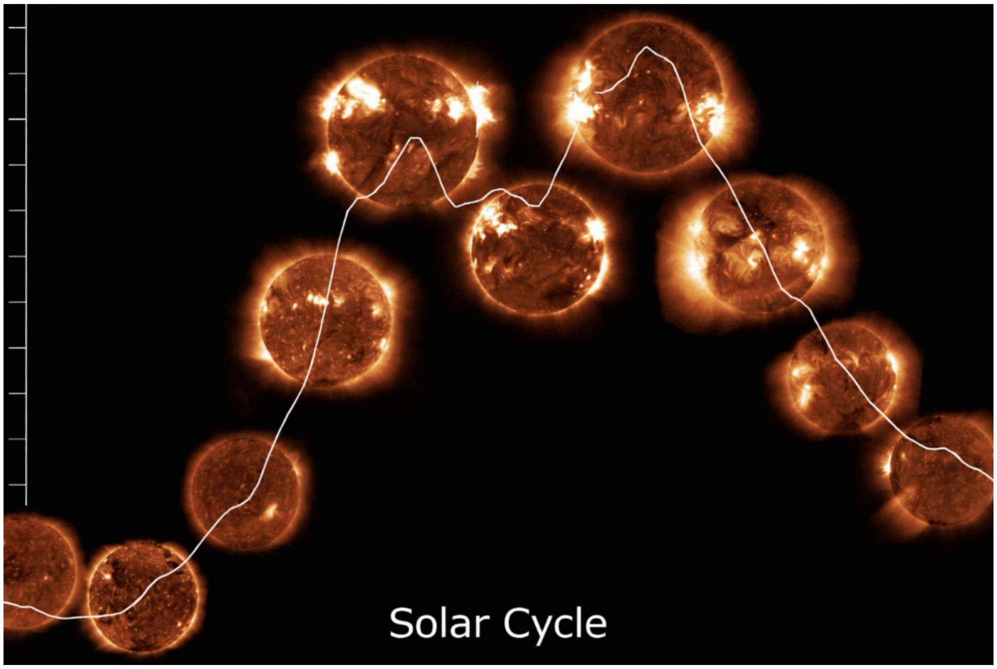
Chasing the Solar Cycle: Parameter Optimisation with Simulated Annealing



Authors:
Penélope Plos, Yeji Huber, Nicole Cieplinski
Zurich University of Applied Sciences (ZHAW)

Course: Optimisation and High Performance Computing
Date: 16.01.2026

1. Introduction

Scientific background

Solar activity is commonly characterised by variations in the Sunspot Number (SN), which exhibits a quasi-periodic behaviour with an average period of approximately 11 years. These solar cycles reflect changes in the Sun's magnetic activity and have been observed over long time periods, enabling detailed empirical analysis of cycle shapes and variability. Understanding and modelling the temporal evolution of solar cycles is important because solar activity influences space weather and can affect Earth-related systems such as satellites and power grids.

Available data

We use the provided dataset data_Team9.csv, containing daily sunspot number measurements. The dataset covers ten consecutive solar cycles (cycle 12 to cycle 21) with a total of 39,220 data points. Each entry consists of a time stamp and the corresponding observed sunspot number.

Goals

The objective is to calibrate a parametric model describing the temporal evolution of solar cycles by minimising the Mean Squared Error (MSE) between observed and modelled sunspot numbers. The work consists of:

1. Hyperparameter tuning for Simulated Annealing (SA) to identify suitable values for the initial temperature $T_0$and proposal variance σ.
2. Model calibration using the chosen hyperparameters via multiple independent SA chains.
3. HPC evaluation: implement a baseline and a parallel version on the Earth cluster (single node, up to 32 cores) by distributing independent SA runs across cores.
4. Scientific evaluation: analysis of calibrated parameters (e.g. correlation between rise and decay parameters) and qualitative comparison with empirical relationships reported in the literature (Hathaway, 2015).

2. Describe the optimisation problem

Objective function (to be minimised): **MSE**

The optimisation problem is formulated as the minimisation of the Mean Squared Error (MSE) between the observed sunspot numbers and the model predictions:

$$\boxed{\frac{1}{N}\sum_{i=1}^{N}\epsilon_i^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x(t_i)\right)^2}$$

where:

- $\boldsymbol{N}$ is the total number of data points (39,220),
- $\boldsymbol{y_i}$denotes the observed sunspot number at time $\boldsymbol{t_i}$,
- $\boldsymbol{x(t_i)}$is the model prediction.

Model function

Each solar cycle k is modelled using an asymmetric rise–decay function based on Volobuev (2009). For times $t$belonging to cycle k, the model is given by:

$$\boxed{x_k(t) = \left(\frac{t - T0_k}{Ts_k}\right)^2 e^{-\left(\frac{t - T0_k}{Td_k}\right)^2}}$$

where:

- $\boldsymbol{T0_k}$= start time of cycle k,
- $\boldsymbol{T_{s,k}}$= rise-time parameter,
- $\boldsymbol{T_{d,k}}$= decay-time parameter.

Problem variables (to be optimized)

The optimisation involves a total of 30 free parameters:

$$\{T_{01}, T_{s1}, T_{d1}, \dots, T_{0,10}, T_{s,10}, T_{d,10}\}$$

corresponding to the start time, rising time, and decay time of each of the ten solar cycles.

Optimisation algorithm: **Simulated Annealing**

Simulated Annealing (SA) is a stochastic optimisation algorithm inspired by physical annealing processes. Starting from an initial parameter vector, new candidate solutions are generated by adding random perturbations.

If a proposed solution improves the MSE, it is accepted. If it worsens the MSE, it can still be accepted with a probability given by the Metropolis criterion:

$$p = \exp\left(\ -\frac{\Delta E}{T}\right)$$

where $\Delta E$is the change in MSE and $T$is the current temperature.

$$\boxed{\Delta E = f(X') - f(X_n) \rightarrow X' = X_n + z \rightarrow z \sim D = N(0,\boxed{\sigma})}$$

σ: step size

$$\begin{cases} 1 & \text{if } \Delta E \leq 0 \\ e^{-\Delta E/T} & \text{if } \Delta E > 0 \end{cases}$$

if the candidate $\mathbf{x}'$ is better, accept it

if the candidate $\mathbf{x}'$ is worse, accept it with a probability that depends on the temperature T and the difference ΔE

Annealing schedule and hyperparameters

The temperature decreases linearly during the optimisation:

$$T_n = T_0\left(1-\frac{n}{n_{\max}}\right)$$

The main hyperparameters of the algorithm are:

- **Initial temperature** T_0: controls how easily worse solutions are accepted at the beginning.
- **Step size** σ: controls how large the random parameter changes are in each iteration.

Constraints

for 10 cycles:

$$x_1(t) = \left(\frac{t-T0_1}{Ts_1}\right)^2 e^{-\left(\frac{t-T0_1}{Td_1}\right)^2} \quad \text{with} \quad T0_1 \leq t < T0_2$$

$$x_2(t) = \left(\frac{t-T0_2}{Ts_2}\right)^2 e^{-\left(\frac{t-T0_2}{Td_2}\right)^2} \quad \text{with} \quad T0_2 \leq t < T0_3$$

$$x_3(t) = \left(\frac{t-T0_3}{Ts_3}\right)^2 e^{-\left(\frac{t-T0_3}{Td_3}\right)^2} \quad \text{with} \quad T0_3 \leq t < T0_4$$

$$\ldots$$

$$x_{10}(t) = \left(\frac{t-T0_{10}}{Ts_{10}}\right)^2 e^{-\left(\frac{t-T0_{10}}{Td_{10}}\right)^2} \quad \text{with} \quad t \geq T0_{10}$$

Outline of the initial (non-parallel) solution

The baseline workflow consists of:

1. Loading and preprocessing the data.
2. Defining the model and MSE objective.
3. Hyperparameter tuning via independent SA runs on a grid of (T_0, σ)values.
4. Calibrate the model using multiple independent SA chains with the optimal hyperparameters obtained in the previous step.
5. Aggregation of chain results by averaging the final parameter vectors (center-of-mass estimate).

3. Parallel programming techniques

Parallelisation opportunities in the problem

The optimisation problem involves 30 model parameters and 39,220 daily observations, making each MSE evaluation during a simulated annealing (SA) iteration computationally expensive. Although a single SA run is inherently sequential due to its state-dependent updates, different SA runs are completely independent. The evaluation of multiple hyperparameter combinations also consists of independent runs, which provides a clear opportunity for parallel execution.

By distributing these independent SA executions across 32 cores on a single compute node in the Earth partition, we significantly reduced the overall execution time and enabled a much broader exploration of possible hyperparameter configurations. In addition to the speed-up, this parallelisation strategy improved the robustness of the results by allowing efficient evaluation of multiple initial conditions and independent stochastic realisations.

Parallelization techniques: Multiprocessing

The parallelization required for this project is implemented using a multi-process configuration, running multiple independent SA executions concurrently on the cluster. As specified in the project documentation, both the hyperparameter exploration and the repeated SA optimisation runs are performed in parallel on a single compute node with 32 cores in the Earth partition.

Due to the limitations of thread-based parallelism in Python, parallel execution was achieved by launching multiple processes, each responsible for an independent SA run. Since these runs do not depend on one another, they could be cleanly distributed across the available cores without requiring communication or synchronisation between processes.

This design avoided the need for complex message-passing frameworks such as MPI. Furthermore, data-partitioning strategies were not applicable, as every SA execution required access to the full dataset to compute the loss function. As a result, using multiprocessing alone enabled a substantial reduction in optimisation time and allowed us to efficiently explore a broader range of parameter configurations.

Challenge: Insufficient wall-time allocation in scaling experiments

During the scaling experiments, we initially observed that the calibration runs executed with 1 and 2 cores did not produce any stored results, even though the jobs were successfully launched. Upon investigation, we found that the issue was not related to the calibration procedure itself but to an underestimation of the required wall time.

The original job configuration specified a maximum runtime of #SBATCH --time=0-02:00:00, which was insufficient for low-parallelism runs with long execution times. As a result, these jobs were terminated by the scheduler before completion, preventing the JSON output files from being written. After increasing the time limit to #SBATCH --time=0-04:00:00, the runs completed successfully and their results were correctly stored.

This highlights that the initial runtime estimate was too optimistic, particularly for configurations with a small number of cores, and emphasizes the importance of adapting wall-time allocations when conducting scaling studies on HPC systems.

4. Results and performance metrics

Hyperparameter tuning

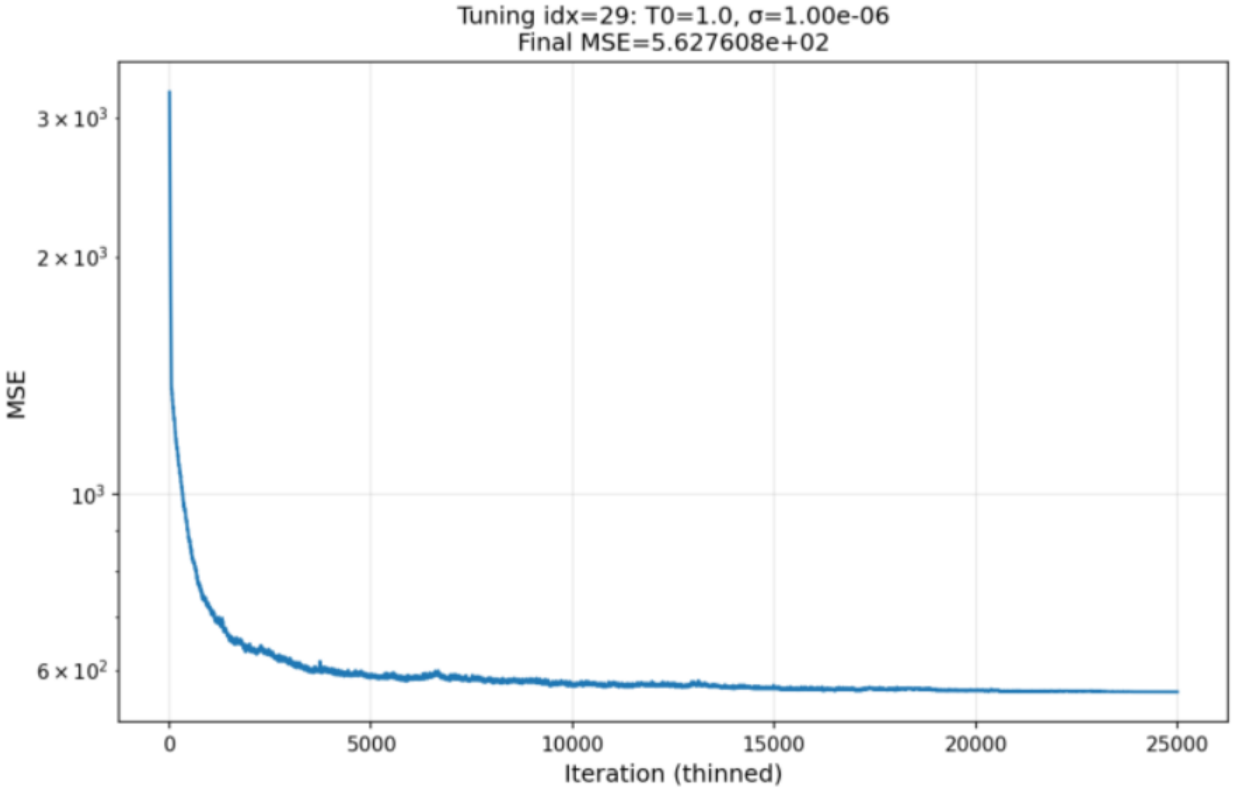
Hyperparameter tuning was performed through exploratory testing of individual (T_0, σ) combinations to identify reasonable ranges for systematic evaluation. Based on these preliminary tests, a final tuning sweep was conducted on the cluster, exploring values of T_0 ranging from 0.1 to 10 and σ ranging from 10^{-4} to 10^{-11}. The explored hyperparameters were the initial temperature T_0 and the proposal variance σ.

Loss curves (MSE vs. iteration) were inspected to identify unstable or stagnating behaviour. From the full set of tuning runs, a subset of configurations (trials 5, 6, 13, 14, 21, 22, 29, 30, 37, 38) showing stable convergence behavior was selected for detailed comparison based on their final MSE values.

The best-performing configuration was:

- Initial temperature: $T_0 = 1.0$
- Step size: $\sigma = 1 \times 10^{-6}$

This configuration achieved the lowest final MSE (≈ 563) and showed smooth, stable convergence. A visual comparison of the corresponding model output with the observed data confirmed a clear improvement compared to the initial model.



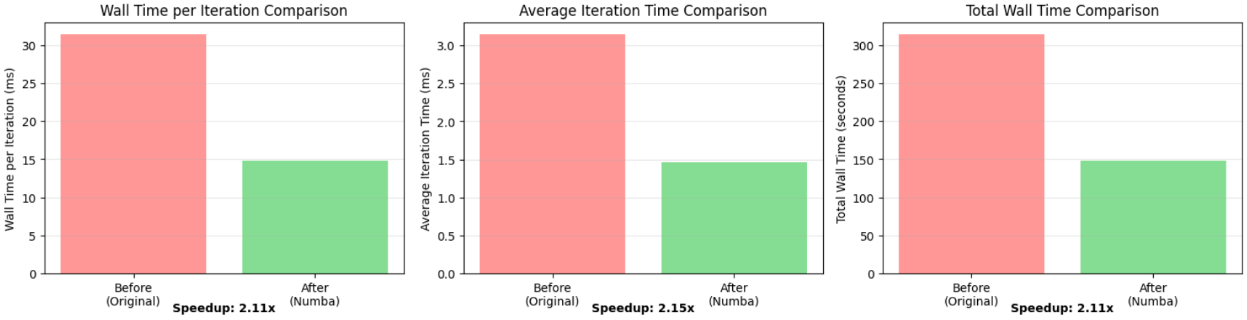
Performance optimization with Numba

To accelerate the calibration process, Numba Just-In-Time (JIT) compilation was applied to the computational bottleneck of the pipeline: the model evaluation. Using identical calibration settings on a single core (n_chains = 10, n_iter = 10,000, burn-in = 5,000, n_workers = 1), the Numba-accelerated implementation was quantitatively compared against the original NumPy-based version.

The results show a consistent and substantial performance improvement. The wall time per iteration was reduced from 31.41 ms to 14.88 ms, corresponding to a speedup of approximately 2.1×. Similar gains were observed in both the average iteration time (2.15× faster) and the total wall time, which decreased from 314.08 s to 148.83 s.

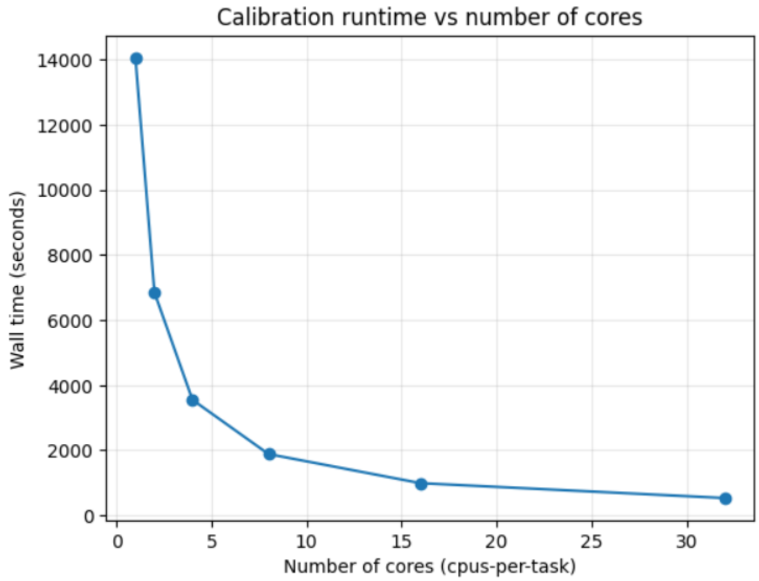
The close agreement across all performance metrics indicates that the observed speedup is primarily due to faster model evaluation rather than reduced overhead or parallelization effects. Importantly, the optimization produces identical numerical results, confirming that the Numba JIT compilation improves computational efficiency without affecting the calibration or optimization outcome.

Overall, this performance gain is particularly relevant for large-scale runs on the HPC cluster, where reducing per-iteration cost directly translates into shorter wall times and more efficient resource utilization.

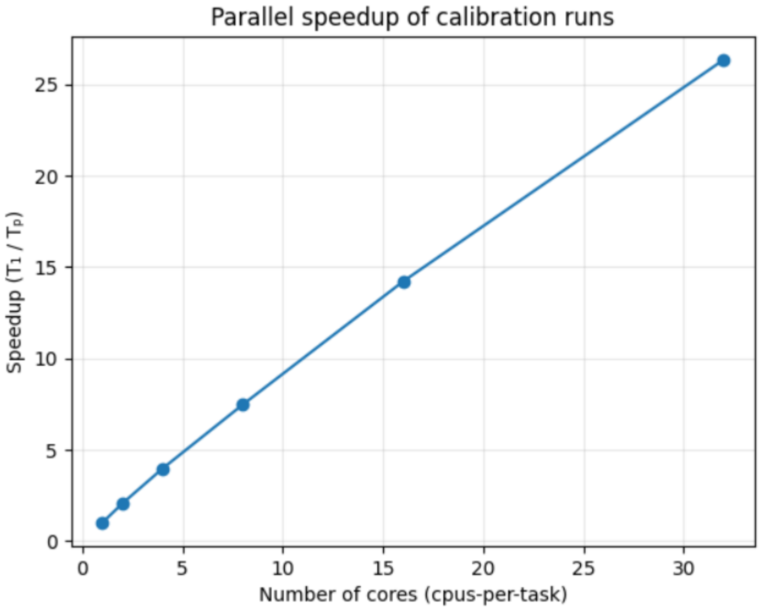


Parallel performance on the cluster

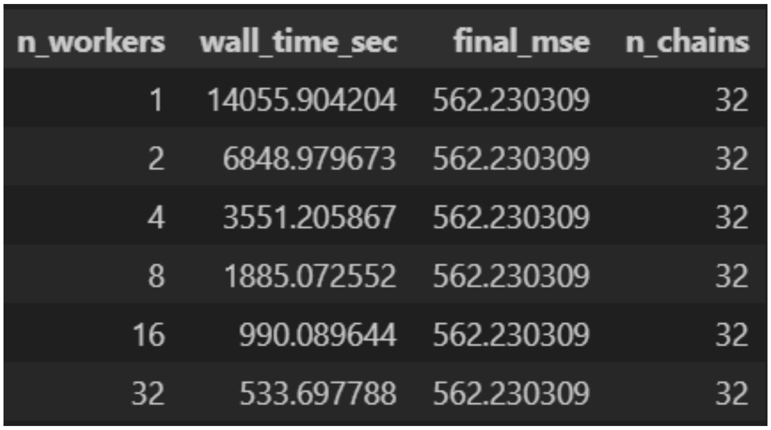
Calibration runs were executed on a single compute node of the HPC cluster using varying numbers of CPU cores (1, 2, 4, 8, 16, and 32), while keeping the total number of Simulated Annealing chains fixed (n_chains = 32). Parallelization was achieved by distributing independent chains across multiple workers using multiprocessing, such that each worker handled a subset of chains.



The wall-clock runtime decreases significantly as the number of allocated cores increases. Specifically, the runtime is reduced from approximately 14,056 seconds on a single core to about 534 seconds when using 32 cores. This corresponds to an overall speedup of roughly 26×, indicating near-linear scaling across a wide range of core counts. The most pronounced runtime reductions occur when increasing the number of cores from 1 to 8, after which the gains become more gradual.



Minor deviations from ideal linear scaling are observed at higher core counts. These effects are attributed to unavoidable multiprocessing overhead, including process management, inter-process communication, and synchronization costs. Nevertheless, the scaling efficiency remains high, and the overall trend clearly demonstrates the benefit of parallel execution.



n_workers	wall_time_sec	final_mse	n_chains
1	14055.904204	562.230309	32
2	6848.979673	562.230309	32
4	3551.205867	562.230309	32
8	1885.072552	562.230309	32
16	990.089644	562.230309	32
32	533.697788	562.230309	32

Importantly, the final mean squared error (MSE) remains identical across all runs, confirming that parallelization affects only the runtime and not the numerical outcome or solution quality of the calibration. All configurations converge to the same fitted parameters, as represented by the aggregated center_of_mass vector.

Overall, these results confirm that distributing independent Simulated Annealing chains across multiple CPU cores is an effective and scalable strategy for accelerating the calibration process on an HPC system. The observed near-linear speedup validates both the parallel design of the algorithm and its suitability for large-scale execution.

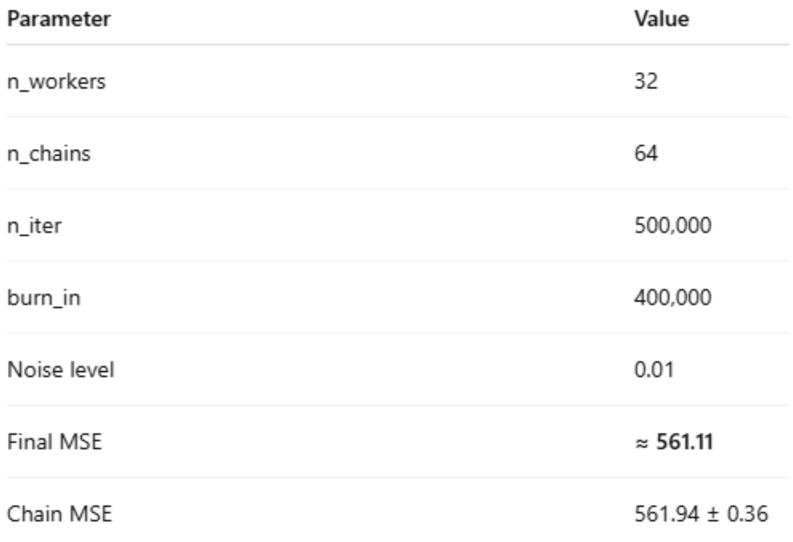
Parameter Selection

To identify the parameter configuration that produces the lowest final MSE, the following conditions were systematically tested:

- Number of iterations: 250,000 (burn_in : 200,000), 500,000 (burn_in : 400,000)
- Number of parallel chains: 32 chains, 64 chains
- Initialization noise levels: 0.1, 0.01, 0.001, 0.0001

These combinations allow a controlled evaluation of how sampling depth (iterations), parallel exploration (chains), and robustness to initialization (noise level) influence the final calibration accuracy.

Selected Final Calibration Parameters

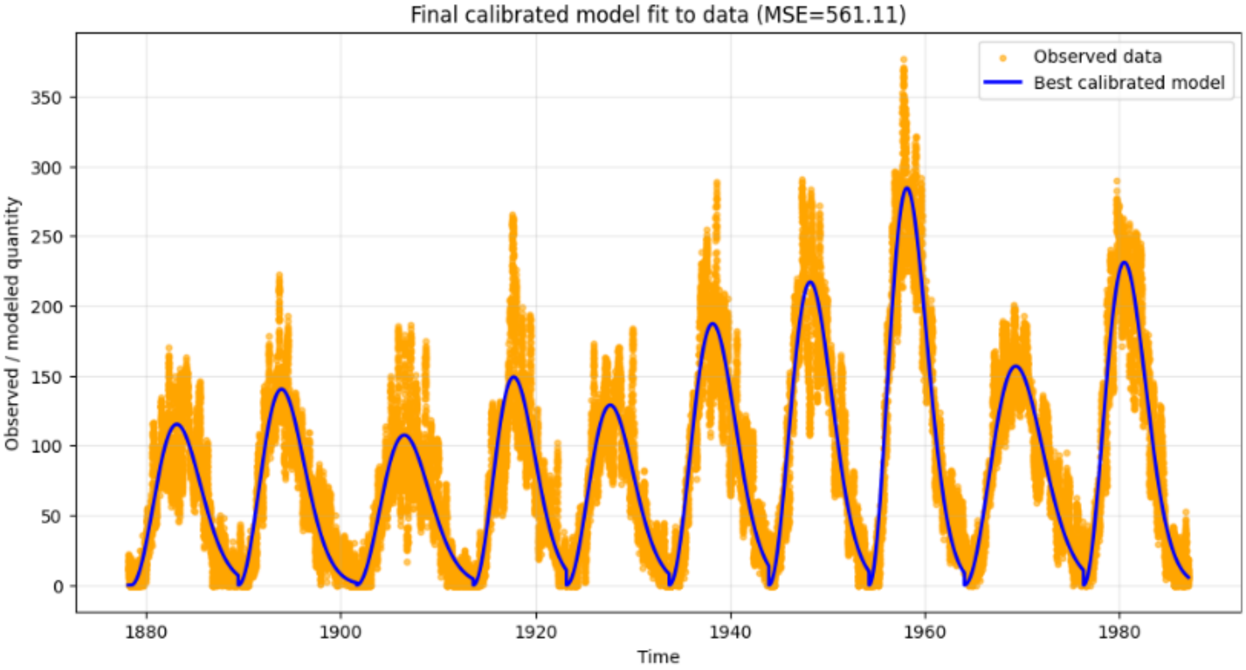


Parameter	Value
n_workers	32
n_chains	64
n_iter	500,000
burn_in	400,000
Noise level	0.01
Final MSE	≈ **561.11**
Chain MSE	561.94 ± 0.36

Based on the combined results of the configuration comparison and the noise robustness analysis, the final calibration parameters are chosen to balance solution quality, stability, and computational cost. The selected setup uses increased iterations and chains together with a moderate initialization noise level of 0.01, achieving a final MSE of approximately 561.11 with low chain-to-chain variability. This configuration represents the best overall trade-off among the evaluated alternatives.

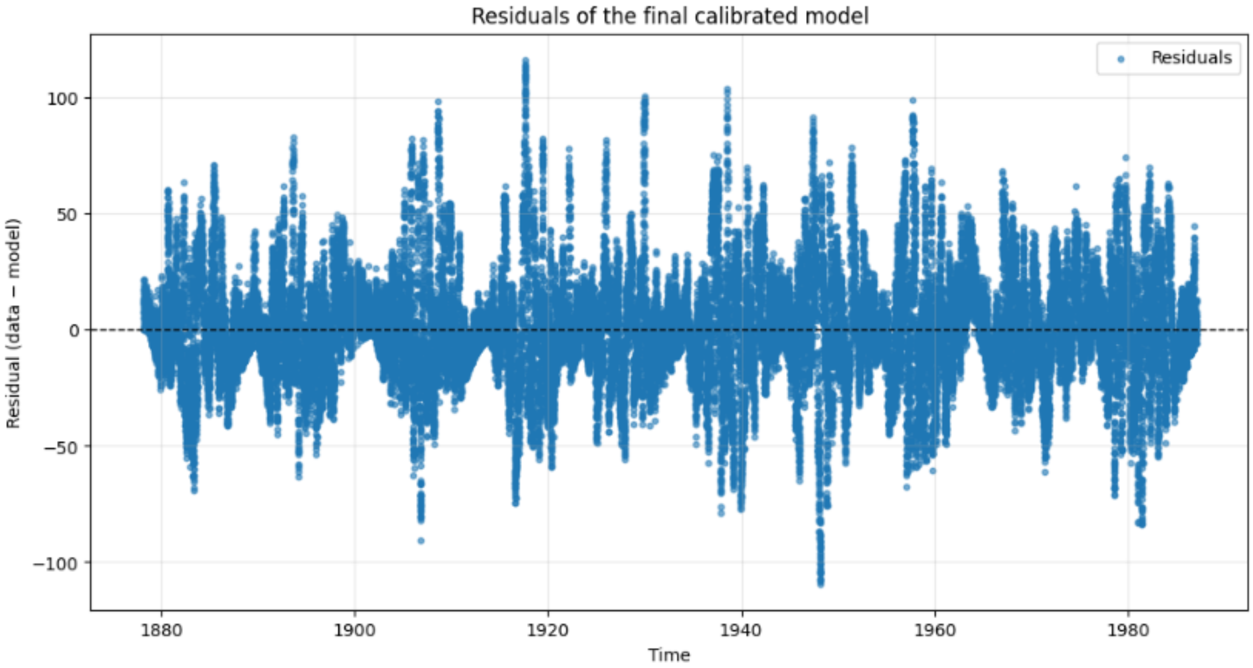
5. Conclusions

Quality of the fit



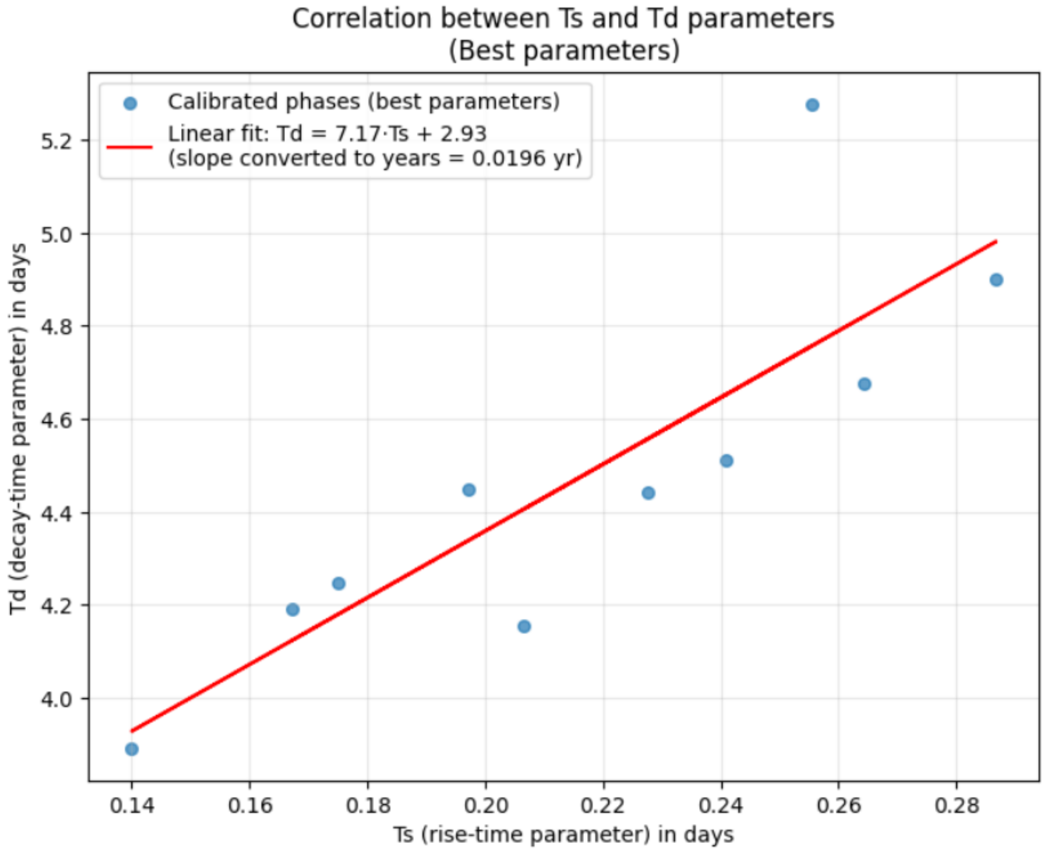
The final calibrated configuration (500 000 iterations, 64 chains, noise level 0.01) achieves a minimum MSE of **561.11** with extremely low chain-to-chain dispersion (± 0.36). This consistency demonstrates excellent convergence of the simulated annealing process and confirms that the optimisation is not driven by stochastic artefacts. Visually and quantitatively, the calibrated model closely reproduces the timing of cycle onsets, the asymmetric rise–decay structure, and the overall envelope of solar activity across all ten cycles. Remaining discrepancies are confined to short-lived, high-activity peaks and are best interpreted as intrinsic variability not represented by the smooth analytical model rather than as calibration failure.

Residual behaviour and model adequacy



Residuals are centred around zero over the full time span, with no detectable long-term drift or systematic bias. The increase in residual variance during high-activity phases is physically expected and reflects enhanced stochasticity in sunspot emergence. This behaviour confirms that the model is well specified at the cycle scale and that its limitations are structural (intentional smoothing) rather than numerical or optimisation-related.

Correlation between rise and decay times



A strong positive linear relationship between the rise-time parameter $T_s$and the decay-time parameter $T_d$is observed. The fitted relation

$$T_d = 7.17\, T_s + 2.93$$

(with parameters expressed in days) yields a coefficient of determination $R^2 = 0.712$, indicating that approximately **71 % of the variance** in decay time is explained by the rise time alone. After unit conversion, the slope corresponds to **≈ 0.0196 years**, which is in excellent agreement with reported literature values (≈ 0.02). This agreement is non-trivial: it shows that the optimisation not only fits the data but also recovers a physically meaningful empirical law of the solar cycle.

Overall assessment

Taken together, the low final MSE, strong convergence diagnostics, well-behaved residuals, and the statistically significant $T_s - T_d$ correlation consistent with published studies demonstrate that the calibrated model provides a robust and physically plausible representation of solar-cycle dynamics. The results support the validity of the optimisation strategy and confirm that simulated annealing, when properly tuned and parallelised, is well suited for high-dimensional, non-convex calibration problems in solar physics.

6. Lessons learned and personal experience

Teamwork

Regarding teamwork, the project benefited from close collaboration and flexible task sharing. Team members supported each other during problem solving and actively searched for solutions when difficulties arose. The division of tasks was dynamic: while one person focused on code implementation and debugging, others updated the notebook documentation, analysed results, or worked on the written report. This parallel workflow improved efficiency and helped maintain consistency between code, notebook, and report

Challenges

Challenge 1: Excessive Runtime Under Full-Scale Calibration Settings

During the initial performance evaluation of the Numba-accelerated implementation, we attempted to use the same full-scale calibration configuration as in the scaling experiments (T0 = 1.0, $\sigma = 1\times10^{-6}$, n_chains = 32, n_iter = 250,000, burn-in = 200,000) to ensure consistency across experiments. However, this setup resulted in excessively long runtimes, with executions exceeding five hours without completion, making it impractical for controlled performance benchmarking.

To obtain meaningful and reproducible performance measurements within a reasonable time budget, we revised the experimental conditions to a reduced yet representative configuration. The final benchmark was conducted using n_chains = 10, n_iter = 10,000, burn-in = 5,000, and n_workers = 1. This configuration preserves the computational structure of the calibration process while allowing a fair and controlled comparison between the baseline and Numba-accelerated implementations.

This challenge highlights the importance of selecting appropriate benchmark configurations when evaluating performance optimizations, particularly when the objective is to isolate computational improvements rather than to reproduce full-scale production runs.

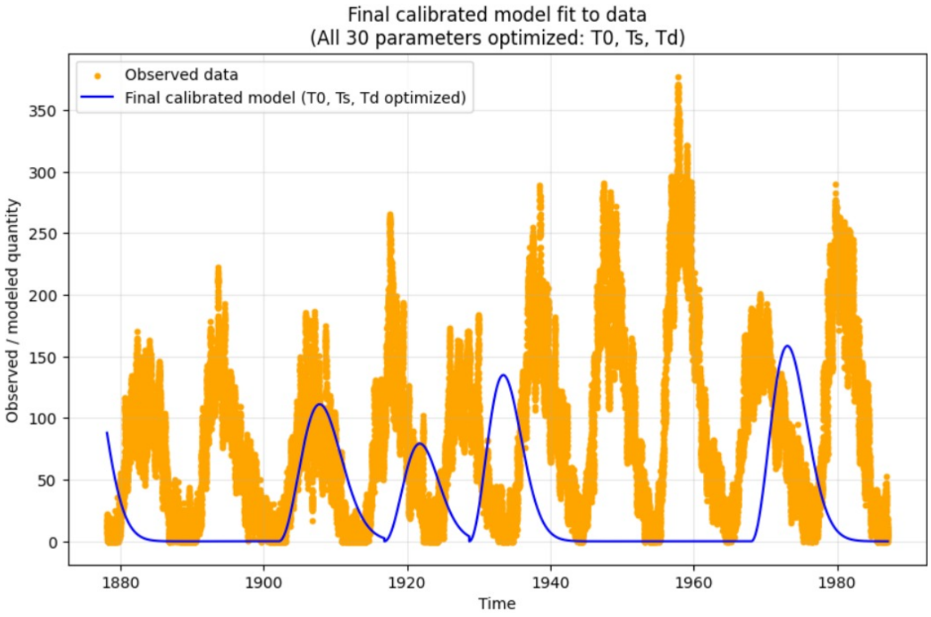
Challenge 2: Calibration Instability Due to Inappropriate Noise Step Size

During the initial calibration attempts following hyperparameter tuning, we encountered unexpected issues, as the mean squared error (MSE) increased from 563 to approximately 1,163 compared to the calibration baseline, and the cause was not immediately clear.

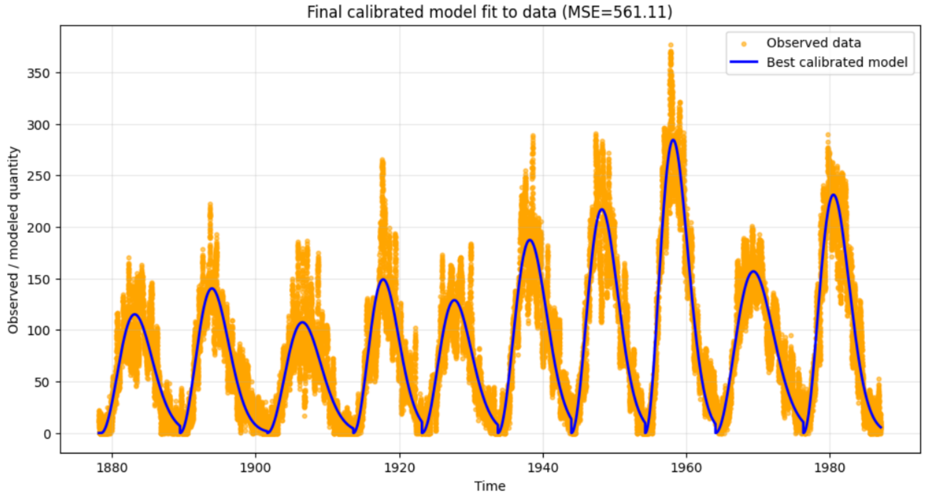
We therefore formulated several hypotheses. First, we hypothesized that the number of iterations (*n_iter*) and the burn-in period (*n_burnin*) were insufficient, causing the optimization to converge to a local minimum. We tested multiple configurations with increased values for both parameters; however, the MSE did not improve.

As a second hypothesis, we considered that the noise step size might be too large. By inspecting the optimization plots, we observed irregular behavior, particularly at the starting points of the cycles. Based on this observation, we experimented with several noise step sizes. Among them, a value of 0.01 yielded the lowest final mean squared error (final MSE) and was therefore selected for the final optimization.

Before adjustement of nose size:



After adjustement of nose size:



Insights

1. Sensitivity to Multiple Parameters

A key insight from this project is that the final calibration quality results from the combined interaction of many factors, including the number of iterations, burn-in length, number of chains, proposal step size, initialization noise, and annealing schedule.

Small changes in these parameters can lead to markedly different optimisation trajectories, convergence behaviour, and final MSE values. This reinforced the importance of systematic exploration and controlled experiments, rather than ad-hoc tuning, when dealing with high-dimensional and stochastic optimisation problems.

2. Critical Role of Visualization

Another major lesson is the central importance of visualization as a diagnostic and problem-solving tool. Numerical outputs alone (e.g. final MSE values) were often insufficient to understand optimisation failures or unexpected behaviour.

Visual tools such as MSE vs. iteration curves, comparisons between observed data and fitted model curves, residual plots, and distributions of Monte Carlo samples across chains proved essential for identifying stagnation, instability, inappropriate noise scales, or premature convergence. In several cases, visualization directly guided corrective actions (e.g. hyperparameter tuning, adjusting noise step size, iteration_n and bruning_n)

3. From Theory to Practice

Through this project, we gained a much deeper understanding of stochastic optimisation, HPC, Numba and multiprocessing, which had previously been covered only at a theoretical level in the lectures. Rather than focusing solely on the mathematical formulation of the algorithms, we were able to observe how these methods behave in practice when applied to real data and executed in realistic computational environments, as well as to understand their practical limitations.

7. Wrap-up

We successfully calibrated a multi-cycle solar activity model using Simulated Annealing and evaluated its performance on an HPC system. The final workflow is scientifically meaningful, computationally efficient, and reproducible.

References / Literature

Hathaway, D. H. (2015).
The solar cycle.
Living Reviews in Solar Physics, 12(4).

Volobuev, D. M. (2009).
The shape of the sunspot cycle: A one-parameter fit.
Solar Physics, 258.

Use of AI-assisted Tools
OpenAI.
ChatGPT (GPT-5.2), Large Language Model.
Used for language refinement, consistency checks, and stylistic corrections of the written report.