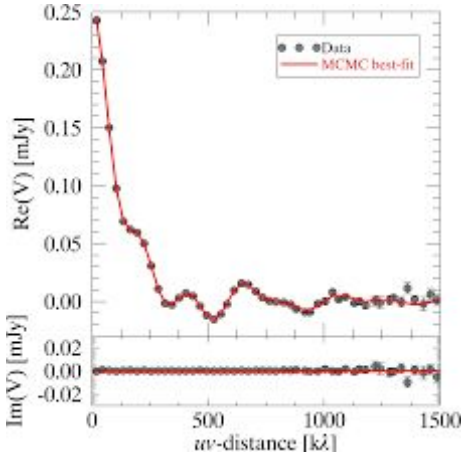
Galario, Frankenstein, and Visibility Modelling

Logan Francis
PhD Student
University of Victoria

Visibility Modelling

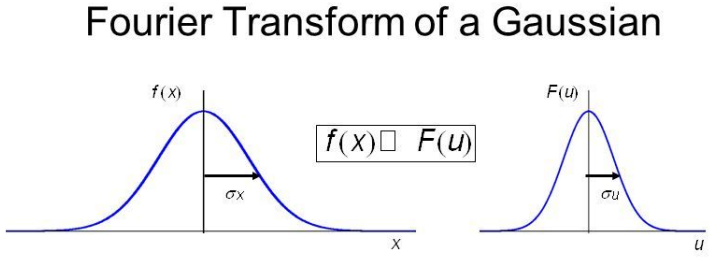
- Our measurements are of the visibility function - we don’t know the true sky brightness, and features in CLEAN images are not always robust.
- We can skip the imaging process completely and instead **compare a model** of the sky **directly** to the uv-plane data.
- Visibility modelling **can also recover structures smaller than the nominal beam*** (super-resolution).

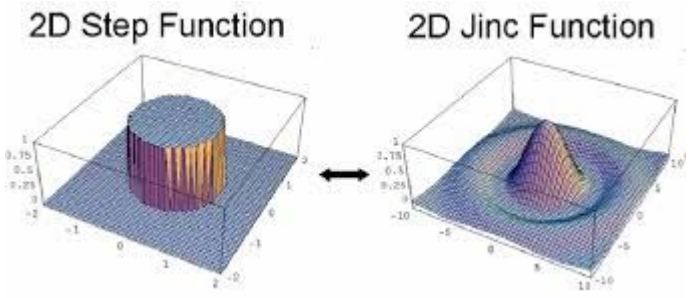
$$V(u,v) = \iint I_\nu(l,m) e^{-2\pi i(ul+vm)} dl dm$$

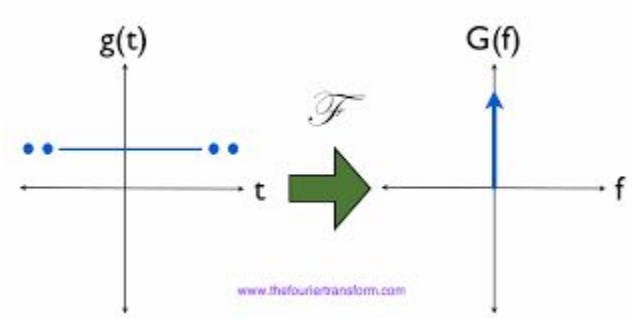


Visibility Modelling

- Option 1: - Use simple analytical functions with exact Fourier transforms (Gaussians, uniform disk, point source, etc.)
- Pro: quick to compute exactly, easy to compare and fit to data.
- Cons: May not adequately describe our data, particularly for complex extended sources.





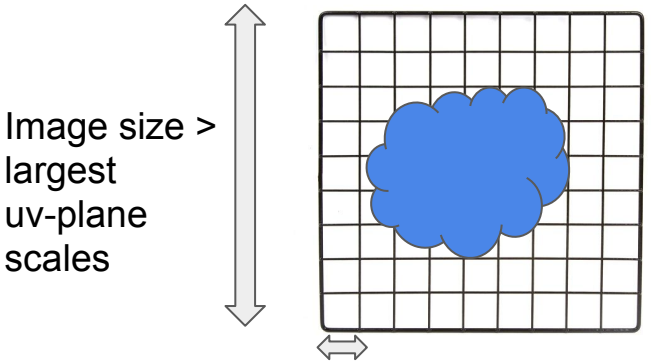


Visibility Modelling

- Option 2: Make any arbitrary model image, and take a fourier transform to get visibilities.
- Pros: Include any physics/structures you want.
- Cons: Need to calculate an FFT of image.
- Model image **must adequately sample smallest/largest scales in uv-plane!**
- **FFT expensive** for high-resolution, wide field data

Discrete Fourier transform $O\left(N^2\right)$

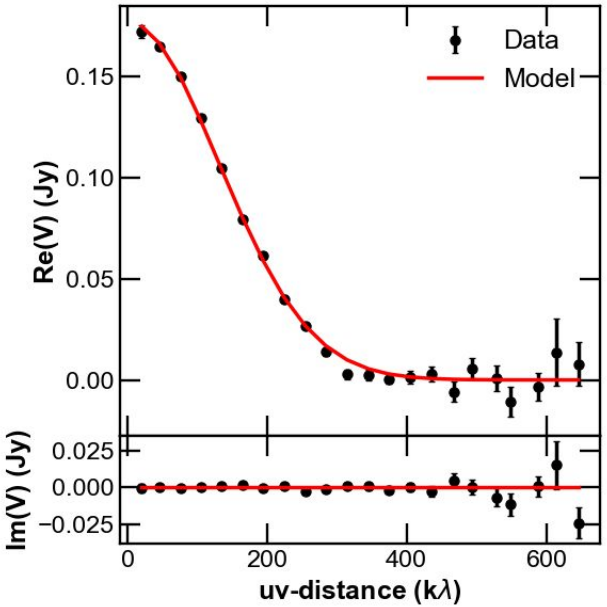
Fast Fourier transform $O\left(Nlog_2N\right)$



What is Galario?

- Galario is a software library for performing visibility modelling with model images (Tazzari 2018).
- Galario can use GPUs to compute an **FFT** fast.
- Moreover, galario has several useful tools for facilitating model fitting.

Galario Features



`galario.double.`**`sampleProfile`**()

Compute the synthetic visibilities of a model with an axisymmetric brightness profile.

The brightness profile `intensity` is used to build a 2D image of the model, which is then Fourier transformed and sampled in the (u, v) locations given in the `u` and `v` arrays.

The image is created as in **`sweep()`** assuming that the x-axis (R.A.) increases from right (West) to left (East) and the y-axis (Dec.) increases from bottom (South) to top (North).

Typical call signature:

```
vis = sampleProfile(intensity, Rmin, dR, nxy, dxy, u, v,
                    dRA=0, dDec=0, PA=0, inc=0, check=False)
```

`galario.double.`**`sampleImage`**()

Compute the synthetic visibilities of a model image at the specified (u, v) locations.

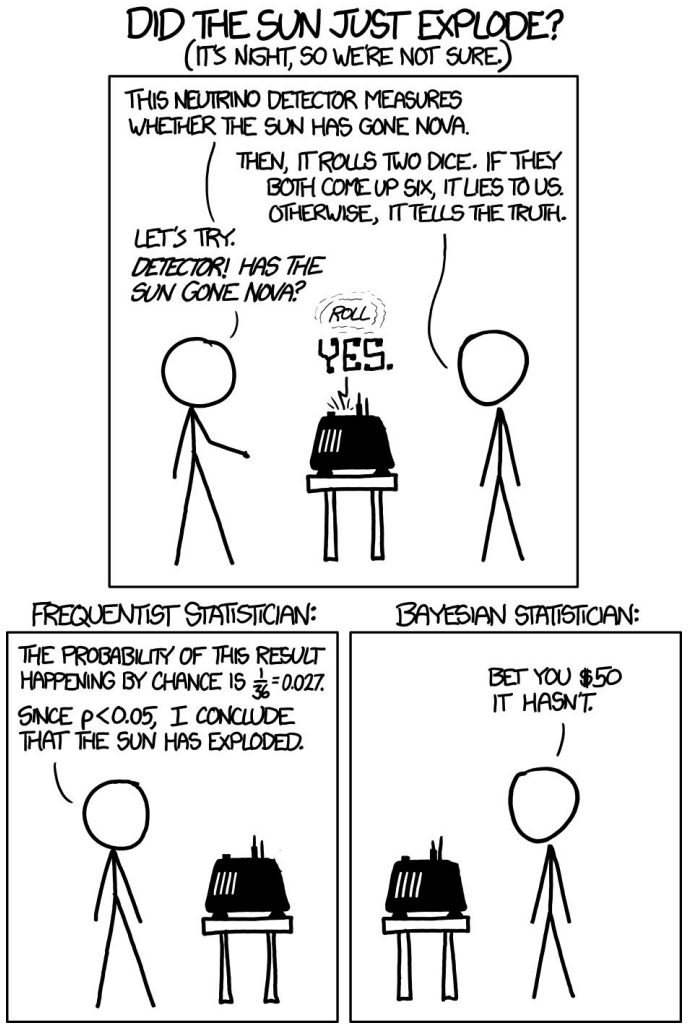
The 2D surface brightness in `image` is Fourier transformed and sampled in the (u, v) locations given in the `u` and `v` arrays.

Typical call signature:

```
vis = sampleImage(image, dxy, u, v, dRA=0, dDec=0, PA=0, check=False, origin='upper')
```

Bayesian MCMC Modelling

- Galario is fast enough that we can use a Bayesian Inference + Monte Carlo Markov Chain exploration to **evaluate how well our model fits the data.**
- This is normally **expensive** as 1000s of model calculations are required.



Bayes Theorem

- Defined as:

$$\text{prob}(M|D) = \frac{\text{prob}(D|M) * \text{prob}(M)}{\text{prob}(D)}$$

- $\text{prob}(M|D)$ = **posterior probability** = probability our model is right, given the data
- $\text{prob}(D|M)$ = **likelihood** = probability of seeing our data, given our model (*and its parameters*)
- $\text{prob}(M)$ = **prior probability** on M = probability of our model (and its parameters)
- $\text{prob}(D)$ = **evidence** = probability of our data (usually ignored)

Bayes Theorem II

$$\text{prob}(M|D) = \frac{\text{prob}(D|M) * \text{prob}(M)}{\text{prob}(D)}$$

- If we assume that the observed visibility samples $V\text{obs}$ are independent and random samples of V, we can derive the following form of the ln likelihood:

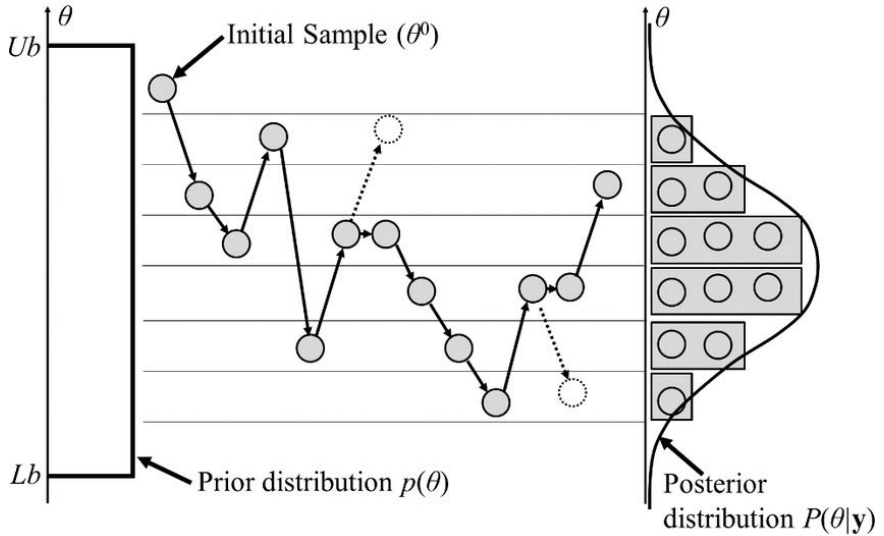
$$\ln(\text{prob}(M|D)) = -0.5 \sum_i \frac{V(u_i, v_i) - V_{\text{mod}}(u_i, v_i)}{\sigma(u_i, v_i)} = -0.5\chi^2.$$

- For a uniform prior, we can set:

$$\ln(\text{prob}(M)) = \begin{cases} -\infty & \vec{p} < \vec{p}_{\min}, \vec{p} > \vec{p}_{\max} \\ \sum_i \ln\left(\frac{1}{p_{i\max} - p_{i\min}}\right) & \vec{p}_{\min} <= \vec{p} <= \vec{p}_{\max} \end{cases}$$

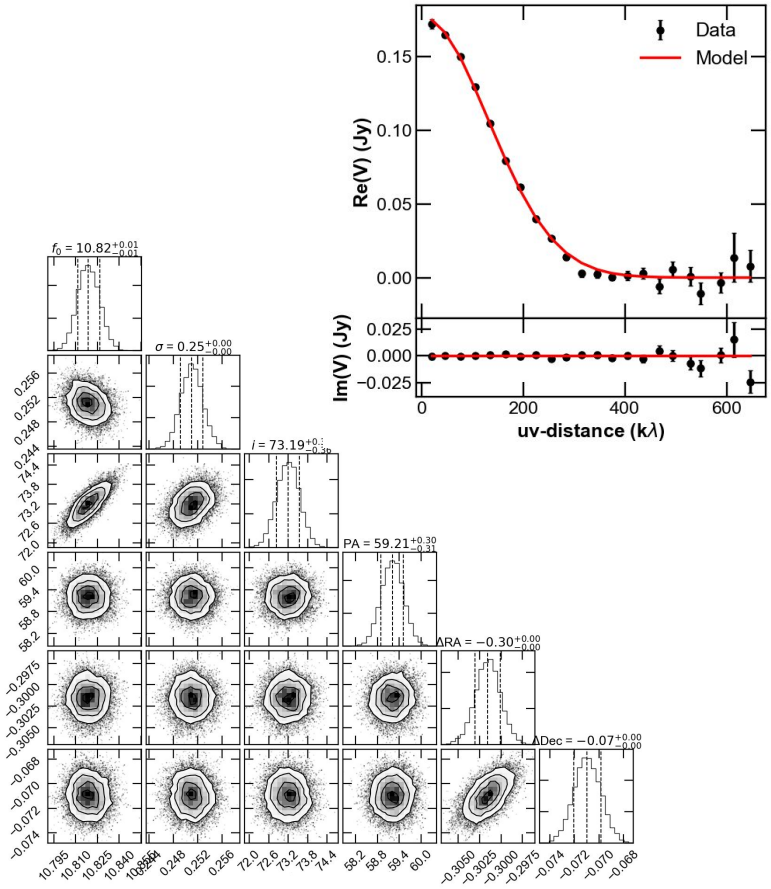
MCMC Sampling

- We could simply brute force calculations of the posterior probability
- This becomes very expensive as we add more parameters (curse of dimensionality.
- MCMC sampling uses “walkers” that sample the Posterior randomly with steps determined by Markov Chain Probabilities



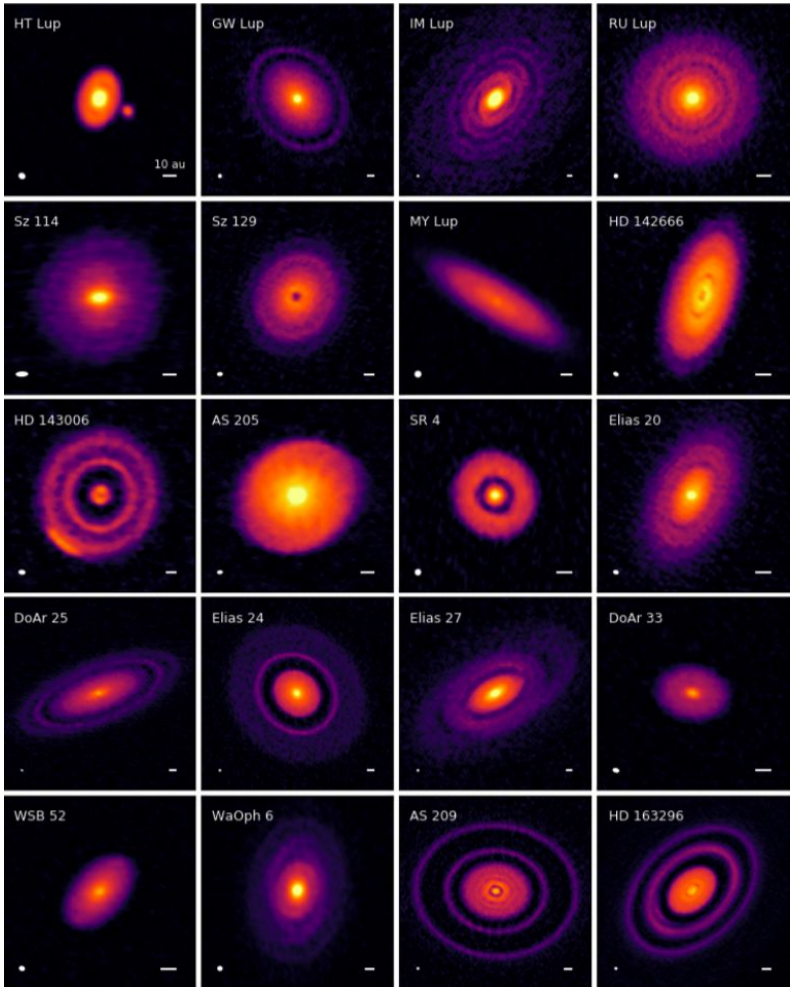
Bayesian Modelling + MCMC

- With a well explored posterior, we can obtain excellent “corner plot” descriptions of the model fit.
- **MCMC is not required!** This is just a way of making the sampling problem tractable.
- These plots are great for seeing correlations, poorly constrained parameters, and more.



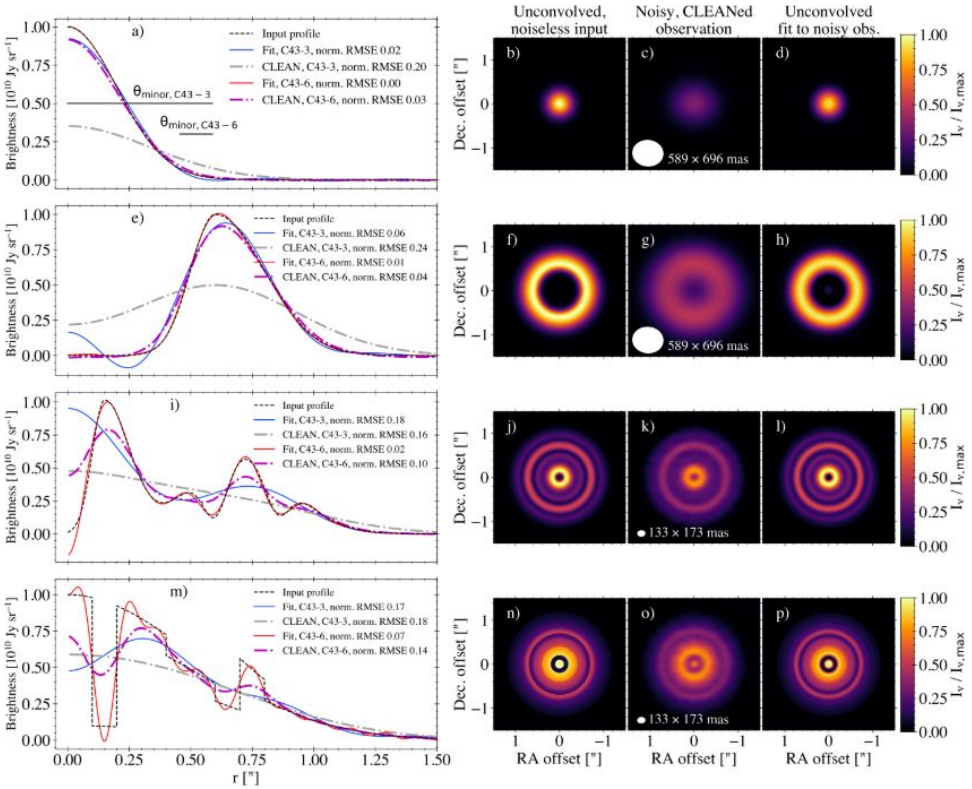
What is Frankenstein?

- Frankenstein (Jennings 2020) is a code which uses a **Gaussian process** to quickly reconstruct **radial** disk brightness profiles.
- Pros: extremely fast, flexible, model fitting in minutes. **Non-parametric** (almost).
- Cons: ONLY suitable for symmetric disks (e.g. DSHARP).



Frankenstein Model Fitting

- Based on **Regularized Maximum Likelihood** methods.
- Find the most probable model radial profile using radial profile **smoothness** as a regularizer.
- Can recover structures < beam size in mock observations.



Resources

Galario Docs: https://mtazzari.github.io/galario/index.html

MCMC Without all the BS: https://jeremykun.com/2015/04/06/markov-chain-monte-carlo-without-all-the-bullshit/

Frankenstein Paper: https://ui.adsabs.harvard.edu/abs/2020MNRAS.495.3209J/abstract

Frankenstein Docs: https://discsim.github.io/frank/