PHY607 Project 3:

Julia Baumgarten, Julia Jones, Lauren Sdun
(Dated: November 21, 2025)

In this project, we apply Bayesian inference and Markov Chain Monte Carlo (MCMC) methods to estimate the temperature and amplitude parameters of a blackbody radiation model. Using experimental spectral intensity data, we construct a two-parameter model based on Planck's law and sample from the corresponding posterior distribution. Two sampling approaches are implemented and compared: a hand-written Metropolis–Hastings algorithm and the affine-invariant ensemble sampler provided by the emcee library. We evaluate convergence using trace plots, posterior histograms, the integrated autocorrelation time, and the Gelman–Rubin statistic. Both samplers recover consistent parameter values, with the ensemble sampler exhibiting faster convergence and more efficient exploration of parameter space. The resulting best-fit model accurately reproduces the observed blackbody spectrum. This project demonstrates how Bayesian methods and MCMC sampling can be used to obtain acceptable parameter estimates and quantify uncertainties for physical models.

I. INTRODUCTION

Many problems in experimental and computational physics require estimating model parameters from noisy or incomplete data. In such situations, classical least–squares methods can fail to capture parameter correlations and are highly sensitive to outliers. For complex model systems, Bayesian inference is widely used as a method of statistical inference. This method applies Bayes' Theorem to return a full posterior probability distribution for the parameters of interest by combining a physical likelihood model with prior information. A common application of Bayesian inference is Markov Chain Monte Carlo (MCMC). MCMC methods are especially powerful in this context, as they allow one to draw samples from complicated probability distributions that cannot be integrated analytically.

In this project, we perform Bayesian analysis and apply Markov Chain Monte Carlo (MCMC) methods to sample from a multidimensional distribution describing the parameters of a two-component blackbody model. The spectral radiance of a blackbody at temperature T is described by Planck's law:

$$I(\lambda; T) = \frac{2hc^2}{\lambda^5}\left[\exp\left(\frac{hc}{\lambda k_B T}\right) - 1\right]^{-1}, \tag{1}$$

which describes the power per area, or intensity, per wavelength λ. Because our recorded intensities are in arbitrary units, an overall amplitude factor is included as an additional free parameter. The goal of the analysis is therefore to infer the posterior distribution of the two-dimensional parameter vector $\theta = (T, A)$, where A denotes the absolute value of intensity amplitude and T the blackbody temperature.

To sample from the posterior distribution $p(\theta \mid \text{data})$, we implemented two complementary MCMC methods. First, we constructed a manual Metropolis-Hastings sampler, which uses a Gaussian distribution. The sampler generated new candidate points from the Gaussian centered at the current point and then accepts or rejects it based on an acceptance probability. Second, we applied the ensemble sampler provided by the `emcee` Python library, which uses multiple walkers to explore parameter space more efficiently. The convergence behavior of these two methods was compared.
In addition to the sampling algorithms themselves, we implemented functions for loading and preprocessing the experimental data, computing the Planck model, evaluating the likelihood and prior, generating posterior samples, and producing summary statistics. Diagnostic tools such as trace plots, posterior histograms, autocorrelation estimates, and the Gelman-Rubin $\hat{R}$ statistic were used to quantitatively evaluate convergence. Visualization methods were developed to compare the experimental data with the maximum posterior model spectrum and to illustrate the resulting parameter constraints.

This project demonstrates how Bayesian inference and MCMC techniques can be used to extract meaningful physical information from experimental measurements. The framework developed here applies broadly across physics, where reliable parameter estimation and uncertainty quantification are essential for interpreting data and validating theoretical models.

II. MCMC METHODS

We compare two Markov Chain Monte Carlo (MCMC) approaches to sample from the posterior distribution $p(\theta \mid \text{data})$ of the blackbody model parameters. The first method is a hand-written Metropolis-Hastings (MH) sampler and the second method uses the ensemble sampler implemented in the `emcee` Python library.

A. Hand-Written Metropolis-Hastings Sampler

The Metropolis-Hastings (MH) method obtains random samples from a probability distribution for cases in which direct sampling is difficult. The generic MH algorith is as follows:

1. For each random variable, the sample value is initialized. This value is sually sampled from the variable's prior distribution.
2. A candidate sample x^{cand} is generated from the proposal distribution
3. The acceptance probability ratio α is computed by the acceptance function.
4. A uniform random number u between 0 and 1 is generated:
   - If $u \leq \alpha$, the candidate is accepted.
   - If $u > \alpha$, the candidate is rejected.

The code for this algorithm is found in the `mcmc_manual.py` module. We ran the chain for $N = 5{,}000$ steps, discarding the first 20% as burn-in. Convergence was assessed using visual diagnostics (plots), the integrated autocorrelation time, and the Gelman-Rubin $\hat{R}$ statistic.

A critical component of the MH algorithm is the choice of proposal width, which strongly affects the efficiency and convergence of the Markov chain. If the proposal steps are too small, the chain explores the posterior distribution slowly and becomes highly correlated, resulting in a large integrated autocorrelation time. Conversely, if the proposal width is too large, proposed steps frequently fall in low-probability regions, leading to very low acceptance rates. For this project, we selected Gaussian proposal widths of $\sigma_T = 50$ K for the temperature and $\sigma_A = 0.1$ for the amplitude. These values were chosen based on two criteria. First, we estimated the natural scale of variation in the posterior by computing a rough temperature estimate from Wien's displacement law:

$$\lambda_{\max} T \simeq 2.9 \times 10^{-3}\ \mathrm{m\,K},$$

which suggested that meaningful changes in the spectral shape occur over temperature variations on the order of tens of kelvin. Second, short exploratory runs were performed to empirically tune the proposal widths toward an acceptance rate in the range 25%–40%, which is commonly recommended for effective mixing in low-dimensional parameter spaces. The final acceptance rate of approximately 35% indicates that the chosen proposal widths allow the sampler to make steps that are large enough to explore the posterior efficiently, yet sufficiently small to maintain a reasonable likelihood of acceptance. These settings yield well-behaved trace plots and autocorrelation times, confirming that the proposal distribution is appropriately scaled for this two-parameter blackbody model.

B. Off-the-shelf emcee library

To compare hand-written sampler, we used the affine-invariant ensemble sampler implemented in the `emcee` library. Unlike single-chain methods, `emcee` evolves an ensemble of "walkers", each performing correlated updates that adapt to the geometry of the posterior.

We initialized 32 walkers in a small Gaussian ball around the maximum likelihood estimate and evolved the ensemble for 5,000 steps. The algorithm internally uses the Goodman-Weare stretch move, which involves simultaneously evolving an ensemble of K walkers where the proposal distribution for one walker is based on the current positions of the $K-1$ walkers in the complementary ensemble. This proposal structure is invariant under affine transformations of parameter space, making it especially effective when the posterior exhibits degeneracies or elongated covariance structures. We discarded the first 20% of samples from each walker as burn-in and flattened the remaining ensemble into a single set of posterior samples. Since the walkers evolve independently, the ensemble sampler naturally provides a built-in measure of convergence via the distribution of acceptance fractions and the consistency of the independent walker trajectories.

III. STRUCTURE AND MODULES

The software developed for this project is organized into several dedicated Python submodules. The complete package is installable via `pip` and exposes a command-line interface for running the full analysis.

A. Data Handling: `data.py`

The `data.py` module provides routines for loading and preprocessing the experimental blackbody radiation data. The primary function, `load_data()`, reads CSV files containing wavelength and intensity measurements (and optionally uncertainties, if provided). Because the experimental dataset does not include measurement uncertainties, we assign each intensity value an assumed relative error of 5%, so that

$$\sigma_i = 0.05\, I_i,$$

where I_i is the measured intensity in bin i. The function returns NumPy arrays containing the wavelength, intensity, and corresponding uncertainties, which are then passed directly to the likelihood and sampling routines. By separating the data from the modeling and inference modules, the software is easily adaptable to alternative datasets or file formats.

B. Physical and Statistical Model: `model.py`

The blackbody model module implements the physical and statistical components required for Bayesian inference of blackbody radiation parameters. It evaluates Planck's law, which provides the theoretical spectral radiance of a blackbody as a function of wavelength and temperature. The function `planck_lambda()` computes this quantity. Building on this physical model, the function `model_intensity()` generates predicted intensity values for a given set of parameters. The model introduces a multiplicative amplitude factor A, allowing the theoretical curve to match the overall scaling of the experimental measurements.

To complete the Bayesian framework, the module defines prior distributions and a likelihood function. The function `log_prior()` encodes simple but physically reasonable constraints on the parameters, restricting the temperature to the range 500 K $< T <$ 10,000 K and requiring the amplitude to be positive. Outside these bounds, the prior probability is zero. The function `log_likelihood()` evaluates the Gaussian likelihood of the data given a parameter set by comparing observed and predicted intensities and weighting the residuals by the measurement uncertainties.

Finally, the function `log_posterior()` combines the prior and likelihood to produce the unnormalized posterior probability, which serves as the target density for MCMC sampling.

C. Hand-Written MCMC Sampler: `mcmc_manual.py`

The `mcmc_manual.py` module contains a complete implementation of a Metropolis-Hastings sampler. The `metropolis_step()` function performs a single Gaussian-proposal update, while `run_mcmc()` manages the construction of the Markov chain, storing parameter states, log-posterior values, and acceptance decisions. The module returns an `MCMCResult` object that has the chain and acceptance rate. Because the sampler is implemented explicitly, it provides full transparency regarding proposal mechanics, burn-in, autocorrelation, and convergence.

D. Ensemble Sampler Interface: `mcmc_library.py`

The `mcmc_library.py` module wraps the affine-invariant ensemble sampler provided by the `emcee` library. The `run_emcee()` function initializes a set of walkers around an initial guess and evolves them. This module returns an `EmceeResult` object containing the walker trajectories, log probabilities, and acceptance fractions.

E. Convergence Diagnostics: `diagnostics.py`

The `diagnostics.py` module provides tools for assessing the convergence and statistical quality of the Markov chains. The module includes an autocorrelation estimator, which is used to compute the integrated autocorrelation time and thereby the effective sample size. It also implements the Gelman-Rubin $\hat{R}$ statistic, which quantifies convergence across multiple independent chains or chain segments. These diagnostics allow us to evaluate mixing, detect nonstationary behavior, and justify the selection of burn-in fractions and chain lengths.

F. Visualization and Output: `plotting.py`

The `plotting.py` module contains all routines for producing visual summaries of the inference results. This includes plotting the raw data with the best-fit model curve, trace plots showing the evolution of each parameter, and posterior histograms. These visualizations are essential both for qualitative convergence assessment and for understanding the final results.

G. Command-Line Interface: `cli.py`

The `cli.py` module defines the command-line interface for the project. The script parses arguments for data file paths, sampler settings, proposal scales, and output directories. It then orchestrates the complete workflow: loading data, running both MCMC samplers, computing diagnostics, generating plots, and printing numerical summaries of the posterior distributions.

IV. RESULTS AND ANALYSIS

To asses the accuracy of our MCMC methods, we test the convergence our MCMC chains by examining the plots of the parameter evolution as well as posterior histograms. We also use the Gelman-Rubin statistic to asses convergence. Then we compare our model to the physical data.

A. Convergence Testing

Figure 1 displays the trace plots for the temperature T and amplitude A after discarding the initial burn-in samples using the manual MCMC method. The initial distribution and the distributions of the subsequent terms of the chain were most likely very different from the target distribution, but then the chain slowly converged to the target distribution, around T ≈ 7200 K and A ≈ 0.62.

The flat, stable appearance of both traces indicates good mixing for this two-parameter problem, and suggests that the retained samples are representative draws from the posterior distribution.

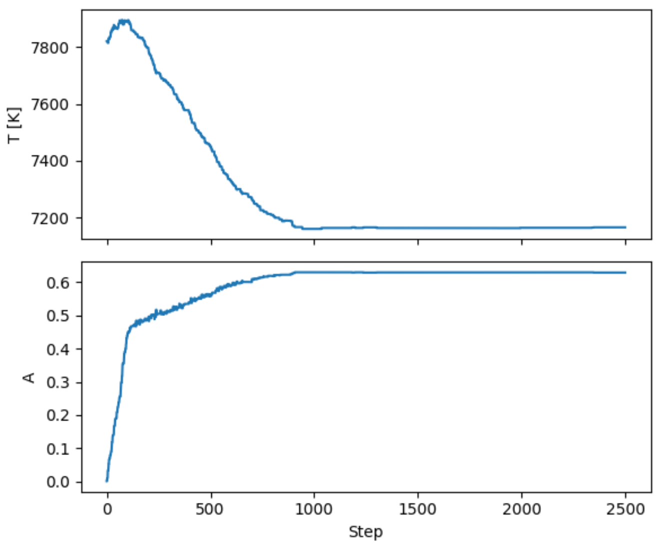

FIG. 1. Trace plots of the hand-written Metropolis-Hastings sampler showing the parameter evolution for the temperature T (top) and amplitude A (bottom) as a function of iteration number. The early portion of each chain exhibits rapid movement as the sampler approaches the high-probability region of parameter space, after which both parameters settle into a stable regime with small fluctuations, indicating that burn-in has completed and the chain has reached its stationary distribution. These traces provide visual evidence of convergence for the manual MCMC method.

We found that the Gelman-Rubin statistic when using the manual MCMC was

$$\hat{R}_T = 1.194, \hat{R}_A = 1.155$$

and when using the emcee method

$$\hat{R}_T = 1.010, \hat{R}_A = 1.007.$$

Both indicate reasonable convergence, however the emcee method appears to be more efficient. This is further demonstrated in Figure 2 and Figure 3. The emcee method produces smooth, Gaussian-like distributions for both parameters, while the manual method yields noisier histograms with significant fluctuations, which indicates a less extensive exploration of the posterior. These results confirm that the emcee method converged more reliably.

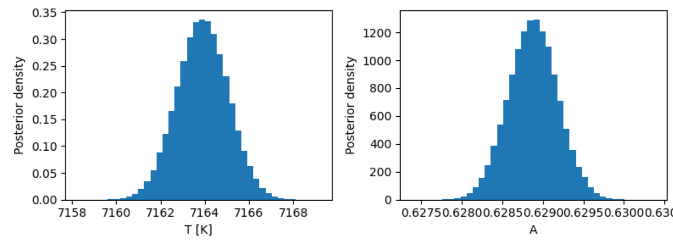

FIG. 2. Posterior distributions for the temperature T (left) and amplitude A (right) obtained using the `emcee` ensemble sampler. Both posteriors exhibit smooth, approximately Gaussian shapes with well-defined peaks, indicating good mixing and convergence of the sampler. The narrow spread in each distribution reflects strong constraints imposed by the blackbody data, and the peak values agree closely with the results from the manual Metropolis-Hastings sampler, demonstrating consistency between the two independent inference methods.

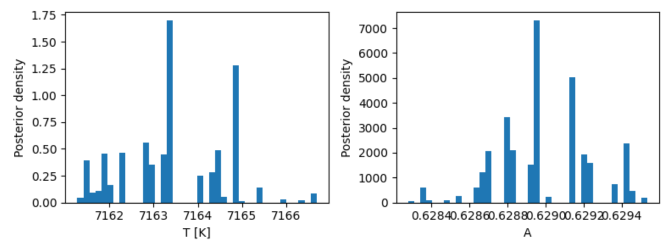

FIG. 3. Posterior distributions for the temperature T (left) and amplitude A (right) obtained from the hand-written Metropolis-Hastings sampler. The posteriors are noticeably noisier and less smooth than those produced by the `emcee` ensemble sampler, reflecting the slower mixing and higher autocorrelation of the manual algorithm. Despite the increased sampling noise, the posterior peaks align with the values recovered by `emcee`, demonstrating that the manual sampler converges to the correct region of parameter space, albeit with reduced efficiency and poorer sampling of the true posterior shape.

B. Autocorrelation Analysis

To further assess the statistical efficiency of each sampler, we computed the integrated autocorrelation time τ_{int} for both parameters. The autocorrelation time provides an estimate of how many steps are required for the chain to produce an effectively independent sample. Smaller values indicate better mixing. For example, if $\tau_{\text{int}} = 50$, 1 independent sample is obtained for every 50 steps. Thus, reducing the number of steps required to obtain a desired number of independent samples makes the simulation more efficient.

For the `emcee` ensemble sampler, we obtained

$$\tau_{\text{int}}(T) = 54.2 \text{ steps}, \qquad \tau_{\text{int}}(A) = 36.0 \text{ steps}.$$

For the manually implemented Metropolis–Hastings sampler, the values were

$$\tau_{\text{int}}(T) = 74.0 \text{ steps}, \qquad \tau_{\text{int}}(A) = 60.9 \text{ steps}.$$

These results indicate that the `emcee` sampler is mixing more efficiently, requiring fewer steps per sample compared to the manual sampler.

C. Validating Results

The two plots shown in Figures 4 show the best-fit blackbody spectra obtained using our two sampling methods: the manually implemented Metropolis–Hastings algorithm and the `emcee` ensemble MCMC sampler. Both fits reproduce the characteristic shape of the blackbody spectrum, capturing the sharp rise toward the peak near $\lambda \sim 0.4\,\mu$m and the exponential decay at longer wavelengths. The inferred peak location and overall spectral curvature are consistent, indicating that both samplers identify similar parameter values for the temperature T and amplitude A.

V. DISCUSSION

In this project, we compared two MCMC methods to estimate the temperature and amplitude parameters of a blackbody radiation model. Both the hand-written Metropolis–Hastings sampler and the affine-invariant ensemble sampler implemented in `emcee` successfully explored the posterior distribution and recovered physically meaningful parameter values. The consistency of the inferred temperatures (around $T \approx 7200$K and $A \approx 0.62$) across both methods demonstrates that the posterior is well constrained by the data and that the likelihood and prior specifications are appropriate. Overall, the `emcee` sampler exhibited higher efficiency in its convergence to the relevant region of the parameter space. This was evident based on the trace plots, the autocorrelation times, and the Gelman-Rubin statistics. Overall, this project illustrates the power and flexibility of Bayesian inference for physical parameter estimation. MCMC methods allow one to examine correlations and rigorously compare models.

[1] M. Taboga, Markov chain monte carlo (mcmc) diagnostics, StatLect.
[2] D. Navarro, The Metropolis-Hastings algorithm.
[3] I. Pasha, Writing a metropolis-hastings mcmc sampler, Online (2020).
[4] I. Yildirim, Bayesian inference: Metropolis-hastings sampling, OnlinePDF (2012).
[5] R. Bakker, Introduction to applied bayesian modeling, Online.
[6] G. Elert, The physics hypertextbook (1998).
[7] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, emcee: The mcmc hammer, PASP 10.1086/670067 (2013).

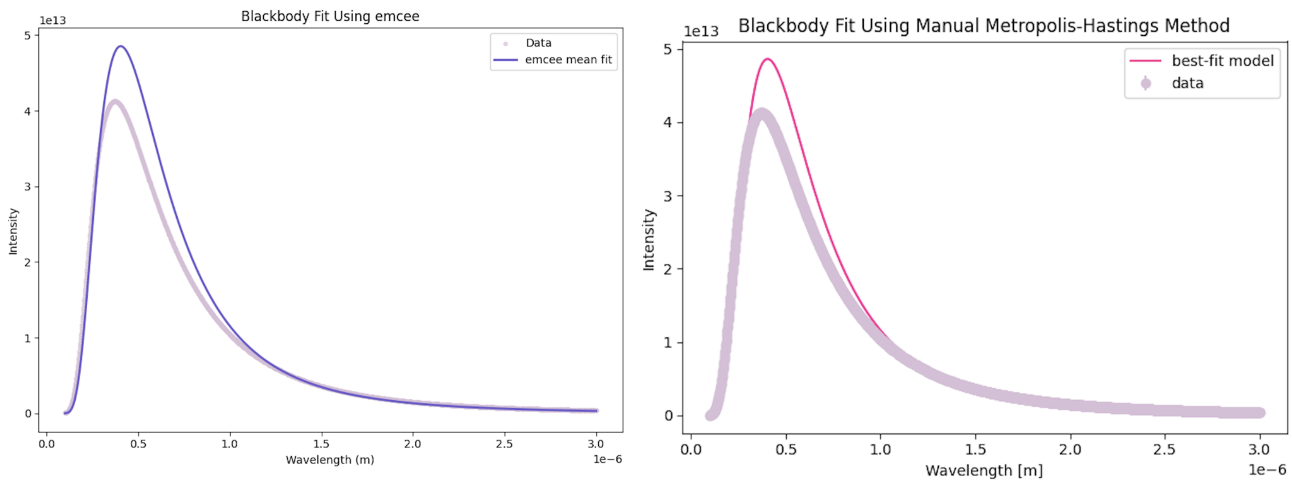

FIG. 4. Comparison of blackbody model fits obtained using the `emcee` ensemble sampler (left) and the hand-written Metropolis-Hastings method (right). In both panels, the measured spectral intensity data are shown as points, while the solid curves represent the mean posterior model predictions. The `emcee` fit exhibits a smoother and more stable reconstruction of the blackbody spectrum, reflecting efficient sampling and low autocorrelation within the ensemble sampler. The manual Metropolis-Hastings fit reproduces the same overall spectral shape and peak location. The close agreement between the two fits validates the correctness of both inference approaches and confirms that the recovered temperature and amplitude parameters are consistent across methods.