Zurich University of Applied Sciences
Institute of Applied Simulation

Bayesian inference methods for the calibration of stochastic dynamo models

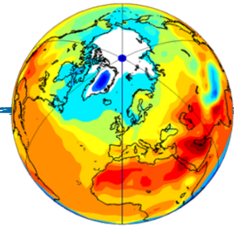
Simone Ulzega, Carlo Albert

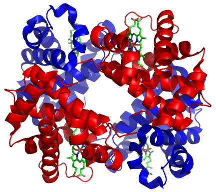
4th Dynamo Thinkshop, Rome, November 2019

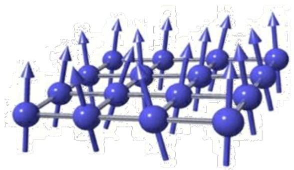
eawag aquatic research

SDSC

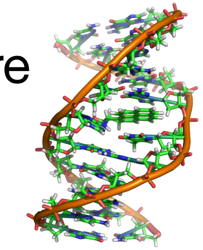
Parameter inference







Simulating and understanding **complex system dynamics** require building **conceptual models**



- Parameterised models need to be **calibrated to measured data**
- **Data-driven model calibration / parameter inference**

  Estimation of system parameters, **with their uncertainties**, given measured data

Parameter inference for non-linear stochastic models can become mathematically and computationally very challenging

The Bayesian framework

- The Bayesian framework formalises **learning** as an **update process** of our knowledge in the light **of new data**

- **Knowledge (belief)** is quantified in the form of **probability distributions**

- **Learning** means **conditioning** these distributions to **observed data** (belief depends on the available information)

- We always need **prior knowledge** in the form of a **probability distribution** for our unknowns (data and parameters)

Thomas Bayes
(1701 - 1761)

The Bayesian framework

The Bayesian prior

$\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ Observables (measured, e.g., monthly sunspots number)

$\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_m\}$ Parameters (to be inferred, e.g., dynamo number, diffusion timescale, time delay, noise amplitude)

$$f(\mathbf{y}, \boldsymbol{\theta}) = f(\mathbf{y} \mid \boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$$

Joint probability distribution for observables and parameters

Probability distribution for observables given parameters (our probabilistic model)

Prior knowledge about parameters

The Bayesian prior

Example: observables $\mathbf{y}$ are expected to be normally distributed around a mean value μ, with a spread defined by a variance σ^2

Observables: $\mathbf{y} = \{y_i\}_{i=1,\ldots,n}$ Parameters to be inferred: $\boldsymbol{\theta} = \{\mu, \sigma\}$

$$f(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)$$

$$f(\boldsymbol{\theta}) = \chi(\mu_{\min} < \mu < \mu_{\max})\,\chi(\sigma_{\min} < \sigma < \sigma_{\max})$$

$$f(\mathbf{y}, \boldsymbol{\theta}) = f(\mathbf{y} \mid \boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})$$

Joint probability distribution for **observables** and **parameters**

The Bayesian posterior

We measure $\mathbf{y}^{(\mathrm{obs})}$, which is believed to be a realisation of our model, $f\left(\mathbf{y} \mid \boldsymbol{\theta}^*\right)$, for a “true” set of parameters $\boldsymbol{\theta}^*$

Bayes Equation

$$f\left(\boldsymbol{\theta} \mid \mathbf{y}^{(\mathrm{obs})}\right) \propto f\left(\mathbf{y}^{(\mathrm{obs})}, \boldsymbol{\theta}\right) = f\left(\mathbf{y}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right) f\left(\boldsymbol{\theta}\right)$$

Posterior distribution:
probability of model parameters given measured data

Likelihood function:
probability that model produces data $\mathbf{y}^{(\mathrm{obs})}$ for given parameters $\boldsymbol{\theta}$

The Bayesian framework

The Bayesian posterior

We measure $\mathbf{y}^{(\mathrm{obs})}$, which is believed to be a realisation of our model, $f\left(\mathbf{y} \mid \boldsymbol{\theta}^*\right)$, for a "true" set of parameters $\boldsymbol{\theta}^*$

$$f\left(\boldsymbol{\theta} \mid \mathbf{y}^{(\mathrm{obs})}\right) \propto f\left(\mathbf{y}^{(\mathrm{obs})}, \boldsymbol{\theta}\right) = f\left(\mathbf{y}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right) f\left(\boldsymbol{\theta}\right)$$

Bayesian inference:
Drawing a sufficiently large parameter sample from the posterior $f\left(\boldsymbol{\theta} \mid \mathbf{y}^{(\mathrm{obs})}\right)$

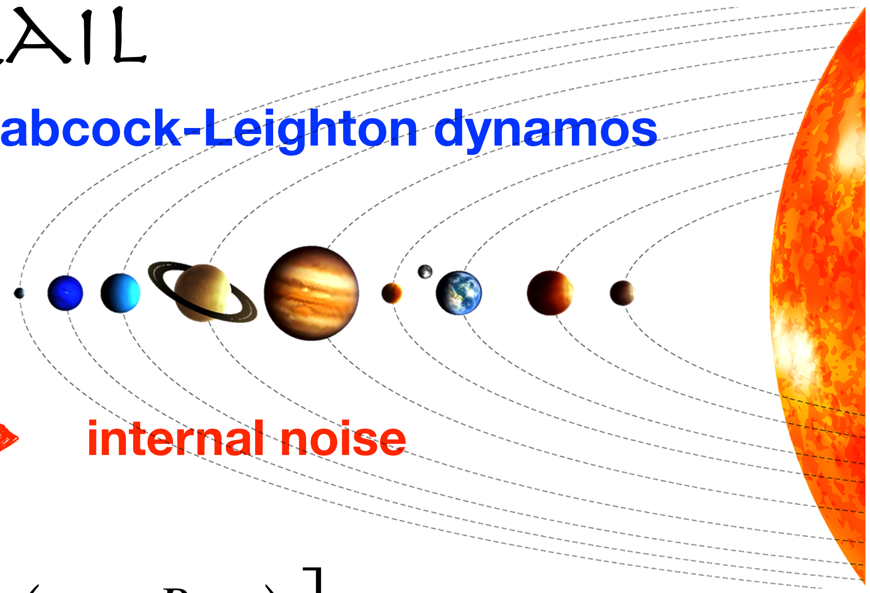
probabilistic predictions

learn something about the system and the mechanisms that lead to observed features (with physically interpretable parameters)

SR in B-L Dynamos?

The Quest for the Holy Grail

Numerical evidence for **stochastic resonances** in **Babcock-Leighton dynamos**



Model: stochastic iterative map

(*Charbonneau et al., Astrophys. J 658, 2007*)

$$p_{n+1} = \alpha f_n\left(p_n\right) p_n + \epsilon_n \qquad \epsilon_n \in \mathcal{U}\left[0,\epsilon\right] \longrightarrow$$ **internal noise**

$$f_n\left(p_n\right) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{p_n - B_{\min}(1 + aT_n)}{W_1}\right)\right]\left[1 - \mathrm{erf}\left(\frac{p_n - B_{\max}}{W_1}\right)\right]$$

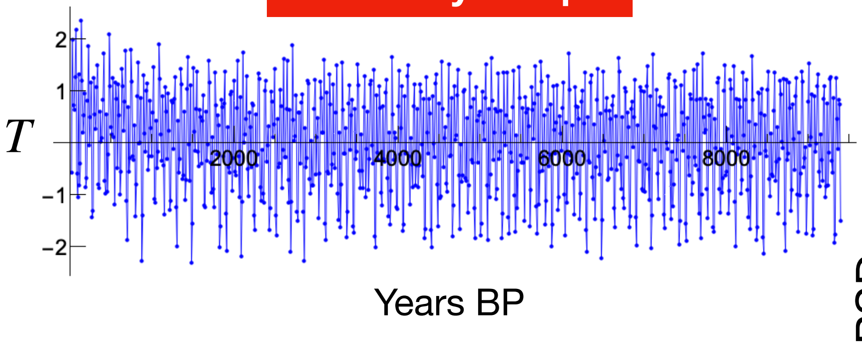
$\mathbf{T} = \left\{T_n\right\}$ Planetary tidal torque, with (small) amplitude a $\longrightarrow$ **external periodic forcing**

$\boldsymbol{\theta} = \{\alpha, a, \epsilon\}$ Parameters to be inferred

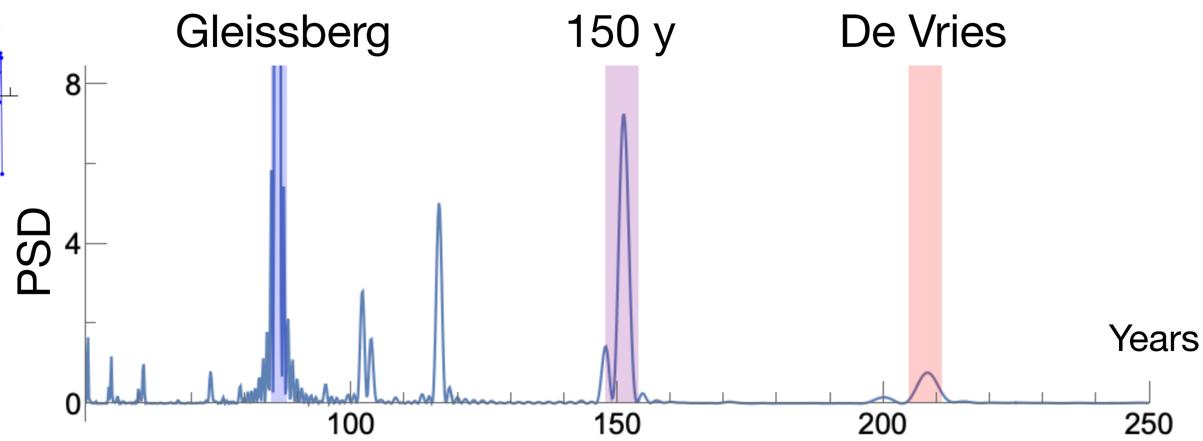
Last but not least, we need **data**

SR in B-L Dynamos?

Planetary torque

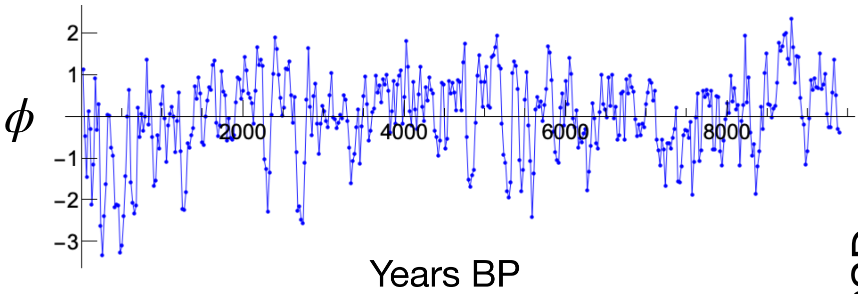


→ Power Spectral Density (PSD)

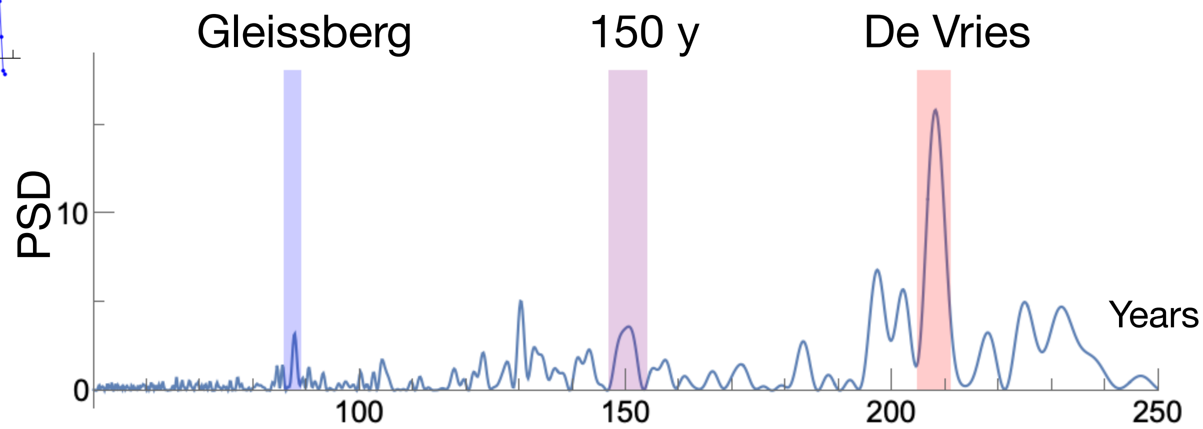


Solar modulation potential

(from ^{10}Be and ^{14}C records)



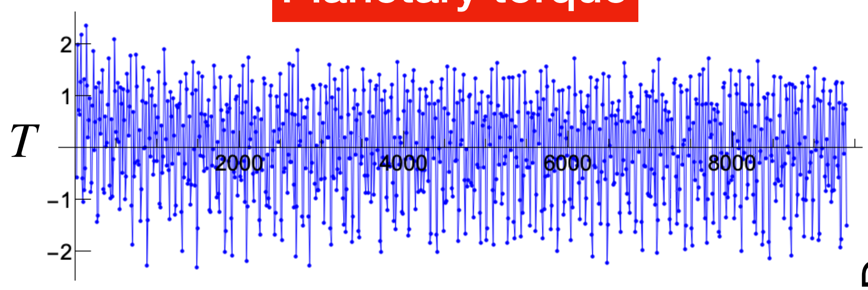
→ Power Spectral Density (PSD)



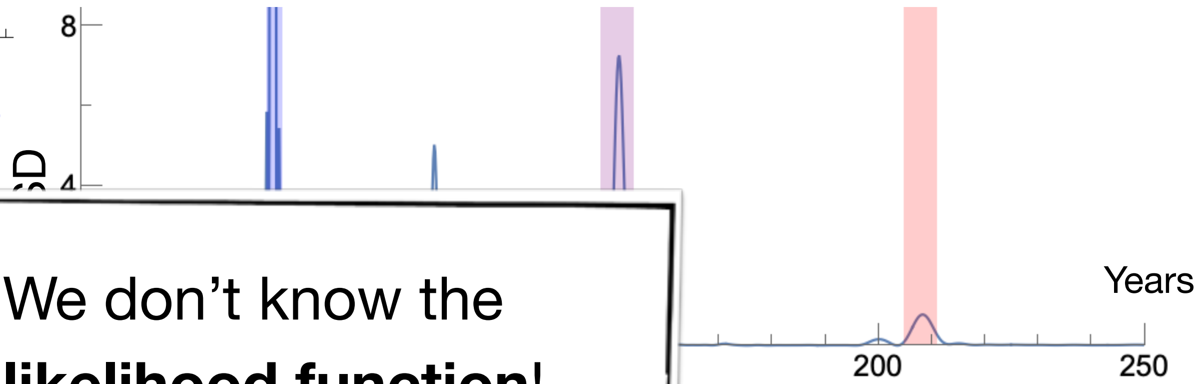
Abreu et al., A&A 548, 2012

SR in B-L Dynamos?

Planetary torque



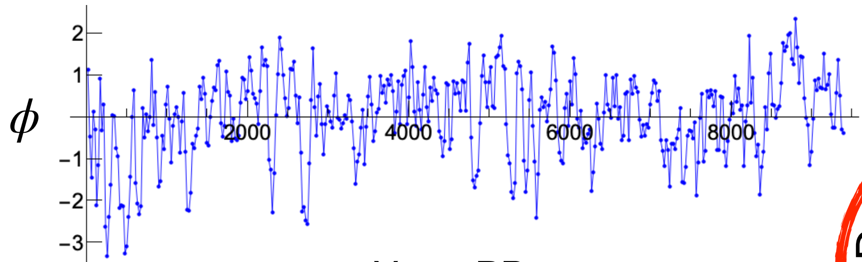
→ Power Spectral Density (PSD)



We don't know the **likelihood function**!

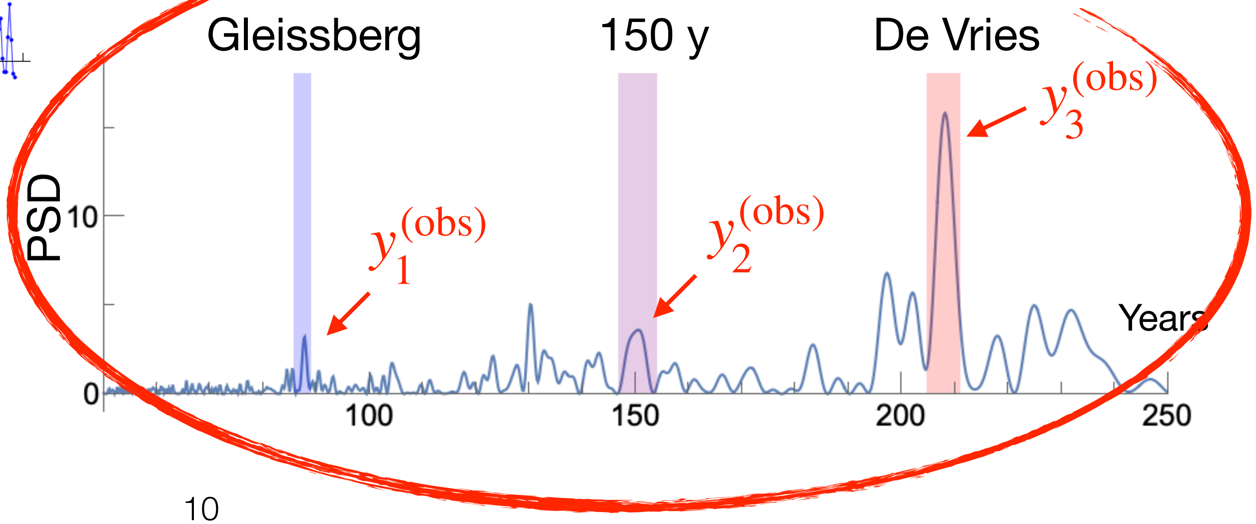
Solar modulation

(from [10]Be and [14]C records)



→ Power Spectral Density (PSD)

Data $\mathbf{y}^{(\mathrm{obs})}$



Abreu et al., A&A 548, 2012

Likelihood-free Bayesian inference

- **Approximate Bayesian Computation (ABC)** methods **bypass the evaluation of the likelihood** function $f\left(\mathbf{y}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right)$

- All ABC-based methods **approximate** the likelihood function by **simulations,** the outcomes of which are **compared** with the observed data

- The simplest **ABC rejection algorithm** iterates the following steps:

1. Sample a parameter set $\hat{\boldsymbol{\theta}} = \{\alpha, a, \epsilon\}$ from the prior $f\left(\boldsymbol{\theta}\right)$

2. **Simulate** a data set $\hat{\mathbf{y}}$ from the model $\mathscr{M}(\hat{\boldsymbol{\theta}})$

   Model $\mathscr{M}$: $p_{n+1} = \alpha f_n\left(p_n\right) p_n + \epsilon_n$

3. The parameter set $\hat{\boldsymbol{\theta}}$ is accepted with tolerance $\delta > 0$ if

$$\boxed{\rho\left(\hat{\mathbf{y}}, \mathbf{y}^{(\mathrm{obs})}\right) \leq \delta}$$

where the **distance** ρ quantifies the discrepancy between $\hat{\mathbf{y}}$ and $\mathbf{y}^{(\mathrm{obs})}$ according to a given metric (e.g., Euclidean distance)

Likelihood-free Bayesian inference

- The outcome of the ABC algorithm is a **sample of parameter values approximately distributed according to the desired posterior distribution**

- …obtained without explicitly evaluating the likelihood function!

Great, but…

very inefficient!

- Broad variety of more sophisticated ABC-based algorithm. We use **Simulated Annealing ABC (SABC)**

Albert et al., Stat. Comput. 25, 2015

In a nutshell…

SABC algorithm

- Replace the distance $\rho\left(\hat{\mathbf{y}}, \mathbf{y}^{(\text{obs})}\right)$ with the **Boltzmann factor**

$$\exp\left[-\rho\left(\mathbf{y}, \mathbf{y}^{(\text{obs})}\right)/T\right]$$

distance measure (energy)

tolerance (temperature)

and **gradually lower the temperature** T during the execution of the algorithm (= reminiscent of **annealing** processes in metallurgy).

- That way, an ensemble of "particles" $\{\mathbf{y}_j, \boldsymbol{\theta}_j\}$ is evolved such that the observables $\mathbf{y}_j$ get more and more concentrated around $\mathbf{y}^{(\text{obs})}$, and the parameters $\boldsymbol{\theta}_j$ represent more and more the posterior $f\left(\boldsymbol{\theta} \mid \mathbf{y}^{(\text{obs})}\right)$

Practically…

SABC algorithm

- **Initialisation**: sample an ensemble of particles $\{\mathbf{y}_j, \boldsymbol{\theta}_j\}$ from the prior $f\left(\mathbf{y}, \boldsymbol{\theta}\right)$
- Then, the **basic SABC loop** is:

  1. Draw a random particle $(\mathbf{y}_j, \boldsymbol{\theta}_j)$ from the ensemble
  2. Make a jump in parameter space $\boldsymbol{\theta}_j \rightarrow \boldsymbol{\theta}_j^*$
  3. **Simulate** a data set $\mathbf{y}_j^*$ from the model $\mathscr{M}(\boldsymbol{\theta}_j^*)$

     Model $\mathscr{M}$: $p_{n+1} = \alpha f_n\left(p_n\right) p_n + \epsilon_n$

  4. Accept the move with probability

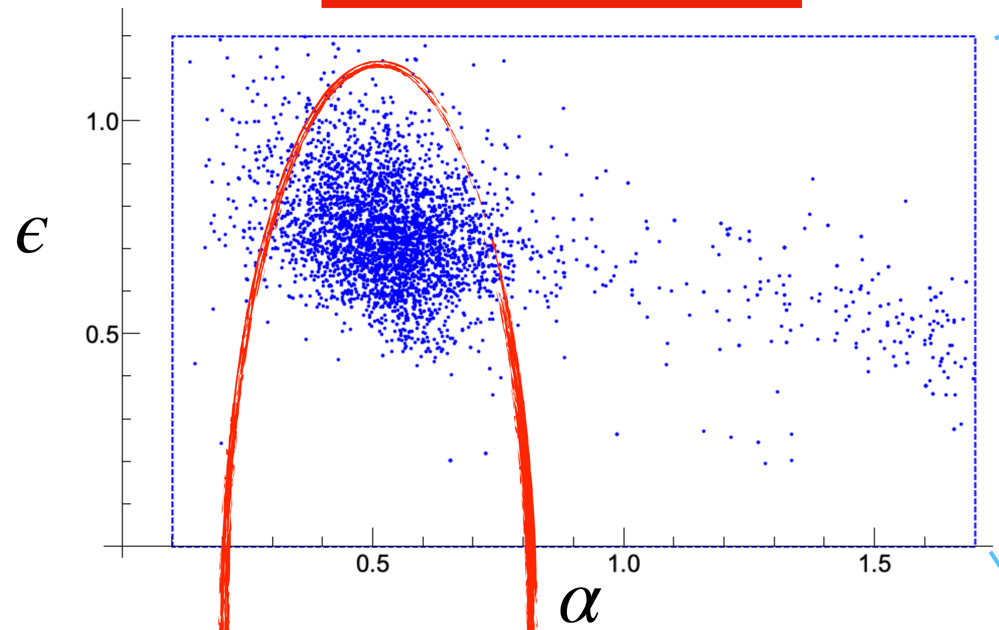
$$\min\left(1, \frac{f(\boldsymbol{\theta}_j^*)}{f(\boldsymbol{\theta}_j)} \exp\left[-\frac{\rho(\mathbf{y}_j^*, \mathbf{y}^{(\mathrm{obs})}) - \rho(\mathbf{y}_j, \mathbf{y}^{(\mathrm{obs})})}{T}\right]\right)$$

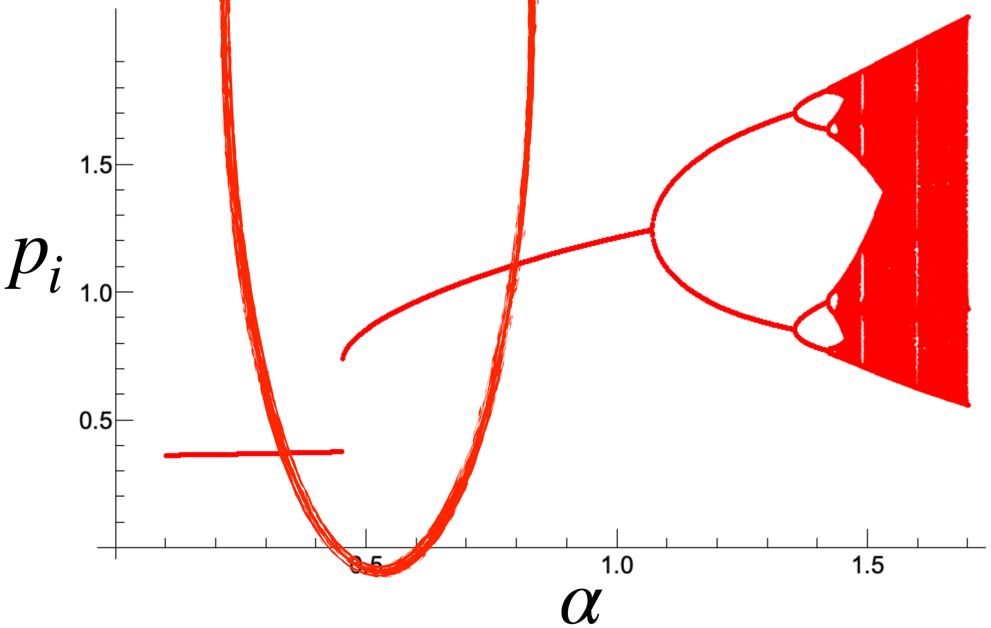
  5. Lower the temperature T a “little bit” (also **adaptively**, depending on the average distance of the ensemble from the target $\mathbf{y}^{(\mathrm{obs})}$)

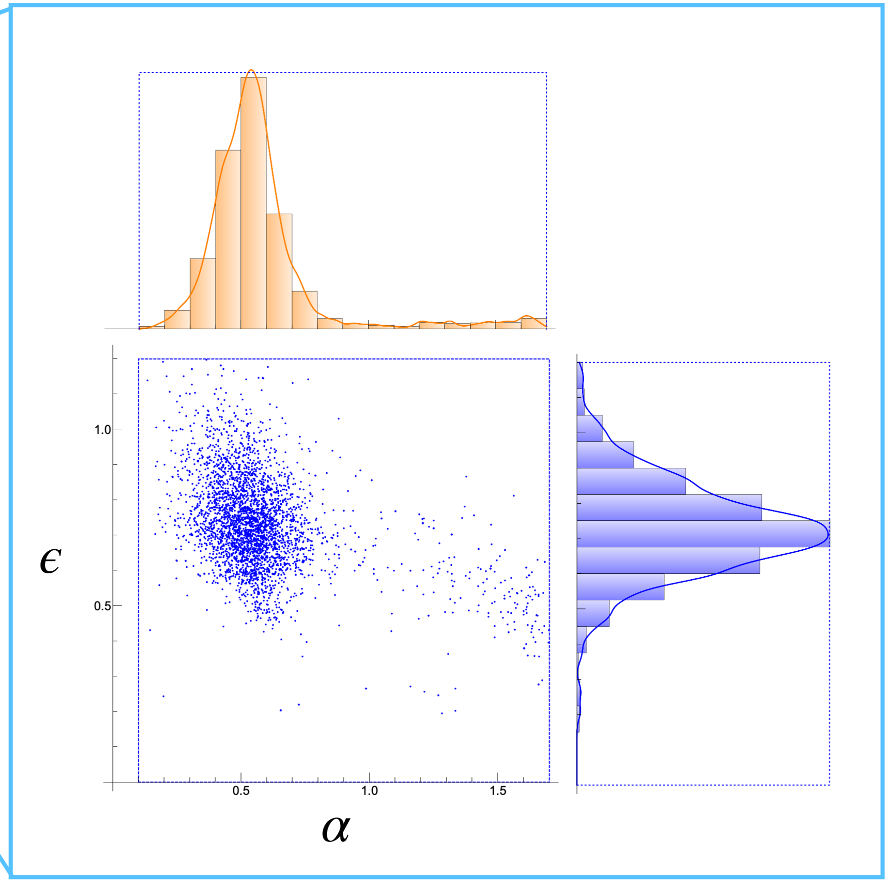
- **SABC** is highly **parallelisable**, and scales **linearly** with the number of particles

Back to Stochastic Resonances

ABC posterior sample





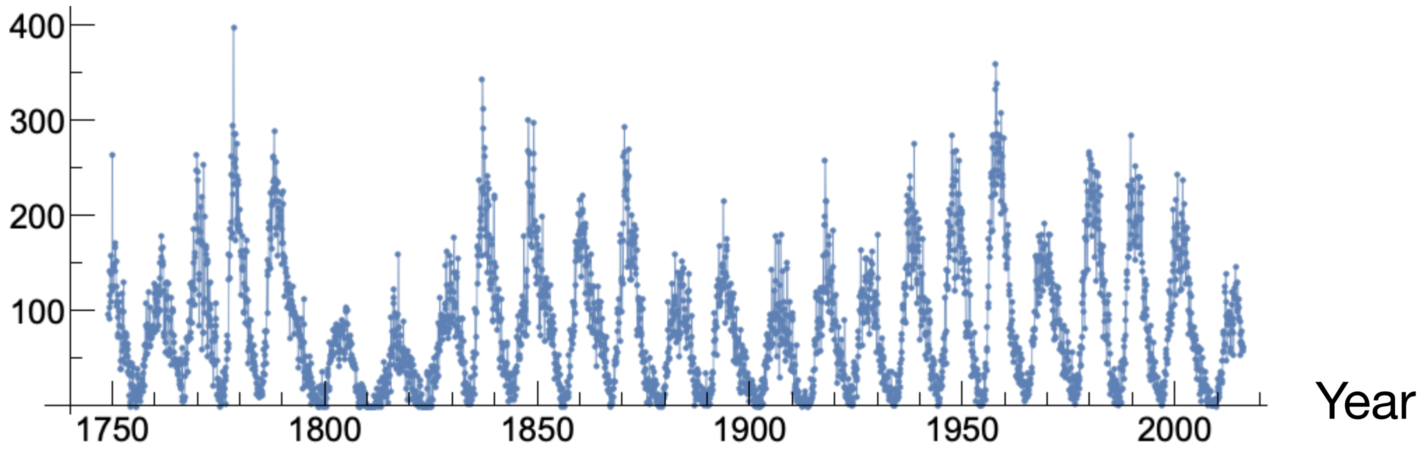


Is this the fingerprint of a Stochastic Resonance?

Pitfalls and remedies

- Data set (= spectral amplitudes of 3 periods) is not very informative
- Proxy data from radionuclides are affected by a variety of “non-solar” processes that are not considered in dynamo models
- The iterative map model is a brute approximation (loss of information)

→ **Data**: use **sunspots number**, the longest direct record of solar magnetic activity. Data available since 1749 with **monthly resolution**.



→ **Model**: use **time-continuous time delay ODE** model

(*Wilmot-Smith et al., Astrophys. J 652, 2006*)

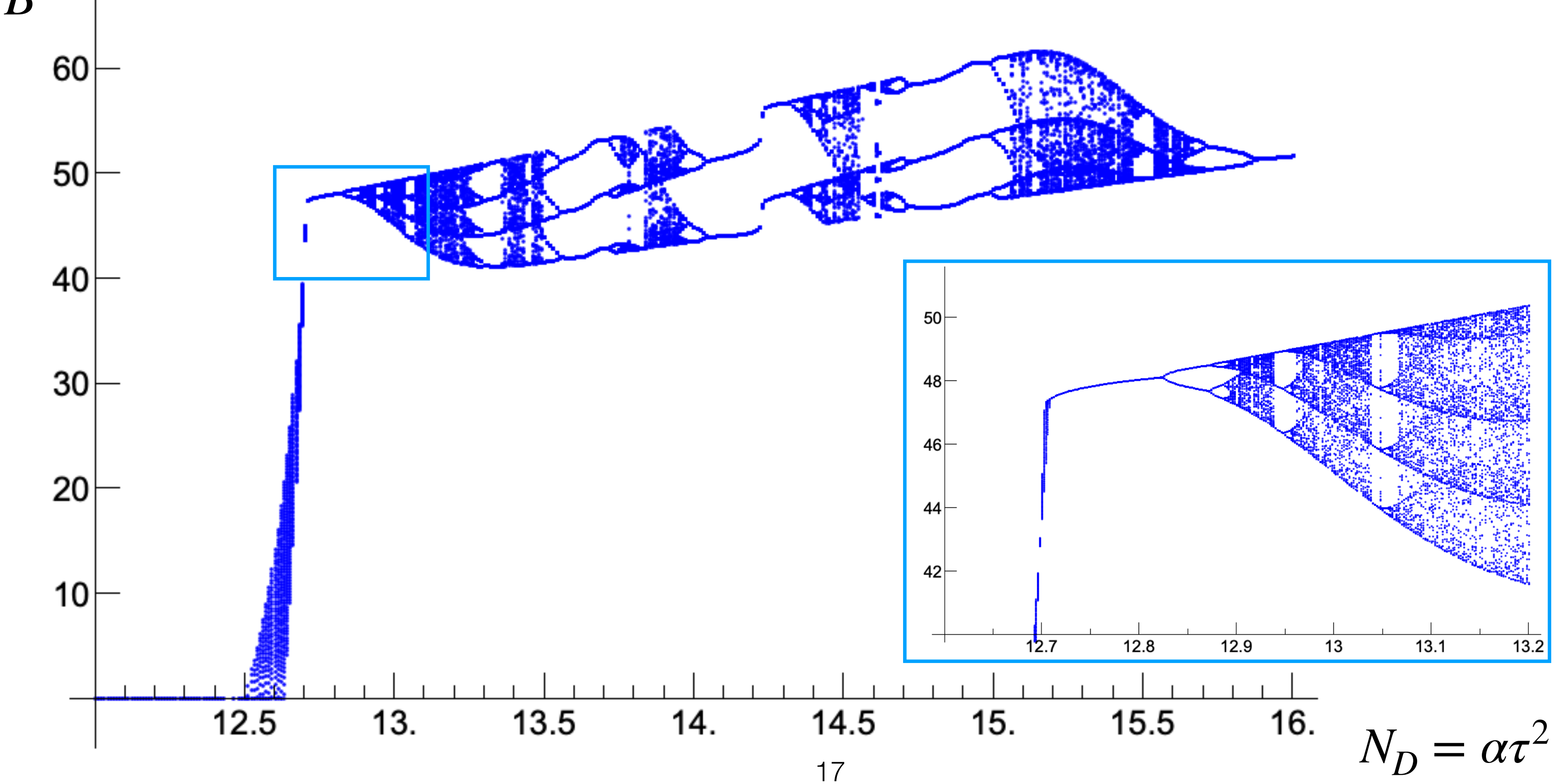
Time delay ODE model

$$\boxed{\ddot{B}(t)+\frac{2}{\tau}\dot{B}(t)+\frac{1}{\tau^2}B(t)=\alpha B(t-T)g(B(t-T))}$$

$$\alpha=\frac{\omega\alpha_0}{L} \qquad T=T_0+T_1$$

$$g(B)=\frac{1}{4}\left(1+\mathrm{erf}(B^2-B_{\min}^2)\right)\left(1-\mathrm{erf}(B^2-B_{\max}^2)\right)$$

Is the sun close to a transition point? Which one?



Stochastic time delay ODE model

- We replace α with $\alpha(t) = \alpha[1 + \eta(t)]$ $\qquad \eta(t) =$ white noise with variance σ^2

- We assume: sunspots number $\propto B^2$

- We can write the likelihood: $f\left(\mathbf{B}^{(\mathrm{obs})} \mid \boldsymbol{\theta}\right) \propto \exp\left[-n\log(\sigma) - \mathcal{S}\right]$

  with the **action** $\mathcal{S}$ defined as:

$$\mathcal{S} = \sum_{i=1}^{n} \frac{\Delta t}{2N_D^2\sigma^2 g^2(B_i)} \left[\tau^2 \frac{B_{i+1} - 2B_i + B_{i-1}}{\Delta t^2} + 2\tau \frac{B_i - B_{i-1}}{\Delta t} + B_i + N_D g(B_i)\right]^2$$

  and the parameters to be inferred, $\boldsymbol{\theta} = \{\tau, N_D, T, \sigma\}$

- Moreover, $\Delta t = 1$ month (observations time step). No need to integrate system dynamics between consecutive observation points.

EMCEE sampler: the MCMC hammer

- Markov Chain Monte Carlo (MCMC) Ensemble sampler

  (*Goodman and Weare, Comm. App. Math. And Comp. Sci. 5, 2010*)

- MCMC generates a **random walk** in parameter space **drawing a representative set of samples** form the posterior distribution $f\left(\boldsymbol{\theta} \mid \mathbf{y}^{(\mathrm{obs})}\right)$

  1. Make a jump in parameter space $\boldsymbol{\theta} \rightarrow \boldsymbol{\theta}^*$ based on a **proposal distribution**
  2. Accept the move with probability $\min\left(1, \frac{f(\boldsymbol{\theta}^* \mid \mathbf{B}^{(\mathrm{obs})})}{f(\boldsymbol{\theta} \mid \mathbf{B}^{(\mathrm{obs})})}\right)$ **Metropolis algorithm**

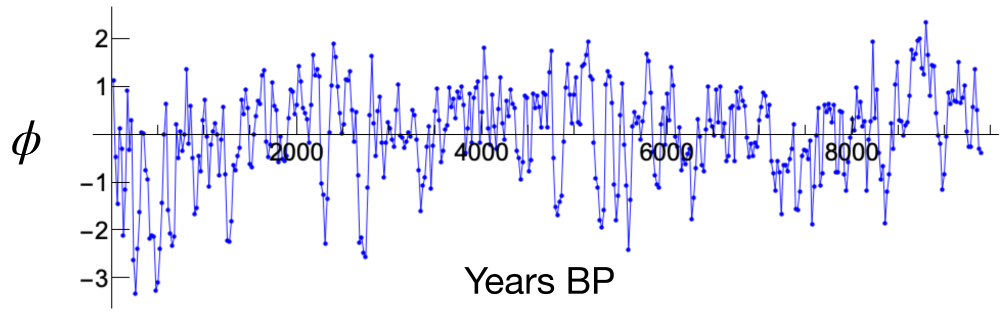
- **EMCEE** involves simultaneously propagating an ensemble of K **walkers** $S = \{X_k\}$ where the proposal distribution for one walker k is based on the current position (in parameter space) of the other $K-1$ walkers

- Very efficient, very few tuning parameters, well-suited for parallel computing

EMCEE sampler: the MCMC hammer

Moreover…

- Radionuclides proxy data (much longer time scale) in combination with time-continuous ODE model



- We need to integrate system dynamics between consecutive data points

- A new ace up our sleeve: **Hamiltonian Monte Carlo (HMC)** (with time-scale separation)

Albert et al, PRE 93, 2016