Select Applications of Bayesian Data Analysis and Machine Learning to Flow Problems

Naoki Seryo*, John J. Molina†, Takashi Taniguchi*

*Department of Chemical Engineering, Graduate School of Engineering, Kyoto University – Katsura
Nishikyo-ku, Kyoto 615-8510, Japan
(Received: November 30, 2020)

This review focuses on the use of Bayesian Data Analysis and Machine Learning Techniques to study and analyze flow problems typical to polymer melt systems. We present a brief summary of Bayesian probability theory, and show how it can be used to solve the parameter estimation and model selection problems, for cases when the model(s) are known. For the more complex non-parametric regression problem, in which the functional form of the model is not known, we show how Machine-Learning (through Gaussian Processes) can be used to learn arbitrary functions from data. In particular, we show examples for solving steady-state flow problem as well as learning the constitutive relations of polymer flows with memory.

Keys Words: Bayesian data analysis / Machine-learning / Constitutive equations

1. INTRODUCTION

Experiments and simulations of Soft-Matter systems, such as polymers and colloids, produce a large amount of (noisy) data. This data must be suitably processed and analyzed before we can begin our work of distilling the physical processes responsible for the phenomena under study. Unfortunately, based on our experience, the required data analysis tools and techniques required to carry out this preliminary step are not (usually) covered as part of the standard science curriculum, and barely mentioned in physics textbooks. With the recent explosion in big data and machine learning applications within the physical sciences, this issue requires an urgent re-dress. We believe that Bayesian probability theory provides a principled and robust way of dealing with data, both big and small.1,2) While a full Bayesian solution is not always computationally tractable, it is a good starting point from which to develop approximate solutions. In this work, we demonstrate a few simple examples drawn from standard flow problems, that illustrate this point. We start by considering the simplest data analysis problem, that of parameter estimation or curve-fitting, before moving on to more complex applications, such as solving the Stokes equation and learning the constitutive relation of polymeric flows with memory. For the latter, we use Gaussian Processes (GP), one of the standard Machine Learning techniques.3) While known for over 50 year, it was only recently that efficient methods for exactly computing GP on large datasets have been developed.4) Likewise their application in solving partial differential equations, used to express most fundamental laws of physics, is a recent realization.5,6)
For simplicity, we will drop the qualifier \( X \), using \( X \) to refer to both the random variable and its value; furthermore, we will write the density distribution as \( P(X) \), as it should be clear from the context which random variable we are referring to.

The two basic rules of probability theory are the sum and product rules. The sum rule quantifies the fact that assigning from the context which random variable we are referring to. \( P(X) \) write the density distribution as both the random variable and its value; furthermore, we will have a value between \(-\infty\) and \(\infty\)

\[
\int_{-\infty}^{\infty} P(X|I)dX = 1 \tag{3}
\]

The product rule allows us to express the joint distribution for \( X \) and \( Y \), in terms of the individual (conditional) distributions of either variable, given knowledge of the other

\[
P(X,Y|I) = P(X|Y,I)P(Y|I) \tag{4}
\]

\[
P(Y|X,I)P(X|I) \tag{5}
\]

As a corollary of the sum and product rules, one can easily derive the marginalization rule\(^1,2\)

\[
P(X|I) = \int_{-\infty}^{\infty} P(X,Y|I)dY \tag{6}
\]

as well as Bayes’ theorem

\[
P(X|Y,I) = \frac{P(Y|X,I)P(X|I)}{P(Y|I)} \tag{7}
\]

Marginalization will prove to be incredibly useful in cases where our model contains parameters that are unknown, or uninteresting to us (so-called nuisance parameters), but which are still required to evaluate the probabilities. In such cases, the answer is simply to integrate out these unwanted degrees of freedom. Likewise, Bayes theorem allows us to rewrite our probabilities into forms that are easier to compute. To understand the relevance of Bayes theorem within the context of data analysis, consider that we are given a series of data measurements \( D \), together with a model \( M \), parametrized by \( \Theta \), that aims to explain this data. Performing the following substitution within Eq.(7), \( X \rightarrow \Theta, Y \rightarrow D \), we obtain

\[
P(\Theta | D, I) = \frac{P(D | \Theta, I)P(\Theta | I)}{P(D | I)} \tag{8}
\]

The different terms appearing in this equation have been given standard names within the Bayesian literature. Our belief in the value of the model parameters \( \Theta \), \( P(\Theta | I) = \Pi(\Theta) \), given only the prior information (i.e., before we have seen any of the data), is called the prior. The probability of observing the data we have actually measured, given knowledge of the model parameters, \( P(D | \Theta, I) = L(\Theta) \) is called the likelihood. The normalization constant in the denominator on the right hand side, \( P(D | I) = Z \) is referred to as the evidence. By definition, the evidence is nothing but the probability of observing the data, given the particular model, but irrespective of the model parameters. Finally, the term on the left-hand side, the probability distribution for the model parameters given the measured data, which is usually what we are interested in, is called the posterior. We will usually write Bayes theorem in the following simplified form,

\[
P(\Theta | D, I) \propto \frac{L(\Theta)\Pi(\Theta)}{Z} \tag{9}
\]

\[
\propto L(\Theta)\Pi(\Theta) \tag{10}
\]

where in the last step, we have dropped the normalization constant, as it doesn’t depend on \( \Theta \). Thus, given the specification of the prior, and a probabilistic model to compute the likelihood, Bayes’ theorem is all we need to infer the values of the model parameters \( \Theta \). If we are satisfied with a single point-estimate, we can look for the optimum value of \( \Theta \), as that which maximizes the posterior distribution,

\[
\Theta^* = \arg \max_{\Theta} \ln P(\Theta | D, I) \tag{11}
\]

Since the logarithm is a monotonically increasing function, formally it doesn’t matter whether we maximize the posterior or its logarithm, but numerically it is more convenient to work with the logarithm, particularly when we are dealing with non-normalized distributions. For the full solution we can use
standard Monte-Carlo techniques (Metropolis Monte-Carlo, Hamiltonian Monte-Carlo, etc.) to generate samples that follow this distribution. This solves the parameter estimation problem.

If all we are interested in is the value of the parameters for one given model, then we can safely ignore the evidence integral. However, if we need to solve the model selection problem, and choose the best model among several candidates, then the corresponding evidence integrals become the main quantity of interest. To see how this can be quantified we rewrite the integral as the product of the (optimum) likelihood \( L^* = L(\Theta^*) \) and the Occam factor \( W \), defined as

\[
Z = \int L(\Theta) \Pi(\Theta) d\Theta = L^* W
\]

Since the prior integrates to unity, the ratio \( L / L^* (0 \leq L / L^* \leq 1) \) will pick out the fraction of prior mass within the high-likelihood regions. As stated by Jaynes, this gives “the ratio by which the parameter space is contracted by the information in the data”.

Consider the case where we are comparing two models, \( M_1 \) and \( M_2 \), and wish to decide which model best explains our data. This can be evaluated by the odds ratio \( O_{21} \), defined as

\[
O_{21} = \frac{P(M_2 \mid D, I)}{P(M_1 \mid D, I)} = \frac{P(D \mid M_2, I)}{P(D \mid M_1, I)} \frac{P(M_2)}{P(M_1)}
\]

where in the second step we have used the product rule to rewrite the posterior distribution for model \( M_i \) to be correct, in terms of the likelihoods for the given model, \( P(M_i \mid D, I) = P(D \mid M_i, I) P(M_i \mid I) / P(D \mid I) \). The likelihood \( L^* \) quantifies the goodness of fit, favoring models with higher \( L \), whereas the Occam factor measures the robustness of the model and prevents against over-fitting. Typically, increasing the number of parameters in a model will allow us to provide a better fit, yielding a larger value of \( L^* \), however, this is only part of the equation. We must also consider to what extent the parameter space as a whole is consistent with our data, in order to determine whether or not the price of introducing a more detailed description is worth it. This is what the Occam factor measures. Roughly speaking, models that produce larger error estimates during the parameter estimation will be better, i.e., have a higher Occam factor, as the fraction of the parameter space that explains the data will be larger. In other words, while the likelihood provides a local measure of how well the model can explain the data, the Occam factor provides a global measure. As such, it is much more difficult to compute. In fact, the standard Monte-Carlo methods used to estimate the posterior are useless for evaluating the evidence integrals, and brute-force approaches are only possible for simple models (i.e., those with less than 3-4 parameters). For this, more sophisticated methods, such as the recently developed Nested Sampling are required (see Appendix B).

To close this section, we would like to point out the close analogy that can be made between Bayesian probability theory and statistical mechanics. We consider the case of classical statistical mechanics in the canonical ensemble. The distribution function for a system of \( N \) interacting particles is given by

\[
\rho\left( r^N, p^N \right) = \frac{1}{h^{3N} N!} e^{-\beta H(r^N, p^N)}
\]

with \( Q_N \) the canonical partition function. Comparing with our definition of the posterior, likelihood, prior, and evidence given above, we can map the model parameters \( \Theta \) onto the (generalized) coordinates \( r^N \), with the interaction potential given by the logarithm of the likelihood function, such that the likelihood plays the role of the Boltzmann factor, the prior is a constant (determined by the number of states), and the evidence integral is the partition function (see Table I). To exploit this analogy, we will rewrite the posterior and partition functions in terms of this energy function, as

\[
U(\Theta) = -\ln L(\Theta)
\]

\[
P(\Theta \mid D, I) = \frac{e^{-U(\Theta) \Pi(\Theta)}}{Z}
\]

\[
Z = \int e^{-U(\Theta) \Pi(\Theta)} d\Theta
\]

If the prior is not constant, we can always redefine the energy function \( U_T = U - \ln \Pi \). Then, instead of talking about maximizing the posterior, we can talk about minimizing the energy function. This analogy is at the heart of Hamiltonian Monte-Carlo methods, which assign conjugate momenta to each of the model parameters, and solve Hamilton’s equations in order to evolve the parameters within the energy landscape specified by the likelihood (and prior). This approach is considerably faster than using Metropolis Monte-Carlo, which relies on random-walks to sample the parameter space. If quick (approximate) calculations are required, it is customary to employ the Laplace approximation (see Appendix A), which amounts...
to a harmonic approximation of the energy function around the minimum. This forces the posterior distribution into the form of a multi-variate Gaussian, allowing us to compute the evidence integral analytically. Obviously, these approximations are not valid for multi-modal distributions, or distributions with a strong degree of asymmetry.

2.2 Gaussian Processes

If the functional form of the model(s) is known, we are faced with either the parameter estimation or model selection problems detailed above. However, in many situations we must deal with the more difficult problem of trying to learn a function when all that we have are noisy (possibly incomplete) data measurements, and maybe knowledge of the underlying physical equation. For example, we might be given the value of the fluid velocity at some points in space and time, and told that they follow a particular constitutive equation. Can we infer the velocity at other positions and/or times? As another example, consider that we have a microscopic polymer model, which we can simulate under arbitrary conditions. Can we use the microscopic data generated from this model to infer the constitutive equation for the stress? These are problems which fall under what has been called “non-parametric regression”, even though this is a misnomer, as there are actually (many) parameters involved. Among the methods that have been developed to approach this family of problems, here we will focus exclusively on Gaussian Processes, as they fit nicely within the Bayesian formalism we have adopted.\(^3\)\(^1\)

Formally, a Gaussian Process (GP) is defined as “a collection of random variables, any finite number of which have a joint Gaussian distribution”.\(^3\) In practice, this means that the values of our unknown function \(a_i = f(x_i)\), discretized over any arbitrary selection of points \(\{x_i\}\), are treated as correlated random variables. We posit that the joint distribution for these variables \(a = (a_1, \cdots, a_N)\) has the form of a multi-variate Gaussian distribution, given as

\[
P(a | \mu, K) = \frac{\exp\left[-\frac{1}{2} \delta a^T K^{-1} \delta a\right]}{\sqrt{\det(2\pi K)}}
\]

\[
a \sim \mathcal{N}(\mu, K)
\]

with \(\mu\) and \(K\) the average and covariance matrix, respectively, and \(\delta a_i = a_i - \langle a_i \rangle\) the fluctuations around the mean. By definition, the statistical properties of this ensemble of random variables is completely defined by these first two moments

\[
\langle a_i \rangle = \langle f(x_i) \rangle = \mu(x_i)
\]

\[
\langle \delta a_i \delta a_j \rangle = \langle (f - \mu(x_i))(f - \mu(x_j)) \rangle = k(x_i, x_j) = K_{i,j}
\]

The covariance matrix is specified in terms of a covariance function, which depends only on the input values \(x_i\), not on the function values themselves. This encodes the idea that if two points are “close” together in \(x\), then their function values \(f(x)\) should also be close together. Without loss of generality, and in the absence of any prior information stating otherwise, we can take the average function to be zero \(\mu = 0\), which leaves only the kernel function \(k\) to be specified.

A valid kernel function needs to be symmetric and positive semi-definite. Kernel functions can be added or multiplied with each other to yield other valid kernels.\(^3\) Among the kernels that have been discovered, three of the most popular are the Squared-Exponential, Rational Quadratic and Matern (family), all of which are stationary, and isotropic, as they are functions of the distance between the points, \(k(x_i, x_j) = k(r = |x_i - x_j|)\).

\[
k_{\text{SE}}(r) = \eta^2 \exp\left[-\frac{r^2}{2\ell^2}\right]
\]

\[
k_{\text{RQ}}(r) = \eta^2 \left(1 + \frac{r^2}{2\sigma^2}\right)^{-\alpha}
\]

\[
k_{\text{MT}}(r) = \eta^2 \left(1 + \sqrt{\frac{r^2}{\sigma^2}} + \frac{5r^2}{3\sigma^4}\right) \exp\left[-\frac{\sqrt{5r}}{\sigma}\right]
\]

These kernels produce random functions whose covariance scales with \(\eta\) and which show variations over a length-scale \(\ell\) (the correlation length-scale). These two parameters, \(\eta\) and \(\ell\), are hyper-parameters or nuisance parameters that must also be learned from the data. An example of a non-stationary kernel is the following Neural-Network kernel,

\[
k_{\text{NN}}(x, x') = \frac{2}{\pi} \arcsin\left[\frac{x^T \Sigma x}{\sqrt{(1 + 2\Sigma^{1/2} x)(1 + 2\Sigma^{1/2} x')}}\right]
\]

where \(\Sigma = \text{diag}(\sigma_0^2, \sigma^2)\), with \(\sigma_0\) and \(\sigma\) the hyper-parameters.

To learn functions from data using GP, we start with two functions, \(a\) and \(b\), and assume that their joint distribution has a GP form. This is written as

\[
\begin{bmatrix} a \\ b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}, \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba}^T & K_{bb} \end{bmatrix}\right)
\]
By definition, the marginal distribution of both \( a \) and \( b \), obtained by integrated over the remaining variable is a GP

\[
a \sim \mathcal{N}(\mu_a, K_{aa})
\]

\[
b \sim \mathcal{N}(\mu_b, K_{bb})
\]  

(31)  

(32)

Not only this, but the conditional distribution of \( a \) given knowledge of \( b \) (or vice-versa) is also a GP

\[
a | b \sim \mathcal{N}\left(\mu_a + K_{ab}K_{bb}^{-1}(b - K_{aa}^{-1}K_{bb}^{-1}a)\right)
\]  

(33)

In addition, any linear transformation of GP results in yet another GP

\[
Aa + Bb + c \sim \mathcal{N}(A\mu_a + B\mu_b + c, AA^T + BKbB^T)
\]  

(34)

Crucially, while the random variables \( a \) and \( b \) can refer to different functions, say \( a_i = f(x_i) \) and \( b_j = g(x_j) \), they can just as well refer to the same function. In fact, we can consider the \( b_j \) to be the ensemble of known data, at the so-called “training” points, while the \( a_i \) refer to the unknown “test” points at which we want to predict the values of the function.

As an example, let us consider the case where we perform noisy measurements of a function, \( y = f(x) + \xi \). Assuming a Gaussian measurement error for \( \xi \), our samples will be drawn from \( y \sim \mathcal{N}(f(x), \sigma^2) \), with \( \sigma^2 \) the variance. Let \( x_i \) refer to the training points, \( x_i \) to the test points, and denote by \( X = (x_1, \cdots, x_n) \) and \( X_s = (x_{1s}, \cdots, x_{ms}) \) the corresponding design matrices. Placing a GP distribution on \( f \) will give us a joint GP distribution for \( f \) and \( y \) of the form

\[
\begin{bmatrix}
f \\
y
\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_f \\ \mu_y \end{bmatrix}, \begin{bmatrix} K(X_s, X) & K(X_s, X) \\ K(X_s, X) & K(X, X) + \sigma^2 \end{bmatrix}\right)
\]  

(35)

Notice that the covariance matrix for \( y \) has an additional term on the diagonal which comes from the uncertainty in the measurement process. The prediction of the function values at the unknown test points is given by the conditional distribution \( f_s(y) \), as specified by Eq. (33). Recall however, that the covariance kernel contains hyper-parameters \( \Theta \) that must be learned from the data. For this, we revert back to the parameter estimation problem and compute the posterior distribution

\[ P(\Theta | y, I) \propto P(y | \Theta, f)P(\Theta) \]

(36)

where the likelihood is given by the marginalized GP obtained from Eq.(35), as \( y \sim \mathcal{N}(\mu_f, K(X, X) + \sigma^2) \). We have assumed that \( \sigma \) is constant, equal for all values, but this restriction can be trivially lifted. Furthermore, if \( \sigma \) itself is unknown in can be treated as an additional hyper-parameter, together with the kernel hyper-parameters. A full Bayesian will average the predictions over this posterior distribution, but for simplicity we can also take a point-wise estimate, using the optimum value \( \Theta^* \) that maximizes this posterior. We should also note that it is customary to add a small artificial noise term to the diagonal of the covariance matrix \( K(X, X) \), mimicking an additional source of error, in order to improve the numerical stability in the calculations involving the matrix inverse.

3. BASIC APPLICATIONS

3.1 Parameter Estimation

Let us start by considering the simplest possible application, that of estimating the parameters of a model given some data measurements. For this, let us focus on data generated from one of the popular constitutive relations for the stress of polymeric liquids, typically written in non-dimensionalized differential form as

\[
\sigma_{\alpha\beta} = f(\sigma, \kappa, t)
\]  

(37)

with \( \sigma_{\alpha\beta} \) the time-derivative of the stress tensor and \( \kappa = \nabla \cdot u \) the velocity gradient tensor. Among the variety of models that have been used, the Maxwell model can be considered the canonical toy-model for viscoelastic flows, as it possesses a strict theoretical foundation, corresponding to a system of non-interacting Hookean dumbbells in the thermodynamic limit. The Maxwell model is given by

\[
\frac{d}{dt} \sigma = \kappa \cdot \sigma^\top + \sigma \cdot \kappa^\top + \frac{1}{\text{De}} (\kappa + \kappa^\top) - \frac{1}{\text{De}} \sigma
\]

(38)

where \( \text{De} \) is the Deborah number, defined as the ratio of the microscopic polymer relaxation time-scale \( \tau \) and the macroscopic fluid advection time-scale \( \ell_0 \). This is a one-parameter viscoelastic model which is linear in the stress. All variables have been given in dimensionless form, using as reference units the characteristic (macroscopic) time scale \( \ell_0 \) and length scale \( \ell_0 \) (channel width). More elaborate models have been developed, which introduce additional non-linear terms, allowing them to better describe non-linear rheological behavior. Examples include the two parameter Giesekeus and Larson models, defined respectively as
Fig. 1 (top) Stress trajectories for systems under simple-shear flow, obtained from three constitutive equations, starting from an isotropic zero stress state \( \sigma(t = 0) = 0 \). (bottom) The corresponding constitutive map, showing the relation between the time-derivative of the stress and the stress, for the Giesekus model. A small Gaussian noise has been added to \( \dot{\sigma}_{\alpha\beta} \) to simulate the effect of the measurement error. The black curves show 400 individual trajectories obtained for different parameter values \((D_e, \alpha)\), sampled according to the posterior distribution \( P(\Theta | D, I) \) using Hamiltonian Monte-Carlo simulations.

Let us generate some artificial data from these models to show how the Bayesian parameter estimation works in practice.

Without loss of generality we consider 2D systems, starting from an initial isotropic condition \( \sigma = 0 \), and assume a fixed

\[
\frac{d}{dt} \sigma = \frac{d}{dt} \sigma_{\text{Maxwell}} - \alpha \sigma \cdot \sigma
\]

\[
\frac{d}{dt} \sigma = \frac{d}{dt} \sigma_{\text{Maxwell}} - \frac{\alpha}{3} \left( \kappa + \kappa^T : \sigma \right) (D_e \sigma + I)
\]

(39) (40)
shear rate, such that

$$\kappa = \begin{pmatrix} 0 & \gamma \\ 0 & 0 \end{pmatrix}$$  \hspace{1cm} (41)

The stress trajectories for the three constitutive models (Maxwell, Giesekus, and Larson), together with the corresponding constitutive relations, are given in Fig. 1. Now, consider that we had sampled this data, either from experiments or simulations, and obtained noisy measurements for $\sigma_{\alpha\beta}$ as a function of $\sigma_{\alpha\beta}$ and $\kappa_{\alpha\beta}$. Given prior knowledge of the constitutive relation, which could come from previous experiments establishing the validity of that particular model for the system under consideration, our task is to find the parameter values $\Theta = (D_0, \alpha, \cdots)$ that best explain the data. The answer to this problem is given by Eq. (9), the posterior distribution for $\Theta$, given the data $D = \sigma(\sigma, \kappa)$. This requires that we compute the prior $\Pi(\Theta)$ and the likelihood $L(\Theta)$. Both will vary on a case-by-case basis, and will generally depend on our prior experience and information. Here, let us assume that the priors are independent, such that

$$\Pi(\Theta) = \prod_i \Pi(\Theta_i)$$ \hspace{1cm} (42)

and consider that the prior information specifies that we are working within the low Deborah limit. As such, we place a weakly informative prior on the Deborah number, given by a Half-Normal distribution (standard deviation of 0.5), and a uniform prior on $\alpha$, constrained to lie in the interval $[0, 1/2]$. For the likelihood, we should specify the probabilistic data generation model, including the measurement process. We will assume that $\sigma$ and $\kappa$ can be measured/controlled exactly, but that the measurement of $\sigma$ is subject to Gaussian measurement error. As such, the likelihood of measuring a particular trajectory $\delta(t)$, will be given by

$$P(\delta_{\alpha\beta} | \Theta, I) = \prod_i \mathcal{N}(\delta_{\alpha\beta}^{\text{Model}}(t_i), \epsilon_{\alpha\beta}^2)$$ \hspace{1cm} (43)

which says that the measurements will be normally distributed around the exact value, obtained from the solution to the differential equation specified by the constitutive model. Here, we have assumed independent measurements, with a constant measurement error $\epsilon_{\alpha\beta}$, although we allow this error to vary for the different components of the stress. If the measurement errors were known, the optimal point-wise solution, obtained by maximizing the log-posterior (Eq.(36)), would coincide exactly with the least-squares solution. However, the Bayesian approach provides a more robust framework, allowing us to consider correlated measurements and non-gaussian noise, as well as missing data or the presence of outliers.1,2)

Figure 1 (bottom) shows the artificially generated data from the Giesekus model, using a Gaussian noise term whose amplitude is equal to 20% of the standard deviation in the data. As specified here, we know that we are dealing with noisy data, but we are not given the corresponding error bars. In this case, the $\epsilon_{alpha}$ should be treated as additional hyper-parameters, and integrated over to obtain the marginal distributions for the parameters of interest, $D_0$ and $\alpha$. We use the PyMC312 python package to perform the Hamiltonian Monte-Carlo simulations (with the no-U-turn sampler13), required to draw parameter samples according to this distribution. The results, in the form of a joint probability density map, are given in Fig. 2. Here, we see that the distribution is well approximated by a 2D Gaussian distribution with a non-diagonal covariance matrix (i.e., non-zero correlations between the two parameters). In contrast to the “standard” least-squares solution that minimizes an ad-hoc loss function, this Bayesian solution presents a robust and principled framework under which to perform this analysis. Furthermore, we are not limited to a single point-wise estimate (although we can derive such approximations), but obtain the full probability distribution.
3.2 Model Selection

We now consider the model selection problem using the same example of the previous section. Instead of trying to find the optimal model parameters, we must decide which of the three models (Maxwell, Giesekus, Larson) is more likely, given the measured data for the time derivative of the stress tensor $\dot{\sigma}$, as a function of the stress $\sigma$ and velocity gradient tensor $\kappa$. To obtain the training data, we first generate 10 independent trajectories for each model ($\text{De} = 10^{-2}, \alpha = 0.3$), using a simple shear flow ($\kappa_{xy} = 20$) with initial condition $\sigma = 0$. Second, the average and standard deviation are taken as the measurement data for the evidence calculations. In contrast to the setup of the previous section, we now have knowledge of the parameter $\epsilon$.

We first compute the optimum energy (likelihood) $U^* = 0$ for the Maxwell model all three fits yield the same optimum parameter values. Using only these energy values we would not favor any of the three models. We could, however, favor the Maxwell model on the basis of it having fewer parameter values (Occam’s razor). For the Giesekus and Larson data sets, the Maxwell model produces an incredibly poor fit, as expected, whereas the optimum energy of the Giesekus and Larson models are at least of the same order of magnitude. In summary, we see that for all the data sets considered, the “true” model indeed gives the lowest (highest) energy (log-posterior).

Table II Optimum energy values ($U = -\log L(\theta)$) obtained by fitting the Maxwell, Giesekus, and Larson data sets to each of the three models. For a given dataset, the best fit is given by the model with the smallest (largest) energy (likelihood).

| Model | $U_{\text{Maxwell}}$ | $U_{\text{Giesekus}}$ | $U_{\text{Larson}}$ |
|-------|----------------------|----------------------|----------------------|
| $D_{\text{Maxwell}}$ | $4.05 \times 10^2$ | $4.05 \times 10^2$ | $4.05 \times 10^2$ |
| $D_{\text{Giesekus}}$ | $1.37 \times 10^4$ | $4.82 \times 10^2$ | $7.40 \times 10^2$ |
| $D_{\text{Larson}}$ | $1.19 \times 10^4$ | $6.33 \times 10^2$ | $4.74 \times 10^2$ |

Table III Evidence values obtained using the Maxwell, Giesekus, and Larson models on training data generated from each of the three models. For a given dataset, the most likely model is the one with the largest value of $Z$, hence the smallest value of the Free Energy $F = -\log Z$.

| Model | $F_{\text{Maxwell}}$ | $F_{\text{Giesekus}}$ | $F_{\text{Larson}}$ |
|-------|----------------------|----------------------|----------------------|
| $D_{\text{Maxwell}}$ | $4.14 \times 10^2$ | $4.23 \times 10^2$ | $4.22 \times 10^2$ |
| $D_{\text{Giesekus}}$ | $1.37 \times 10^4$ | $4.96 \times 10^2$ | $7.54 \times 10^2$ |
| $D_{\text{Larson}}$ | $1.19 \times 10^4$ | $6.48 \times 10^2$ | $4.89 \times 10^2$ |

\[ O_{ij,k} = \frac{P(M_i \mid D_k, I)}{P(M_j \mid D_k, I)} = \frac{Z_i(D_k)}{Z_j(D_k)} \frac{L_i^*(D_k)}{L_j^*(D_k)} \quad (44) \]

where $i$ and $j$ refer to the models we are comparing, and $k$ to the model used to generate the training data set ($D_k$). Following Jaynes,\(^1\) we will measure the odds ratio in units of decibels (dB)

\[ e_{ij,k} = 10 \log_{10} (O_{ij,k}) \quad \text{dB} \quad (45) \]

\[ = 10(\log_{10} Z_i(D_k) - \log_{10} Z_j(D_k)) \quad \text{dB} \quad (46) \]

Where every 3 dB corresponds (roughly) to a factor of two (i.e., $P(M_i \mid D, I) \approx 2P(M_j \mid D, I)$), while 10 dB corresponds to a factor of ten (i.e., $P(M_i \mid D, I) \approx 10P(M_j \mid D, I)$). As a rule of thumb,\(^{14}\) we can say that 5 dB provides substantial evidence for model $i$ in favor of model $j$. 10 dB is strong evidence, and anything over 20 would be decisive ($P(M_i \mid D, I) \geq 10^2P(M_j \mid D, I)$). However, we should always keep in mind that we are dealing with probabilities conditioned on our data and prior information. If either of these change, we must update our beliefs accordingly.

We first compute the optimum energy (likelihood) $U^* = -\log L^*$ by fitting the three different data sets to each of the three models. In this case, with a Gaussian measurement error and uniform priors for the model parameters, the energy is proportional to the sum of the squared residuals normalized by the errors, i.e., $U^* \propto \chi^2/2$.
we performed Nested Sampling simulations using the Dynasty python package.\textsuperscript{15} We now observe a clear difference between the suitability of the models to explain the Maxwell data set, even if they all provide exactly the same optimum energies (likelihoods). Here, the increased complexity of the Giesekus and Larson models results in significantly smaller Occam factors without increasing their goodness of fit, thus favoring the Maxwell model. For the Giesekus and Larson data sets the situation is unchanged with respect to the optimum likelihood results (\textit{i.e.}, the Occam factors play only a minor role). In this way, we are able to provide a clear, interpretable, and quantifiable measurement about our relative degrees of belief in any of the models (given the data). The odds ratios we have obtained are summarized in Table IV. In all cases we are able to correctly identify the “true” model. For the Maxwell data, for which it was not possible to select a single best model using only the optimum energy, we see that the odds ratios with respect to the Giesekus and Larson models are $\approx 40$, such that $P(M_{\text{Maxwell}}|D,I) \approx 10^4 P(M_{\text{Giesekus/Larson}}|D,I)$. Thus, given the measured data, we can unequivocally choose the Maxwell model in this case. The odds ratios are even more heavily skewed towards the “true” model when considering the Giesekus or Larson training data sets.

\section{ADVANCED APPLICATIONS}

\subsection{Solving the Stokes Equation}

Here, we will use GP to solve the Stokes equation, taking as training data the values of the velocity at the boundaries, together with the forces (\textit{i.e.}, the right-hand side of the Stokes equations) sampled within the domain. While we will focus on the Stokes equation in 1D, for simplicity, we note that this method can be applied to arbitrary partial differential equations for $\mathbf{u}$, of the form

\begin{equation}
\mathcal{L}\mathbf{u} = f
\end{equation}

where $\mathcal{L}$ is any linear operator, and $f$ is an arbitrary function of space and time but independent of $\mathbf{u}$. For the specific case of the Stokes problem, $\mathcal{L}$ is the diffusion operator, and $f$ is minus the pressure gradient

\begin{align}
\mathcal{L} &\equiv \frac{\partial^2}{\partial x^2} \\
f &\equiv -\Delta P
\end{align}

Now, without loss of generality, we can place a GP prior on the velocity, $\mathbf{u} \sim \mathcal{N}(0,K_{uu})$. Then, thanks to the linearity of $\mathcal{L}$, together with Eq. (47), we have a joint GP prior for $\mathbf{u}$ and $f$

\begin{equation}
\begin{bmatrix} \mathbf{u} \\ f \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mu_u \\ \mu_f \end{bmatrix}, K_{uu} \begin{bmatrix} \cdot & \cdot \\ \cdot & K_{ff} \end{bmatrix} \right)
\end{equation}

Now, we can encode our knowledge of the physics of the problem, \textit{i.e.}, the Stokes equation, into the covariance matrix.\textsuperscript{51} In this way, we will express all quantities solely in terms of $\mu_u$ and $K_{uu}$. Consider first the average of $f$,

\begin{equation}
\mu_f(x) \equiv \langle f(x) \rangle = \langle \mathcal{L}u(x) \rangle = \mathcal{L}\langle u(x) \rangle = \mathcal{L}\mu_u(x)
\end{equation}
which is obtained by applying the differential operator on the average function $\mu_u$. In a similar way, we can rewrite the covariance terms as

$$k_{uf}(x, x') \equiv \langle (u - \mu_u) \cdot (f - \mu_f) \rangle(x, x')$$

$$= \langle (u - \mu_u) \cdot (L u - L \mu_u) \rangle(x, x')$$

$$= \langle (u - \mu_u) \cdot L(u - \mu_u) \rangle(x, x')$$

$$= L_2 \langle (u - \mu_u)(u - \mu_u) \rangle(x, x')$$

$$= L_2 k_{aa}(x, x')$$

where we have adopted a functional notation, using the subindex $i$ on the $L_i$ operator to specify which of the two input variables it is operating on. The fact that we can incorporate knowledge of the underlying physical equation into the covariance kernel presents substantial advantages. In particular, we need not worry about any discretization errors for the differential operators $L$, as we can exactly compute their effect on the kernel functions. In addition, if we use a numerical package with automatic differentiation capabilities (e.g., Python’s JAX\(^{16}\)), we can avoid having to perform these tedious calculations by hand.

Assuming that the boundary values $u_b$ are known, we write them out explicitly, and separate them from the unknown velocity values within the domain, by expressing the joint distribution as

$$\begin{bmatrix} u \\ u_b \\ f \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{ss} & q \\ q^T & Q \end{bmatrix}\right)$$

$$K_{ss} = K(X_s, X_s)$$

$$q = \begin{bmatrix} K(X_s, X_b) \\ L_2 K(X_s, X) \end{bmatrix}$$

$$Q = \begin{bmatrix} K(X_b, X_b) \\ L_2 K(X_b, X) \\ L_1 L_2 K(X, X) \end{bmatrix}$$

with $X = (x_1, \ldots, x_n)$ and $X_b = (x_{1b}, \ldots, x_{nb})$ the design matrices for the training points, and $X_s = (x_{1s}, \ldots, x_{ms})$ that of the test points. This expression has exactly the same form as Eq.\((30)\), with $a \leftarrow u$ and $b \leftarrow \begin{bmatrix} u_b \\ f \end{bmatrix}^T$. Then, the conditional distribution for the velocity, given the boundary values and the values of $f$ at arbitrary locations $X$, is again determined by Eq.\((33)\), which we rewrite as

$$\begin{bmatrix} u \\ u_b \\ f \end{bmatrix} \sim \mathcal{N}\left(q Q^{-1} \begin{bmatrix} u_b \\ f \end{bmatrix}, K_{ss} - q Q^{-1} q^T\right)$$

The kernel hyper-parameters $\Theta^*$ are determined by maximizing the log-posterior, conditioned on the training data set $\{u_b, f\}$, as given in Eq.\((36)\).

For the case of a pressure-driven flow between flat parallel plates at a separation $L = 1$ apart, the velocity at the boundary is $v(0) = v(1) = 0$. The exact solution is given as $v = -\frac{\Delta \phi}{\Delta x} (x - 1)$. Figure 3 shows the predictions obtained from solving Eq.\((62)\); which reduces to computing the average and covariance matrix of this conditional distribution. We only require 4 points to obtain an accurate prediction for the velocity profile, with all kernels that we have considered producing a similar degree of accuracy. To check this accuracy we can look at the prediction uncertainty (given by the covariance matrix), as well as the absolute error with respect to the exact solution. As mentioned above, we have added a small artificial noise term $\epsilon^2$ to the diagonal of the covariance matrix for the training data, in order to improve the numerical stability. In other words, the velocity at the boundary and the forces in the fluid are allowed to deviate from their exact values of 0 and 1, respectively, by amounts of the order of $\epsilon$. As can be seen in Fig. 4, this uncertainty in the training data is propagated through to the test data, and largely determines the scale of the uncertainty in the velocity predictions. A smaller $\epsilon$ is preferred, but it cannot be chosen too small or the calculations will not converge; it should be set so that it is much smaller than the magnitude of the quantities that are being measured.

Fig. 4 (top) Prediction uncertainty, as given by the covariance matrix of the conditional GP, and (bottom) absolute error in the velocity as a function of the position along the channel. Different colors correspond to results for different values of $\epsilon$, the artificial variance. The dashed lines in the top plot indicate the value of $\epsilon$ that was used.
determined by Eq. (33), which we rewrite as covariance terms as dious calculations by hand. e.g. Python’s JAX16)), we can avoid having to perform these tech
effect on the kernel functions. In addition, if we use a numer-
differential operators \( L \), as we can exactly compute their
an unknown
\[
\begin{align*}
\Theta_0 &= \arg \max_\Theta \left( \log [p(f | x, \Theta)p(\Theta)] \right) \\
p(f_* | x_*, x, f, \Theta_0) &
\end{align*}
\]

\[
\begin{align*}
\text{Input} & \quad x = (\kappa, \sigma) \\
\text{Output} & \quad f(x) = \hat{\sigma}(\kappa, \sigma) \\
\text{Model Parameters} & \quad \Theta_0 = \arg \max_\Theta \left( \log [p(f | x, \Theta)p(\Theta)] \right) \\
\text{Posterior Distribution} & \quad p(f_* | x_*, x, f, \Theta_0)
\end{align*}
\]

\[
\begin{align*}
\text{Generate Training Data} \\
\text{Train the Model} \\
\text{Predict New Data}
\end{align*}
\]

Fig. 5 Schematic representation of the learning strategy used to infer the constitutive relations of polymer melt flows. First, the data is generated from microscopic simulations performed at fixed values of the strain-rate \( \kappa \). From this, stress trajectories are used to estimate the time rate of change of the stress, and thus the training data \( (\kappa, \sigma, \hat{\sigma}) \). Second, the model is trained in order to learn the optimal kernel hyper-parameters \( \Theta \), given by the maximum of the log-posterior. Finally, these kernel hyper-parameters are used within the conditional distribution for the stress test points, allowing us to predict the value of the stress for any input point \( (\kappa, \sigma) \). Figure taken from Ref. 23

This can also be seen in Fig. 4, where the absolute error \( |u^{GP} - u^{Exact}| \) is plotted. Here, we can appreciate that the error is negligible \( \leq 10^{-6} \) for \( \epsilon \leq 10^{-3} \).

To summarize, we have seen how a simple application of GP allows us to solve an arbitrary (linear) partial differential equation of the form \( Lu = f \). We can interpret the differential equation as specifying the nature of the correlations between \( u \) and \( f \), and as such, can directly encode this information into the covariance matrix for the joint distribution of \( u \) and \( f \). For the training data we use the boundary values, together with the forcing values \( f \) at arbitrary locations within the domain. The distribution for the test variables, here the unknown velocities inside the domain, conditioned on the training data, is the solution to the problem. This method is an implementation of a numerical Gaussian Process, as pioneered by Raissi and Karniadakis, who showed how to use GP to solve partial differential equations for time-dependent processes, obtain their parametrization, and merge multi-fidelity data in the solution.5,6,17–20)

\[ \frac{d}{dt} \sigma \equiv \sigma(\sigma, \kappa) \sim N(0, K) \] (63)

A schematic representation of the learning strategy we have used is given in Fig. 5. To generate the required training data, we perform several microscopic simulations, at fixed \( \kappa \), corresponding to different simple shear and elongational flow setups. From these simulations, we take the stress trajectories \( \sigma(t) \) and estimate the corresponding time-rate of change of the stress \( \sigma(t) \). In practice, we use a simple finite-difference approximation to compute \( \sigma \). This last step requires special care, as the stress is subject to considerable statistical fluctuations due to the use of a finite number of polymer chains in the simulations. In low dimensions, i.e., simple 1D flow problems, it is possible to perform repeated measurements to obtain reliable measures of \( \sigma \), together with their error bars. However, for most non-trivial problems, which require that we learn a function in a high-dimensional space, this is not suitable. Thus, we introduce this unknown measurement error as an additional hyper-parameter that will also be learned from the data. Once the training data has been generated, the optimal hyper-parameters \( \Theta \) are computed as usual, by maximizing their posterior distribution. Finally, we can write down the conditional distribution for the stress at arbitrary “loca-
tions” in the \((\kappa,\sigma)\) space, \(\tilde{\sigma}, |\kappa, \sigma, \tilde{\sigma}\). Our best estimate for the local constitutive relation is given by the average of this distribution. This function can then be used to update the stress within macroscopic flow simulations, in a way that still satisfies the underlying microscopic dynamics of the particular polymer model used to generate the training data.

The validity of this method and its ability to correctly capture the visco-elastic behavior of polymer melt flows that show memory effects has been presented in Ref. [23]. In that work, we used a system of non-interacting Hookean dumbbells as a microscopic model. This allowed us to perform stringent error checks, as the exact constitutive relation is known to be given by the upper convected Maxwell model. Comparison with the exact solution, as well as full multi-scale simulations, which directly couple the microscopic and macroscopic degrees of freedom, showed excellent agreement for both simple-shear flow and pressure driven flow setups. As further proof of the ability of our learning procedure to capture the memory effects, we have considered the transient flow between flat parallel plates, one of which is oscillating at a fixed angular velocity \(\omega\). The velocity of the bottom wall is \(v(y = 0) = 0\), while that of the top wall is \(v(y = 1) = \cos(\frac{\text{Wo}^2}{\text{Re}})t\), with \(\text{Re}\) the Reynolds number, and \(\text{Wo}\) the Womersley number, which is defined as the ratio of the transient inertial forces and the viscous forces. The same constitutive relation used in Ref. [23] to study steady simple-shear and pressure driven flows is applied. Even though the training data was generated assuming a constant strain-rate, our constitutive relation can correctly capture the memory effects of the deformation.

We performed 1D simulations using our learned constitutive relation, together with full multi-scale simulations (MSS) of Hookean dumbbells. Both simulation methods show a similar degree of accuracy, in excellent agreement with the exact Maxwell constitutive relation. However, we would like to point out that the simulations using the learned constitutive relations (GP-MSS) are roughly two orders of magnitude faster than the standard MSS. We have also compared our results with those obtained from a generalized Newtonian model, which are not able to reproduce these memory effects. Simulation snapshots from systems at \(\text{Wo}^2 = 20\) and \(\text{Wo}^2 = 50\).

De = \(10^{-2}\) and \(\alpha = 0.2\). Figure 7 shows the non-trivial components of the constitutive map, expressing \(\sigma_{\alpha\beta}\) as a function of \(\sigma_{\alpha\beta}\) for various values of \(\kappa\). As in the case of the Maxwell model, we are able to accurately learn the underlying constitutive relation, with a larger training data set resulting in more accurate predictions.

5. CONCLUSION

We have presented a Bayesian solution to several typical problems encountered when studying the flow of complex fluids, with an emphasis on polymer melts. In all cases, the analysis can be thought of as a repeated application of the basic rules of probability theory, the sum and product rules, together with their corollaries, Bayes’ theorem and the marginalization rule. No other operations are permitted if one wishes to maintain a consistent interpretation of the probability distributions. The standard “parameter estimation” problem is then reduced to evaluating the posterior distribution, which can be easily
done by employing the same type of Monte-Carlo methods ubiquitous in Statistical and Computational Physics. If the model is not known, but instead, a collection of candidate models are presented as possible candidates, we are faced with the more demanding "model-selection" problem, requiring us to compute the normalization factor in the posterior distribution of the parameters. This is equivalent to evaluating the partition function in statistical mechanics. Until very recently, there was no general computational method capable of performing such multi-dimensional integrals, but the discovery of Nested Sampling by Skilling,\(^{10}\) has opened up a whole new avenue of research, not only in Bayesian data analysis but in Statistical Physics and Material Science as well.\(^{7,24–29}\) Finally, in cases where no model is available, we must solve the so-called "non-parametric regression" problem, in which we learn functions from data. This is the standard problem in Machine Learning. This is a vast and rapidly expanding field, and we could not possible provide a summary of the relevant methods. Therefore, we have chosen one particular approach, based on Gaussian Processes (GP), which naturally fits into a Bayesian framework. Briefly stated, a GP provides a probability distribution over functions, in the form of a multi-variate Gaussian for the values of the unknown function. This allows us infer the values of any arbitrary function, given some known (training) data, by exploiting the latent correlations within the data itself. We showed how one can formulate and solve the Stokes flow problem as an inference problem using GP. As a more powerful example, we described recent work to learn the constitutive relations of polymer flows with memory using GP. This has allowed us to drastically reduce the computational cost, compared to state-of-the-art multi-scale simulations, while still maintaining a faithful representation of the underlying microscopic dynamics. In the future, we will consider more complex microscopic models, in order to describe the flow of entangled polymer melts relevant to industrial flow problems.

Acknowledgements

NS acknowledges the "J. Soc. Rheol. Jpn. Encouragement Award" from the 15th International Workshop for East Asian Young Rheologists (IWEAYR15), which motivated him to write this article. The authors appreciate the kind encouragement of Prof. Y. Masubuchi in writing this review, and would like to thank Simon Schnyder and Mika Akagi for useful discussions. This work was partially supported by the Japan Society for the promotion of Science (Grants-in-Aid for Scientific Research Kakenhi No. 19H01862 and Wakate B No. 17K17825), as well as the SPIRITS 2020 of Kyoto University. Calculations were performed using the NumPy (v1.19),\(^{30}\) JAX (v0.2.5),\(^{16}\) and PyMC3 (v3.9.3)\(^{12}\) python libraries. All figures were created using the matplotlib\(^{31}\) python library. The python jupyter notebooks used to generate figures 1-4 (parameter estimation, model selection, and flow prediction) are available through a public repository on https://github.com/johnjmolina/IWEAYR-15.

A THE LAPLACE APPROXIMATION(S)

In exploratory situations, where fast calculations are required, and when we wish to avoid performing Monte-Carlo simulations to estimate the full posterior distribution, we can employ the Laplace approximation.\(^{2,11}\) For this, let us recall the definition of the posterior, as proportional to the likelihood and the prior, and express it as proportional to \(h(\theta)\)

\[
P(\theta | D, I) \propto e^{-U(\theta)} \Pi(\theta)
\]

\[
e^{-U_T(\theta)} = h(\theta)
\]

where \(U_T = U - \ln \Pi\) is the “total” energy of the system. Now, we Taylor expand this energy function to second order around the minimum \(\theta^*\)

\[
U_T(\theta) \approx U_T^* + \nabla U_T|_{\theta^*} \cdot (\theta - \theta^*) + \frac{1}{2}(\theta - \theta^*) \cdot \nabla^2 U_T|_{\theta^*} (\theta - \theta^*)
\]
The zeroth order term can be absorbed into the normalization constant for \( h(\Theta) \), and by definition, the first order term, proportional to the gradient of \( U_T \), will vanish, leaving only the second-order term. If we use this approximation for the energy, and place it back into the definition of the posterior, we obtain a Gaussian approximation for \( h \),

\[
\begin{align*}
  h(\Theta) &\approx e^{-U_T \frac{1}{2}(\Theta - \Theta^*)^T \Sigma(\Theta^*)^{-1} (\Theta - \Theta^*)} \\
  &= h^* e^{-\frac{1}{2}(\Theta - \Theta^*)^T \Sigma(\Theta^*)^{-1} (\Theta - \Theta^*)}
\end{align*}
\]

where the average is given by the optimum value \( \Theta^* \), and the covariance matrix is given by (minus) the inverse of the matrix of second derivatives (i.e., the Hessian), evaluated at the optimal value. Since the posterior is normalized to one, we necessarily have

\[
\begin{align*}
P(\Theta | D, I) &= N_\Theta(\Theta^*, \Sigma(\Theta^*)) \\
(\Sigma(\Theta^*))^{-1} &= \nabla^2 U_T |_{\Theta^*} \\
&= (-\nabla \nabla \ln h) |_{\Theta^*}
\end{align*}
\]

Thus, within this Laplace approximation, the posterior distribution for the parameters \( \Theta \) of any given model can be expressed as a multi-variate Gaussian distribution. The evidence integral can then be computed as

\[
Z = \int d\Theta h(\Theta)
\]

\[
= h^* \int d\Theta \exp \left[ -\frac{1}{2} (\Theta - \Theta^*)^T \Sigma(\Theta^*)^{-1} (\Theta - \Theta^*) \right]
\]

\[
= h^* \sqrt{\det (2\pi \Sigma(\Theta^*))}
\]

Furthermore, the marginalized distribution for any of the individual parameters \( \Theta_i \) will be yet another Gaussian distribution

\[
P(\Theta_i | D, I) = N_{\Theta_i}(\Theta_i^*, \Sigma_{ii}(\Theta^*))
\]

Unfortunately, this forced symmetrization of the distributions can be problematic if the underlying posterior distribution does not actually exhibit such symmetry. A more accurate approximation, which does not suffer from this problem, can be obtained by starting from the definition of the marginalized distribution for \( \Theta_i \). From Eq. (6), we see that this is obtained by integrating the joint distribution over all other parameter values,

\[
P(\Theta_i | D, I) = \int d\Theta_1 \cdots d\Theta_{i-1} d\Theta_{i+1} \cdots d\Theta_N \times P(\Theta | D, I) = \int d\Theta_{\neg i} P(\Theta | D, I)
\]

where we have adopted the convention that a sub-index of \( -i \) indicates that term \( i \) is missing. Let us now define auxiliary functions \( h^{(i)}_{\Theta} \), obtained from \( h \) by setting one of its \( N \) parameters, in this case the \( i \)-th one, to have a fixed value of \( \nu \). If \( h \) is a function that accepts \( N \) parameters, then \( h^{(i)}_{\Theta} \) will accept only \( N - 1 \) parameters

\[
h^{(i)}_{\Theta}(\Theta_{-i}) = h(\Theta_1, \ldots, \Theta_{i-1}, \nu, \Theta_{i+1}, \ldots, \Theta_N)
\]

Now, we rewrite the marginalized distribution in terms of these auxiliary functions,

\[
P(\Theta_i | D, I) = \int \frac{d\Theta_{-i} h(\Theta)}{d\Theta h(\Theta)} = \int \frac{d\Theta_{-i} h^{(i)}_{\Theta}}{d\Theta h^{(i)}_{\Theta}}
\]

where the normalization constant must be explicitly added as \( h \) is only proportional to the posterior distribution. One can then proceed to use the Laplace approximation for the distributions appearing in both the numerator and denominators. The integrand in the numerator is expanded around \( \Theta_i^*(\nu) \), the maximum of the posterior with \( \Theta_i \) held fixed. Notice that this optimum value depends on \( \Theta_i = \nu \), resulting in a function of \( \Theta_i \). The denominator is expanded around the unconstrained optimum value \( \Theta^* \), as before. Formally, we have

\[
\begin{align*}
\Theta_i^*(\nu) &= \arg\max \left( \ln h^{(i)}_{\Theta} \right) \\
\Theta^* &= \arg\max \left( \ln h \right)
\end{align*}
\]

This results in the following approximation

\[
P(\Theta_i | D, I) \propto \frac{h(\Theta_i^*(\nu) | \Theta_i) \left[ \frac{1}{2} \det (2\pi \Sigma_{ii}(\Theta_i^*)) \right]^{1/2}}{h(\Theta^* \Sigma(\Theta^*))^{1/2}}
\]

\[
= \frac{e^{-U_T(\Theta_i^*(\nu), \Theta_i)}}{e^{-U_T(\Theta^*, \Theta_i)}} \left[ \frac{1}{2} \det (2\pi \Sigma_{ii}(\Theta_i^*)) \right]^{1/2}
\]

\[
(\Sigma_{ii}(\Theta_i^*))^{-1} = -\nabla \nabla \ln h^{(i)}_{\Theta}
\]

which is no longer symmetric in \( \Theta_i \). This extended Laplace approximation was first discovered by Tierney and Kadane.\(^{11,32}\)
who showed that it is as accurate, and in some cases more accurate, than approximations based on third-order expansions of the energy functions.

B NESTED SAMPLING

Here, we present an outline of the Nested Sampling algorithm developed by Skilling. The basic idea is to rewrite the multi-dimensional evidence integral as a one-dimensional integral of the likelihood over the prior mass, this can be understood as a Lebesgue integration. To start, consider the density of states $g(\lambda)$

$$g(\lambda) = \int d\Theta \delta(L(\Theta) - \lambda) \Pi(\Theta)$$

which gives the probability of observing a state with likelihood $\lambda$, sampled from the prior distribution $\Pi(\Theta)$. From this, we can define the integrated density of states $\chi(\lambda)$, which computes the total amount of (prior) probability mass with likelihood greater than $\lambda$

$$\chi(\lambda) = \int_\lambda^\infty d\lambda' g(\lambda') = \int_{L(\Theta) > \lambda} d\Theta \Pi(\Theta)$$

where by definition $\chi(\lambda = 0) = 1$ and $\chi(\lambda = \infty) = 0$. Furthermore, the density of states and integrated density of states are related by

$$\frac{d\chi}{d\lambda} = -g(\lambda)$$

Assuming that the inverse of $\chi(\lambda)$ exists, $\chi^{-1} = \mathcal{L}$, we have

$$\mathcal{L}(\chi(\lambda)) = \lambda$$

$$\chi(\lambda) = \mathcal{L}^{-1}(\lambda)$$

Note that we use a different symbol for the function $L(\Theta)$, which is a function of the parameters $\Theta$ and returns the corresponding likelihood value $L(\Theta) = \lambda$, and the function $\mathcal{L}(\chi)$, which is a function of the integrated density of states $\chi(\lambda')$, and returns the likelihood contour $\lambda'$ that encloses that amount of prior mass.

With this, we can now rewrite the evidence integral as a one-dimensional integral of $\mathcal{L}(\chi)$ over $\chi$, as

$$Z = \int d\Theta L(\Theta) \Pi(\Theta)$$

$$= \int_0^\infty d\lambda g(\lambda) \lambda$$

$$= \int_0^1 d\chi \mathcal{L}(\chi)$$

In the second step we have used the definition of the density of states, Eq. 88, to express the integral over $\Theta$ as an integral over likelihood values $\lambda$, and in the third step we transform variables from $\lambda$ to $\chi$. We can easily evaluate such an integral by using a Riemann sum, such that

$$Z \approx \sum_k A_k \Delta \chi_k \mathcal{L}(\chi_k) = \sum_k A_k$$

with $A_k$ the area of the $k$-th rectangle in the discretization. Unfortunately, we will usually not know the relationship between the likelihood and the integrated density of states, i.e., $\mathcal{L}$ as a function of $\chi$. However, as recognized by Skilling, we can get away with guessing the values of $\chi$ using a stick-breaking process. At each step of this procedure, we will break off the “stick” (here our $\chi$ interval) at some random locations, with known statistical properties, which will provide the values of the abscissa used in the Riemann integration (i.e., the $\chi_k$).

At the beginning, the length of our stick is unity, with endpoints at $\chi(\lambda = \infty) = 0$ and $\chi(\lambda = 0) = 1$. Distribute $N$ “live” points uniformly and independently in $\chi$. Pick the worst of these points, the one with the largest $\chi$ or smallest likelihood $\lambda$, and break the stick at this location, such that the new interval $[0, \chi^*]$ contains $N - 1$ live points, still distributed uniformly in this truncated domain. The previous worst point $\chi^*$ is added to the list of “dead” points that will be used to approximate the integral. Now, re-introduce an additional live point, such that the uniform distribution over $[0, \chi^*]$ is maintained. This procedure is repeated, at each step truncating at the position of the worst point, which is transferred to the list of dead points, before introducing a new live point within the newly updated constraint. The entropy of the distribution can be used to define a stopping criteria for the number of steps required to obtain a reliable estimate for the evidence.

At each iteration, the statistics of the shrinkage ratio $t = \chi^{\text{worst}} / \chi^*$ are given by

$$P(t) = Nt^{N-1}$$

with an average and variance (for the logarithm) of
we finally obtain

\[ \langle \ln t \rangle = -\frac{1}{N} \]  
\[ \langle (\ln t - \langle \ln t \rangle)^2 \rangle = \frac{1}{N^2} \]  

After \( k \) steps, the list of dead points consist of the worst points obtained at each iteration \((\chi_k, \chi_{k-1}, \ldots, \chi_1, \chi_0 = 1)\), where

\[ \chi_k = \chi_{k-1} \prod_{i=1}^{k} t_i \chi_0 \]  
\[ \log \chi_k = \sum_{i=1}^{k} \log t_i \]  

with \( t_i \) the shrinkage ratio for step \( k \). We can use the average value for \( \log t \) to obtain a simple estimate, or sample over \( t \) to obtain independent realization for \( \chi_k \), resulting in more robust statistics for the evidence integral. Thus, provided we can evaluate \( \mathcal{Z} \), or more precisely \( \Pr(Z | t) \).

The second key point to the algorithm, after the use of the stick-breaking procedure to guess the values of \( \lambda \), is to recognize the fact that we can use \( \Theta \) as a proxy for \( \chi \). To see this, consider sampling from \( \Pi(\Theta) \) within the likelihood constraint \( \lambda^* \), such that the probability distribution for a given parameter configuration \( \Theta \) is

\[ P_\Theta(\Theta | \lambda^*) = \begin{cases} \frac{\Pi(\Theta)}{\lambda(\Theta \lambda^*)} & \text{if } L(\Theta) > \lambda^* \\ 0 & \text{otherwise} \end{cases} \]  

which is suitably normalized, \( \int d\Theta P_\Theta(\Theta | \lambda^*) = 1 \), following Eq.(89). The corresponding probability distribution for the likelihood \( \lambda \), is then given by Eq.(88) as

\[ P_\lambda(\lambda | \lambda^*) = \begin{cases} \frac{g(\lambda)}{\lambda(\lambda^*)} & \text{if } \lambda > \lambda^* \\ 0 & \text{otherwise} \end{cases} \]  

Performing the change of variables from \( \lambda \) to \( \chi \), using Eq.(90), we finally obtain

\[ P_\chi(\chi | \lambda^*) = \begin{cases} \frac{1}{\chi} & \text{if } \chi < \lambda^* \\ 0 & \text{otherwise} \end{cases} \]  

where the factor of \( g(\lambda) \) in the numerator of \( P_\lambda \) is cancelled out by the Jacobian of the transformation. From this, we see that sampling uniformly over \( \chi \), within the constraint \( \chi^* \), is equivalent to sampling \( \Theta \) according to the prior \( \Pi(\Theta) \), within the likelihood constraint \( \lambda^* \). Furthermore, since \( \mathcal{Z} \) is a monotonic function of \( \chi \), points with lower likelihood (or larger energy) necessarily have larger values of \( \chi \). This means that we never need to compute \( \chi \), as the worst points can be selected based on their likelihood values.

In summary, the Nested Sampling procedure works by distributing the live points according to the (constrained) prior \( P_\Theta(\Theta | \lambda^*) \) and ordering them according to their likelihoods \( L(\Theta) \). The worst of these, the one with the smallest likelihood, is added to the list of dead points and used to update the likelihood constraint \( \lambda^* \). Then, a new live point is introduced, again sampled following the constrained prior. We know the likelihood of each dead point but not the corresponding amount of prior mass \( \chi \) that it encloses. For this, we use the statistics of the stick breaking procedure, since it is guaranteed that we are also sampling \( \chi \) uniformly. This entire procedure is predicated on the assumption that we are able to sample according to the constrained prior. One possibility, already suggested by Skilling in his original work, is to make a copy of one of the remaining live points, and run a (Monte-Carlo) simulation to produce an uncorrelated candidate point that will be suitably accepted/rejected depending on its likelihood (energy). Recent works have investigated how to tackle this sampling problem in general settings, particularly when multimodal distributions are involved.\(^{15,33–37}\)

**REFERENCES**

1) Jaynes ET, “Probability Theory: The Logic of Science”, 1st ed. (2003), Cambridge University Press, New York.

2) Sivia DS, Skilling J, “Data Analysis: A Bayesian Tutorial”, 2nd ed. (2006), Oxford University Press, Oxford.

3) Rasmussen CE, Williams CKI, “Gaussian Processes for Machine Learning”, (2005), The MIT Press, Cambridge.

4) Wang KA, Pleiss G, Gardner IR, Tyree S, Weinberger KQ, Wilson AG, “Exact Gaussian Processes on a Million Data Points”, in Wallach H, Larochelle H, Beygelzimer A, D’Alché-Buc F, Fox E, Garnett R, eds., “Advances in Neural Information Processing Systems 32”, 14648–14659 (2019), Curran Associates, Inc.

5) Raissi M, Karniadakis GE, J Comp Phys, 357, 125 (2018).

6) Raissi M, Perdikaris P, Karniadakis GE, SIAM J Sci Comput, 40, A172 (2018).

7) Baldock RJN, “Classical Statistical Mechanics with Nested Sampling”, (2017), Springer International Publishing.

8) von der Linden W, Dose V, von Toussaint U, “Bayesian Probability Theory: Applications in the Physical Sciences”, (2014), Cambridge University Press, Cambridge.

9) Murphy KP, “Machine Learning : A Probabilistic Perspective”, 1st ed. (2012), MIT Press, Cambridge.

10) Skilling J, Bayesian Anal, 1, 833 (2006).
we finally obtain
\[
\Theta(\chi) \text{uniformly sample statistics for the evidence integral. Thus, provided we can}
\]
\[
\Theta \text{ is suitably normalized, }
\]
\[
\lambda \text{ is then given by Eq.}(88) \text{ as }
\]
\[
\log P(\Theta | \chi) \text{ is cancelled out }
\]
\[
\sum_{i=1}^{N} (\lambda^i \chi^i) \text{ resulting in more robust }
\]
\[
\log P(\Theta | \chi) \text{ likelihood constraint }
\]
\[
\lambda \text{ constrained prior. One possibility, already }
\]
\[
\lambda^i \chi^i \text{ according to the prior }
\]
\[
\Pi(\lambda) \text{ as a proxy for }
\]
\[
\chi_1 < \chi_2 \text{ for step } k \text{, is given by Eq.}(102) \text{ as }
\]
\[
\log t \text{ as the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ is selected }
\]
\[
\Theta(\chi) \text{ is the worst of these, the one with the smallest likelihood, }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\chi \text{ is the one with the smallest likelihood, }
\]
\[
\Theta(\chi) \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]
\[
\Theta(\chi) \text{ is the worst points can be selected based }
\]
\[
\Theta(\chi) \text{ necessarily have larger values of }
\]
\[
\chi \text{ within the constraint }
\]