APPLICATION OF THE SABC ALGORITHM TO THE SOLAR DYNAMO MODEL

Physics of Data – Laboratory of Computational Physics

Calandra Buonaura Lorenzo

Rossi Lucrezia

Turci Andrea

July 22, 2024

SOLAR DYNAMO THEORY (1)

The **solar dynamo** is the mechanism that generates and maintains the Sun's magnetic field through a dynamic cycle that repeats every approximately 11 years, known as the solar cycle. This process is fundamental for understanding variations in the Sun's magnetic field and the resulting manifestations of solar activity, such as sunspots, flares and solar storms.

Based on **Magnetohydrodynamics** (MHD): combines the equations of fluid dynamics and electromagnetism to describe how magnetic fields interact with conducting fluids.

The solar dynamo is driven by two main mechanisms: the α effect (alpha-effect) and the Ω effect (omega-effect).

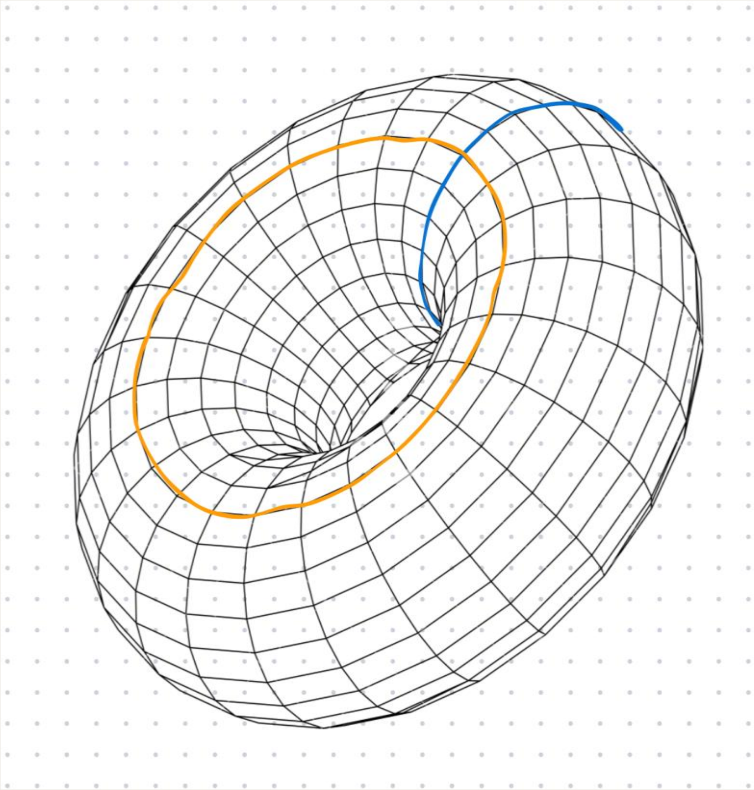
SOLAR DYNAMO THEORY (2)

α Effect (Alpha-Effect):

This effect describes the generation of a poloidal magnetic field (perpendicular to the rotation axis of the Sun) through plasma turbulence. Electric currents induced by plasma movements in the Sun generate magnetic fields that organize and strengthen, producing an overall stronger magnetic field.

Ω Effect (Omega-Effect):

This effect refers to the generation of a toroidal magnetic field (parallel to the rotation axis) through the differential rotation of the Sun. The Sun rotates faster at the equator than at the poles, and this differential rotation stretching the poloidal magnetic fields, transforming them into toroidal fields.



Toroidal configuration, characterized by two circumferences: the plane on which the smaller one lies (blue) is called poloidal, the one on which the larger circumference (orange) is called toroidal.

MAIN CHARACTERISTICS OF THE MODEL

The 11-year solar cycle, known as the Schwabe cycle, is modulated by longer cycles that influence its amplitude and are associated with periods of low solar activity, such as Deep Lows. To better understand this variability, it is useful to consider the **zero-dimensional solar dynamo** model proposed by Wilmot-Smith et al. (2006).

The two main characteristics of this type of dynamo are:

- **Limit of the α Effect**: The α effect, responsible for the regeneration of the poloidal magnetic field, is effective only when the intensity of the toroidal field exceeds a minimum threshold and does not exceed a maximum threshold, as indicated by Ferriz-Mas et al. (1994).

- **Time Lag**: The model incorporates a time lag that takes into account the effect of the meridional circulation, a key element in Babcock-Leighton (BL) type models. This time delay affects the dynamic behavior of the system.

DYNAMICS OF MODES

The model features two stable oscillation modes: Weak Mode and Strong Mode.

- **Weak Mode**
  - Characterized by frequencies and amplitudes subject to periodic or chaotic modulation.
  - Chaos is possible thanks to time delay, which makes the system infinite-dimensional.
  - In this mode, the toroidal magnetic field mostly remains below a certain maximum value, Bmax.
  - This mode can explain Deep Lows, periods during which solar magnetic activity does not completely stop.

- **Strong Mode**
  - Characterized by a stable limit cycle without modulations.
  - The amplitude of this mode can be much larger than Bmax.
  - A twist is observed in the decreasing branch of the cycle, an effect also seen in sunspot data.

BL TYPE DYNAMO MODEL (1)

- The **Babcock-Leighton (BL)** type dynamo model uses two delay-coupled ordinary differential equations to study the behavior of the solar magnetic field.

- The **toroidal field** is influenced by the angular velocity, the characteristic length and the retarded poloidal field.

- The **poloidal field** is regenerated by the α effect and is influenced by the delayed toroidal field.

BL TYPE DYNAMO MODEL (2)

Combining the equations of the two fields we obtain a second order differential equation:

$$\frac{d^2B(t)}{dt^2} + B(t) = N\, f(B(t-q))\, B(t-q)$$

- $\frac{d^2B(t)}{dt^2}$: second derivative of the toroidal magnetic field with respect to time
- $B(t)$: toroidal magnetic field
- N : number of dimensionless dynamo
- $f\big(B(t-q)\big)$: nonlinear function representing the α effect
- $B(t-q)$: delayed toroidal field of q
- $q = \frac{(T_0 + T_1)}{\tau}$: Total delay, with T_0 and T_1 which represent the delays associated with the meridional circulation and the toroidal field, respectively, and τ a constant representing a diffusion time scale.

STOCHASTIC RESONANCE

- Stochastic resonance occurs when a nonlinear system, such as a Babcock–Leighton (BL) type solar dynamo, is influenced by external perturbations such as noise or periodic modulation.
- This phenomenon amplifies the system's response to external stimuli, altering the dynamic behavior of the system and its oscillation modes.
- To explore the effect of stochastic resonance, a small periodic modulation and additive white noise is added to the second-order differential equation:

$$\frac{d^2B(t)}{dt^2} + B(t) = N\, f\big(B(t - q)\big)B(t - q) + \epsilon \cos(\omega_d t) + \eta(t)$$

where ϵ is the amplitude of the periodic modulation and $\eta(t)$ is white noise.

- When the noise is relatively low, the system can switch between strong and weak modes depending on the critical conditions and coexistence interval.
- As noise intensity increases, dynamo behavior can exhibit Grand Minimum-like episodes, i.e., prolonged periods of low solar activity.

SUMMARY

- **BL Dynamo Theory**:
  - Differential equations describe how the magnetic field evolves over time, considering delays caused by physical processes such as meridional circulation.

- **Stochastic Resonance**:
  - By introducing periodic modulation and noise, we analyze how the dynamo's oscillation modes can be influenced and amplified.
  - The system becomes particularly sensitive to small perturbations near critical points, which can lead to complex dynamic behavior, such as modulation of oscillation amplitude and frequency.

STATISTICAL APPROACH

- **Goal** of the project:
  - Infer parameters from the data
  - Without computing the likelihood

- **Simulated Annealing Approximate Bayesian Computation** (SABC):
  - Parameter inference with data sampling from target distribution
  - Simulated annealing techniques

- Suitable field of application:
  - Likelihood hard to evaluate, but…
  - … easy to sample from

FROM ABC TO SABC (1)

➢ **Parameter inference** in the Bayesian framework generating parameter samples from the posterior:

$$f_{post}(\theta|y) = \frac{L(y|\theta)f_{pri}(\theta)}{\int L(y|\theta)f_{pri}\,(\theta)}$$

- Usual approach: huge number of samples with Markov Chain Monte Carlo methods

➢ However, they require many evaluations of the likelihood → For complex models the likelihood is expensive to evaluate

➢ **Our case:** treat stochastic differential equation (SDE), whose output is a time-series.

- Simulating output is very fast
- Evaluating the likelihood is very expensive

So, how to perform Bayesian inference?

FROM ABC TO SABC (2)

- The solution comes from the **Approximate Bayesian Computations** (ABC) algorithms:
  - Instead of computing the likelihood ...
  - ... they simulate outputs from the model likelihood and compare with the data

- Requires **introducing a metric** on the space of outputs
  - An ensemble of particles is iteratively updated in order to converge to the exact posterior along a series of approximations

- ABC framework is generic: no information about the model needed (e.g. the equations)

- A framework of ABC algorithm has been introduced, inspired by **Simulated Annealing** algorithm:

  - SABC: developed for optimization tasks, by minimizing the “temperature” of the system.

SABC ALGORITHM (1)

SABC algorithm: Simulated Annealing Approximate Bayesian Computation

System represented by a moving ensemble of particles, $E = \{\theta_i, x_i\}_{i=1}^{N}$, in the product space of parameters and associated model outputs.

Predictions given by ABC: $f_{post}(\theta|y) = \int L(x|\theta) f_{pri}(\theta) \delta(x-y) dx$

If the output space has high cardinality or is continuous:

- Sampling from $L(x|\theta) f_{pri}(\theta) \delta(x-y)$ becomes inefficient or impossible (dimension = $\Theta x X$)

➢ **Solution** given by the SABC: $f_{post}(\theta|y) = \int L(x|\theta) f_{pri}(\theta) e^{-\frac{\rho(x,y)}{\epsilon}} dx$

with euclidian metric in $\mathbb{R}^n$: $\rho(x,y) = \frac{1}{2} \sum_{i=1}^{n} (x_i - y_i)^2$

SABC ALGORITHM (2)

- **SABC framework**: generate, through many simulations, a canonical ensemble of particles E described by the distribution:

$$\pi_{\epsilon_k}(\theta, x) \approx L(x|\theta) f_{pri}(\theta) e^{-\frac{\rho(x,y)}{\epsilon_k}} \qquad \text{so that} \qquad f_{post}(\theta|y) \approx \int \pi_{\epsilon_k}(\theta, x) dx$$

- **Simulated annealing**:
  - Temperature time-dependent $\epsilon_k = \epsilon_k(t)$, and
  - Reduce it through iterations: from $\epsilon_k \to \infty$ to $\epsilon_k \to 0$.

- **Crucial question:**
  - “How fast we should reduce this temperature in order to have a fast convergence to the correct result?”

SABC – NON-INFORMATIVE PRIORS (1)

We lack prior knowledge about model parameters

➢ SABC algorithm for the case of negligible prior information: $\boldsymbol{f_{pri}(\theta) = const}$

With $\pi_{\epsilon_k}(\theta, x)$ equilibrium distribution, we have detailed balance principle:

$$\pi_{\epsilon_k}(\theta, x) \cdot q_{\epsilon_k}\left((\theta, x) \to \left(\theta', x'\right)\right) = \pi_{\epsilon_k}(\theta', x') \cdot q_{\epsilon_k}\left((\theta', x') \to (\theta, x)\right)$$

And transition rates for particle update:

$$q_{\epsilon_k}\left((\theta, x) \to \left(\theta', x'\right)\right) = k\left(\theta, \theta'\right) L\left(x' \middle| \theta\right) \min\left(1, e^{-\frac{\left(\rho\left(x', y\right) - \rho(x, y)\right)}{\epsilon_k}}\right)$$

- A jump in the parameter space described by the transition rate $k\left(\theta, \theta'\right)$
- A simulation of an output from the model (using $L\left(x' \middle| \theta\right)$) with a Metropolis acceptance/rejection

SABC – NON-INFORMATIVE PRIORS (2)

- **Endoreversibility assumption:**

For sufficiently slow annealing, the system will be approximately described by the equilibrium distribution:

$$\mu(z,t) \approx \pi_{\epsilon(t)}(z)$$

- **Two tuning parameters**
  - Annealing Speed v: speed at which temperature is reduced.
  - Mixing speed β: related to the covariance matrix of $k(\theta,\theta')$

- Endoreversibility assumption is satisfied if $\boldsymbol{v}$ is sufficiently **small** and the $\boldsymbol{\beta}$ is sufficiently **big**
  - In this regime we have the convergence to the correct result.

CODE - JULIA

- Julia: high-level, object-oriented, dynamic programming language (both just-in-time and ahead-of-time computation).

- SimulatedAnnealingABC package, which implements the sABC algorithm.

- Usage of a VM on CloudVeneto:

Flavor Details: cloudveneto.xlarge

ID	cfc7bdd0-bd3e-4644-b175-6c6a956e9291
VCPUs	8
RAM	16GB
Size	25GB

- Parallelization:

```
ThreadPinning.pinthreads(:cores)
ThreadPinning.threadinfo()
```

```
System: 8 cores (no SMT), 8 sockets, 1 NUMA domains

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

# = Julia thread, | = Socket seperator

Julia threads: 8
├ Occupied CPU-threads: 8
└ Mapping (Thread => CPUID): 1 => 0, 2 => 1, 3 => 2, 4 => 3, 5 => 4, ...
```

CODE – MODEL FUNCTIONS

```
# Box-shaped function for the magnetic field range
function f(B, B_max = 10, B_min = 1)
  return 1 / 4 * (1 .+ erf.(B .^ 2 .- B_min ^ 2)) .* (1 .- erf.(B .^ 2 .- B_max ^ 2))
end
```

```
# Drift function for the DDE
function drift(du, u, h, p, t, lags)
  N, T, tau, sigma, Bmax = p
  lags = (T, )
  Bhist = h(p, t - lags[1])[1]
  B, dB = u
  du[1] = dB
  du[2] = - ((2/tau)*dB -
          - (B/tau^2)-(N/tau^2)*Bhist*f(Bhist, Bmax))
end


# Noise function for the DDE
function noise!(du, u, h, p, t)
  N, T, tau, sigma, Bmax = p
  du[1] = 0
  du[2] = (sigma * Bmax)/(tau^(3/2))
end
```

```
# SDDE problem solver
function bfield(θ, Tsim, dt)
  τ, T, Nd, sigma, Bmax = θ
  lags = (T, )
  h(p, t) = [Bmax, 0.]
  B0 = [Bmax, 0.]
  tspan = (Tsim[1], Tsim[2])
  prob = SDDEProblem(drift,noise!,B0,h,tspan,θ;
                     constant_lags=lags)
  solve(prob, EM(), dt = dt, saveat = 1.0)
end
```

CODE – SUMMARY STATISTICS

- Parameter space: small dimension; output space: high dimension → we can reduce the output dimension to the order of the dimension of the parameter space without losing too much information about the parameters, using the so called “summary statistics”.
- **Theorem:** only for probability distribution from the exponential family, there exists a bounded set of sufficient statistics (which is our case).
- We use some Fourier components, the most significant ones: this is done as the Fourier spectrum is easier and the resemblance in the spectrum means a resemblance in the signal.

```julia
# function to compute the summary statistics
function reduced_fourier_spectrum(u::Vector{Float64},
        indices::Union{Vector{Int64}, StepRange{Int64, Int64}} = 1:6:120)
  fourier_transform = abs.(fft(u))
  return fourier_transform[indices]
end
```

CODE - SABC

```julia
function f_dist(θ::Vector{Float64}; type::Int64 = 1, indices::Union{Vector{Int64},
StepRange{Int64, Int64}} = 1:6:120, fourier_data::Vector{Float64})
  sol = bfield(θ, Tsim)
  simulated_data = sol[1,:].^ 2
  fourier_stats = reduced_fourier_spectrum(simulated_data, indices)
  rho = [euclidean(fourier_stats[i], fourier_data[i]) for i in 1:length(fourier_stats)]
  return rho
end
```

```julia
prior = product_distribution(Uniform(0.1, 15), Uniform(0.1, 10.0), Uniform(0.1, 6.0),
                             Uniform(0.01, 0.3), Uniform(1, 15))
n_particles = 1000
n_simulation = 20000000
type = "single"

tmin = data.year[1]; tmax = data.year[end]
Tsim = [tmin, tmax]
dt = 0.1

result = sabc(f_dist, prior; n_particles = n_particles, n_simulation = n_simulation,
type = type, indices = indices, fourier_data = sim_ss)
```

DATA

- **SYNTETHIC DATA:**

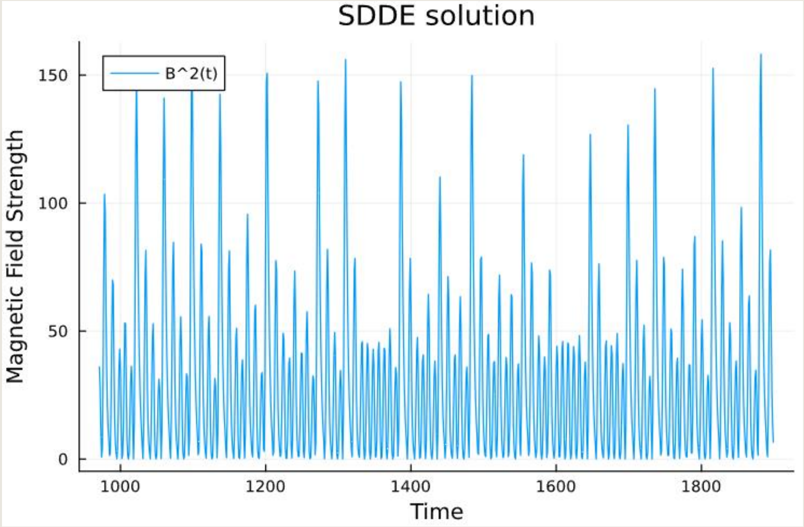
  First we need to test the correct implementation of the model. We use synthetic data (data produced by the model itself, with a specific choice of parameters) and then we run a simulation using the sABC algorithm.

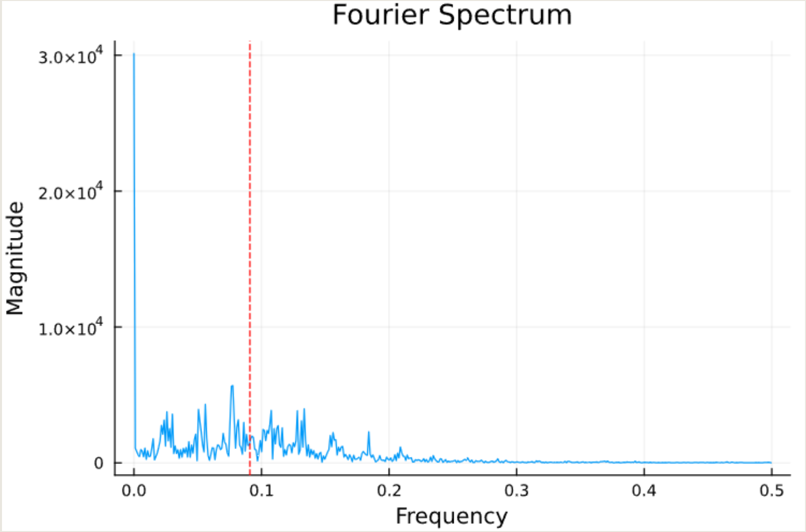
- **REAL DATA:**

  After checking the correctness of the model, we apply the sABC algorithm to the real data. A different choice of summary statistics leads to different results, so we explore different possibilities, looking for the most valid one.

SYNTHETIC DATA

- **Data**: The synthetic data were obtained from the SDDE problem solver. The signal is highly oscillatory and always returns to 0.
- **X-Axis**: Years from 971 to 1899 (929 years), years of real data.
- **Graph 1**: Oscillatory signal.
- **Graph 2**: Fourier Spectrum:
    - **Zero Mode**: Characterizes the return to 0 of the signal.
    - **Peaks**: Clearly visible in the graph.
    - **11-Year Cycle**: Highlighted in red; the highest peak is shifted to a lower frequency (longer period) than expected.
    - **Parameters:** N = 6.2, T = 3.1, τ = 3.5, σ = 0.04, B_{max} = 6.0 (as suggested by Professor Albert).



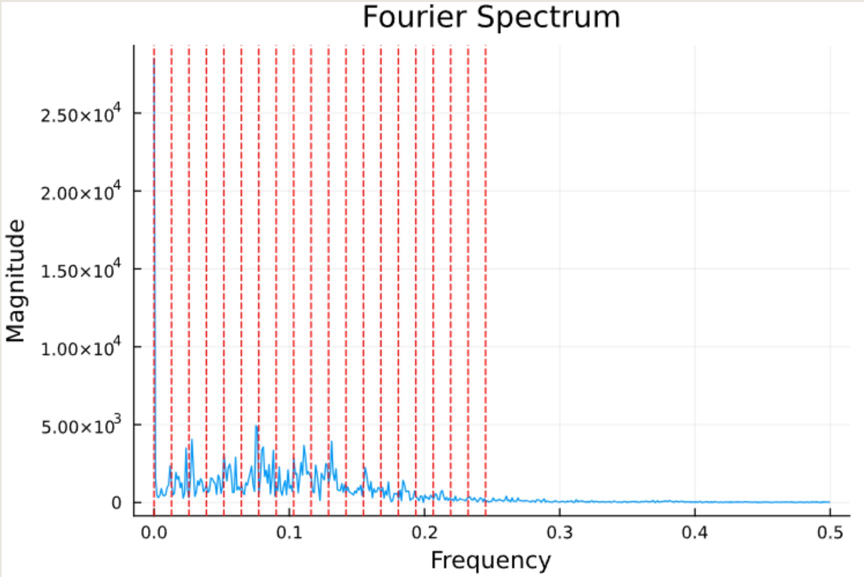


SYNTHETIC DATA – SIMULATION

- **Objective**: Select simulation parameters and summary statistics.

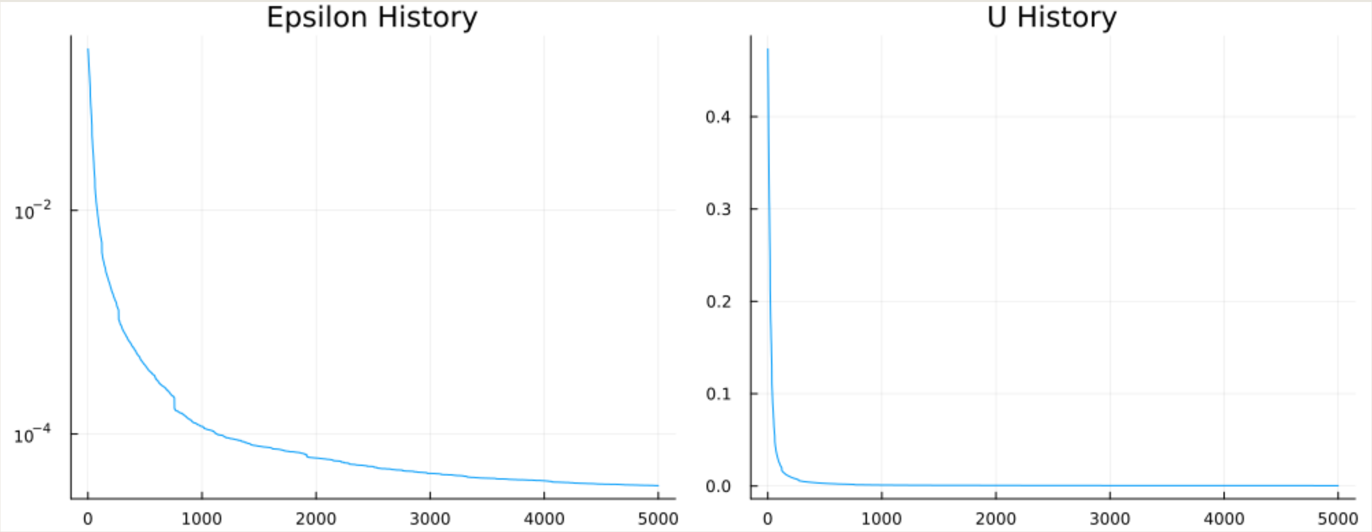
- **Fourier Components**:
  - Selection based on significant peaks in the spectrum.
  - **Random Choice**: For example, choosing one every six components up to a certain limit (1:6:120).

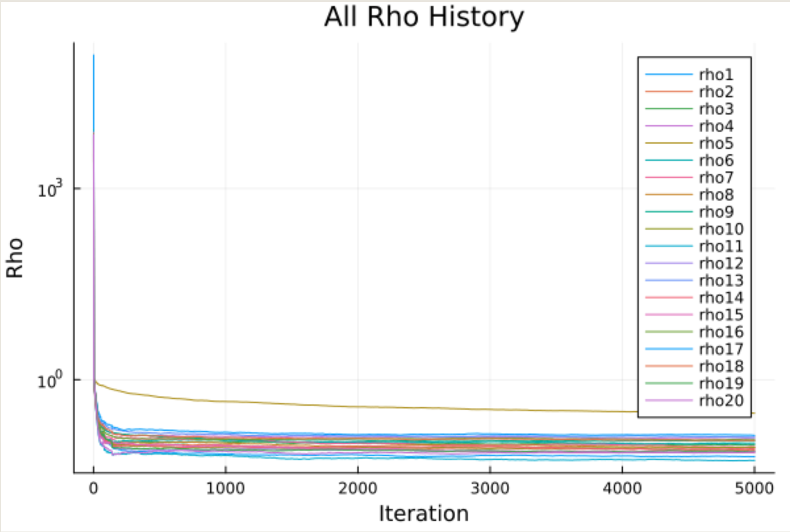
- **Issue**: This choice may capture high peaks (like the zero mode) but could miss other significant peaks.
  - **Reason for Choice**: To reduce the number of compared points and avoid excessive correlation that might bias the spectrum towards one peak.



SYNTHETIC DATA - HISTORY

- **Simulation type**: "Single" (using only one ϵ for all the summary statistic).
- **ϵ and u plots**: Show convergence of the algorithm, 5000 iterations is enough for synthetic data.
  - **Observation**: Both decrease to very small values, indicating that the algorithm is effectively reducing the model's energy.
- **ρ plot**: Represents ρ for each summary statistic versus the number of iterations.
  - **Observation**: All ρ values decrease after a few iterations, suggesting that the choice of summary statistics is appropriate. Slightly higher values might correspond to lower values in the spectrum.

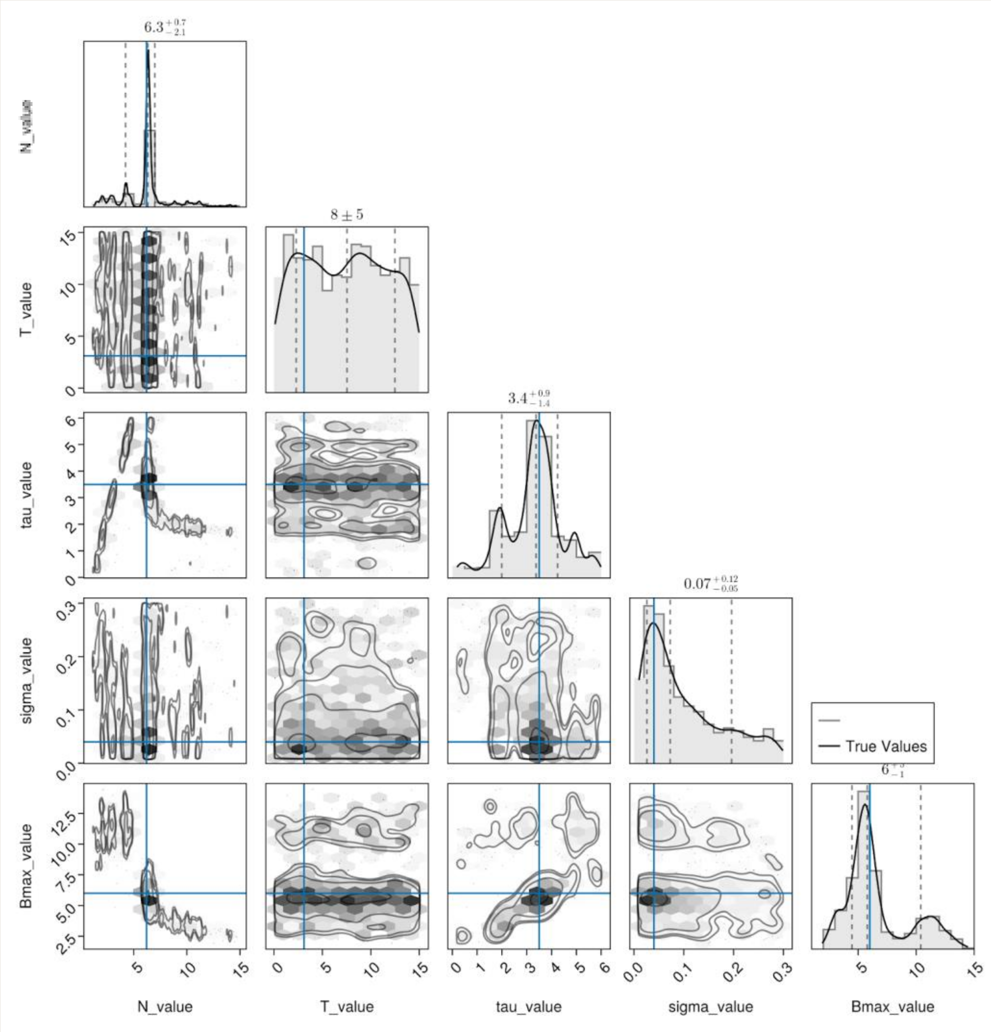




SYNTHETIC DATA - POSTERIORS

Example of Posteriors: Comparison with the chosen parameters used for synthetic data.

•N: Strong peak around 6.3, which is a very good result.
•T (Period): Multimodal distribution with two main peaks (around 3 and 8). This is expected as the period might be confused with its multiples.
•τ: Well-confined around 3.4, but with some smaller peaks, indicating that the algorithm might detect other oscillations with different periods.
•σ (Error): Peaks near small values, which is desirable as we want the error to adjust the main behavior without dominating it.
•B_{max} : Measure of intensity (signal amplitude), which is the least important parameter and could be fixed to ensure comparable spectra.



SYNTHETIC DATA – RESULTS (1)

We got some nice looking posterior. But which values should we choose as the best values for generating the synthetic data?

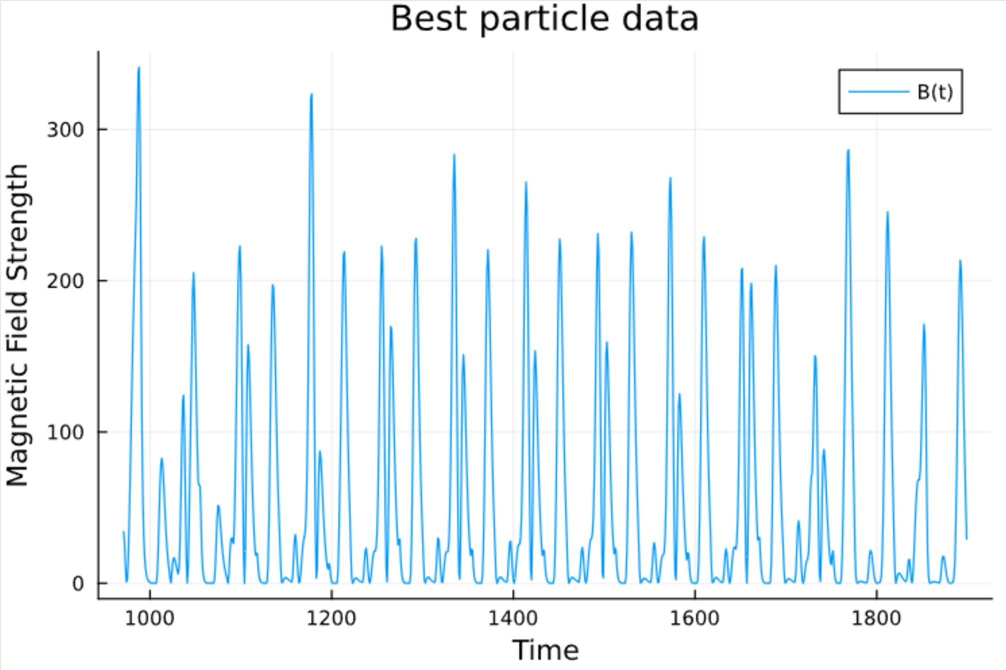
We identified 3 possible choices:

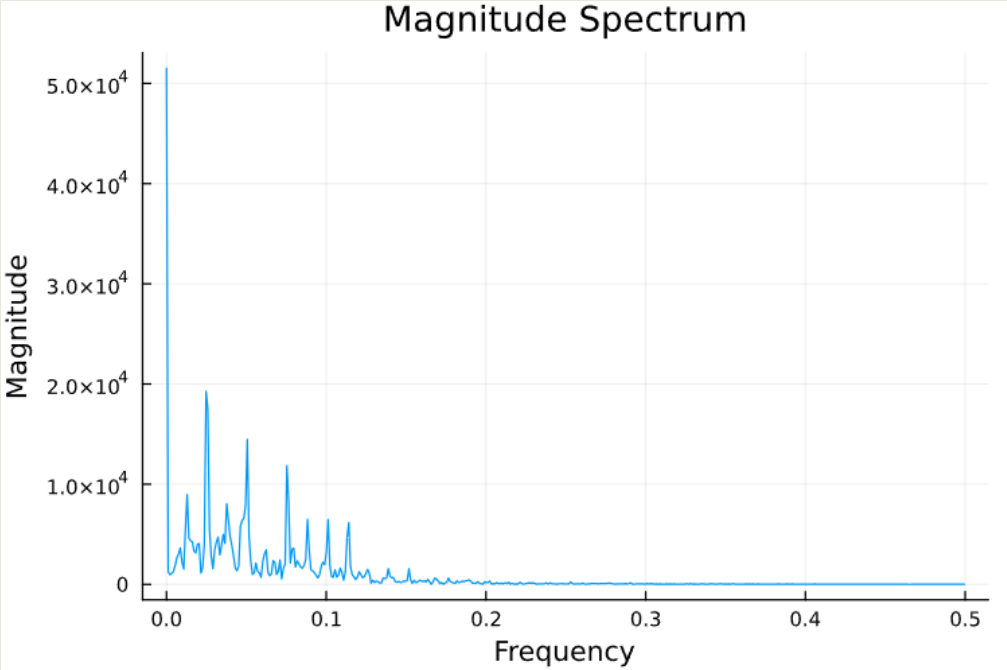
- **Mean** of the posterior: good for highly confined posterior; bad for many peaks posteriors
- **Mode** of the posterior: not always indicative of the best result
- Use the **particle** that at the same time **minimizes the distance** for all the particles together.

SYNTHETIC DATA – RESULTS (2)

➢ **Best Particle** that minimizes the distance

- N = 6.13971
- T = 13.2822 → too high
- tau = 3.60154
- sigma = 0.0394572
- B_max = 5.8472

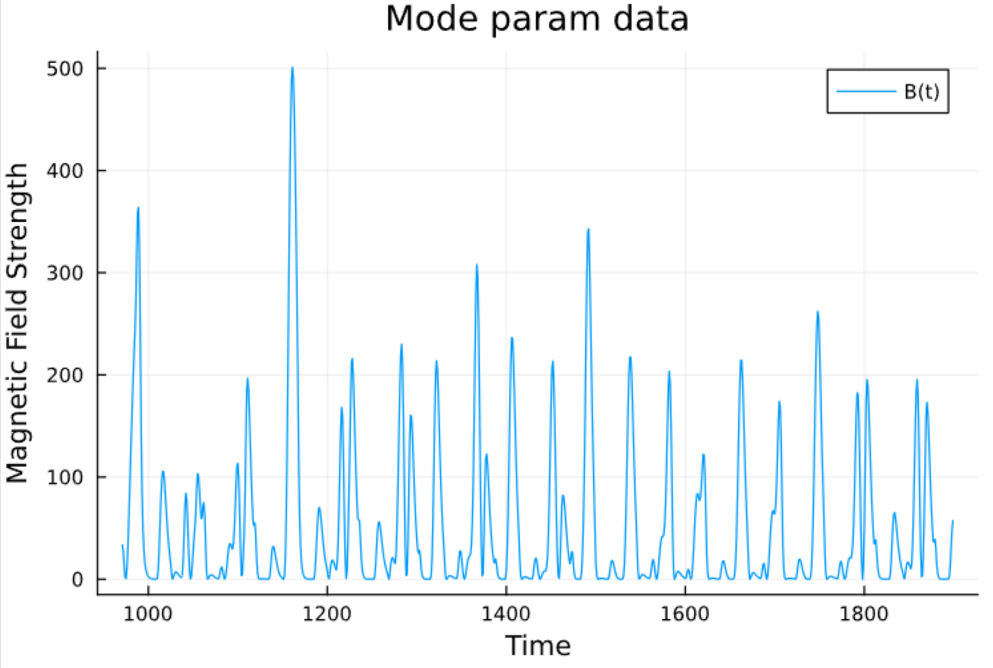


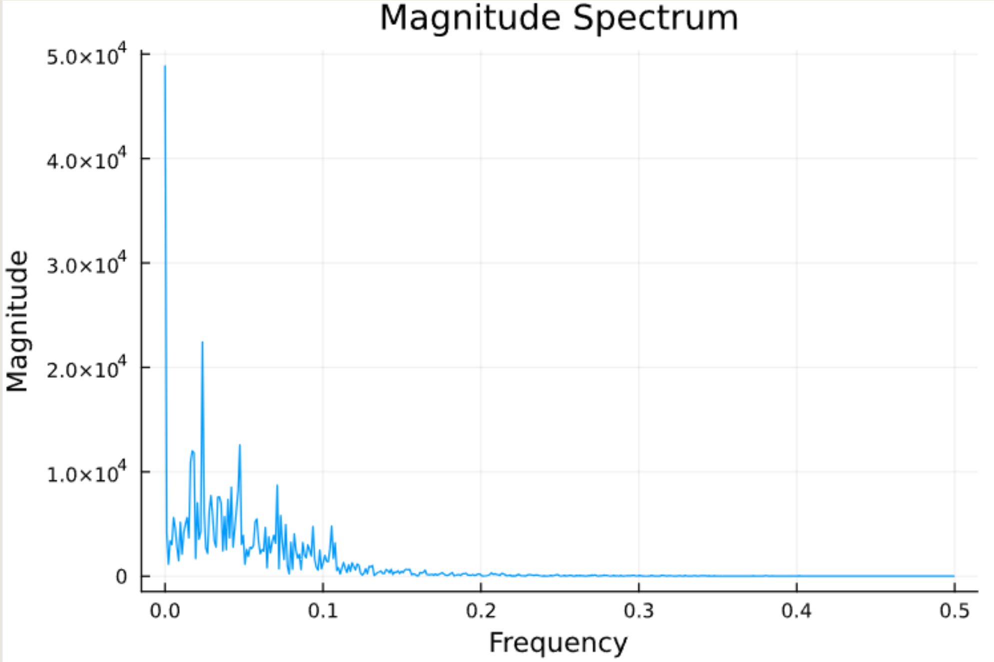


SYNTHETIC DATA – RESULTS (3)

➢ **Mode** of the posteriors

- N = 6.47699
- T = 14.01918 → too high, second peak
- tau = 3.93010
- sigma = 0.04617
- B_max = 5.78465

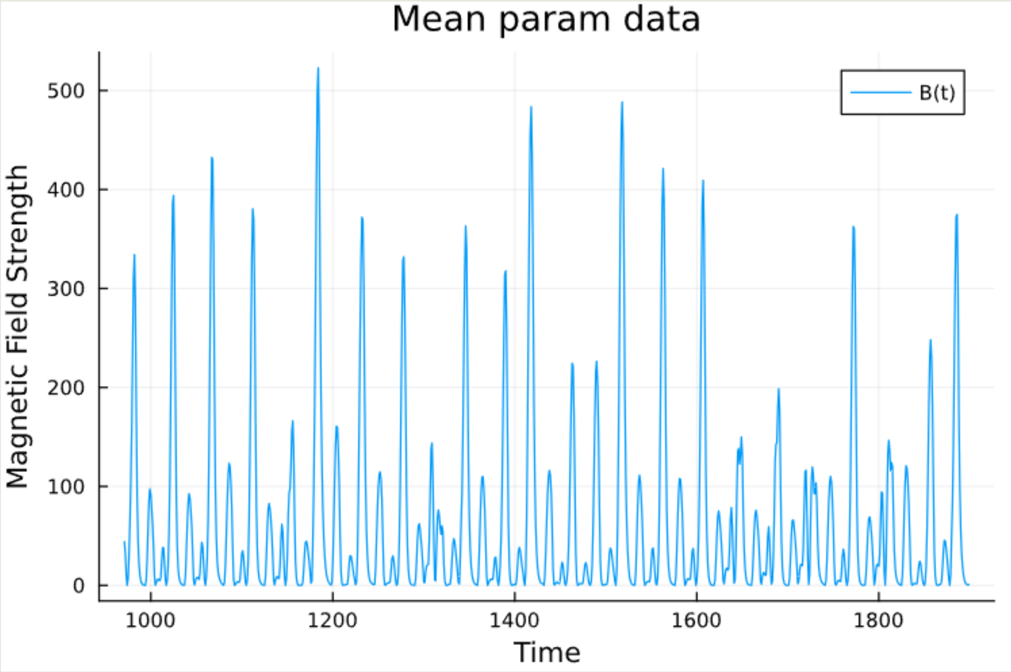


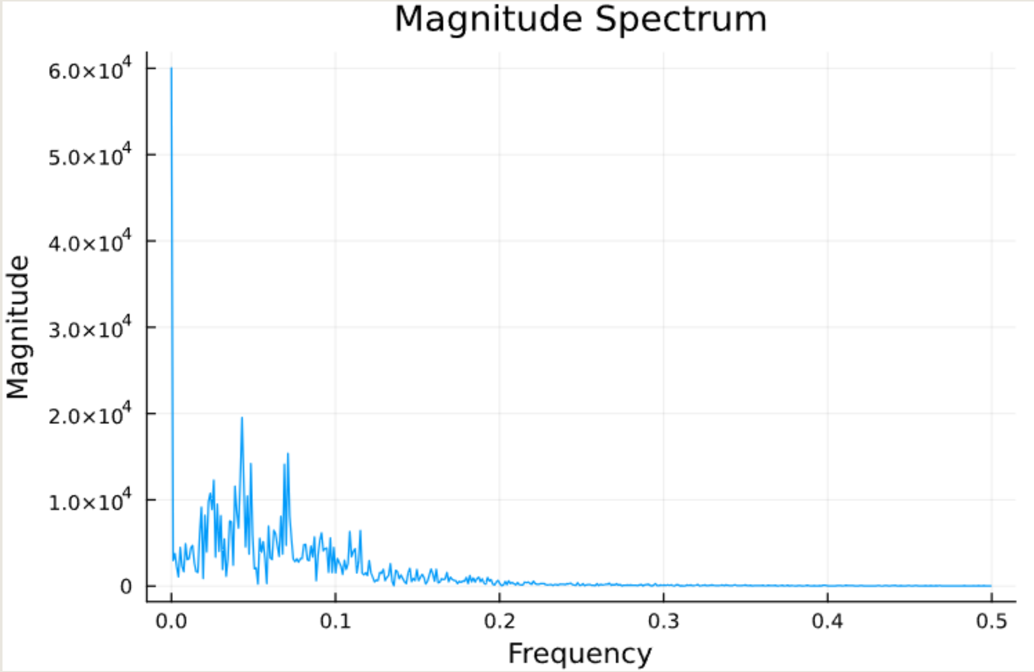


SYNTHETIC DATA – RESULTS (4)

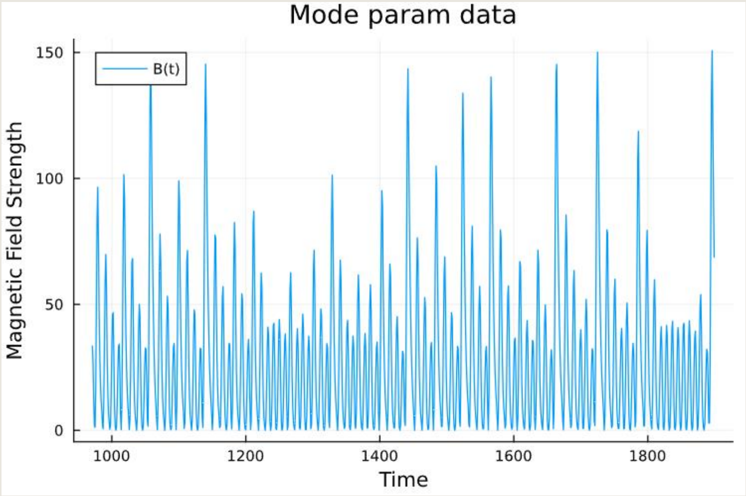
➢ **Mean** of the posteriors

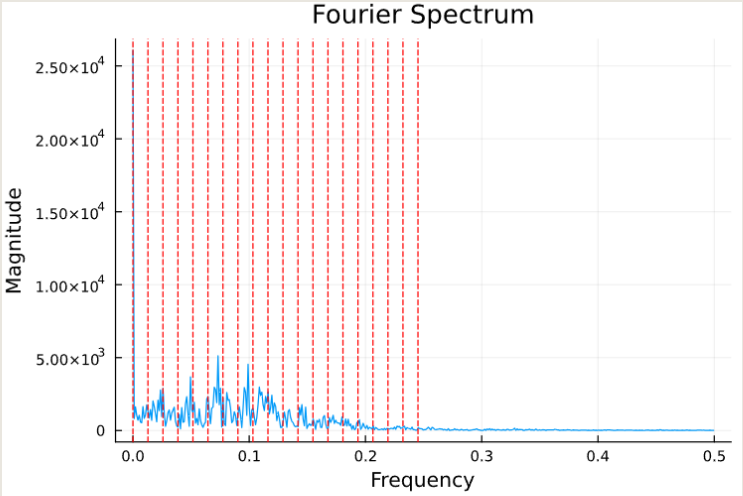
- N = 6.20303
- T = 7.38231 → a little smaller, but still high
- tau = 3.30273
- sigma = 0.10035 → too high, broad distribution
- B_max = 6.63542

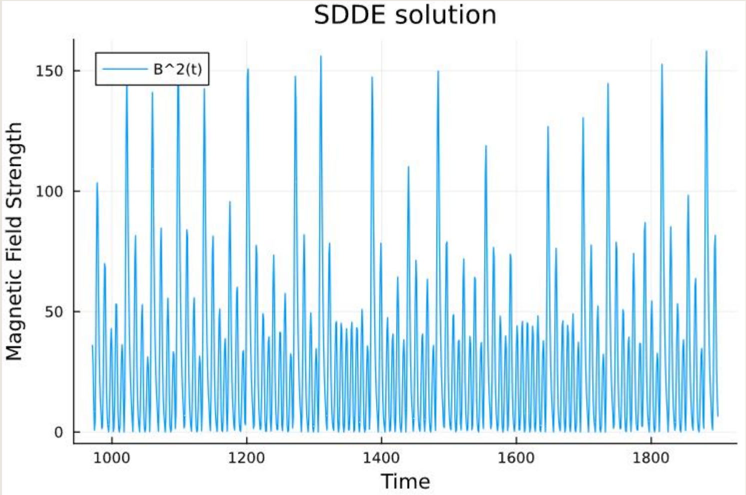


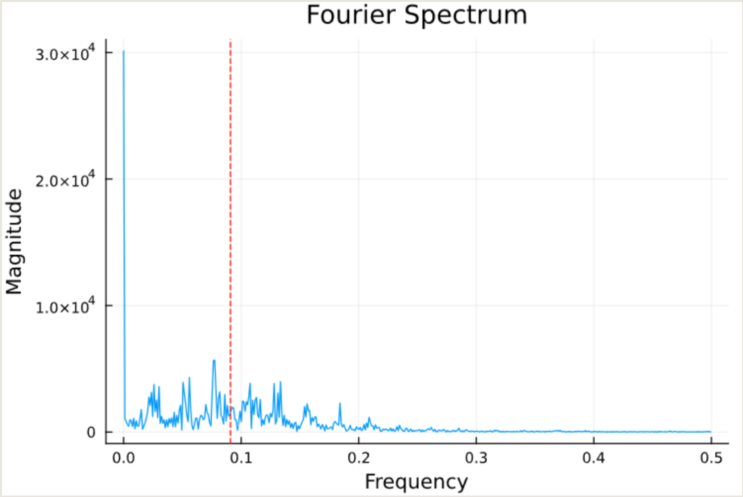


SYNTHETIC DATA – RESULTS (5)









- The main “problems” come from the parameter T → reduce the prior to the region 0.5 < T < 8;
- Considering the mode we get: N = 6.47699, T = 3.30046, tau = 3.93010, sigma = 0.04617, B_max = 5.78465 → **good approximation**.
- Agreement between the initial synthetic data and the final result
- Conclusion: the model is working on the synthetic data

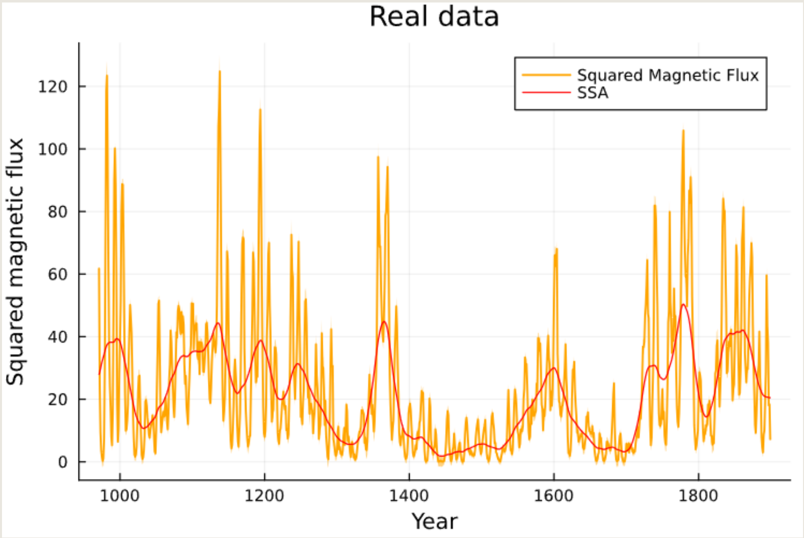
REAL DATA

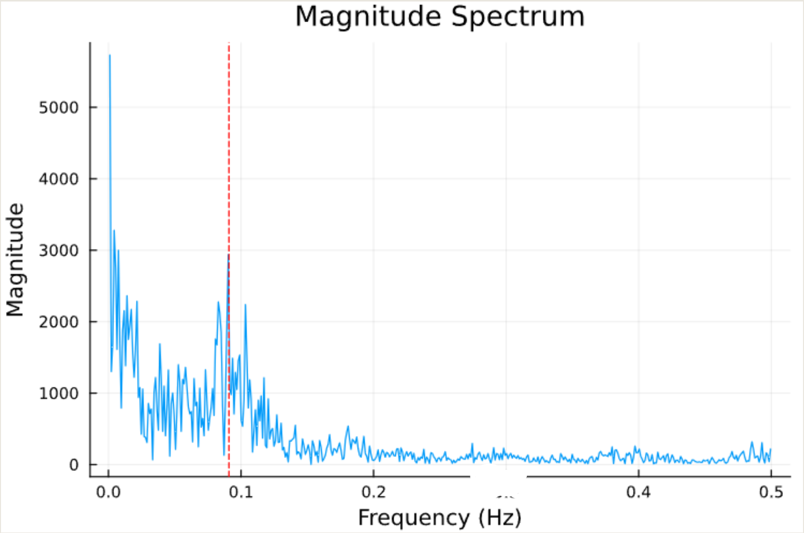
- **Data**: The real data represent the solar cyclic activity over the last millennium, reconstructed from annual ^{14}C data.

- **X-Axis**: Years from 971 to 1899 (929 years).

- **Graph 1**: Squared magnetic flux:
  - Different behaviour from synthetic data: the signal goes to zero much less often.
  - SSA: used to individuate trends (Grand Minima).

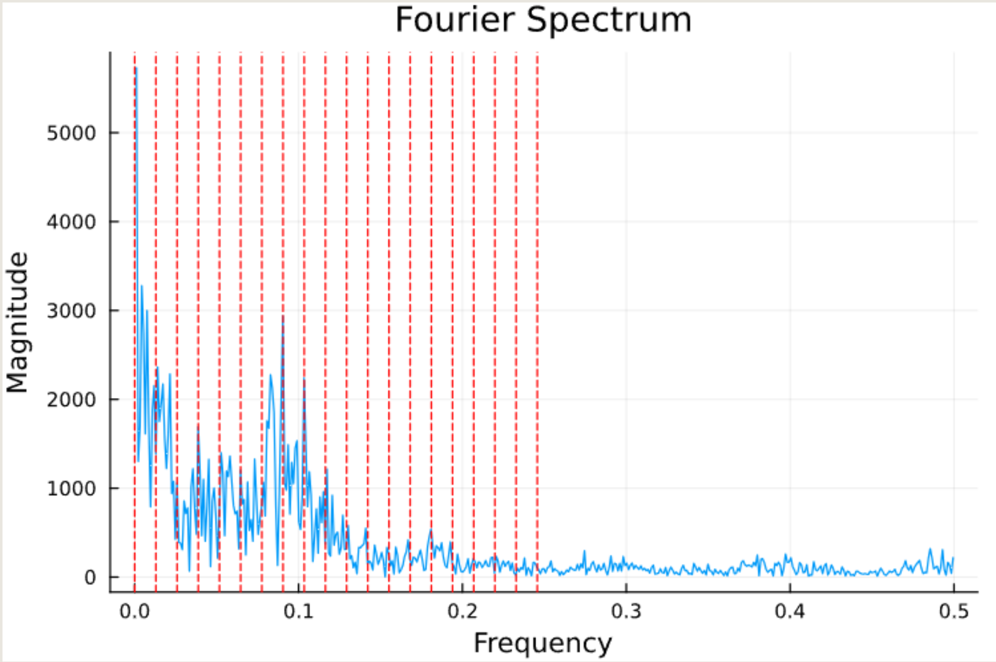
- **Graph 2**: Fourier Spectrum, much more disturbed that in the case of synthetic data
  - **11-Year Cycle**: Highlighted in red, we have an exact peak at this frequency, but also some peaks near to it.

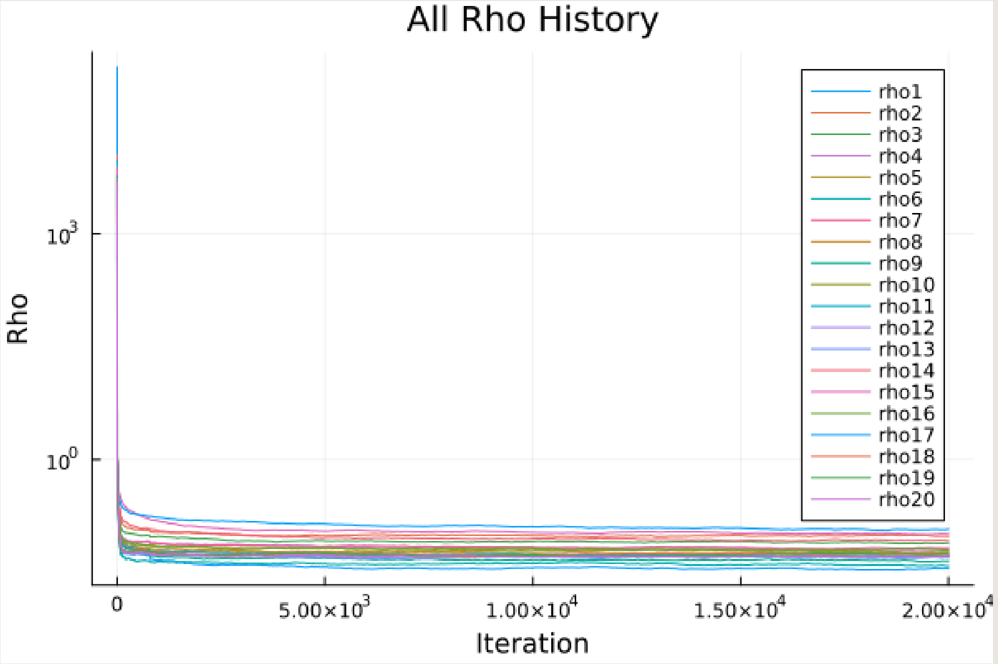




REAL DATA – SIMULATION 1 (1)

```
Prior = "product_distribution(Uniform(0.1, 15.0), Uniform(0.1, 10.0), Uniform(0.1, 6.0),
                              Uniform(0.01, 0.3), Uniform(1.0, 15.0))"

n_particles = 1000; n_simulation = 20000000; type = "single"; indices = 1:6:120
```



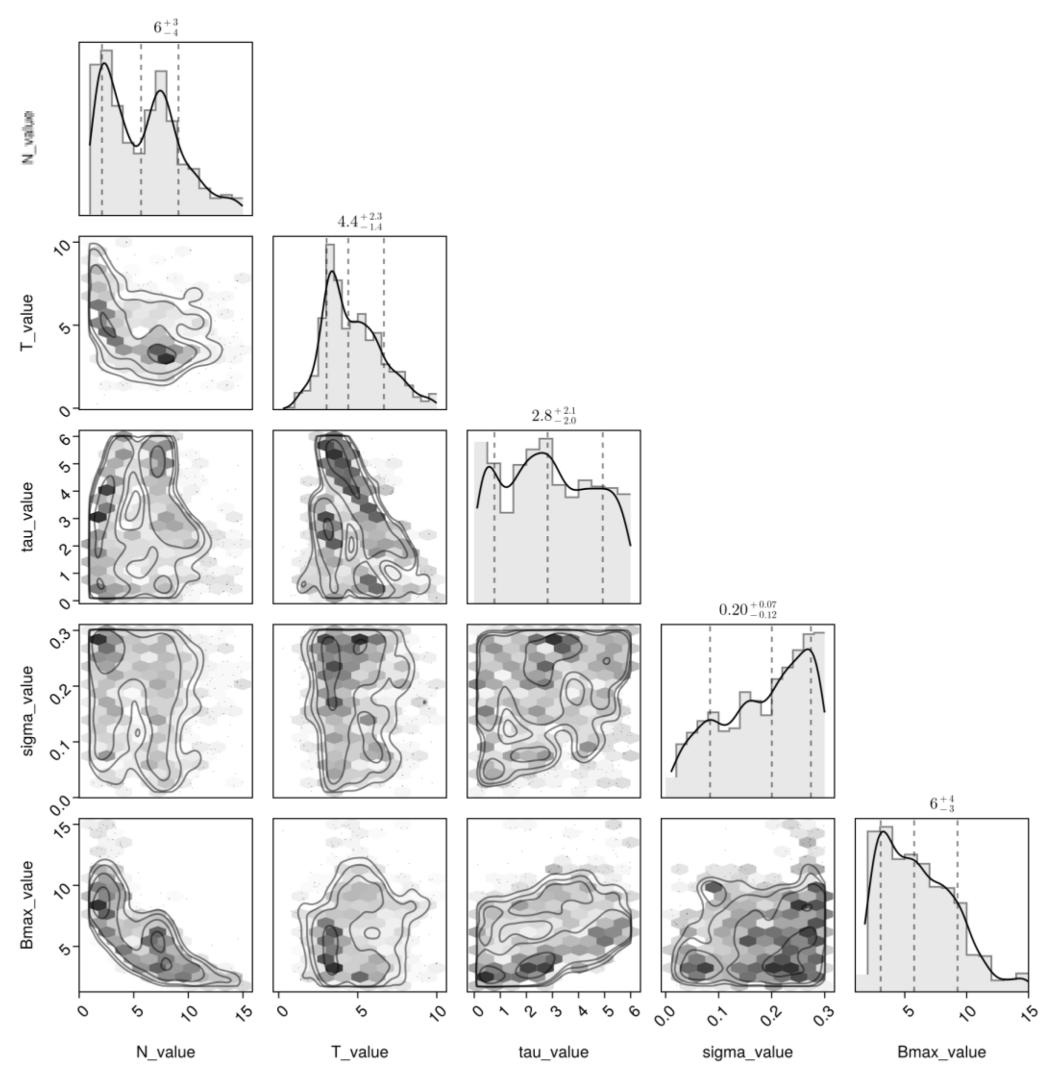


REAL DATA – SIMULATION 1 (2)

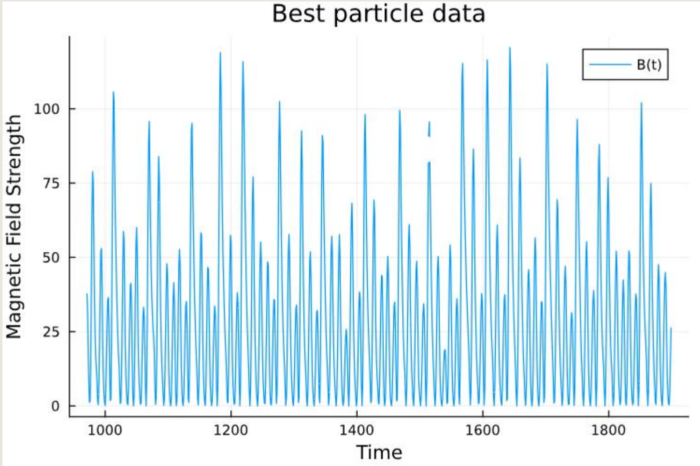
The posteriors in this case are a bit different:

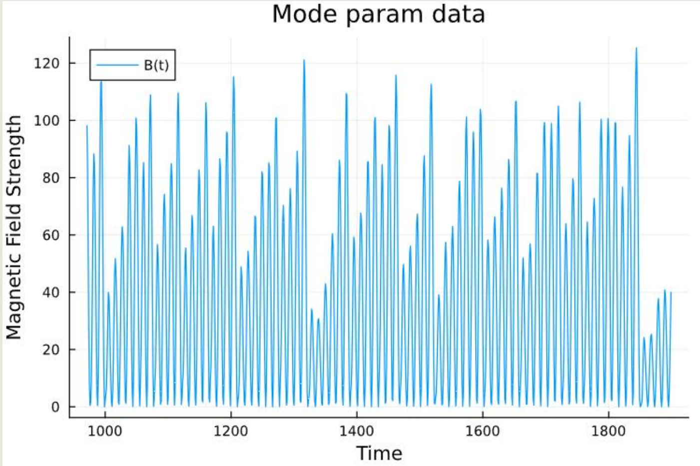
- N: presents two peaks, one at a small value (2) and one at an higher value (7).
- T: doesn’t present the multinomial distribution, which is a bit strange as we expect it from the synthetic analysis.
- Tau: presents a multinomial distribution.
- Sigma: it’s shifted towards high value for the noise, expected for real data.
- B_max: broad distribution, the peak is shifted towards smaller value (low amplitude)

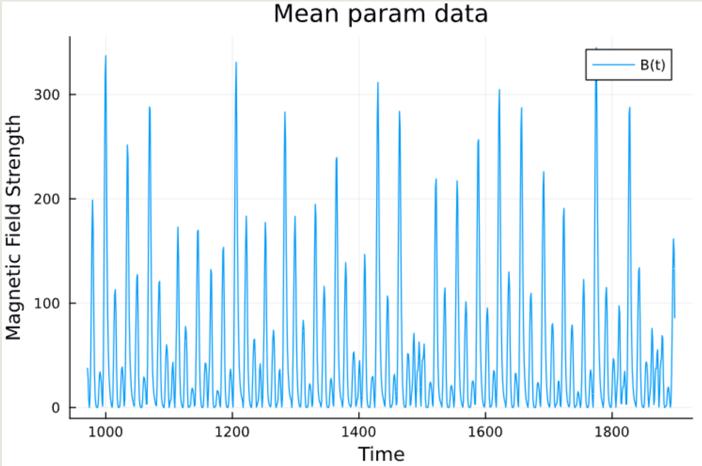
We can conclude that working with real data everything becomes a lot messier and the noise plays a much more important role.

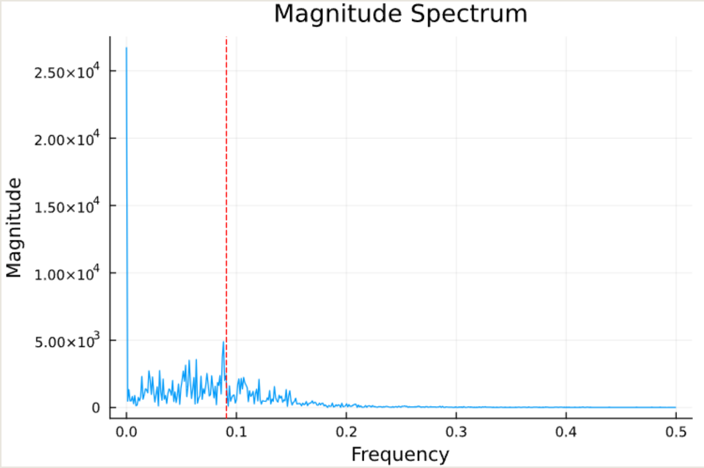


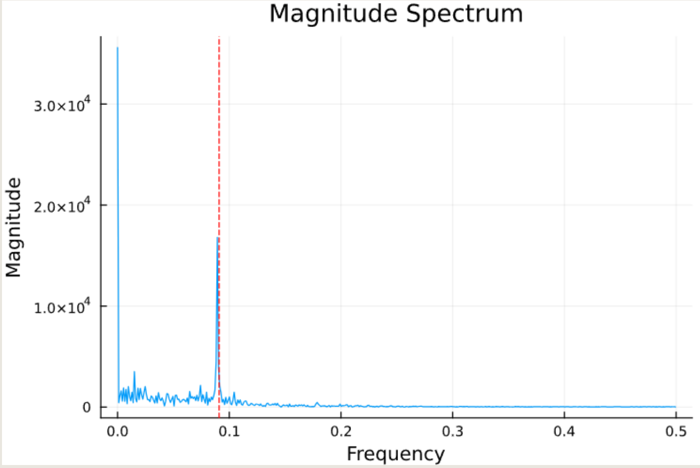
REAL DATA – SIMULATION 1 (3)

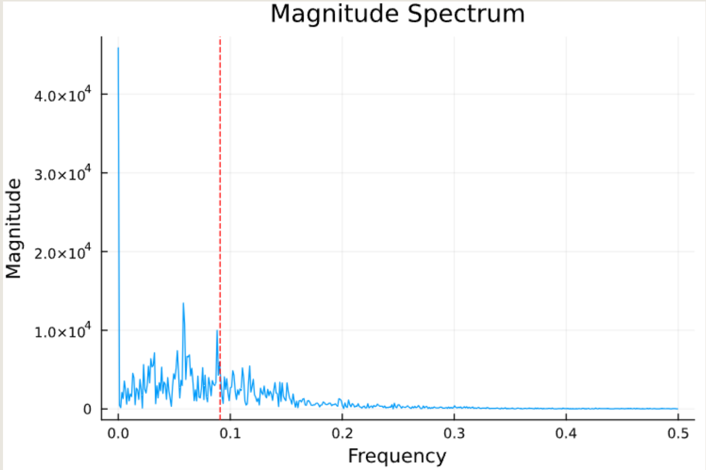












REAL DATA – SIMULATION 1 (4)

Parameter	Best particle	Mode data	Mean data
N	6.16239	1.82915	5.79470
T	3.43561	6.28477	4.72243
Tau	5.01778	2.90133	2.86546
Sigma	0.23343	0.07705	0.18407
Bmax	6.14206	9.90661	6.14657

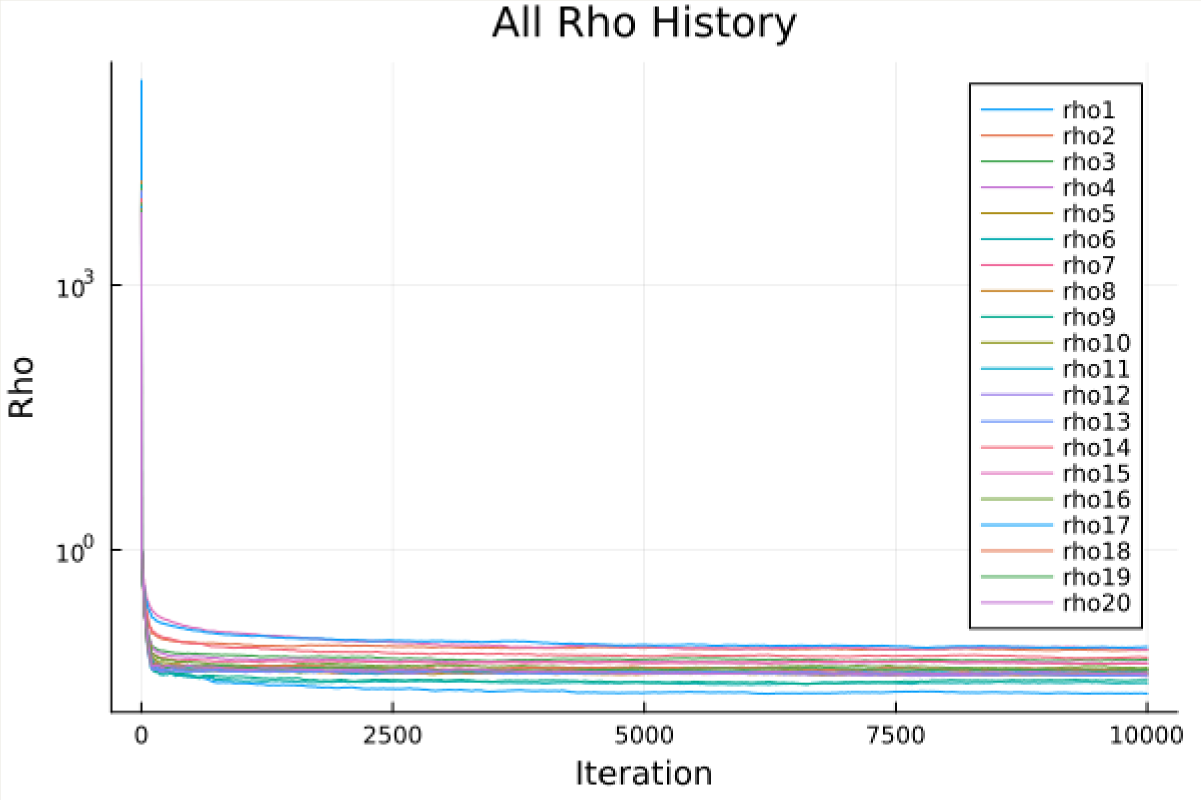
From this first simulation we have seen that the spectra is difficult to reproduce: this time we have a lot of noise and it’s difficult for our model to pick up all the peaks.

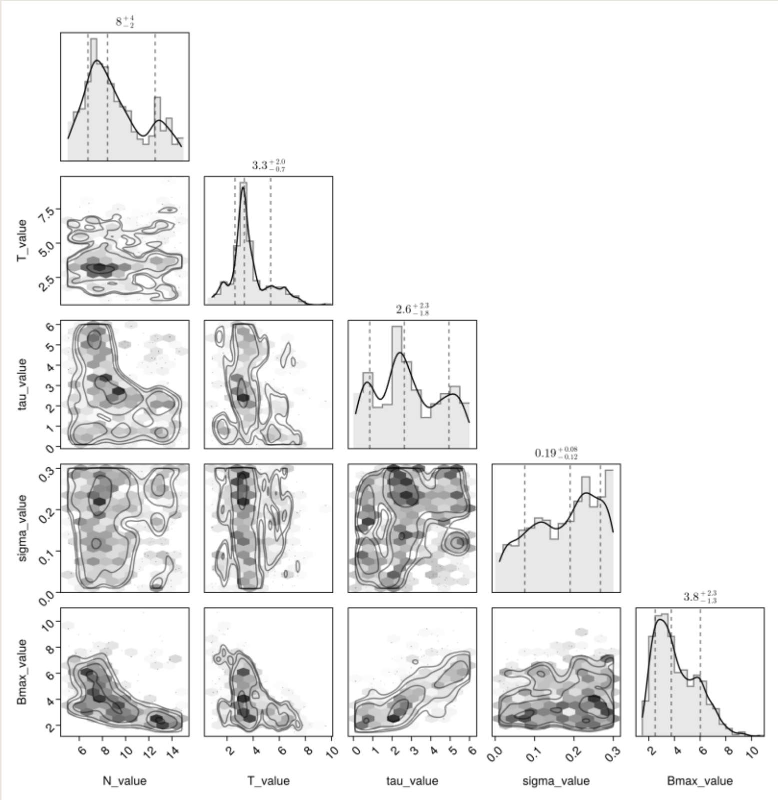
We also have posteriors different from what we expected. In particular, we have a double peak in the posterior of N, which is shifted towards value that don’t produce a good spectra. Thus, we can change the prior and reduce the interval for N.

Moreover, the zero mode seems to be too dominant with respect to the other peaks, so in the following simulations we will neglect it.

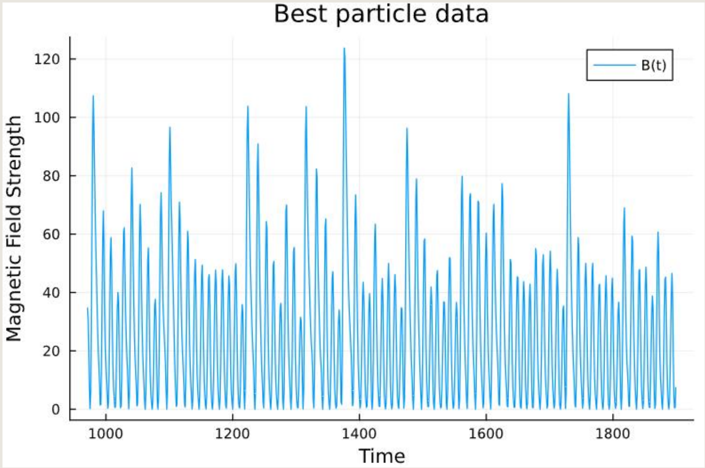
REAL DATA – SIMULATION 2 (1)

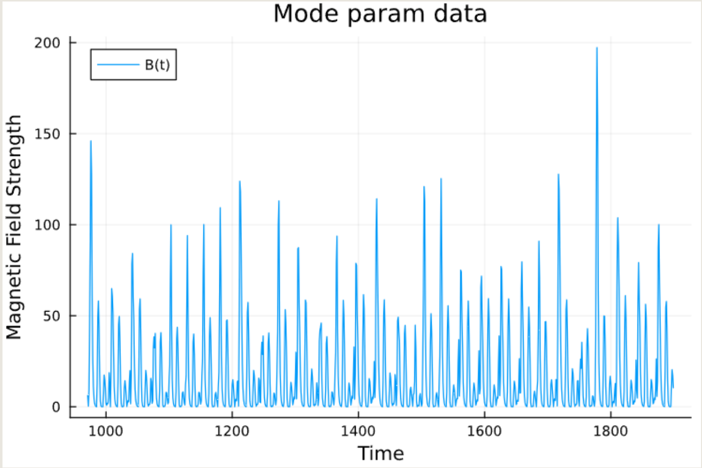
```
Prior = "product_distribution(Uniform(5.0, 15.0), Uniform(0.1, 10.0), Uniform(0.1, 6.0),
                              Uniform(0.01, 0.3), Uniform(1.0, 15.0))"

n_particles = 1000; n_simulation = 10000000; type = "single"; indices = 1:6:120
```

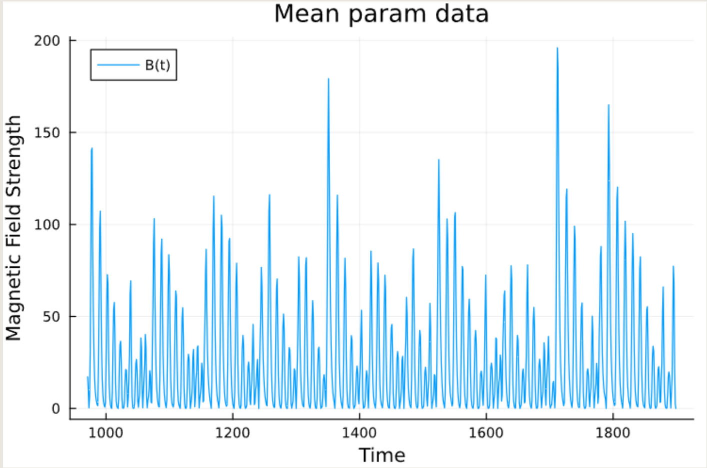


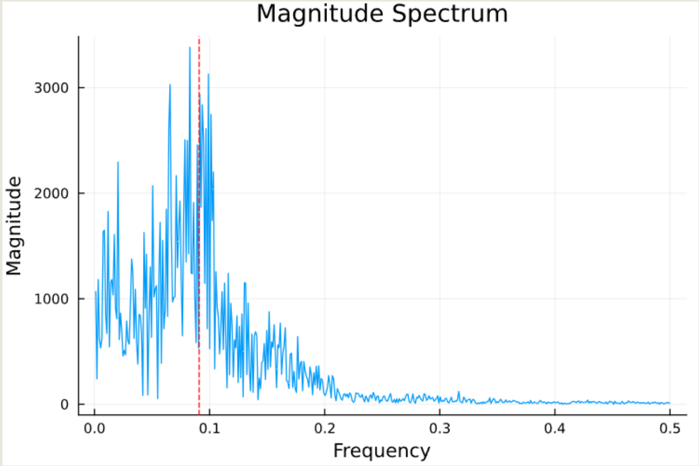


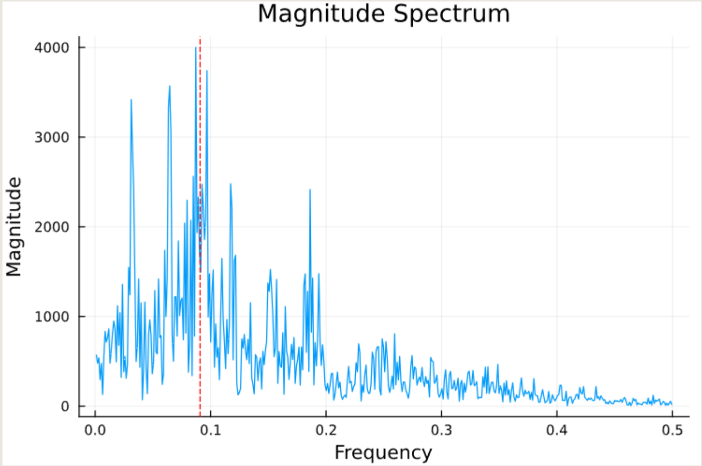
REAL DATA – SIMULATION 2 (2)

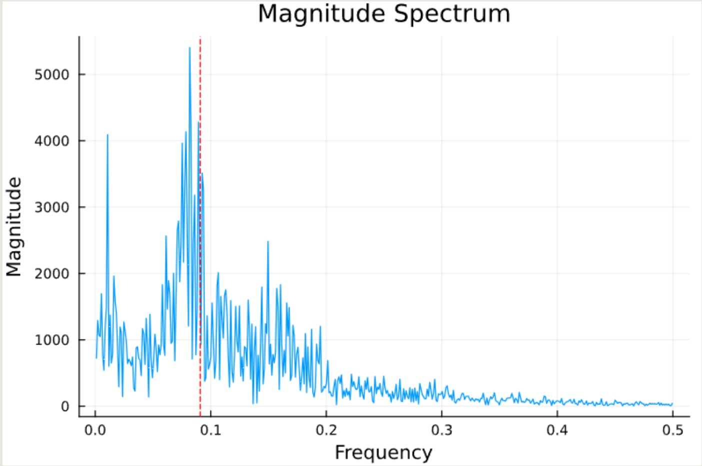












REAL DATA – SIMULATION 2 (3)

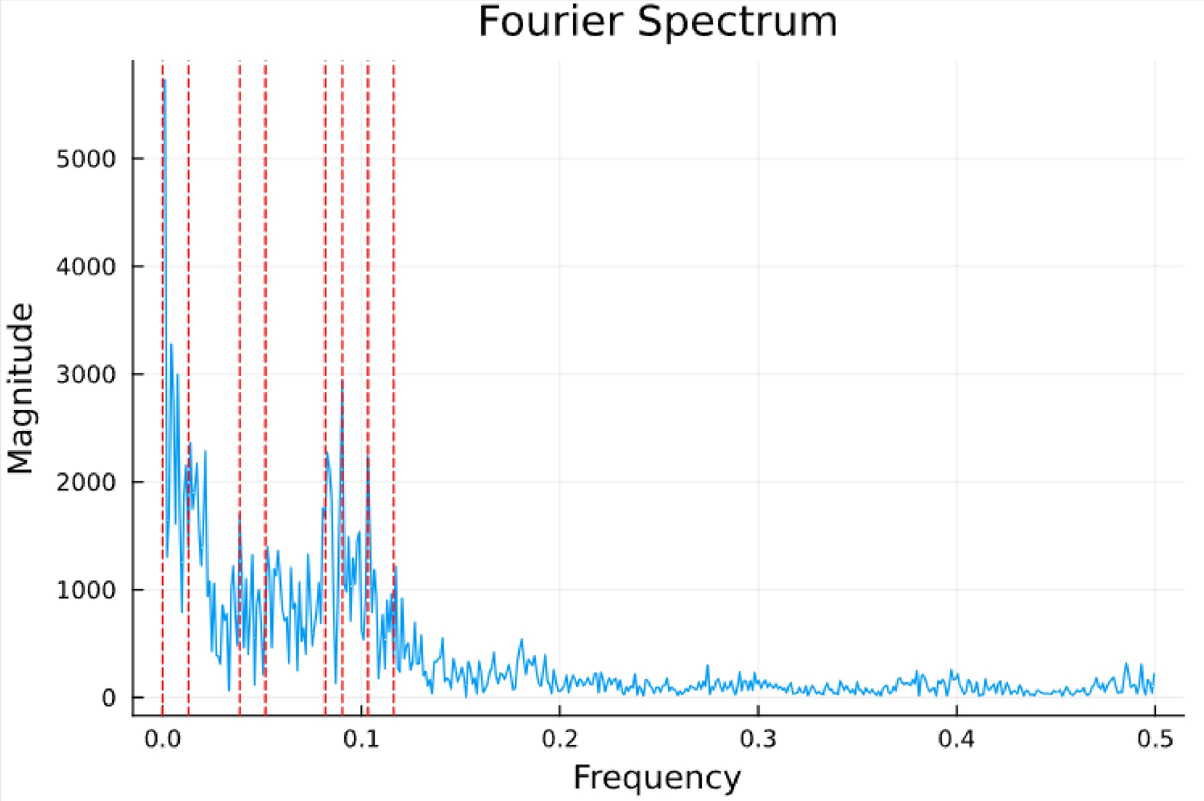
Parameter	Best particle	Mode data	Mean data
N	7.39336	12.54836	9.07687
T	3.21182	4.03001	3.69191
Tau	5.26223	1.79359	2.85687
Sigma	0.29928	0.01493	0.17463
Bmax	5.88976	2.49569	4.15237

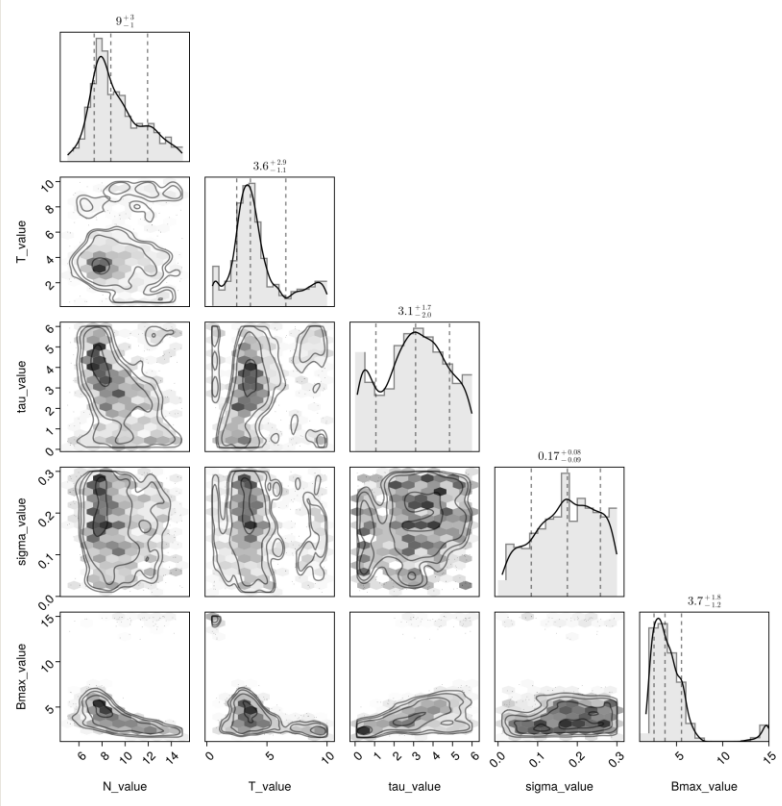
For this simulation, the spectra are better, also in amplitude. The best one is the one for the best particle, while we see that for the mode we get some strange values of the parameters, which are not expected (in particular for Bmax).

In order to get a better result, we can select instead of one each six component, only the components of the most important peaks; this should improve the performance of the sABC algorithm.

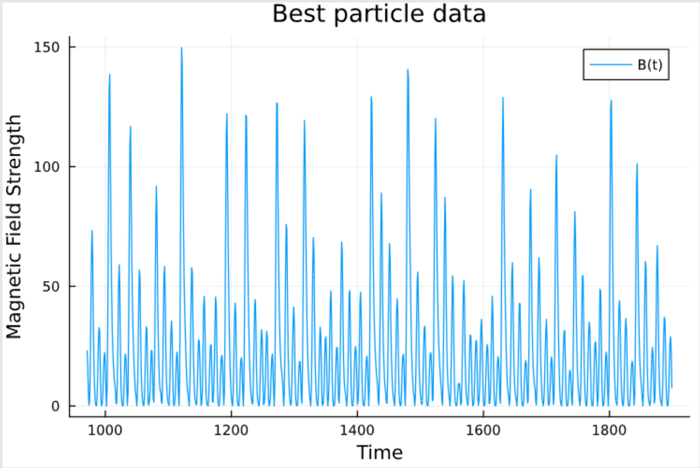
REAL DATA – SIMULATION 3 (1)

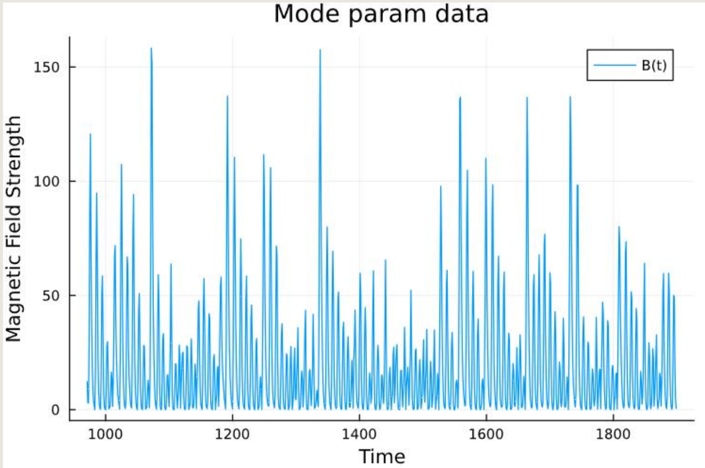
```
Prior = "product_distribution(Uniform(5.0, 15.0), Uniform(0.1, 10.0), Uniform(0.1, 6.0),
                              Uniform(0.01, 0.3), Uniform(1.0, 15.0))"

n_particles = 1000; n_simulation = 10000000; type = "single";
indices = [1, 7, 19, 25, 39, 43, 49, 55]
```

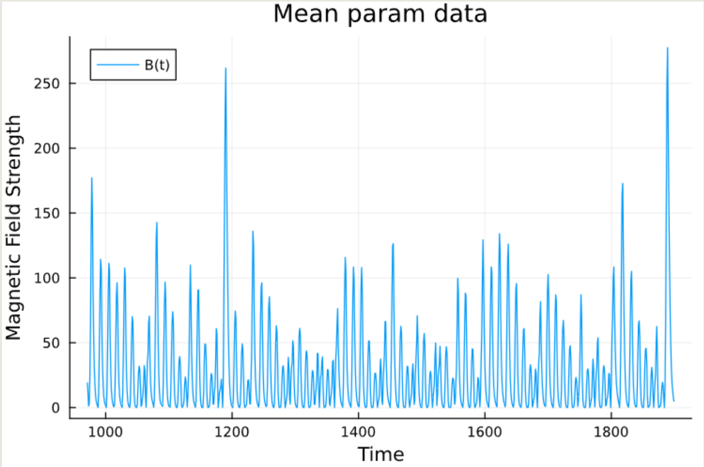


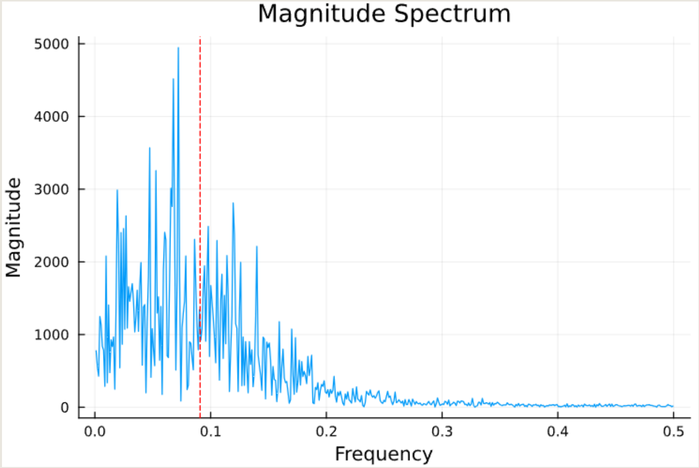


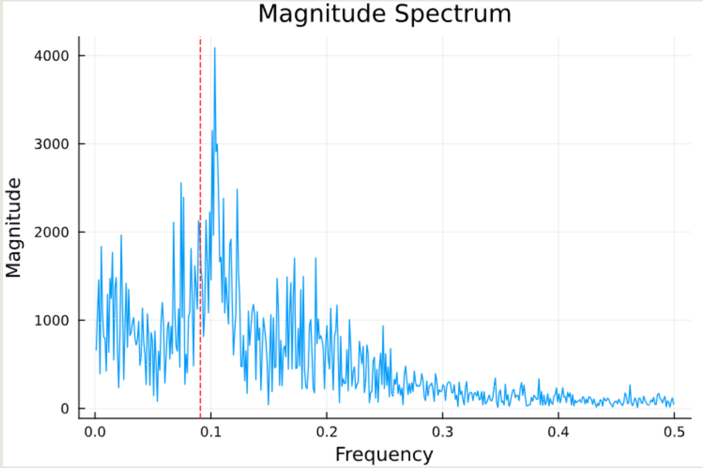
REAL DATA – SIMULATION 3 (2)

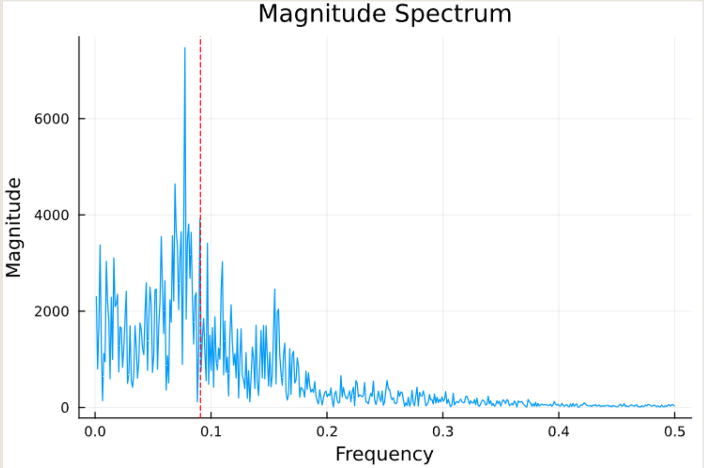












REAL DATA – SIMULATION 3 (3)

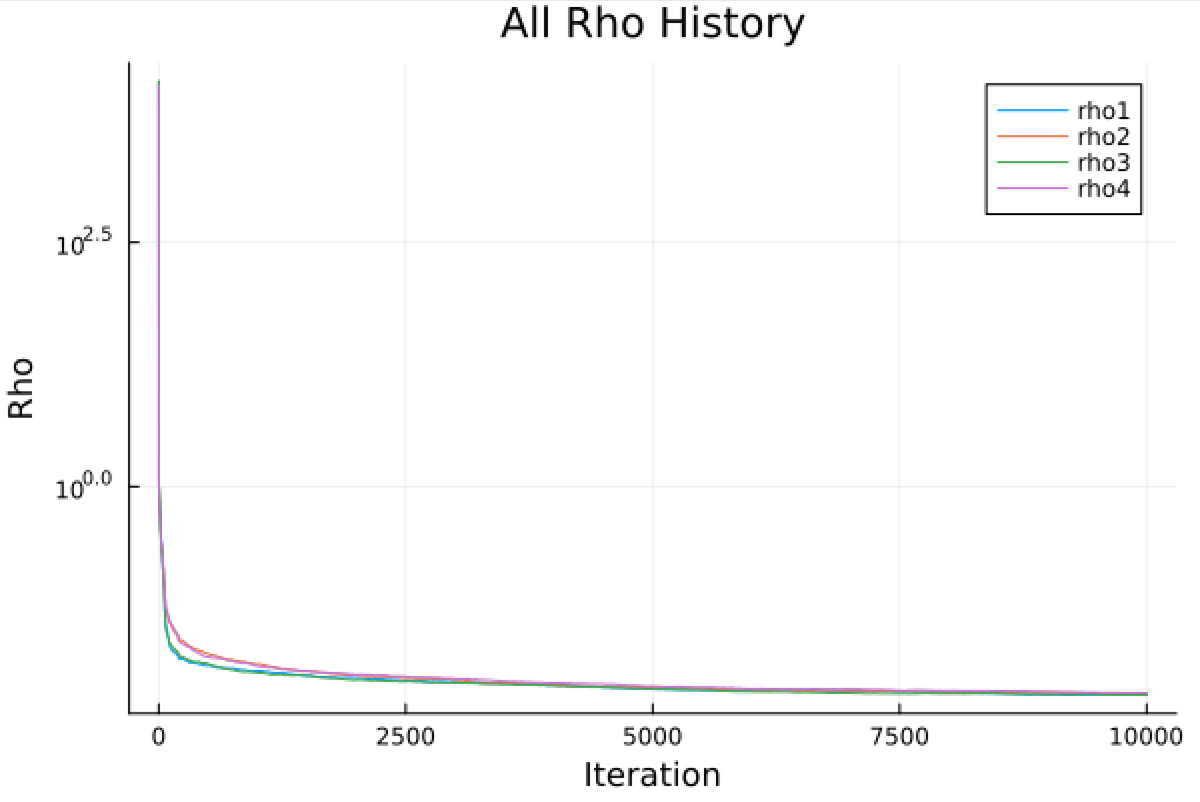
Parameter	Best particle	Mode data	Mean data
N	6.16513	10.25693	9.32215
T	3.67889	2.87033	4.18320
Tau	0.28917	2.16513	0.17128
Sigma	0.23343	0.27441	0.18407
Bmax	4.79926	3.50802	4.36229

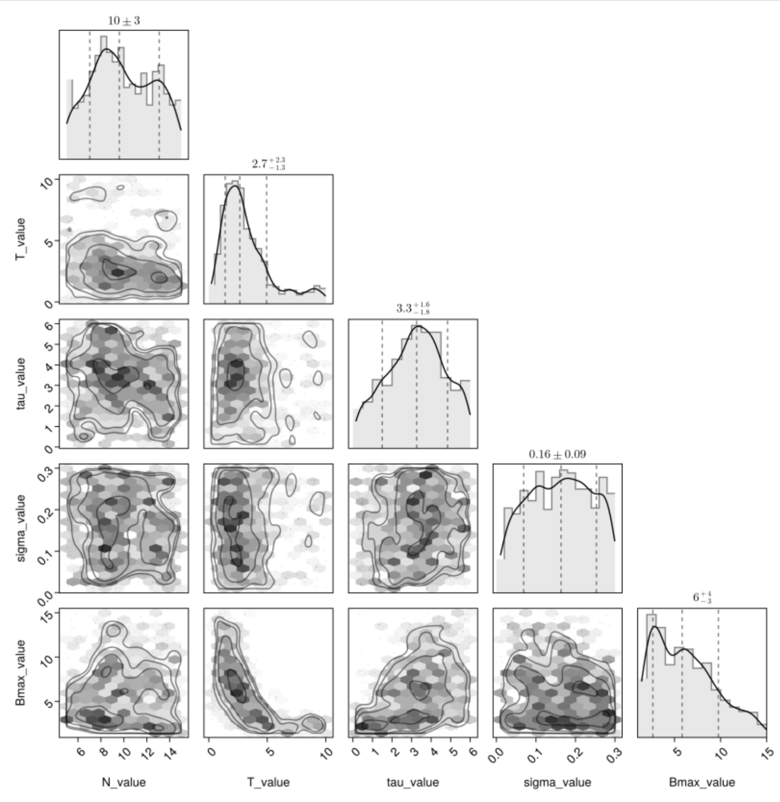
This time we have more peaked distributions; nevertheless, we see that we have tau with two peaks and one of them at very small values, which is strange. Moreover, also Bmax is more shifted to small values.

All the spectra present a peak, which is never centred in the correct position (of the 11 year cycle), which is not good; we should try with less summary statistics, considering only the peaks near the 11 year cycle and forgetting about the others.

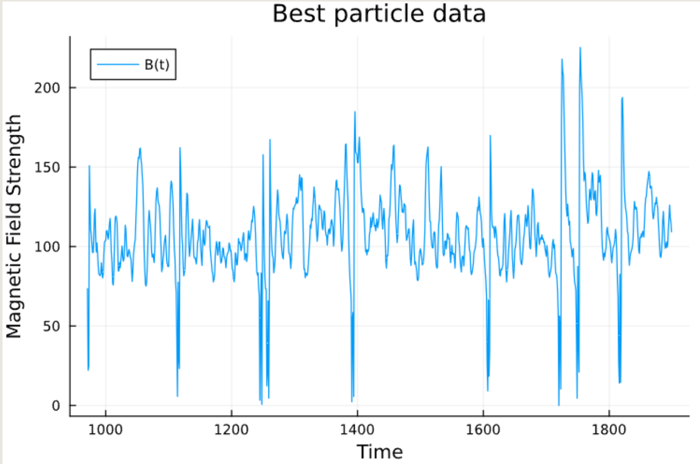
REAL DATA – SIMULATION 4 (1)

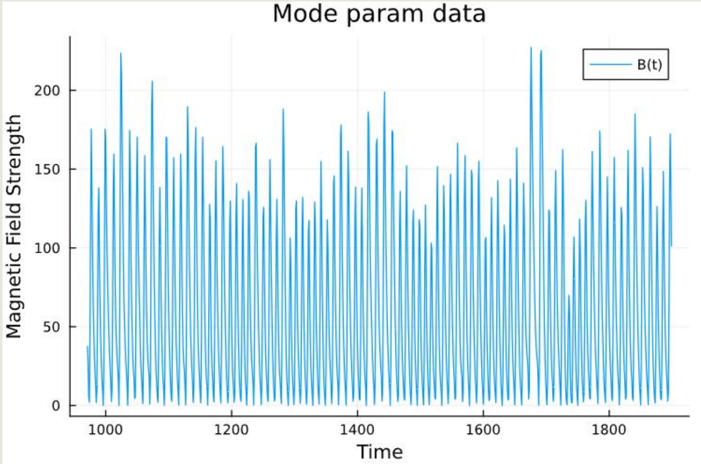
```
Prior = "product_distribution(Uniform(5.0, 15.0), Uniform(0.1, 10.0), Uniform(0.1, 6.0),
                                Uniform(0.01, 0.3), Uniform(1.0, 15.0))"

n_particles = 1000; n_simulation = 10000000; type = "single"; indices = [39, 43, 49, 55]
```

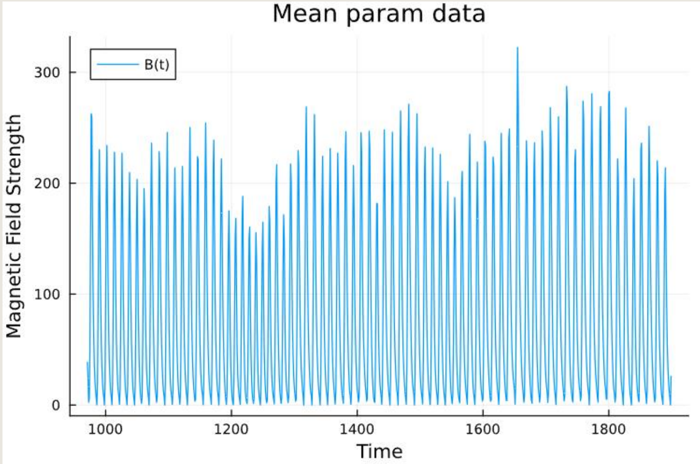


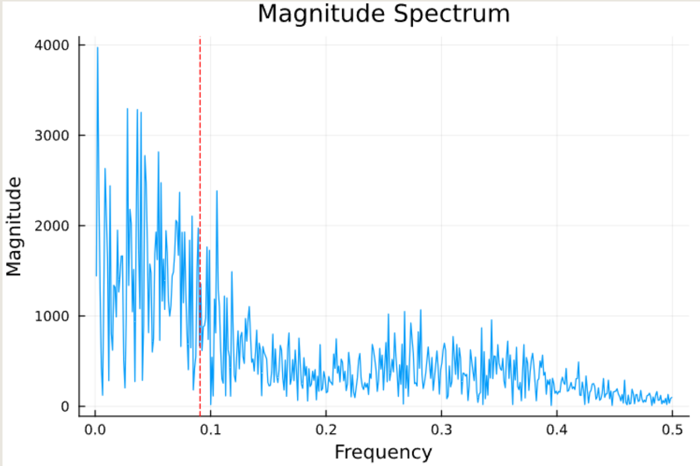


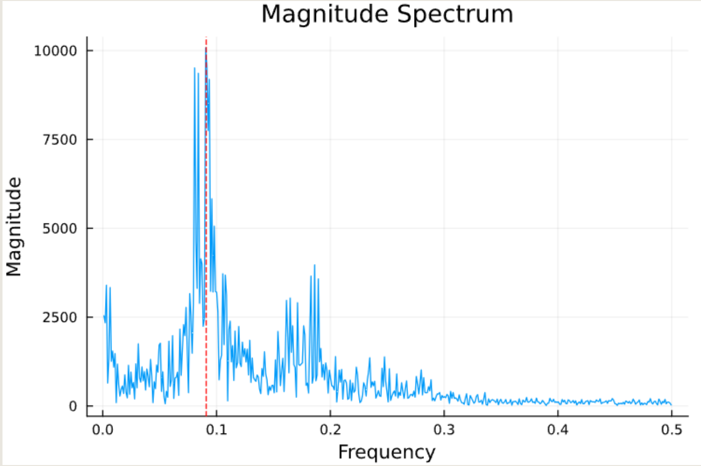
REAL DATA – SIMULATION 4 (2)

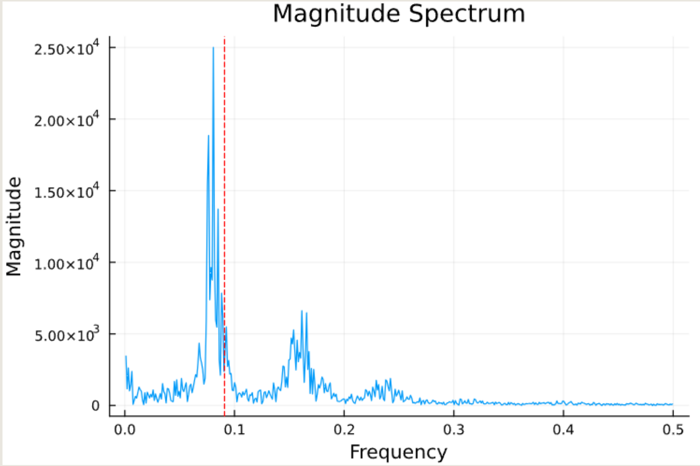












REAL DATA – SIMULATION 4 (3)

Parameter	Best particle	Mode data	Mean data
N	11.99071	9.82563	9.87008
T	0.77433	2.45856	3.25258
Tau	2.39681	3.48089	3.21518
Sigma	0.21858	0.28269	0.16073
Bmax	8.58340	6.12289	6.21297

The last simulation don’t give us a better result: instead, we get a strange low signal for the best particle, with some peaks occurring casually (the spectra is very noisy), due to the small value of T obtained.

For both mode and mean we get instead reasonable values for the parameters and a nice peak around the frequency of the 11 year cycle: nevertheless, the other peaks are not captured by this analysis, only the principal, thus it doesn’t improve the performance of the model.

CONCLUSIONS (1)

➢ **SYNTETHIC DATA:**

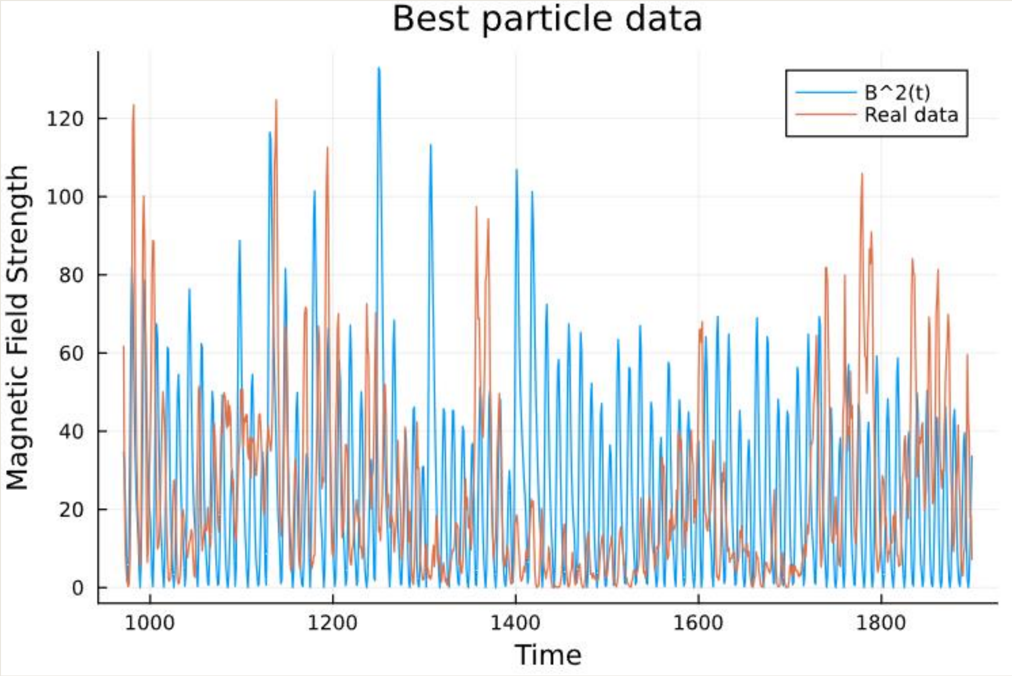
- The model is proven to be working: we get good looking posteriors.
- We are able to obtain the expectd values of the parameters (up to a certain level).

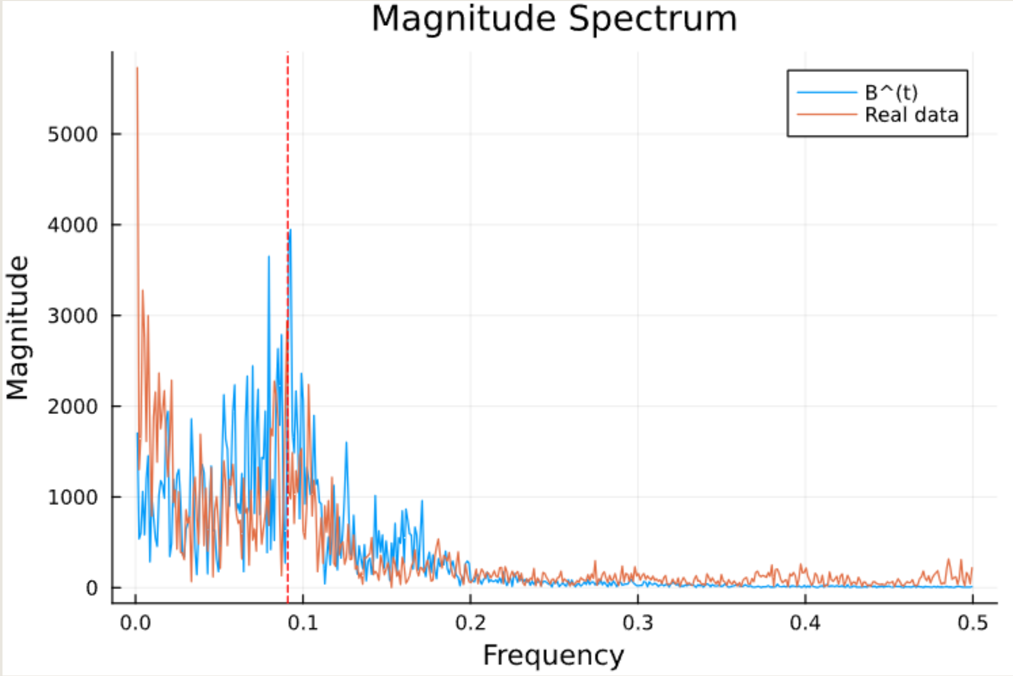
➢ **REAL DATA:**

- Different choices of summary statistics lead to really different results.
- Not always we obtain good looking posteriors: most of the time they follow a strange distribution and we don’t know which method is better to find the correct parameters.
- The zero mode is dominant in the synthetic data, so it’s better not to include in the spectra, otherwise no trend can be seen.
- The best result (the synthetic data that most resamble real data) was obtained with 1:6:120 as indeces and reducing the N window interval.

CONCLUSIONS (2)

Parameter	N	T	Tau	Sigma	Bmax
Best particle	7.39336	3.21182	5.26223	0.29928	5.88976





FUTURE APPROACHES

- **Change the parameters of the simulations:**
  - Using type = «multi» → enables different epsilon for different summary statistics.
  - Higher number of iterations or particles → more statistically significant posteriors.
  - Different choice of prior (different interval or different shape).

- **Change the summary statistics:**
  - Usage of more specific summary statistics → other Fourier components or something else.
  - Good reducing but not too much; we need both minima and maxima (not only peaks).

- **Change the parameters of the model:**
  - Fixing Bmax, so the amplitude of the signal is fixed.
  - Introduction of a shifting in order to reduce the zero mode dominance.

THANKS FOR THE ATTENTION

Physics of Data – Laboratory of Computational Physics

Calandra Buonaura Lorenzo

Rossi Lucrezia

Turci Andrea

July 22, 2024