlinear correlation is no longer present. Without the nonlinear correlation, asymptotic theory may give improved performance; bootstrapping and asymptotic theory are then likely to appear comparable to the Bayesian densities if the parameter densities are normal. We examine this in the next section.

Table 1: Pressure OLS asymptotic results (using earlier MCMC parameter means as the initial guess).

<table>
<thead>
<tr>
<th>Param.</th>
<th>Estimate</th>
<th>SE</th>
<th>CI95</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E)$</td>
<td>5.0708</td>
<td>1.8364</td>
<td>(1.4404, 8.7011)</td>
</tr>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8412</td>
<td>0.2179</td>
<td>(1.4104, 2.2719)</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8804</td>
<td>1.8394</td>
<td>(-4.5166, 2.7558)</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0510</td>
<td>1.9143</td>
<td>(1.2669, 8.8352)</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7539</td>
<td>0.0062</td>
<td>(-3.7662, -3.7416)</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1109</td>
<td>4.4702</td>
<td>(-9.9477, 7.7259)</td>
</tr>
</tbody>
</table>

Table 2: Pressure bootstrap results (using earlier MCMC parameter means as the initial guess).

<table>
<thead>
<tr>
<th>Param.</th>
<th>Estimate</th>
<th>SE</th>
<th>CI95</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E)$</td>
<td>5.0672</td>
<td>0.0432</td>
<td>(4.9817, 5.1526)</td>
</tr>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8399</td>
<td>0.0250</td>
<td>(1.7904, 1.8893)</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8829</td>
<td>0.0484</td>
<td>(-0.9786, -0.7873)</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0518</td>
<td>0.0346</td>
<td>(4.9834, 5.1201)</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7538</td>
<td>0.0057</td>
<td>(-3.7651, -3.7425)</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1123</td>
<td>0.0889</td>
<td>(-1.2882, -0.9365)</td>
</tr>
</tbody>
</table>

Table 3: Pressure optimization results for the following cases: OLS, bootstrapping mean using OLS value to initiate, and DRAM parameter means; data frequency 1024 Hz. $\text{RSS} = J_{\text{ols}}(q)$

<table>
<thead>
<tr>
<th>Method</th>
<th>OLS</th>
<th>Bootstrap mean</th>
<th>DRAM mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E)$</td>
<td>5.0708</td>
<td>5.0672</td>
<td>5.0164</td>
</tr>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8412</td>
<td>1.8399</td>
<td>1.8327</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8804</td>
<td>-0.8829</td>
<td>-0.9300</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0510</td>
<td>5.0518</td>
<td>5.0696</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7539</td>
<td>-3.7539</td>
<td>-3.7539</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1109</td>
<td>-1.1123</td>
<td>-1.1765</td>
</tr>
<tr>
<td>RSS</td>
<td>2.538e-009</td>
<td>2.569e-009</td>
<td>6.136e-009</td>
</tr>
<tr>
<td>Comp. Time</td>
<td>0.1662 mins</td>
<td>25.99 hrs*</td>
<td>52.82 hrs*</td>
</tr>
</tbody>
</table>

*Note: Computations here are serial. Bootstrapping may be easily parallelized, and parallelized versions DRAM and DREAM exist; see the discussion in Section 2.4.
Figure 2: Comparison of density results between the three methods, for each parameter. The MCMC density is computed from the chain values using KDE; for bootstrapping and asymptotic analysis, the mean and variance for each parameter are used to create a normal probability density function which is plotted.
Figure 3: Pressure parameter chain results using DRAM, with data sampled at 1024 Hz, $N_p = 1$. The corresponding OLS estimate from Table 3 is shown as the dashed line.

Figure 4: Pressure parameter pairwise comparisons using DRAM, with data sampled at 1024 Hz, $N_p = 1$, using every 50th chain value. Noticeable patterns in a sub-graph indicate correlation between parameters (e.g., the middle left comparison between $\gamma_1$ and $E$) whereas roughly random relationships indicate less correlations (e.g., the parameter $A$ with any other parameter).
4 Method comparison when holding $E$ constant

Given the strong correlation between $E$ and the other parameters, particularly $\gamma_1$, we next examine what happens when we hold $E$ constant and estimate the remaining parameters. Thus $E$ will not be an element in the covariance matrix, and the strong correlation between it and other parameters will not affect the asymptotic computation (which we suspect does not handle the nonlinear correlations well). Estimated parameter values are shown in Table 4. Asymptotic results are in Table 5, bootstrap results in Table 6, the MCMC chains in Figure 6 (which are mixing even better than before), and densities (sampling for asymptotic and bootstrapping, posterior for Bayesian) in Figure 5.

We now see much more consistent relationships between the different methods. Asymptotic standard errors, though not smaller than bootstrap standard errors, are nearly identical; this is fine, and serves to indicate that the linearizations made when deriving the asymptotic theory are applicable for the restricted inverse problem. Without the highly nonlinear correlation between $E$ and $\gamma_1$, the sampling distribution of the OLS estimate in the asymptotic and bootstrapping results lies very near the posterior parameter density found by the Bayesian method. There are slight differences in the apparent mean of the distribution for each parameter (see Table 4), but the differences are fairly insignificant. The variances of the sampling distributions and of the posterior density are nearly identical across the methods. Thus, now that the parameters are either uncorrelated or only linearly correlated (see plots of Figure 7), the asymptotic theory provides results that are comparable to the other two methods and which are obtained with significantly less computational effort. However, the need to fix either $E$ or $\gamma_1$ would not have been apparent had we not appealed directly to the physical meanings of these parameters or had we not computed the bootstrap confidence intervals and, particularly, the Bayesian posterior densities which demonstrate the strong nonlinear correlation in these parameters. We will comment further on this point in the conclusion.

Table 4: Pressure optimization results for the following cases (with $E$ fixed): OLS, bootstrapping mean using OLS value to initiate, and DRAM parameter means; data frequency 1024 Hz. RSS=$J_{ols}(q)$.

<table>
<thead>
<tr>
<th>Method</th>
<th>OLS</th>
<th>Bootstrap mean</th>
<th>MCMC mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8406</td>
<td>1.8403</td>
<td>1.8408</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8818</td>
<td>-0.8809</td>
<td>-0.8776</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0510</td>
<td>5.0511</td>
<td>5.0511</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7540</td>
<td>-3.7540</td>
<td>-3.7543</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1117</td>
<td>-1.1097</td>
<td>-1.1081</td>
</tr>
<tr>
<td>RSS</td>
<td>2.538e-009</td>
<td>2.539e-009</td>
<td>2.538e-009</td>
</tr>
<tr>
<td>Comp. Time</td>
<td>0.1691 mins</td>
<td>11.33 hrs*</td>
<td>46.70 hrs*</td>
</tr>
</tbody>
</table>

*Note: Here again the computations are serial.

Table 5: Pressure OLS asymptotic results, with $E$ fixed.

<table>
<thead>
<tr>
<th>Param.</th>
<th>Estimate</th>
<th>SE</th>
<th>CI95</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8406</td>
<td>0.0122</td>
<td>(1.8164, 1.8648)</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8818</td>
<td>0.0311</td>
<td>(-0.9433, -0.8203)</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0510</td>
<td>0.0026</td>
<td>(5.0459, 5.0562)</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7540</td>
<td>0.0057</td>
<td>(-3.7654, -3.7427)</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1117</td>
<td>0.0248</td>
<td>(-1.1606, -1.0627)</td>
</tr>
</tbody>
</table>
Table 6: Pressure bootstrap results, with $E$ fixed.

<table>
<thead>
<tr>
<th>Param.</th>
<th>Estimate</th>
<th>SE</th>
<th>CI95</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_{10}(E_1)$</td>
<td>1.8403</td>
<td>0.0127</td>
<td>(1.8153, 1.8654)</td>
</tr>
<tr>
<td>$\log_{10}(\tau_1)$</td>
<td>-0.8809</td>
<td>0.0310</td>
<td>(-0.9423, -0.8196)</td>
</tr>
<tr>
<td>$\log_{10}(\gamma_1)$</td>
<td>5.0511</td>
<td>0.0026</td>
<td>(5.0459, 5.0562)</td>
</tr>
<tr>
<td>$\log_{10}(-A)$</td>
<td>-3.7540</td>
<td>0.0057</td>
<td>(-3.7652, -3.7427)</td>
</tr>
<tr>
<td>$\log_{10}(-\Upsilon)$</td>
<td>-1.1097</td>
<td>0.0247</td>
<td>(-1.1586, -1.0608)</td>
</tr>
</tbody>
</table>

Figure 5: Comparison of density results between the three methods, for each parameter (with $E$ fixed). The MCMC density is computed from the chain values using KDE; for bootstrapping and asymptotic analysis, the mean and variance for each parameter are used to create a normal probability density function which is plotted.
Figure 6: Pressure parameter (with $E$ fixed) chain results using DRAM, with data sampled at 1024 Hz, $N_p = 1$. The vertical dashed line indicates the point at which we consider the run to be burned in.

Figure 7: Pressure parameter (with $E$ fixed) pairwise comparisons using DRAM, with data sampled at 1024 Hz, $N_p = 1$, using every 50th chain value. Note that most relationships are significantly more random than in the previous pairwise plot results, indicating that these parameters are roughly uncorrelated although some linear correlation may still exist (e.g., $\tau_1$ and $\Upsilon$).
5 Conclusion

We have seen that each of the three inverse uncertainty quantification methods can be effective on a problem. If parameters are highly correlated in a nonlinear way, asymptotic analysis alone may provide at best unclear or at worst wrong results. However, one may not suspect the nonlinear correlations without additional computations like the bootstrap or Bayesian methods. The Bayesian approach naturally searches not only for a global cost minimizer, but more importantly, it provides a joint density for the parameters. We found that leaving \( E \) in the model resulted in strong correlation between parameters, which asymptotic theory was unable to discern, bootstrapping suggested, and Bayesian methods clearly described. If we assumed \( E \) was a fixed value and estimated the remaining five parameters, the nonlinear relationships between parameters were reduced or eliminated. At that point, all three methods gave results that were similar. Recalling that the two frequentist methods describe the sampling distribution while the Bayesian method describes the parameter densities directly, this is a fairly convincing indication that the sampling distribution is an acceptable approximation to the parameter distribution if we fix \( E \), reducing complex correlation between parameters. Thus, if our goal had been to propagate parameter uncertainty through the model, in this case we could use the asymptotic results (which, as has been discussed, is not true unless we show the sampling and parameter densities are approximately the same). However, without resorting to the physical meaning of these parameters, this may not have been found (and was not found in our initial calculations) until we used the more computationally expensive bootstrapping and Bayesian techniques. If one wishes to propagate uncertainty through the model, either the parameter densities from the Bayesian method must be used or one must be convinced that the frequentist sampling distribution is an adequate approximation of the parameter density (which requires the Bayesian results for comparison). If one is just concerned with uncertainty in the parameter estimation procedure, then the frequentist methods may be adequate.

We have used the pressure wave propagation model (1.3) and corresponding inverse problem for the model parameters to illustrate how to provide information about parameter uncertainty in our model as well as demonstrate the pitfalls and/or advantages of each of the three uncertainty methods. While no formal definitive conclusions can be made, we suggest that some serious consideration should be given before choosing any single one of these three methods. Each method has strengths and weaknesses, which must be considered and balanced for a problem at hand.

6 Acknowledgements

This research was supported in part (for HTB) by Grant Number NIAID R01AI071915-10 from the National Institute of Allergy and Infectious Diseases, in part (for HTB and ZRK) by the Air Force Office of Scientific Research under grant number AFOSR FA9550-12-1-0188, and in part (for RCS) by the Air Force Office of Scientific Research under grant number AFOSR FA9550-11-1-0152.

References


