Bayesian Inference of the Calendar Ring Hole Count in the Antikythera Mechanism

Advance Statistical Methods for Data Intensive Science

mem97

Lent Term 2025

Contents

Acknowledgment

I would like to acknowledge the work of Thoeni, Andrew, Chris Budiselic, and Andrew in producing the replicated X-ray data of the Antikythera Mechanism in 2019. Their efforts in digitising and segmenting the fractured calendar ring sections have provided an invaluable foundation for this analysis.
The dataset can be found in either my GitHub Repository (see data) or from the original source Harvard Dataverse.

I would like to acknowledge the assistance of ChatGPT-4 and Grammarly in the preparation of this project. The tools played a significant role in:

- Debugging and resolving coding challenges encountered during the development of the Python package.
- Providing insights and suggestions for improving the structure and functionality of the code.
- Generating concise and accurate summaries of complex texts to enhance understanding and clarity.

While ChatGPT-4 contributed significantly to streamlining the development process and improving the quality of outputs, all results were rigorously reviewed, tested, and refined to ensure their accuracy, relevance, and alignment with project objectives.
Word Count: ~ 3200

1 Introduction

In 1901 Captain Dimitrios Kontos and a crew of sponge, divers retrieved numerous large objects from an ancient Roman cargo ship, 45 meters below the sea level, near the island of Antikythera[1] (Greek island located in the Mediterranean Sea). Among the many objects retrieved from the wreckage there was a tool, that is now known as the Antikythera mechanism.

The mechanism was designed by ancient Greek astronomers during the second century BC. This was the first known analog computer in human history. This consists of a calendar ring with holes punched at the extremity of its circumference. Unfortunately, approximately 25% of the ring survived. We used to believe that the full ring contained 365 holes, implying that the mechanism was used as a solar calendar. While a new theory suggests that there were 354 holes overall, i.e. the mechanism was a lunar calendar.

In this project we aim to use an X-ray image of the calendar ring to then infer on the number of holes present in the complete ring, through Bayesian inference[2, 3] with HMC. Bayesian inference is a statistical framework where under the Bayes' theorem one determines a probability of hypothesis, based on prior evidence, this can then be updated when more observations become available. Here, Bayesian inference is used to estimate the number of holes in the Antikythera mechanism's fractured calendar ring, alongside other geometric and alignment parameters. To sample from this high-dimensional posterior distribution, we use Hamiltonian Monte Carlo (HMC), a gradient-based Markov Chain Monte Carlo (MCMC) method. HMC allows for more efficient exploration of the parameter space compared to traditional MCMC techniques, especially when dealing with correlated parameters and complex likelihood surfaces.

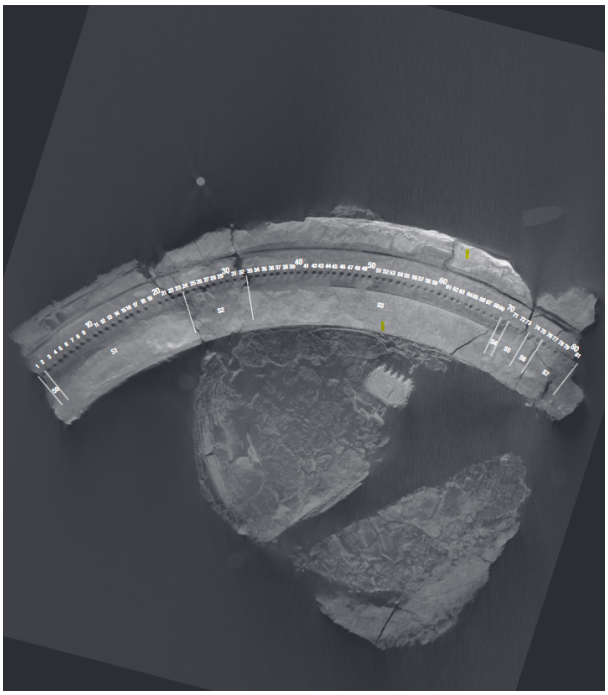

Figure 1: X-ray image, reproduced by Budiselic et al (2020), of the Antikythera Mechanism's ring, showing 81 numbered hole impressions distributed across eight fragmented sections.

2 Data Description and Preprocessing

Before processing the data, we should understand the dataset we are working on and its structure. The dataset is saved as a '.csv' file, to access the data with Python we used Pandas and its module 'pandas.read_csv'.

Once accessed the dataset, we can observe that is divided into 5 columns, of which only 4 are used to fulfil the objectives of this project:

- **Section ID**: Labels each fragments of the remaining ring.
- **Hole**: Label assigned to each hole.

Both Hole and Section ID will be used to identify each hole, we will later see they are labelled as i and j respectively, in the mathematical model.

- **Inter-hole Distance**: Calculated distance between each hole (not used)
- **Mean(X)** & **Mean(Y)**: Many measurements were reported when scanning the arc for the xy-coordinate, Mean(X) and Mean(Y) are the means of all reported measurements (units, mm) (for more information, see ReadMe v2.txt)

Throughout our calculations we excluded two of the Section ID, namely sections 0 and 4, following the footsteps of Woan & Bayley[3] during their research, as these sections contain only one hole each. In Bayesian inference (as we will see later), particularly for a model with many parameters like ring radius, hole count, section offset, etc., any sections with only one hole do not provide enough information to meaningfully estimate parameters such as rotation and translation (this requires at least two points to estimate a direction or offset robustly). Including such sections could add noise or even bias without providing useful constraints.

The following plots (Fig.2 & 3) display location of Mean(X) (x-axis) and Mean(Y) (y-axis). One can observe the misalignment between sections. (For both plots, see code)

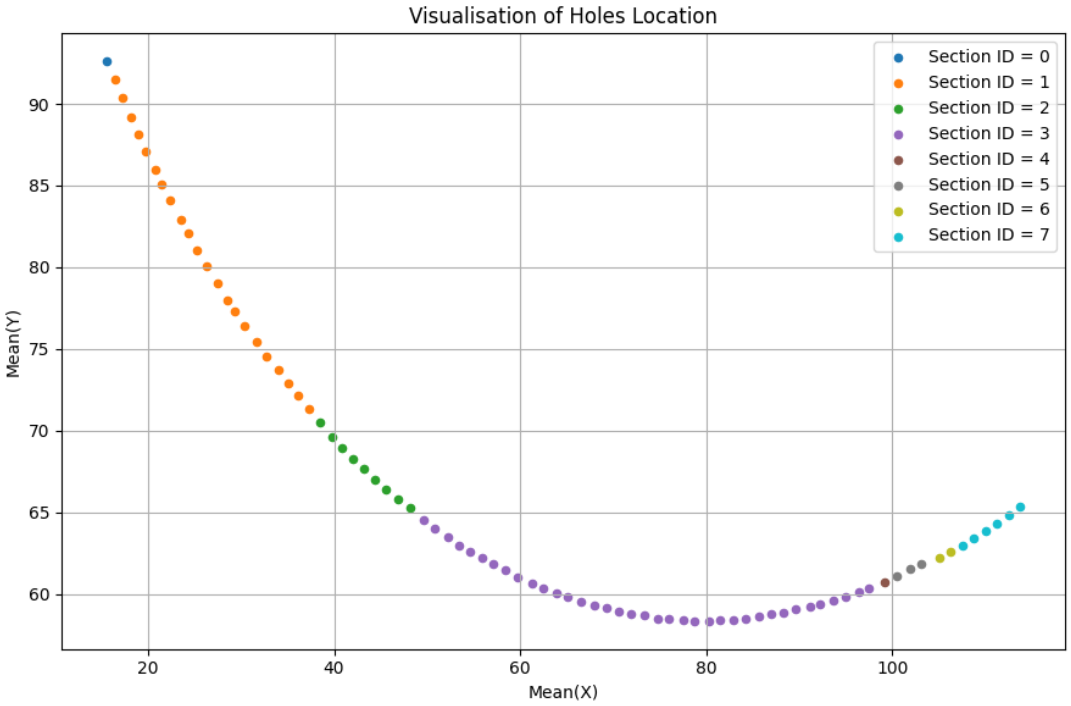

Figure 2: Full dataset plot.

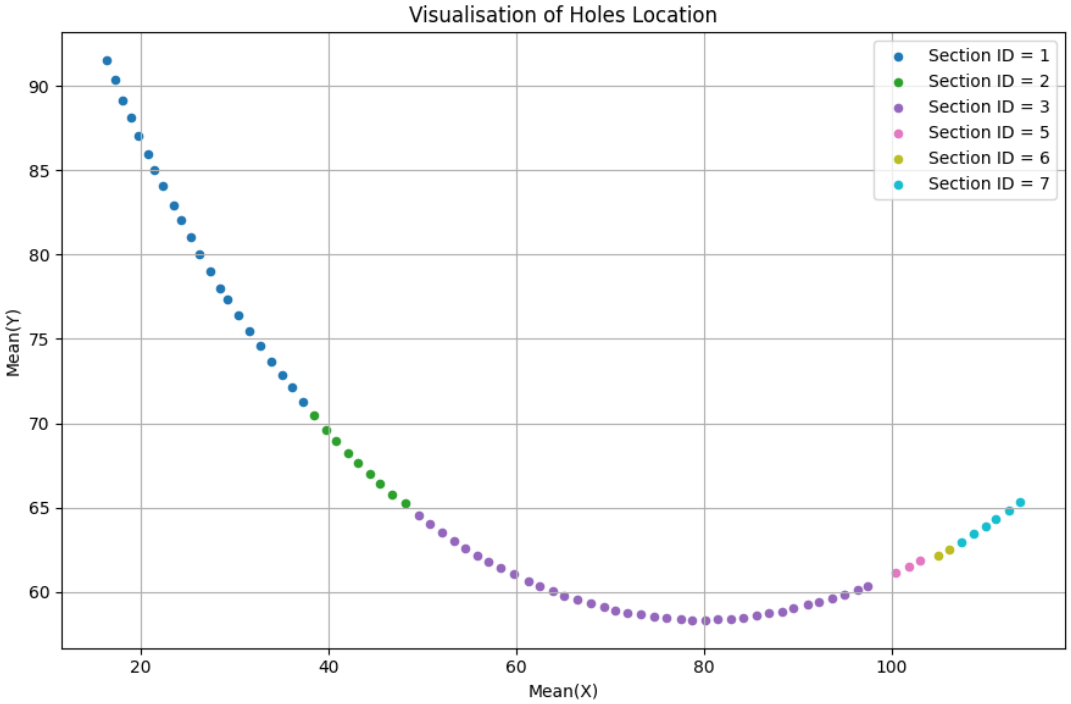

Figure 3: Dataset plot, after removing sections 0 and 4.

3 Modeling Hole Locations

Additionally, to the radial-tangential model presented in An improved calendar ring hole-count for the Antikythera mechanism[3], used for modelling hole location, we are going to implement an isotropic model, to determine which model fits the dataset best.

3.1 Circular Model of Hole Arrangement

The hole positions in the Antikythera Mechanism's are assumed to lie along a perfect circle. Let N[1] be the total number of holes (for simplicity we are going to threat N as a continuous variable, for all models and inference) evenly spread in the full ring of radius r. Thus, the ideal angular position for hole i in section j can be described as:

$$\phi_{ij} = 2\pi \frac{(i-1)}{N} + \alpha_j \tag{1}$$

where α_j (unknown) is the angular position of the first hole in the full circle. These angles are the used to generate the ideal hole coordinate (Cartesian) as:

$$x_{ij} = r\cos(\phi_{ij}) + x_{0j},\ y_{ij} = r\sin(\phi_{ij}) + y_{0j} \tag{2}$$

Adjust for section misalignment using the parameters $(x_{0j}, y_{0j}, \alpha_j)$, where x_{0j}, y_{0j} represent the the unknown location of each section circle-centres. Thus, we need to find parameter estimates $(x_{0j}, y_{0j}, \alpha_j)$ for each section.

3.2 Isotropic Gaussian Error Model

The isotropic model assumes the same uncertainty in all directions. The error between predicted and observed hole locations is modelled using a 2D Gaussian distribution with an isotropic covariance matrix $\Sigma_{\text{iso}} = \sigma^2\mathbb{I}$:

$$\vec{d_i} \sim \mathcal{N}(\vec{m}_i, \Sigma_{\text{iso}}) \tag{3}$$

Where $\vec{d_i} = (x_i, y_i)$ is the observed data, and $\vec{m}_i = (x_{ij}, y_{ij})$ is the predicted data. This simplicity assumes circular uncertainty blobs around predicted points, regardless of the hole's angular orientation.

[1] what we aim to infer

3.3 Radial-Tangential Gaussian Error Model

In contrast, the radial-tangential model accounts for direction specific uncertainty. Measurement errors are decomposed into radial (along the direction of the circle) and tangential (along the circumference) components. We still use the a 2D Gaussian distribution to model the errors, but we define Σ_{rt} as follows:

$$\Sigma_{\text{rt}} = R(\phi_{ij})^T \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_t^2 \end{bmatrix} R(\phi_{ij}) \tag{4}$$

Where (ϕ_{ij}) is the rotation matrix:

$$R(\phi_{ij}) = \begin{bmatrix} \cos\phi_{ij} & \sin\phi_{ij} \\ \sin\phi_{ij} & -\cos\phi_{ij} \end{bmatrix} \tag{5}$$

Therefore, the Gaussian error is distributed as follows:

$$\vec{d_i} \sim \mathcal{N}(\vec{m}_i, \Sigma_{\text{rt}}) \tag{6}$$

Where $\vec{d_i}$ and $\vec{m}_i$ are the same as before.

This allows is to model elongated uncertainty ellipses oriented along the hole direction, which, we can assume without inference, better reflects possible imaging or segmentation bias.

3.4 Coding

The models described in the previous subsection subsections, are coded with NumPy, JAX and NumPyro[2], please see Python files original_model.py and model.py.

- original_model.py: This Python file contains seven different functions, each for specific purposes. They will be used for MLE, Nested Sampling and compute the derivatives log-likelihood function numerically w.r.t. all parameters. I would like to acknowledge that some of the functions in original_model.py took inspiration from jcbayley/Antikythera_holecount.
- model.py: This Python file also contains seven different functions, each designed for each specific purpose. This is where models are coded with JAX and NumPyro. Here we call functions to; perform HMC, generated predicted hole location using parameter estimates, and unpack the log-likelihood functions from models coded with NumPyro.

For more information about the above Python files, the reader is invited to view my GitHub Repository, as all functions in each file are well documented and illustrate their specific purposes.

4 Likelihood Function and Derivatives

In this section we are going to derive the expression for the likelihood that is used for inference in this project. The likelihood functions for both radial-tangential and isotropic model, are both coded in different ways and with different Python packages (NumPy, JAX, NumPyro), in original_model.py and model.py, for MLE and checking code differentiability.

4.1 2D Gaussian Likelihood Derivation

For starter, let us recall the PDF expression for a multivariate Gaussian distribution:

$$p_{G_k}(\vec{x}; \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\det\Sigma}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T \cdot \Sigma^{-1} \cdot (\vec{x}-\vec{\mu})\right) \tag{7}$$

Where k represent the number of dimensions (i.e. $k = 2$ for us), $\vec{\mu}$ is the mean vector and Σ is the covariance matrix.

[2]NumPyro is a programming probabilistic language PPL, that can be used for Bayesian inference in particular to run different types of MCMC including HMC

In Bayesian inference, the likelihood is the probability of the observed data given a specific set of parameters. It quantifies how well the model with those parameters explains the data. Mathematically $Likelihood = P(Data|Parameters)$. This is used to update prior beliefs into the posterior distribution. We will denote the likelihood function as $\mathcal{L}(D|\vec{\theta})$, where $\vec{\theta}$ are the parameters and D is the observed data:

$$\begin{aligned}\mathcal{L}(D|\vec{\theta}) &= \prod_{i=1}^{n} p_{G_2}(\vec{d_i}; \vec{m}_i, \Sigma) \\ &= \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{2}{2}}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\vec{d_i} - \vec{m}_i)^T \cdot \Sigma^{-1} \cdot (\vec{d_i} - \vec{m}_i)\right)\end{aligned}$$

Thus the general likelihood function for both models is:

$$\mathcal{L}(D|\vec{\theta}) = \frac{1}{(2\pi\sqrt{\det \Sigma})^n} \prod_{i=1}^{n} \exp\left(-\frac{1}{2}(\vec{d_i} - \vec{m}_i)^T \cdot \Sigma^{-1} \cdot (\vec{d_i} - \vec{m}_i)\right) \tag{8}$$

From eq.(8) we can derive the log-likelihood function, by taking the logarithm:

$$\log \mathcal{L}(D|\vec{\theta}) = -n \log\left(2\pi\sqrt{\det \Sigma}\right) - \frac{1}{2}\sum_{i=1}^{n}(\vec{d_i} - \vec{m}_i)^T \cdot \Sigma^{-1} \cdot (\vec{d_i} - \vec{m}_i) \tag{9}$$

Once more we recall that $\vec{d_i} = (x_i, y_i)$ represents the observed data, and $\vec{m}_i = (x_{ij}, y_{ij})$ is the estimated hole location.

4.2 Covariance Models

By substituting the Eq.(3, 6), i.e. isotropic and radial-tangential covariances, into Eq.(9) log-likelihood we have:

Isotropic log-likelihood model

Here $\vec{\theta} = N, r, \sigma, x_{0j}, y_{0j}, \alpha_j$, for six sections that correspond to 21 parameters. The log-likelihood expression can be written as:

$$\log \mathcal{L}_{\text{iso}}(D|\vec{\theta}) = -n \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n} ||\vec{d_i} - \vec{m}_i||^2 \tag{10}$$

Radial-Tangential log-likelihood model

Here $\vec{\theta} = N, r, \sigma_r, \sigma_t, x_{0j}, y_{0j}, \alpha_j$, for six sections that corresponds to 22 parameters. The log-likelihood expression can be written as:

$$\log \mathcal{L}_{\text{rt}}(D|\vec{\theta}) = -n \log\left(2\pi\sqrt{\det \Sigma_{\text{rt}}}\right) - \frac{1}{2}\sum_{i=1}^{n}(\vec{d_i} - \vec{m}_i)^T \cdot \Sigma_{\text{rt}}^{-1} \cdot (\vec{d_i} - \vec{m}_i) \tag{11}$$

4.3 Computed Gradient of Coded Log-Likelihoods

In Bayesian inference using HMC (Hamiltonian Monte Carlo), efficient sampling requires computing the gradient of the log-likelihood with respect to parameters in $\vec{\theta}$. This is handled automatically in NumPyro, which uses JAX's automatic differentiation (jax.grad under the hood). To check the differentiability of the NumPyro model, we "extracted" the likelihoods, computed the likelihood given some arbitrary chosen values of the parameters, and compared them to NumPy and JAX coded likelihoods, and these are the results:

Input parameter values:

```
theta_init_rt = jnp.array([80.0, 0.5, 0.5, -2.0, -3.0, -2.0, -1.0, 0.0, -2.0, 86.0, 78.0, 89.0, 90.0,
79.0, 74.0, 140.0, 150.0, 145.0, 148.0, 149.0, 151.0])

theta_init_is = jnp.array([80.0, 0.5, -2.0, -3.0, -2.0, -1.0, 0.0, -2.0, 86.0, 78.0, 89.0, 90.0, 79.0,
74.0, 140.0, 150.0, 145.0, 148.0, 149.0, 151.0])
```

Output:

```
Extracted jax log-likelihood:    -571927.6
Coded jax log-likelihood:  -571927.6
Coded np log-likelihood:  -571927.6
```

Note; for this particular case all input parameters are the same for both radial-tangential and isotropic models (including $\sigma = \sigma_r = \sigma_t = 0.5$, i.e. radial-tangential $\implies$ isotropic), hence the output is the same for both models (as expected).

The next step is to compute the derivatives of all coded log-likelihoods. Both "extracted" (from NumPyro) and JAX coded log-likelihoods can be differentiated with jax.grad(), while the derivatives of the NumPy coded log-likelihoods are computed numerically (using scipy.optimize.approx_fprime). All derivatives were computed with the same input parameters and evaluated at those points.

Derivative output:

```
Max abs difference for RT model (jax-likelihood vs unpacked likelihood gradients): 0.1875
Relative L2 error: 9.809903e-08

Max abs difference for isotropic model (jax-likelihood vs unpacked likelihood gradients): 0.25
Relative L2 error: 1.0427481e-07

Max abs difference for RT model (numerical vs automatic derivative): 326.04687501487206
Relative L2 error: 0.00017058603903772284

Max abs difference for isotropic model (numerical vs automatic derivative): 605.3541670546401
Relative L2 error: 0.00025249275897155015
```

4.3.1 Interpretation

We need to keep in mind that:

- This is a high-dimensional Bayesian model (20+ parameters).
- Comparing gradients from different sources (e.g. JAX autodiff vs. manual vs. numerical)
- The limitations of floating-point precision (typically ∼1e-7 to 1e-16 depending on operations).

JAX (autodiff) vs "Extracted" Likelihood Gradients

These differences are tiny and well within the expected machine precision limits, especially when dealing with 64-bit floats. The relative L2 error confirms that the overall gradient vectors are extremely close. The slightly larger max abs difference (∼0.25) is not alarming in high dimensions, as this could be due to a few outlier entries, possibly sensitive parameters or under/overflow in specific terms.

JAX (autodiff) vs Numerical Finite Difference

There are relatively large absolute differences in some of the components, but the relative L2 errors are still very low, suggesting that most of the gradient vectors are still close. We were not expecting a perfect match, this was aimed to see if we get similar (but very inaccurate) results, to confirm and show that our functions are differentiable with 'jax.grad()'. Furthermore, finite difference gradients are notoriously unstable in high-dimensional problems, small perturbations in a single parameter can easily propagate to large changes in likelihood, especially with noisy or correlated data!

5 Maximum Likelihood Estimation

The Maximum Likelihood Estimator (MLE) is a procedure utilised to estimate model's parameters by maximising the likelihood function, i.e. the probability of observing the given data under the model. The procedure involves:

- Defining a parametric model for the data
- Compute the likelihood (or log-likelihood) as a function of the parameters
- Use an optimisation technique to determine the parameter values that yield the likelihood

To perform the last step we are going use 'iminuit.Minuit'. This optimiser is a minimiser, but we want to maximise our likelihood, hence we need to define a negative log-likelihood (NLL), thus when minimising NLL we are maximising the log-likelihood.

5.1 Results

Table 1: MLE Results Isotropic Model

Parameter	Mean	Hessian Error
N	352.0	± 2.2
r	78.0	± 0.5
σ	0.323	± 0.018
α_1	-2.531	± 0.0004
α_2	-2.5	± 1.0
α_3	-2.5	± 1.0
α_5	-2.5	± 1.0
α_6	-2.6	± 1.0
α_7	-2.6	± 1.0
x_{01}	80.27	± 0.16
x_{02}	79	± 1.0
x_{03}	79	± 1.0
x_{05}	81	± 1.0
x_{06}	81	± 1.0
x_{07}	83	± 1.0
y_{01}	136.3	± 0.5
y_{02}	135	± 1.0
y_{03}	135	± 1.0
y_{05}	136	± 1.0
y_{06}	135	± 1.0
y_{07}	136	± 1.0

Table 2: MLE Results Radial-Tangential Model

Parameter	Mean	Hessian Error
N	353.8	± 0.9
r	78.39	± 0.17
σ_r	0.113	± 0.009
σ_t	0.45	± 0.04
α_1	-2.5283	± 0.0019
α_2	-2.5	± 1.0
α_3	-2.5	± 1.0
α_5	-2.5	± 1.0
α_6	-2.5	± 1.0
α_7	-2.5	± 1.0
x_{01}	80.46	± 0.06
x_{02}	80	± 1.0
x_{03}	80	± 1.0
x_{05}	80	± 1.0
x_{06}	80	± 1.0
x_{07}	80	± 1.0
y_{01}	136.71	± 0.17
y_{02}	135	± 1.0
y_{03}	135	± 1.0
y_{05}	135	± 1.0
y_{06}	135	± 1.0
y_{07}	135	± 1.0

With the above parameter estimates, one can use the designed function `hole_prediction`, which is essentially Eq.(1) and Eq.(2) coded up, (see `model.py`). Then input the predicted hole locations into the function `plotting_comparison` (this function outputs a plot comparison between predicted and observed hole location), see `plotting.py`.

5.2 Interpretation

It appears, from the plots in Figures 4 and 5, that the MLE did not predict hole locations that well, even without checking errors for each of the parameter estimates. When the number of parameters in a model is large relative to the number of data-points (just like in our case, i.e. 21/22 parameters vs 79 data-points), MLE becomes problematic for several key reasons.

5.2.1 Overfitting

MLE tries to find parameters that make the observed data most likely. With many parameters, the model can over-fit, i.r. bending itself too closely to the data, including its noise or measurement error, rather than capturing the underlying structure. This leads to poor generalisation and results, as we see

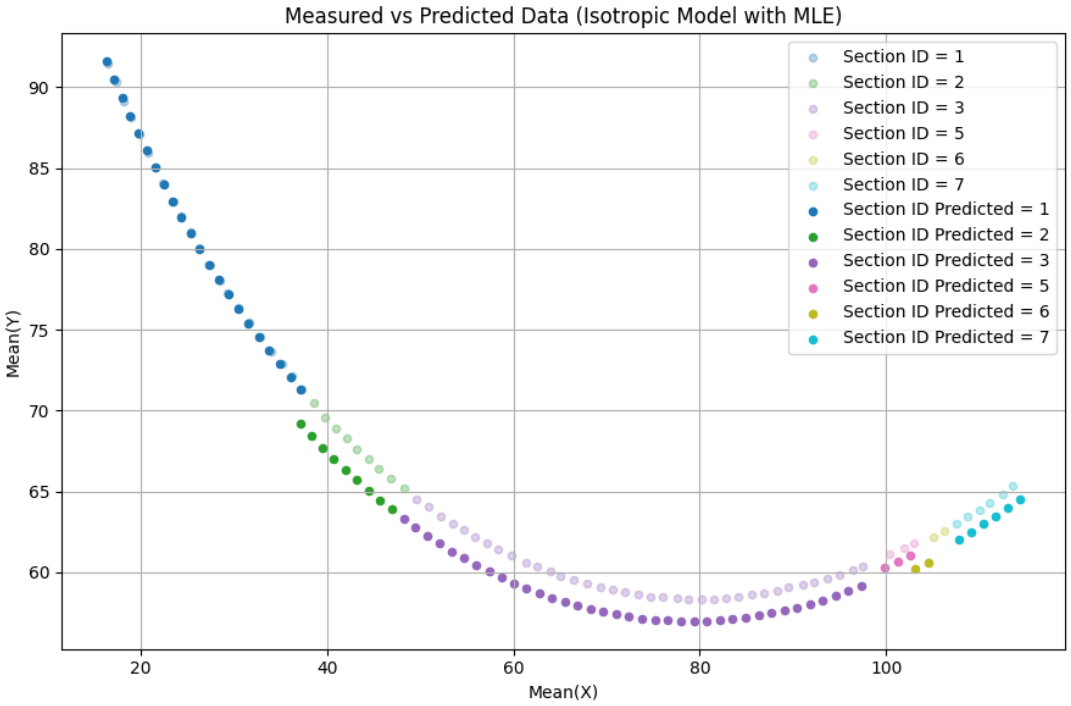

Figure 4: MLE computed with the isotropic log-likelihood model.

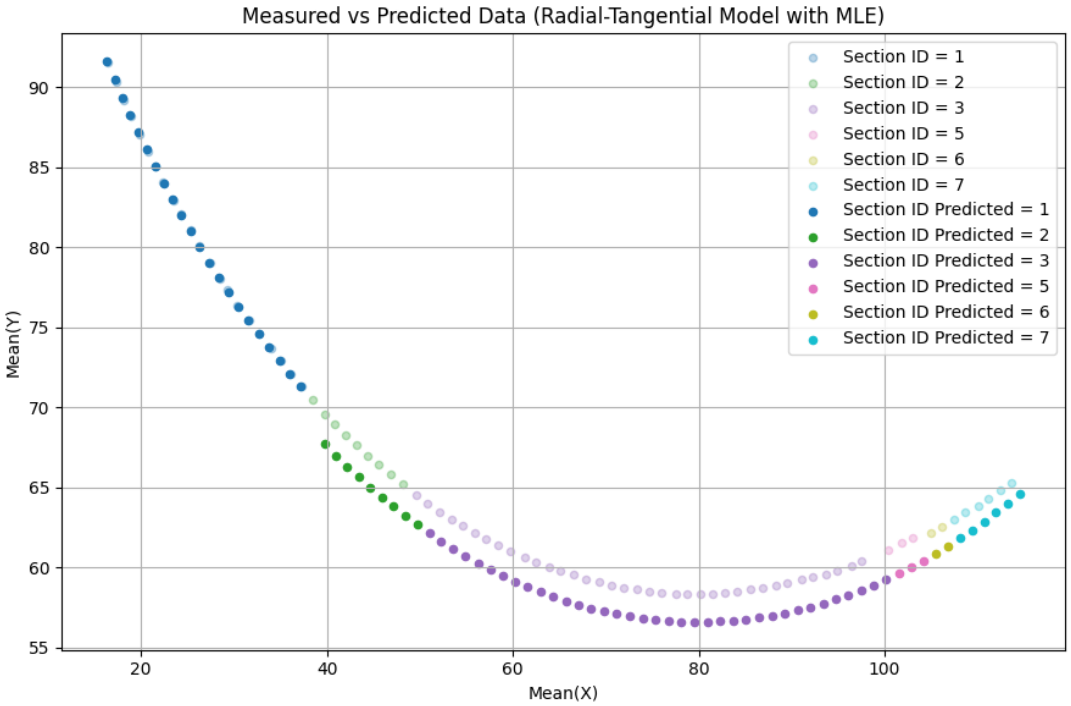

Figure 5: MLE computed with the radial-tangential log-likelihood model.

from the plots and parameter estimates, in particular N (number of holes in the full circle) estimate is ~ 352 and ~ 353 for each models, while we expect $N \sim 354$.

5.2.2 Identifiability and Degeneracy

In high-dimensional spaces with limited data, multiple parameter combinations may explain the data equally well. This means the likelihood surface can have flat regions or multiple maxima, making it hard (or even impossible) to identify a single "best" set of parameters reliably. Perhaps if you have a close look to the `Minuit` codes for both models, you will see that the initialisation of the parameters is already very close to the global minimum (i.e. global maximum for log-likelihood). And I also had to set constraints to few of the parameters, otherwise `Minuit` could potentially unwanted minimums.

5.2.3 No Prior Knowledge

MLE relies solely on the data and ignores any prior knowledge of constraints (hence as already mentioned, I set some constraints with `Minuit`). In underdetermined problems (just like in our case), this can lead to implausible or extreme parameters values simply because they happen to fit well (here is were I invite the reader to remove some of the constraints and run '`Minuit` again, what do you notice? DO you still get "okay" estimates?).

5.2.4 Hessian Error

The small Hessian errors in both models indicate strong parameter identifiability and high confidence in the estimates. The radial-tangential model shows particularly tight uncertainties in key parameters like σ_r (± 0.009) and x_{01} (± 0.06), reflecting its more nuanced treatment of anisotropic error. In contrast, the isotropic model's higher error in σ (± 0.018) suggests reduced sensitivity to directional error, confirming the radial-tangential model captures hole position variability more precisely.

6 Bayesian Inference and Posterior Sampling

Our objective with HMC, is to sample the posterior distribution from the observed data using isotropic and radial-tangential models. A posterior distribution is defined as follows:

$$P(\vec{\theta}|D) = \frac{\mathcal{L}(D|\vec{\theta})\pi(\vec{\theta})}{Z} \tag{12}$$

Where, $P(\vec{\theta}|D)$ is the posterior, $\mathcal{L}(D|\vec{\theta})$ is the likelihood, $\pi(\vec{\theta})$ is the prior and Z is the evidence.

For both models, we chose a uniform prior distribution for all parameters except σ, σ_r and σ_t, which prior are log-uniform distributions. As mentioned above, here N is assumed to be continuous.

6.1 HMC Algorithm

HMC introduces a fictitious physical system:

- Consider parameters in the model as positions in space ($\vec{x}$),
- Introduce momentum variables ($\vec{\rho}$) for each posterior.
- Define a Hamiltonian function that combines:
  - Potential energy: $E(\vec{x}) = -\log P(\vec{x})$
  - Kinetic energy: $K(\vec{\rho}) = \frac{1}{2}\vec{\rho}^T M^{-1}\vec{\rho}$
  - The total energy (Hamiltonian): $H(\vec{x}, \vec{\rho}) = E(\vec{x}) + K(\vec{\rho})$

Sampling steps:

1. Initialise position $\vec{x}_0$
2. Sample momentum $\vec{\rho}\, N(0, M)$ from a multivariate Gaussian.

3. Simulate dynamics: Use leapfrog integration to evolve $(\vec{x}, \vec{\rho})$ over time using the gradient of the potential energy E.

4. Accept/Reject: Use the Metropolis criterion with energy difference to accept or reject the proposal.

This procedure allows the sampler to explore the posterior more efficiently, especially in regions of high curvature.

6.2 Performing HMC with NumPyro

To perform HMC, we prepared two different models (RT and Isotropic), coded up in `model.py`, these models are `rad_tang_model` and `isotropic_model`. Those are constructed to be compatible with `numpyro` and `jax`. `NUTS` and `MCMC` are the two modules chosen from `numpyro` used to run the HMC.

6.2.1 What is `NUTS`?

`NUTS`‘ = No-U-Turn Sampler, this is a smart, automatic version of HMC. HCM is great for sampling from complicated posteriors, but one should specify how far to go on each simulated trajectory (i.e. number of leapfrog steps). This could lead either to going too short (the sampling algorithm does not explore much) or too far (i.e. wasting of time or looping back on itself). Here is where `NUTS` comes handy. `NUTS` avoids this by stopping automatically when the procedure starts to "turn back" (thus the name No-U-Turn). It builds a tree of proposals, doubling its depth each time, until it detects a U-Turn in the trajectory. Then it samples from the points it has seen.

See documentation on Bayesian Regression Using NumPyro

6.2.2 Sampling Results

For both models, the HMC 20000 number of samples, and 1000 warm-ups. Note, when running the codes, expect radial-tangential model to be significantly slower, this is because of its complexity compared to the isotropic model.

Table 3: HMC Summary Result on Isotropic Model

Parameter	Mean	Std	Median	5.0%	95.0%	n_eff	r_hat
N	355.59	4.37	355.62	348.37	362.66	3755.57	1.00
$\alpha_{\text{pred}}[0]$	-2.54	0.00	-2.54	-2.55	-2.54	6029.49	1.00
$\alpha_{\text{pred}}[1]$	-2.54	0.01	-2.54	-2.56	-2.52	7836.81	1.00
$\alpha_{\text{pred}}[2]$	-2.54	0.01	-2.54	-2.56	-2.52	3820.21	1.00
$\alpha_{\text{pred}}[3]$	-2.56	0.05	-2.56	-2.64	-2.48	9573.40	1.00
$\alpha_{\text{pred}}[4]$	-2.54	0.07	-2.54	-2.65	-2.43	7240.51	1.00
$\alpha_{\text{pred}}[5]$	-2.58	0.02	-2.58	-2.62	-2.54	5724.02	1.00
r	77.42	0.94	77.42	75.94	79.02	3754.07	1.00
σ	0.09	0.01	0.09	0.09	0.10	16707.40	1.00
$x_{\text{centre}}[0]$	79.74	0.67	79.74	78.64	80.83	3910.44	1.00
$x_{\text{centre}}[1]$	79.93	0.76	79.93	78.71	81.21	7650.07	1.00
$x_{\text{centre}}[2]$	79.87	0.12	79.87	79.69	80.07	6213.00	1.00
$x_{\text{centre}}[3]$	81.41	3.52	81.47	75.76	87.36	10697.59	1.00
$x_{\text{centre}}[4]$	80.62	5.08	80.78	72.80	89.20	7293.40	1.00
$x_{\text{centre}}[5]$	83.18	1.26	83.17	81.12	85.25	12145.36	1.00
$y_{\text{centre}}[0]$	136.11	0.69	136.12	134.97	137.24	3832.79	1.00
$y_{\text{centre}}[1]$	135.79	0.89	135.79	134.31	137.20	4121.65	1.00
$y_{\text{centre}}[2]$	135.78	0.96	135.79	134.28	137.40	3756.89	1.00
$y_{\text{centre}}[3]$	136.09	1.34	136.14	133.94	138.32	5480.47	1.00
$y_{\text{centre}}[4]$	135.43	2.03	135.63	132.08	138.64	5462.16	1.00
$y_{\text{centre}}[5]$	136.48	1.00	136.48	134.89	138.17	4685.32	1.00

Table 4: HMC Summary Result on Radial-Tangential Model

Parameter	Mean	Std	Median	5.0%	95.0%	n_eff	r_hat
N	355.28	1.38	355.29	352.97	357.50	4059.85	1.00
$\alpha_{\text{pred}}[0]$	-2.54	0.00	-2.54	-2.55	-2.54	6571.33	1.00
$\alpha_{\text{pred}}[1]$	-2.54	0.00	-2.54	-2.55	-2.54	8244.07	1.00
$\alpha_{\text{pred}}[2]$	-2.54	0.00	-2.54	-2.55	-2.53	4128.65	1.00
$\alpha_{\text{pred}}[3]$	-2.56	0.02	-2.56	-2.59	-2.53	11923.79	1.00
$\alpha_{\text{pred}}[4]$	-2.55	0.03	-2.55	-2.61	-2.50	13374.91	1.00
$\alpha_{\text{pred}}[5]$	-2.58	0.01	-2.58	-2.59	-2.57	6181.88	1.00
r	77.35	0.28	77.35	76.88	77.80	3879.84	1.00
σ_r	0.03	0.00	0.03	0.02	0.03	16159.54	1.00
σ_t	0.13	0.01	0.13	0.11	0.15	19536.17	1.00
$x_{\text{centre}}[0]$	79.70	0.20	79.70	79.37	80.03	4003.23	1.00
$x_{\text{centre}}[1]$	79.91	0.23	79.91	79.52	80.28	9329.83	1.00
$x_{\text{centre}}[2]$	79.86	0.03	79.86	79.80	79.92	6010.10	1.00
$x_{\text{centre}}[3]$	81.44	1.11	81.44	79.60	83.22	14013.08	1.00
$x_{\text{centre}}[4]$	81.56	2.43	81.54	77.54	85.57	13873.82	1.00
$x_{\text{centre}}[5]$	83.22	0.39	83.22	82.57	83.84	13735.37	1.00
$y_{\text{centre}}[0]$	136.04	0.21	136.04	135.70	136.38	3972.94	1.00
$y_{\text{centre}}[1]$	135.72	0.27	135.72	135.28	136.16	4193.92	1.00
$y_{\text{centre}}[2]$	135.72	0.29	135.72	135.24	136.18	3884.22	1.00
$y_{\text{centre}}[3]$	136.10	0.41	136.11	135.45	136.78	6460.71	1.00
$y_{\text{centre}}[4]$	135.82	0.84	135.85	134.40	137.16	11390.88	1.00
$y_{\text{centre}}[5]$	136.43	0.30	136.43	135.93	136.91	4839.08	1.00

6.2.3 Error Interpretation

From the standard deviations in Tables 3 and 4, we can infer that the **Radial-Tangential model generally yields lower uncertainty** for most parameters, particularly the global ones such as N, r, and σ. This indicates that incorporating direction-specific error components (radial and tangential) results in a more confident and tighter fit to the data compared to the simpler isotropic model, which assumes uniform uncertainty in all directions.

6.2.4 Plots

Using the parameter estimates from the HMC summaries, we can once again plot predicted vs observed hole displacement, following the same procedure as for MLE. In this section we also present corner plots of predicted global parameter distribution for both isotropic and radial-tangential models. Both plots display the uncertainty for each parameter prediction, as well as the confidence interval with quantiles $(0.15, 0.5, 0.85)$.

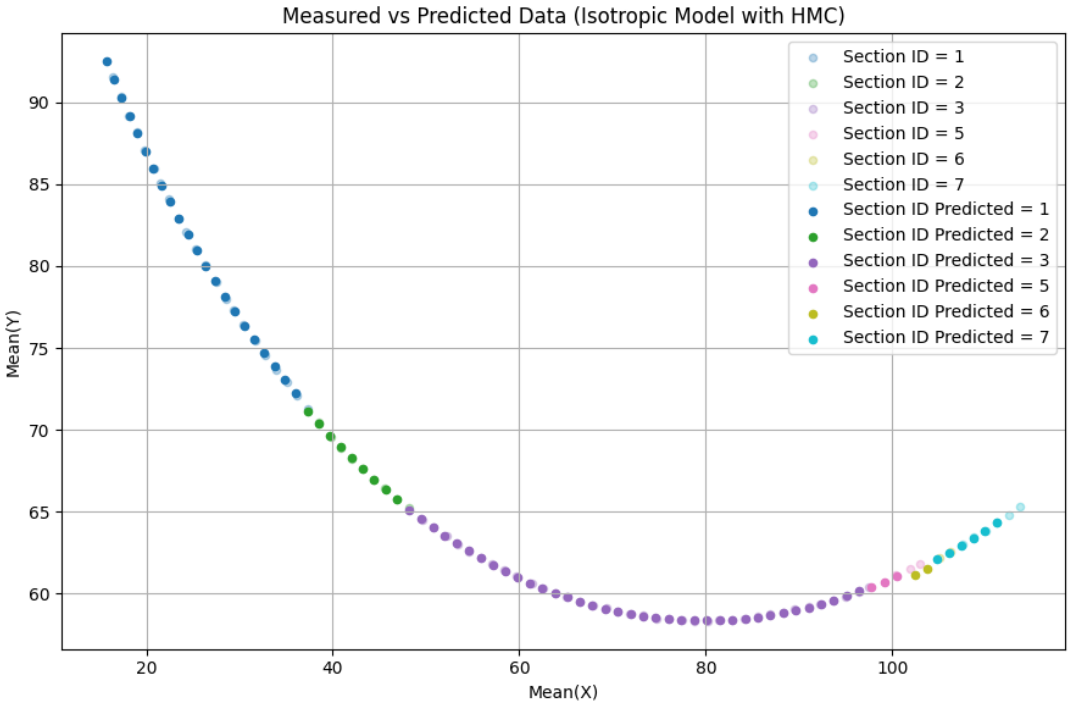

Figure 6: Measured vs predicted hole positions using the Isotropic model. Predictions are accurate, though slightly less aligned compared to the radial-tangential model.

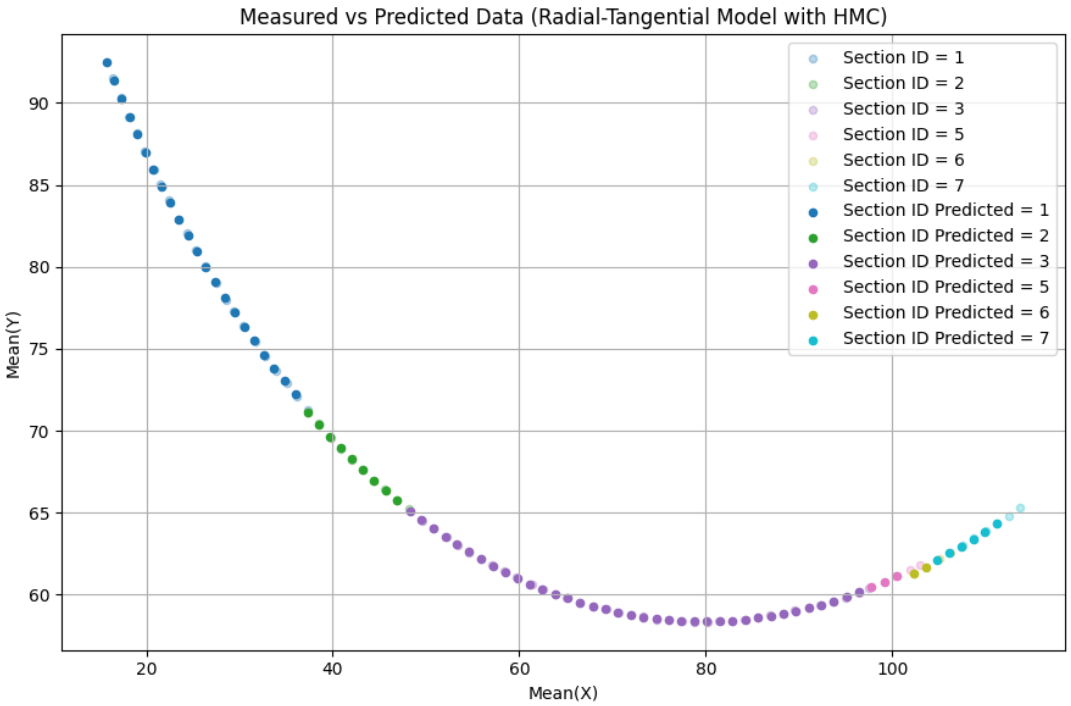

Figure 7: Measured vs predicted hole positions using the Radial-Tangential model. Predictions closely align with observations across all fractured ring sections.

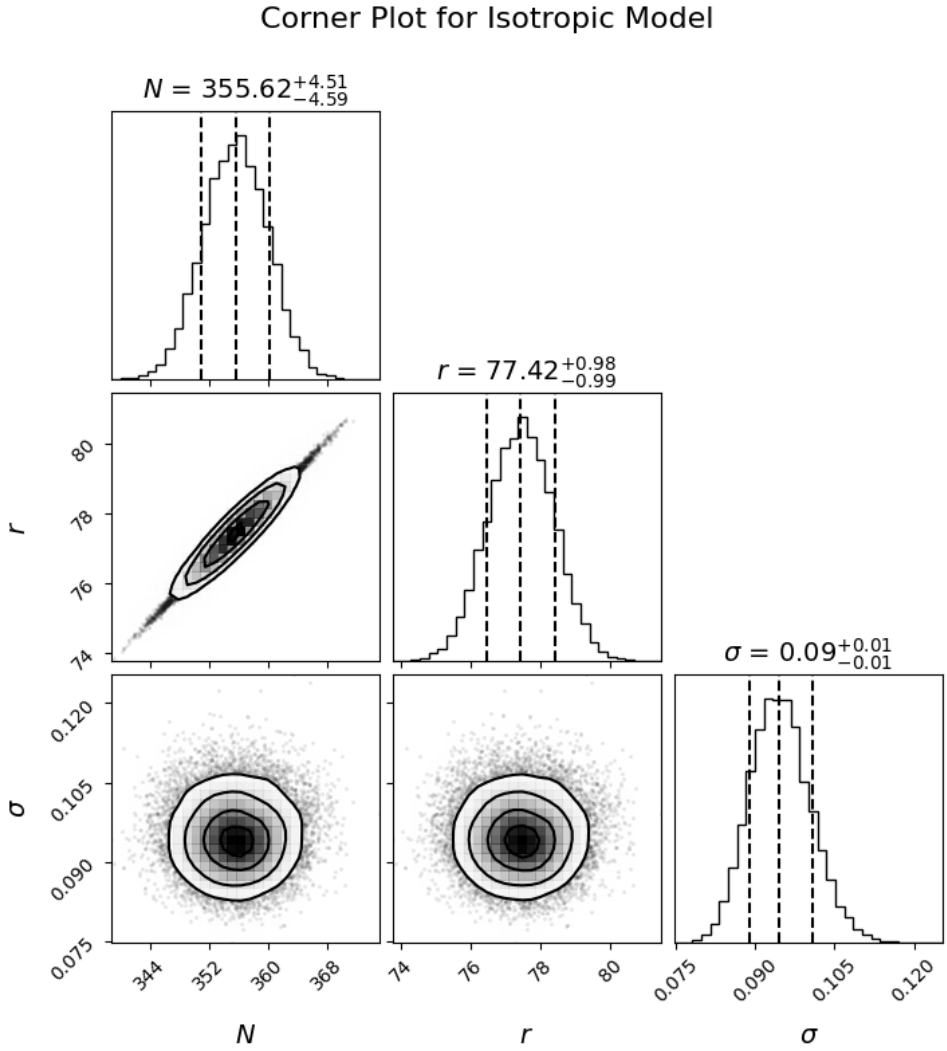

Figure 8: Corner plot of the Isotropic model revealing broader uncertainties and stronger correlation between N and r compared to the other model.

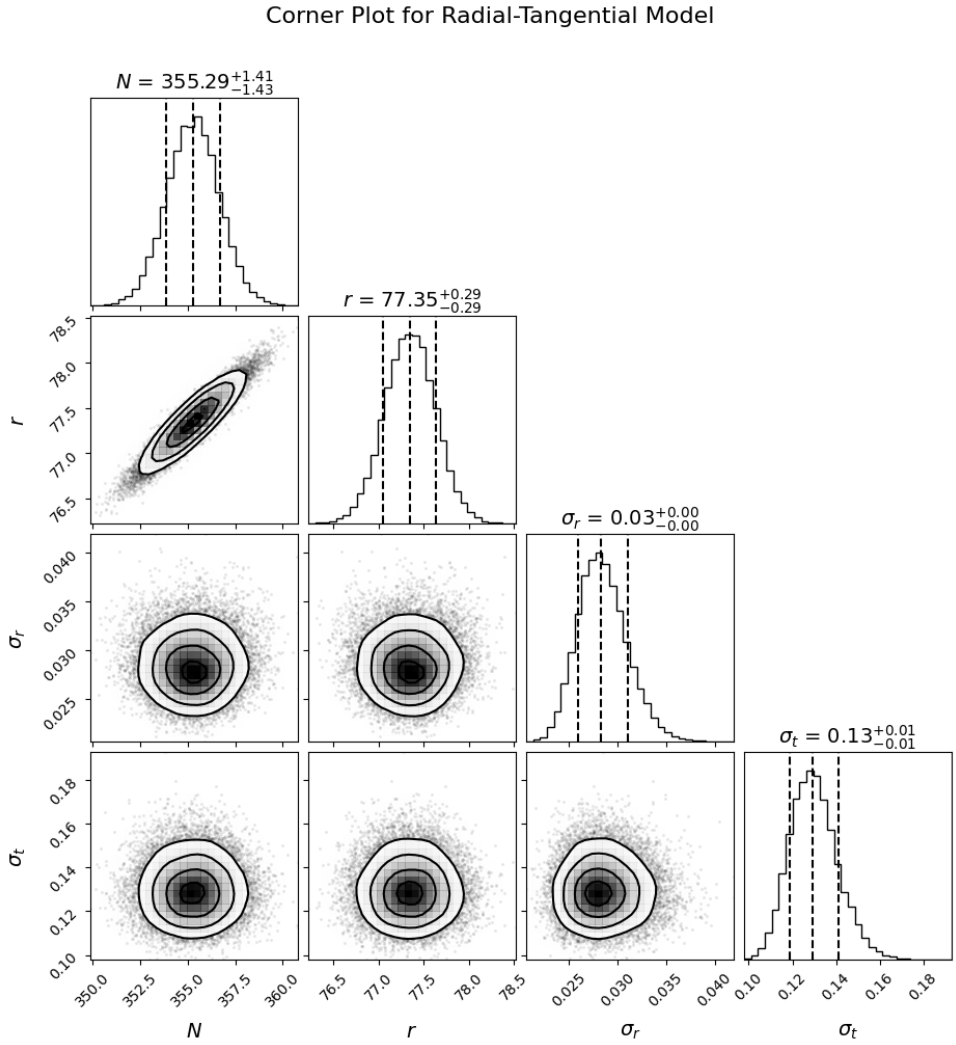

Figure 9: Corner plot showing posterior distributions and parameter correlations for the Radial-Tangential model, with tight uncertainties and minimal skew.

6.3 Posterior Predictive Distribution for a Hole

For this part we want to determine the posterior predictive distribution of a single hole. The way one can do this, is to implement a dynamic code, allowing the reader to choose a specific hole by selecting index value, i.e. outputs are reproducible for every hole.

6.3.1 Theory

Let us recall posterior predictive distribution from the lectures,

$$p(\tilde{x}|D) = \int \mathcal{L}(\tilde{x}|\theta)P(\theta|D)d\theta \tag{13}$$

where,

- $\tilde{x}$ is the predicted observation,
- $p(\tilde{x}|D)$ is the predictive posterior distribution,
- $\mathcal{L}(\tilde{x}|\theta)$ is the likelihood of new the data given parameters,
- $P(\theta|D)$ is the posterior distribution.

6.3.2 Coding & Plots

Find posterior predictive samples for both Isotropic and Radial-Tangential Models, this can be achieved with Python when using the module `Predictive` from `numpyro.infer`. From the predictive samples, we drew the posterior predictive distribution for each single hole. To plot the posterior predictive distributions for one hole at the time, there is a coded function in `plotting.py` called `joint_plot_sns` that outputs a 2D joint plot together with marginals distribution. The posterior predictive plots (Figs. 10, 11) show that the Radial-Tangential model captures directional uncertainty more effectively, producing an elongated distribution aligned with local geometry. In contrast, the Isotropic model assumes uniform uncertainty, resulting in a circular spread. This highlights the Radial-Tangential model's advantage in modelling anisotropic errors present in the fractured ring structure.

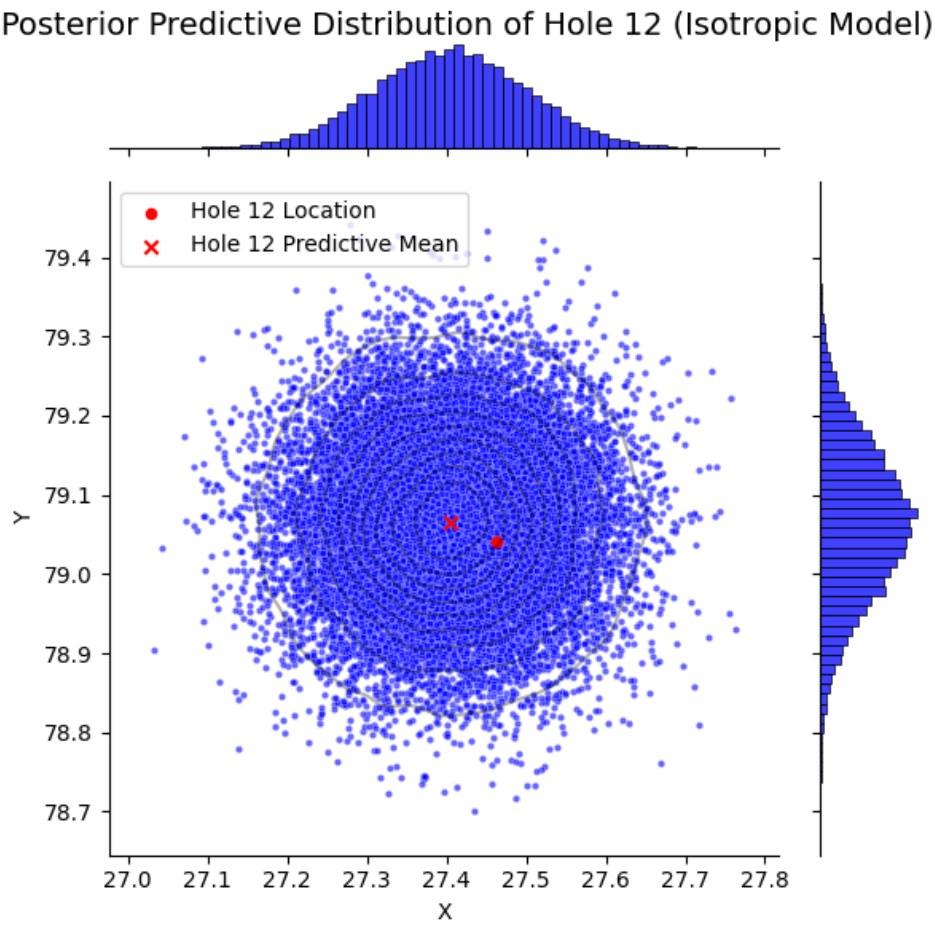

Figure 10: Posterior predictive distribution of Hole 12 using Isotropic model displays circular symmetry, reflecting equal uncertainty in all directions.

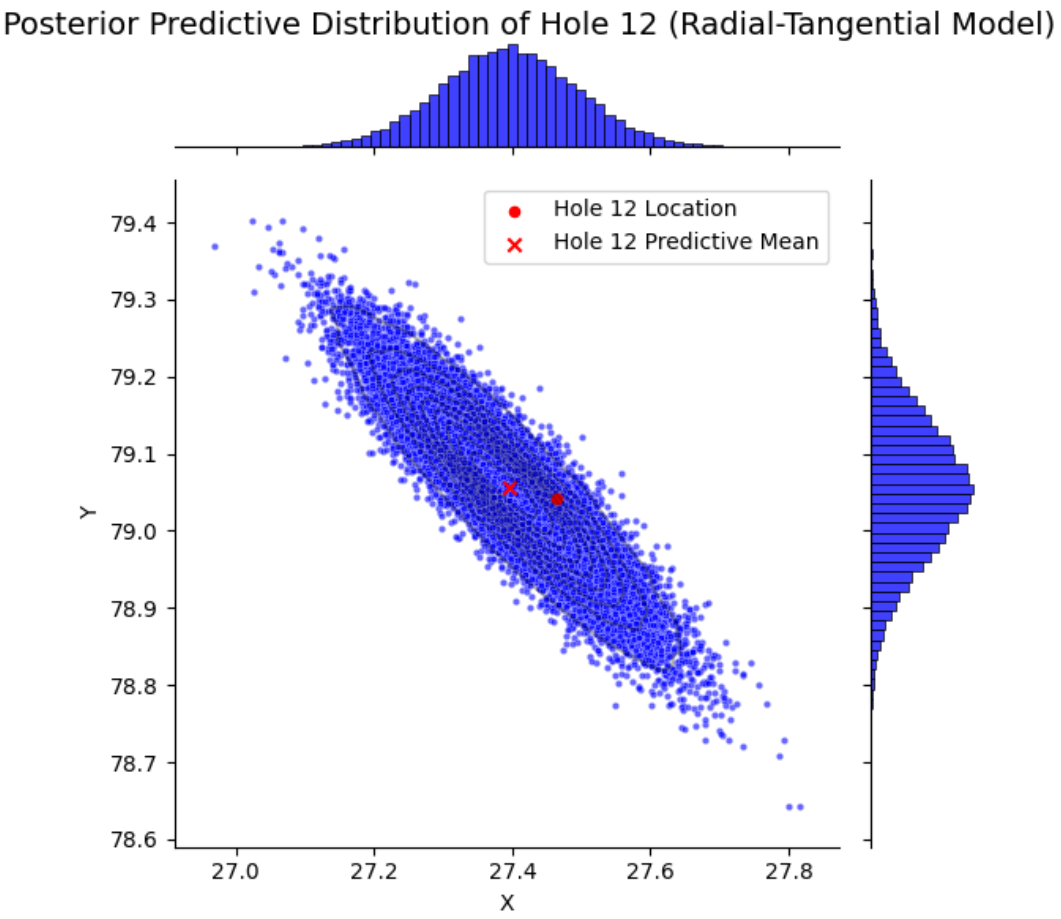

Figure 11: Posterior predictive distribution of Hole 12 using Radial-Tangential model shows strong directional uncertainty aligned with error covariance structure.

7 Model Comparison and Discussion

So far we collected enough information to declare Radial-Tangential as the favourite model, both MLE and HMC showed a much narrower uncertainty with all (especially global parameters) parameter predictions. Let us pretend that what we have done so far is not enough, then an ulterior test that we can perform is the Bayesian Factor (BF), which requires the evidence (Z_{iso} and Z_{rt}) of both posterior distributions. The evidence acts like a normalisation factor thus:

$$Z = \int \mathcal{L}(D|\vec{\theta})\pi(\vec{\theta})d\vec{\theta} \tag{14}$$

Conditions for preferred model:

- If $Z_{\text{rt}} >> Z_{\text{iso}}$ then the data clearly favour the radial-tangential model.
- Otherwise, data favour isotropic model

The evidence can be determined both numerically and analytically (rare cases, when the posterior is "nice"). We can compute the evidence numerically through nested sampling.

7.1 Nested Sampling

To tackle this task, we are going to rely on Nested Sampling Algorithm. In the Python file `original_model.py` there are two functions `prior_transform_isotropic` and `prior_transform_rt` (for Isotropic and Radial-Tangential Models respectively), designed to serve as inputs for our Nested Sampling Algorithm. To execute the algorithm, one can us the module `NestedSampler` and its features from the Python package `dynasty`.

7.1.1 `NestedSampler` Results

We ran both models for 2000 nlive points, providing the following output:

```
Log Evidence (Radia-Tangential Model): 110.27
Log Evidence (Isotropic Model): 55.23
```

We also extracted the parameter estimates and their uncertainties and these are the results: If we

Table 5: Radial-Tangential Model: `NestedSampler` global parameter estimates

Parameter	Mean	Std	Median	5%	95%
N	355.271	1.393	355.271	353.004	357.549
r	77.346	0.284	77.346	76.880	77.811
σ_r	0.028	0.003	0.028	0.025	0.033
σ_t	0.129	0.011	0.129	0.113	0.148

Table 6: Isotropic Model: `NestedSampler` global parameter estimates

Parameter	Mean	Std	Median	5%	95%
N	355.497	4.347	355.497	348.520	362.908
r	77.398	0.936	77.398	75.903	78.981
σ	0.095	0.006	0.095	0.086	0.105

compare the results from Tables 3, 4 to the result displayed above, we can see that they are very close.

7.2 Bayesian Factor

Once the evidence of both models is determined, one can compute the **Bayes Factor (BF)** as follows:

$$BF = \frac{Z_{rt}}{Z_{iso}} \tag{15}$$

Or, more commonly (and more numerically stable), using the log-evidence:

$$\log BF = \log Z_{rt} - \log Z_{is} \tag{16}$$

And interpret it using **Jeffreys' scale**:

Table 7: Jeffreys' Scale for Interpreting $\log BF$

$\log BF$ **(log-evidence difference)**	**Strength of Evidence (RT vs Isotropic)**
< 1	Not worth more than a bare mention
[1, 2.5]	Substantial
[2.5, 5]	Strong
> 5	Decisive

The difference in log-evidence $\log Z_{rt} - \log Z_{is} \simeq 55.03 > 5$, this suggests that, the evidence from Radial-Tangential model is decisive, we can confidently select the Radial-Tangential model over the Isotropic model.

8 Conclusion

The covariance matrix Σ plays a central role in quantifying measurement uncertainty. The isotropic model assumes uniform uncertainty in all directions, simplifying inference but failing to capture the directional nature of measurement errors in fractured components. The RT model introduces anisotropic covariance, better reflecting realistic uncertainty aligned with the geometry of the ring.

Quantitative results from both MLE and HMC sampling showed tighter uncertainty bounds for the RT model, especially in global parameters like N and r. Posterior predictive plots confirmed that the RT model captures directional spread more accurately, as seen in the elongated predictive clouds compared to the circular spread of the isotropic model.

Nested Sampling further confirmed this preference, with the RT model yielding significantly higher Bayesian evidence ($\log Z\_\text{rt} = 110.27$ vs $\log Z\_\text{is} = 55.23$), indicating strong support for the RT model under Bayes Factor analysis.

In summary, the data clearly favours the RT model, both theoretically—due to its more physically grounded covariance structure—and empirically—based on predictive performance and evidence metrics. This affirms the RT model as the more appropriate choice for modelling fractured components of the Antikythera calendar ring.

References

[1] Chris Budiselic et al. “Antikythera Mechanism shows evidence of lunar calendar”. In: (2021).

[2] John Hugh Seiradakis and MG Edmunds. “Our current knowledge of the Antikythera Mechanism”. In: *Nature Astronomy* 2.1 (2018), pp. 35–42.

[3] Graham Woan and Joseph Bayley. “An improved calendar ring hole-count for the Antikythera mechanism”. In: *arXiv preprint arXiv:2403.00040* (2024).