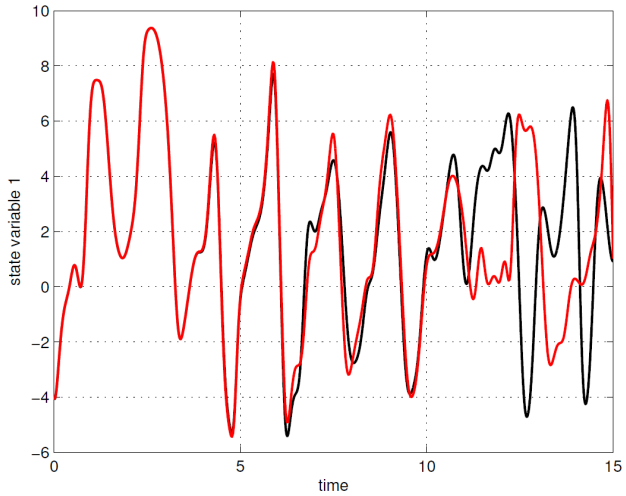
MCMC for UQ: high CPU models

H. Haario

DTU 18.12.2018

Index of the presentation

Chaotic Systems

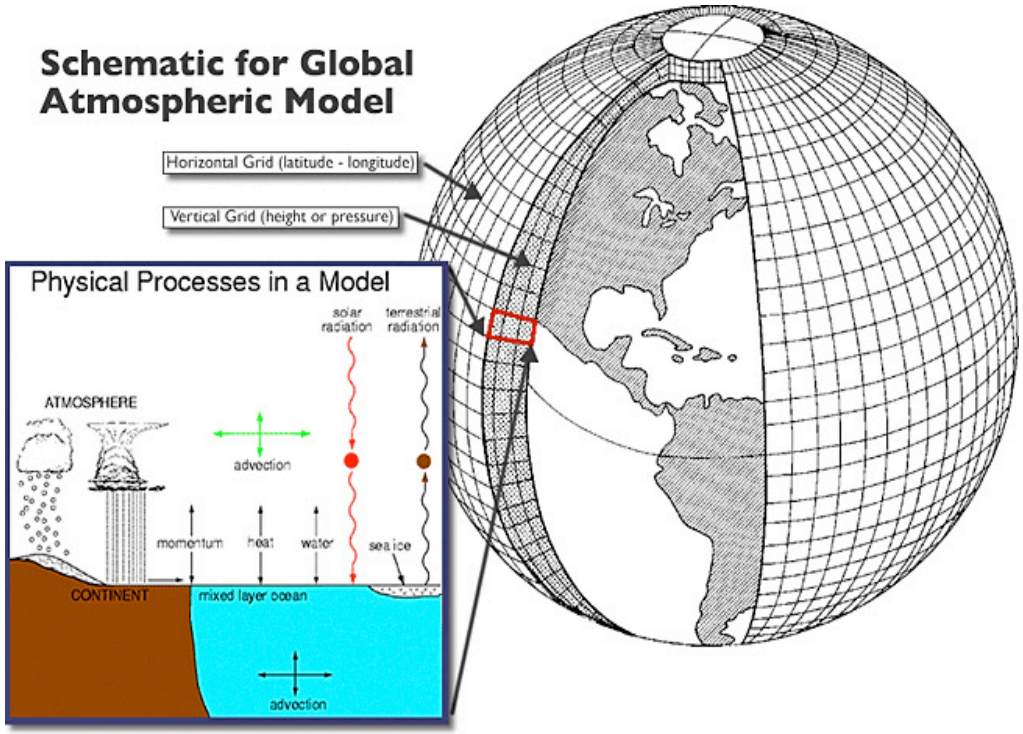


After a predictable interval, any changes (initial values, parameters, solver settings) lead to unpredictable deviations. Options:

- deal with predictable time intervals only (Weather)
- deal with behaviour after predictability (Climate).
- **How to create cost functions for parameters of a chaotic system? Distinguish chaotic variability of a fixed system from systematic change between different systems?**

Long time: summary statistics for climate models ?

In order to find 'typical' behaviour of a chaotic climate system, observations and simulations may be averaged in space and time to create 'summary statistics'.



Long time: summary statistics for climate models ?

- If the statistics of the summary expression is known, a likelihood is formulated which yields the posterior for the model parameters.

Long time: summary statistics for climate models ?

- If the statistics of the summary expression is known, a likelihood is formulated which yields the posterior for the model parameters.
- The approach was implemented for the ECHAM5 climate model, using likelihoods based on monthly global and zonal net radiation averages.

Järvinen, H., Räisänen, P., Laine, M., Tamminen, J., Ilin, A., Oja, E., Solonen, A., H.H.: *Estimation of ECHAM5 climate model closure parameters with adaptive MCMC*, Atmos. Chem. Phys., Vol. 10, nro. 2, 9993-10002, 2010.

MCMC for climate models

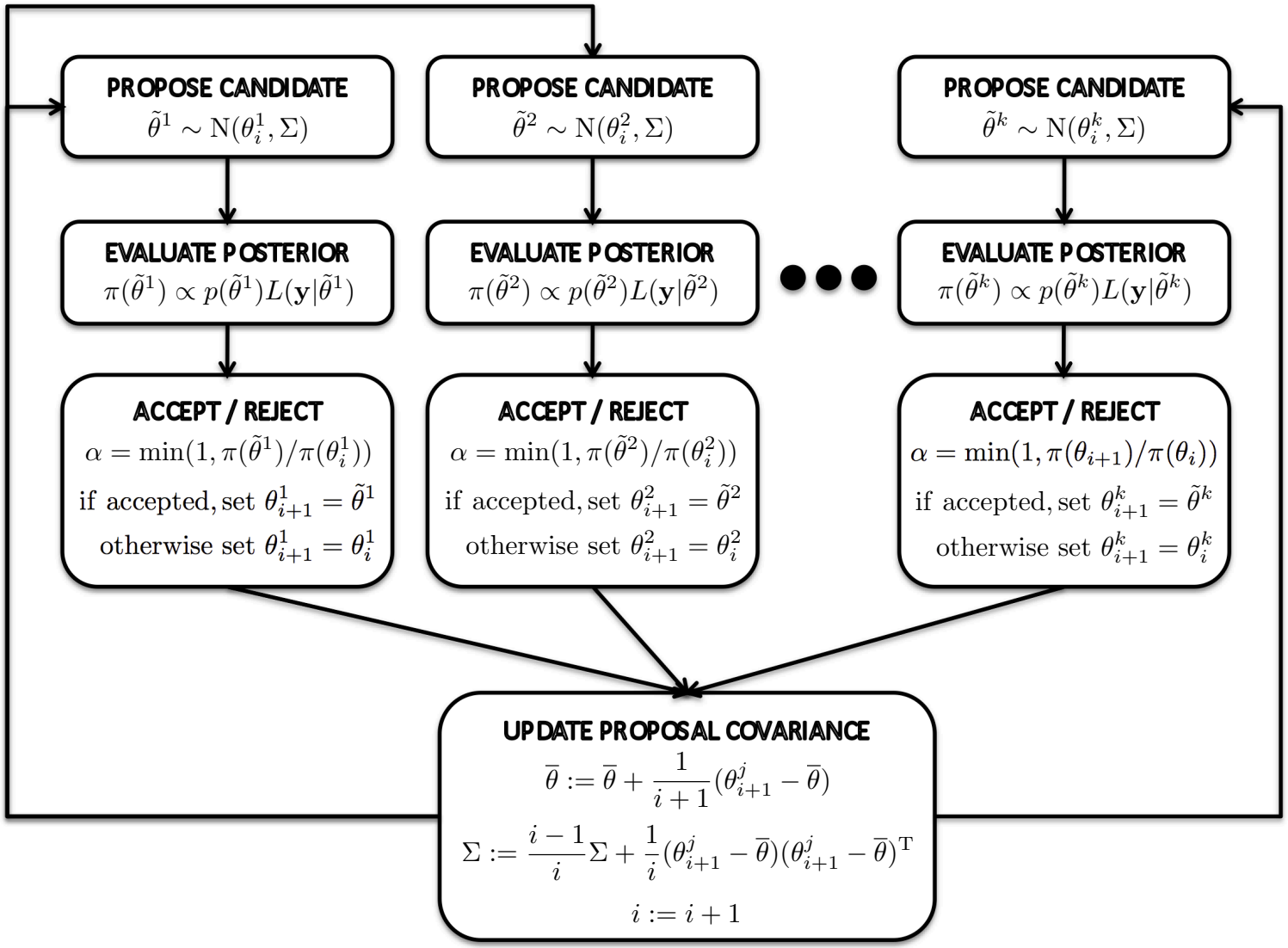
MCMC samples for the 'closure' (subgrid–scale) parameters of climate models (ECHAM5). Technically possible,

- No good initial proposal: use DRAM (adaptive 2 proposals)
- High CPU, short chains: parallel chains
- High CPU: minimize calculations by Early Rejection

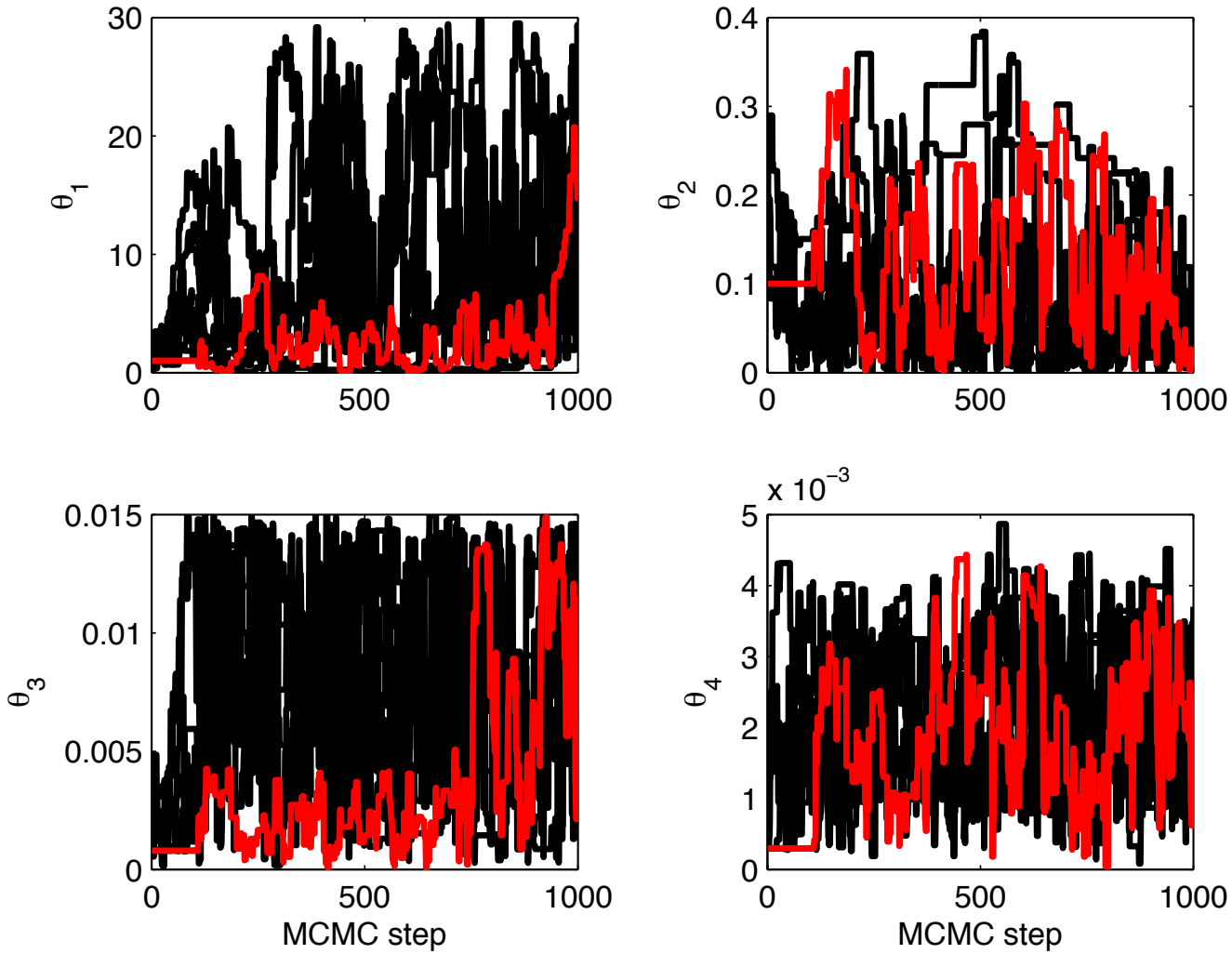
A. Solonen, P. Ollinaho, M. Laine, H. Haario, J. Tamminen, H. Järvinen: *Efficient MCMC for Climate model Parameter Estimation: Parallel Adaptive chains and Early Rejection*. Bayesian Analysis, 7, Number 2, pp 1–22, 2012.

Next: minimize likelihood evaluations by surrogate construction of the parameter posterior. LA-MCMC (Local Approximation MCMC).

Faster MCMC: parallel AM with inter-chain adaptation



Faster MCMC: parallel AM with inter-chain adaptation

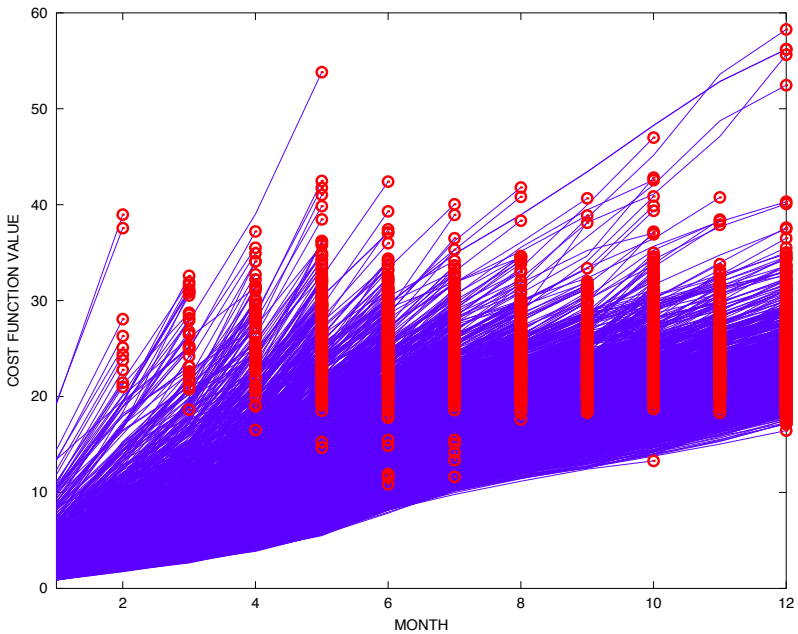


Faster MCMC: early rejection

Evaluate the likelihood in parts and check after each part if the proposed parameter value can be rejected.

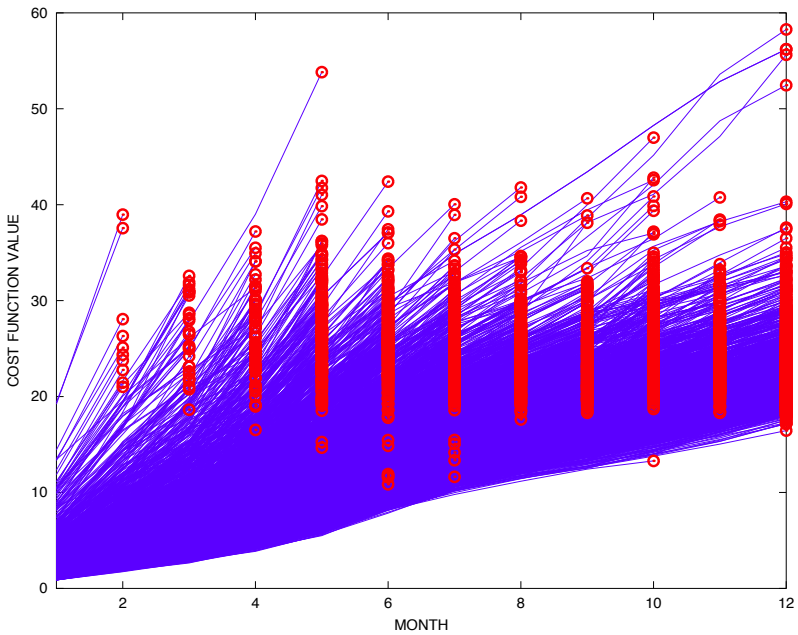
Faster MCMC: early rejection

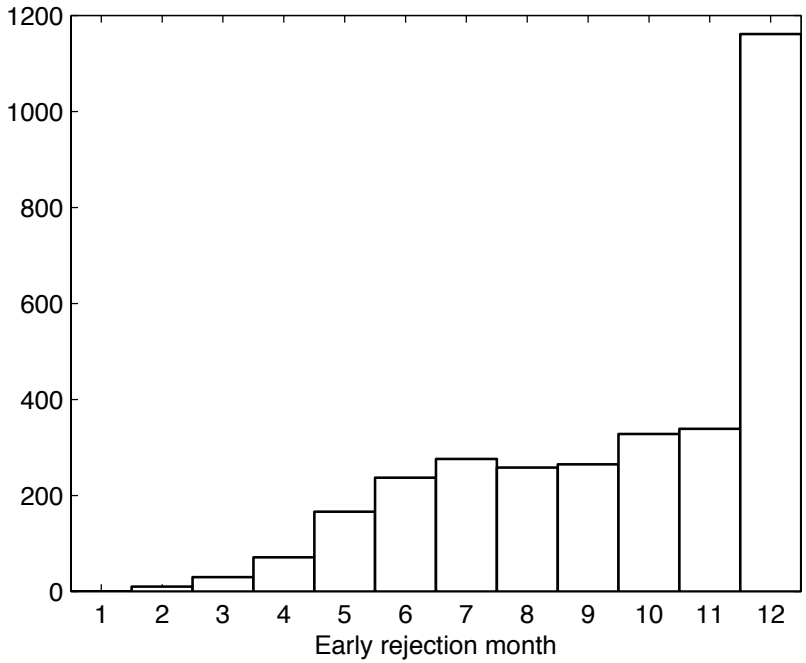
Evaluate the likelihood in parts and check after each part if the proposed parameter value can be rejected.



Faster MCMC: early rejection

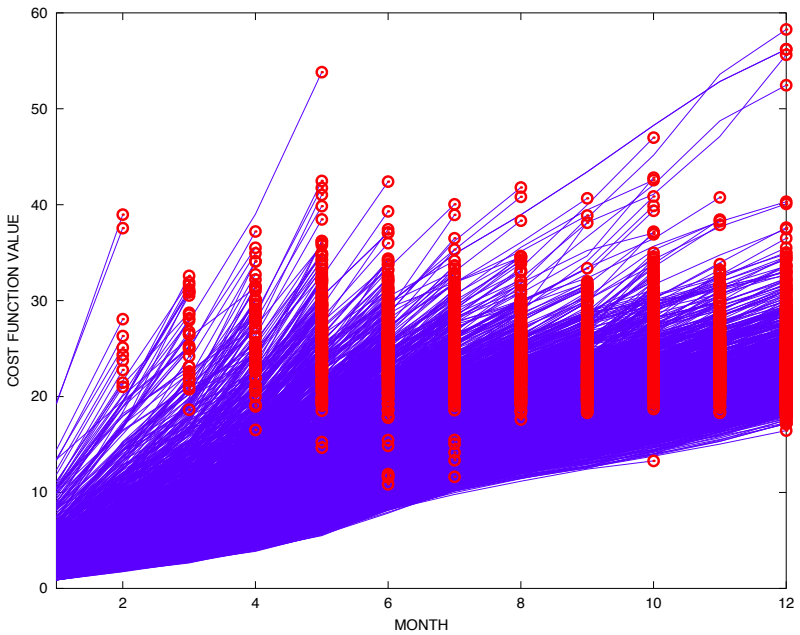
Evaluate the likelihood in parts and check after each part if the proposed parameter value can be rejected.

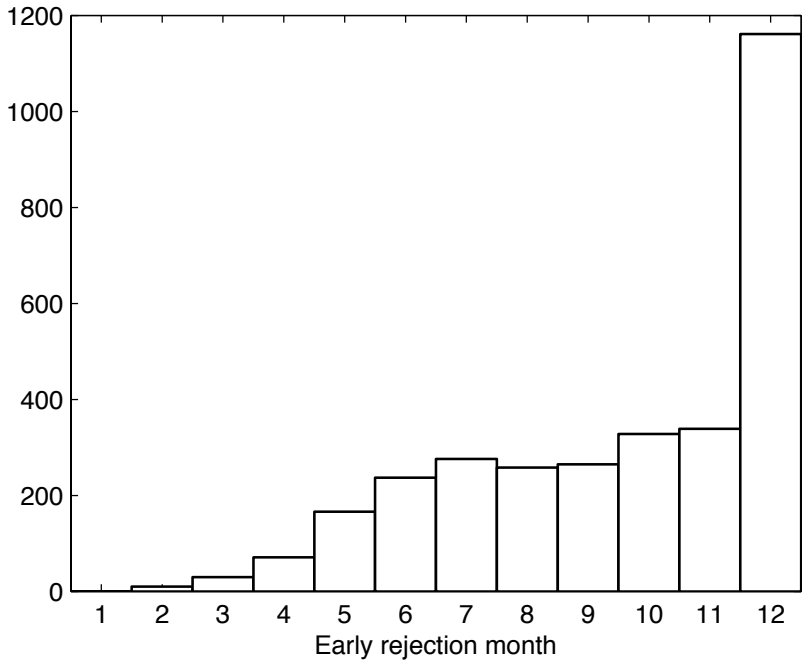




Faster MCMC: early rejection

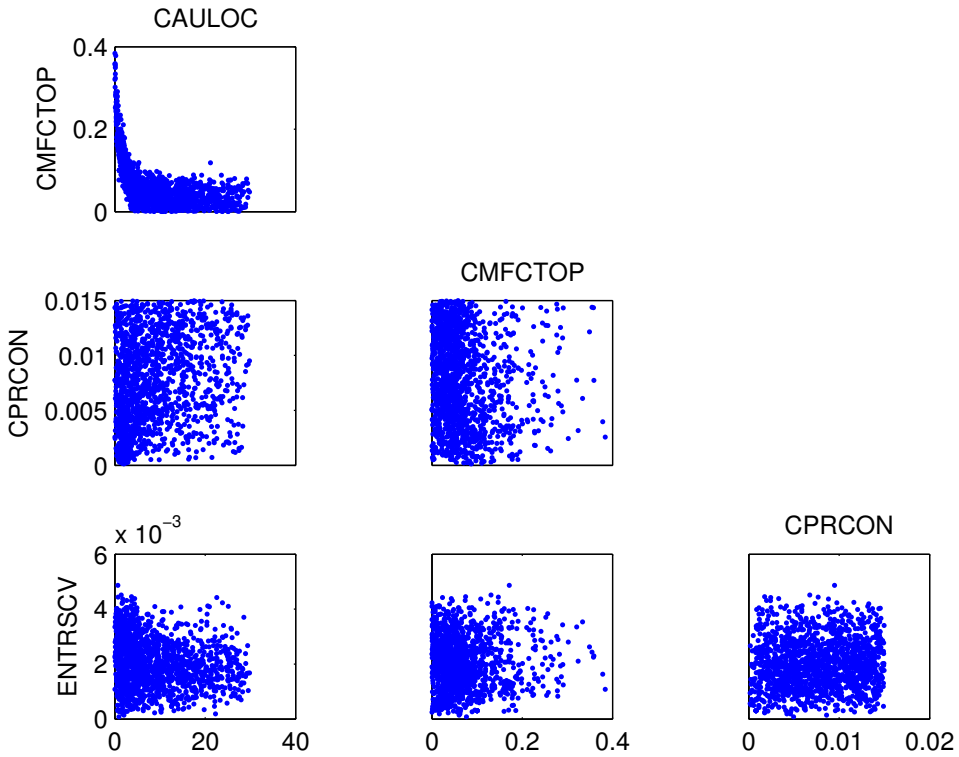
Evaluate the likelihood in parts and check after each part if the proposed parameter value can be rejected.



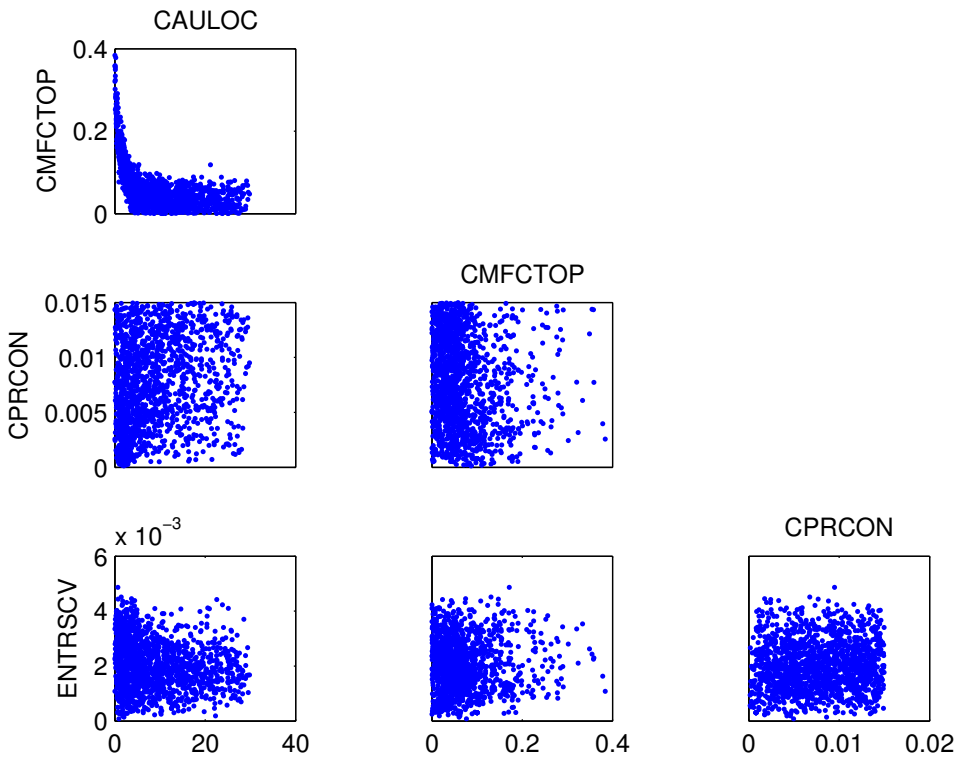


This simple trick can save 10%-80% of CPU, case depending.

Example: climate model MCMC results



Example: climate model MCMC results



- Direct, naive summary statistics (projections) do not identify the parameters, i.e., characterise the simulated trajectories.

Parameter estimation, standard methods: state augmentation, filter likelihood

- Using Kalman filter, integrate out the state space, what remains gives a likelihood for parameters.
- Standard way for linear time series (DLM, Dynamical Linear Models) and SDE (stochastic differential equations) systems. Less standard for chaotic dynamics, but can be implemented with EKF.
- BUT:
  - Filtering not available for large scale systems, due to memory and CPU.
  - Each filter algorithm has built-in 'tuning parameters' (model error covariance, linearization ...). The amount of bias introduced by them ?
  - Only for 'short' assimilation time integrations.

Without filtering: heuristic sampling of parameters using existing ensembles of predictions. 'EPPES' implemented at ECMWF.

Chaotic dynamics: likelihood by attractor

Example: 3D Lorenz, Unpredictable Data. Initial part of a time series:

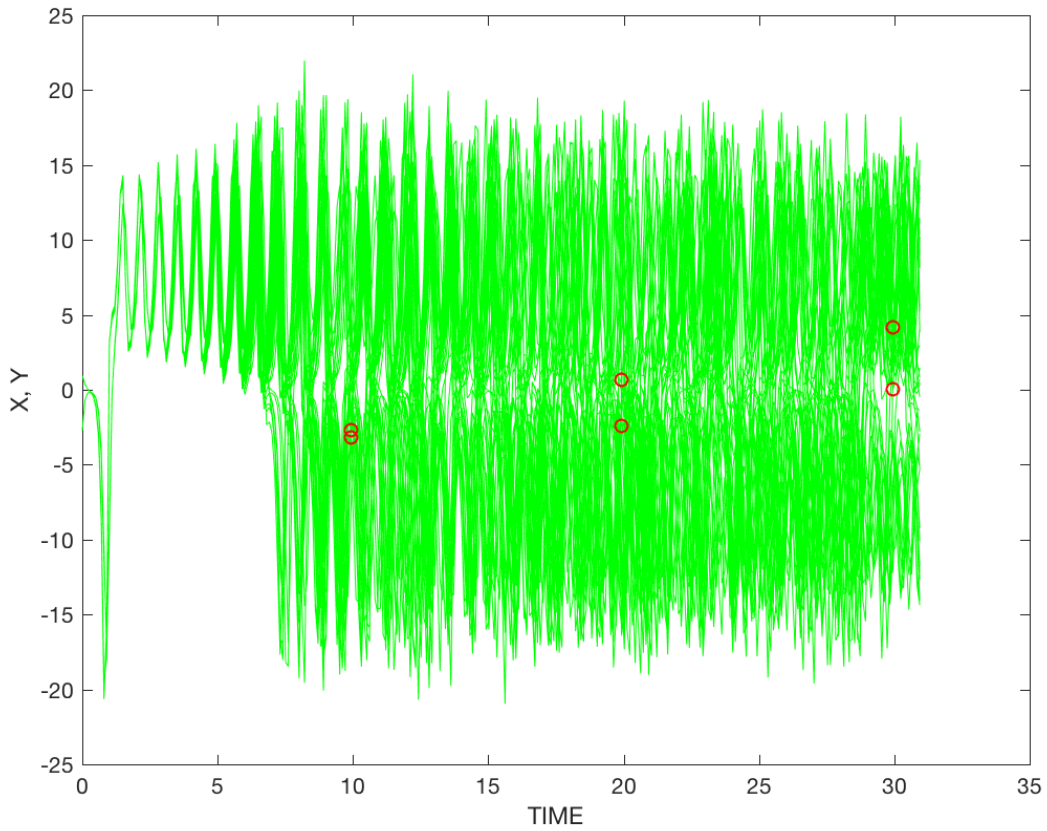


Figure: Observation samples (the red circles) for 3D Lorenz.

Likelihood based on fractal concepts

Fractal dimensions of chaotic attractors, such as the Hausdorff dimension or box-counting, approximate the **internal properties** of the underlying attractor via numerically simulated trajectories. How to employ them to **define a distance between chaotic trajectories**?

We want to separate the **model variability due to initial values etc, but with fixed model parameters** from that **due to different model parameters**.

Key idea: interprete the **time-varying, chaotic** trajectories as samples from a **fixed** attractor.

Idea of Likelihood

Simulations of a model give samples from the underlying 'strange' attractor.

Create a training set of simulations – or one long enough time series – to characterize statistically the variability of the points, to define a likelihood for the 'internal' variability.

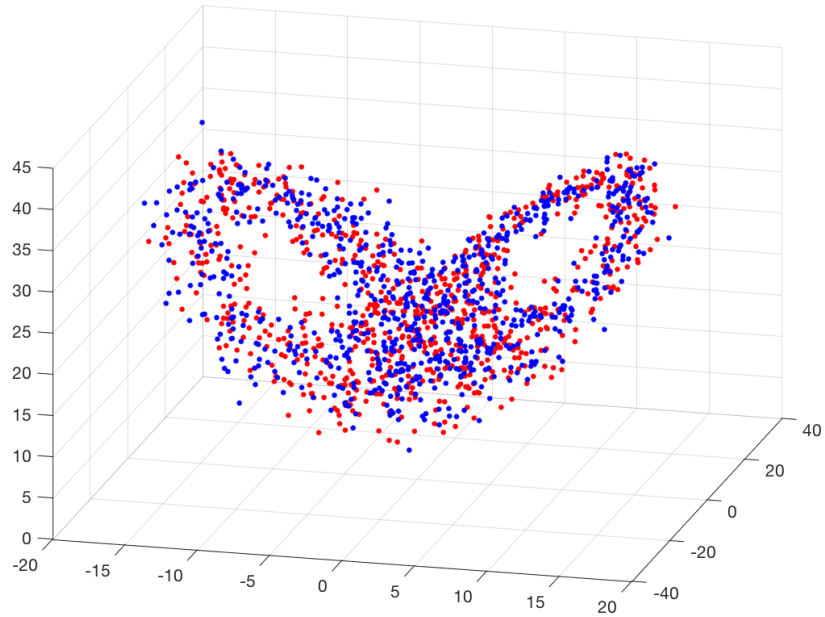


Figure: Two sets of samples, with given number (N=800) of points

Correlation dimension for fractal sets

Denote by $s_i, i = 1, 2, ..., N$ points of a trajectory vector $s \in R^n$, evaluated at time points t_i. For $R > 0$ set

$$C(R, N) = 1/N^2 \sum_{i,j} \#(||s_i - s_j|| < R)$$

and define then the correlation integral as the limit $C(R) = \lim_{N\to\infty} C(R, N)$. So we take the total number of points closer than R, normalize by the number of pairs N^2 and take the limit. Not that for each N we have $1/N \leq C(R, N) \leq 1$.

If ν is the dimension of the trajectory, we should have

$$C(R) \sim R^\nu$$

and the Correlation Dimension ν is defined as the limit

$$\nu = \lim_{R\to 0} \log C(R)/\log(R).$$

Distance via a generalized correlation sum

- Fix a radius R_0, large enough for each ball $B(s_i, R)$ to contain all the points $s_j, j \neq i$
- Select smaller radii by $R_k = b^{-k} R_0$, with $k = 1, 2, ..., M$. Select M and the base b (e.g., $M = 10$, $b = 2$).

The generalized correlation vector $y = (y_k), k = 1, ..., M$, between trajectories $s = s(\theta, x)$ and $\tilde{s} = s(\tilde{\theta}, \tilde{x})$ is given by

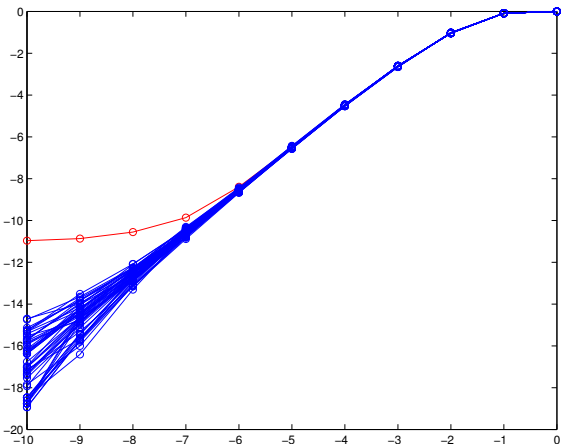
$$y_k = C(R_k, N, \theta, x, \tilde{\theta}, \tilde{x}) = 1/N^2 \sum_{i,j} \#(||s_i - \tilde{s}_j|| < R_k), \qquad (1)$$

where $\theta, \tilde{\theta}$ denote the respective model parameters and $x, \tilde{x}$ the initial values. For $\tilde{\theta} = \theta, \tilde{x} = x$ the formula reduced to the original definition of the correlation sum.

Likelihood by Correlation Vector

Characterize the 'within variability' of a fixed chaotic system

- Create an ensemble of point sets from the attractor (subsamples of a time series, or simulated values $s = s(\theta_0, x)$ if θ_0 known).
- Compute the distance matrix between (all) different trajectory pairs, to get the values y_k .
- The stochastic vector $(y_k), k = 1, ...M$ (the empirical CDF of distances) turns out to be **Gaussian** (by CLT, Donsker's theorem, etc)



Likelihood by Correlation Vector

We treat the above vectors $y = C(R_k, N, \theta_0, x, \theta_0, \tilde{x}), k = 1, ..., M$ as 'measurements' of the variability of a chaotic trajectory with a given fixed model parameter. Construct the respective likelihood:

- Obtain $y = C(R, N, \theta_0, x, \theta_0, \tilde{x})$ from repeated simulations (or, get them from a long enough empirical time series).
- Create the empirical likelihood function: compute **mean and covariance.**

For any other parameter θ and trajectory $s(\theta)$ compute the distance matrix from the reference trajectory, and the respective $C(R_k, N, \theta, x, \tilde{\theta}, \tilde{x})$, to evaluate the likelihood for θ.

HH, Leonid Kalachev, Janne Hakkarainen *Generalized Correlation integral vectors: A distance concept for chaotic dynamical systems*. Chaos, 25, 2015.

Example: Likelihood for 3D Lorenz

- Test the Gaussianity of values $y_k = C(R_k, N, \theta_0, x, \theta_0, \tilde{x})$, by the usual χ^2 test: calculate the mean value μ_0 and covariance matrix Σ_0 of the training set.
- The statistics of the expression $(\mu_0 - y)\Sigma_0^{-1}(\mu_0 - y)$ should obey the χ^2_M distribution for a Gaussian y,

$$(\mu_0 - y)\Sigma_0^{-1}(\mu_0 - y) \sim \chi^2_M \tag{2}$$

Example: Likelihood for 3D Lorenz

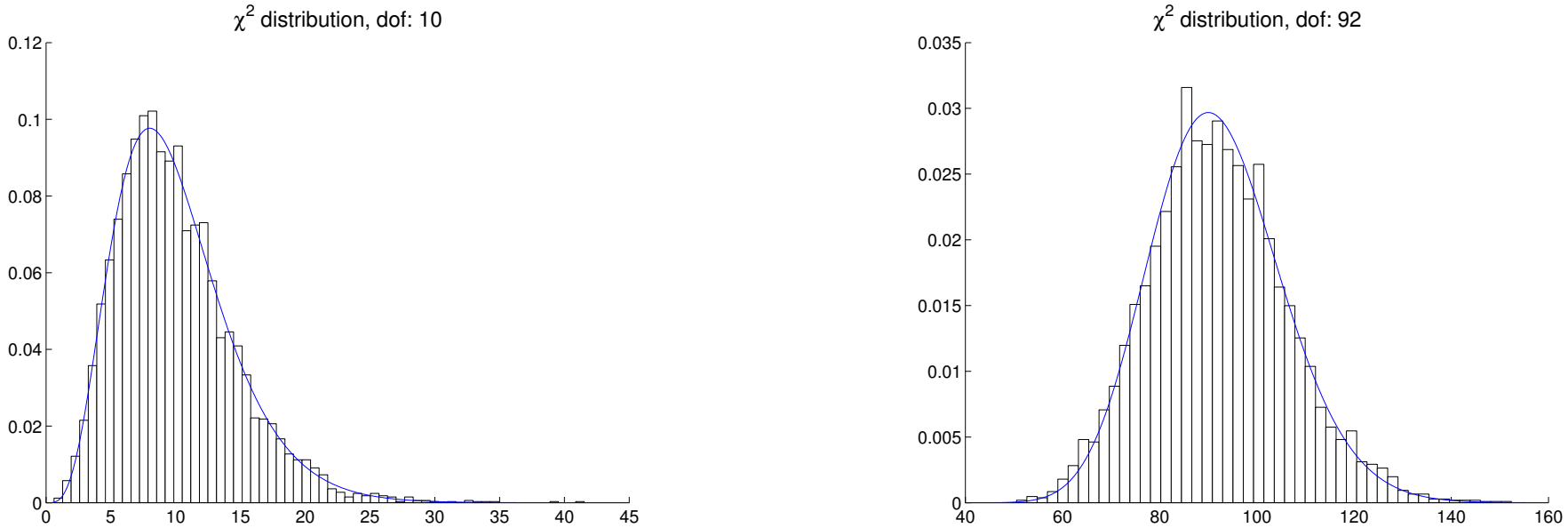


Figure: **Normality check** of the correlation integral vector by the χ^2 test for the Lorenz 63 system. Left: with 10 radius values used. Right: with 92 radius values

Inference as a pseudo-marginal MCMC algorithm

Due to chaoticity and randomised x the likelihood is non-deterministic. Sampling can be interpreted as from the joint distribution of the initial values and model parameters.

Denote the likelihood function of y, evaluated for an arbitrary θ by $T_{\theta_0}(\theta, x)$. The target distribution for θ is given as

$$\pi(\theta) = \int T_{\theta_0}(\theta, x)\lambda(x)\mathrm{d}x,$$

where $\lambda(x)$ is the distribution of the initial values x. Here, $T_{\theta_0}(\theta, x)$ is unknown, but an empirical approximation can be created as above.

The method is a bivariate Markov chain, $(\theta_n, T_n)_{n\geq 0}$, where T_n are auxiliary variables that are non-negative, unbiased estimators of the underlying intractable target density $\pi(\theta_n)$: a **pseudo-marginal algorithm**.

Example: Chua7

$$
\begin{cases}
\dot{X} &= \alpha \cdot (Y - h); \\
\dot{Y} &= X - Y + Z; \\
\dot{Z} &= -\beta \cdot Y; \\
h &= m_7 \cdot X + 0.5 \\
&\quad \times \ ((m_0 - m_1) \cdot (|X + c_1| - |X - c_1|) \\
&\quad + \ (m_1 - m_2) \cdot (|X + c_2| - |X - c_2|) \\
&\quad + \ (m_2 - m_3) \cdot (|X + c_3| - |X - c_3|) \\
&\quad + \ (m_3 - m_4) \cdot (|X + c_4| - |X - c_4|) \\
&\quad + \ (m_4 - m_5) \cdot (|X + c_5| - |X - c_5|) \\
&\quad + \ (m_5 - m_6) \cdot (|X + c_6| - |X - c_6|) \\
&\quad + \ (m_6 - m_7) \cdot (|X + c_7| - |X - c_7|)) .
\end{cases}
$$

$(\alpha, \beta, m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, c_1, c_2, c_3, c_4, c_5, c_6, c_7) =$
$(14, 20, 0.9/7, -3/7, 0.5, -0.3429, 0.36, -0.24, 0.36, -0.24,$
$-0.3429, 1, 2.15, 6.2, 9, 14, 25)$, $(X_0, Y_0, Z_0) = (0.1, 0.1, 0.1)$;
$ft = 240000$, $no = 16000$, $ns = 4000$,
$M = M_{vel} = 10$, $R_0 = 2.9$, $b = 1.88$, $R_{0,vel} = 2.49$, $b_{vel} = 1.82$.

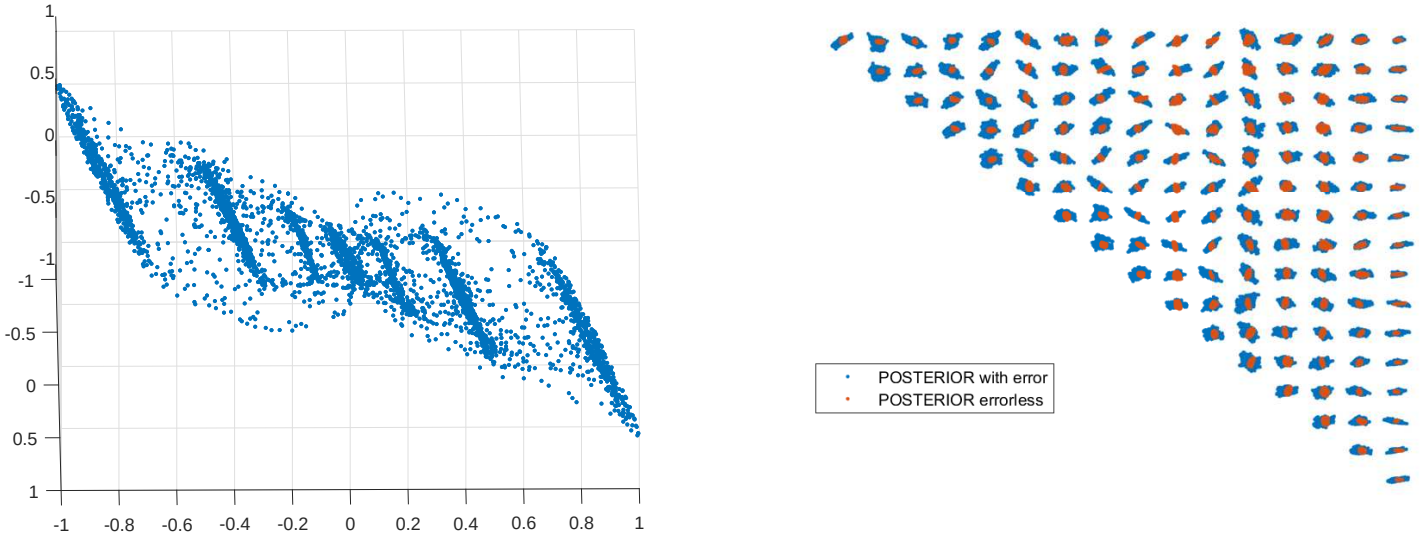


Figure: Chua 7 model: measurements and 2D posterior marginal distributions.

- X, Y, Z measured, without and with 1% relative error. Results with 2% and 6% relative accuracies.
- Note:**Extended state vector** used, $\tilde{s} = (s, ds/dt)$.

High dimensional, high CPU systems

So far 3D chaotic systems. But the challenges are

- High state space dimension: need for effective, parallel distance calculations
- High model simulation CPU: need for efficient parallel calculations for
  - model simulation numerics
  - ensemble computations
  - parallel MCMC sampling
- Greatly benefit from the LA posterior surrogate sampling approach

LA-MCMC

- Evaluating the likelihood at every step can became prohibitively expensive when the dimensionality and the complexity of the system increase;
- The key idea in the LA-MCMC is to build a cheap local polynomial approximation of the likelihood that will converge to the true one as the chain proceeds;
- The local polynomial approximation is built on a subset (neighborhood) of the set of full likelihood evaluations defined as *support points*;
- To guarantee the convergence of the polynomial approximation to the "true" likelihood, the number of support points must increase as the chain proceeds;
- The refinement strategy, i.e. when and where to make a new likelihood evaluation and add it to the support points, is the key to obtain the optimal efficiency of the algorithm.

LA-MCMC

In our case, the likelihood is stochastic

The refinement strategies for deterministic cost functions might lead to over refinement when applied to stochastic likelihoods as CIL.

Davis et al 2018, proposes an optimal strategy for stochastic likelihood (ongoing joint work with Y. Marzouk et al, MIT)

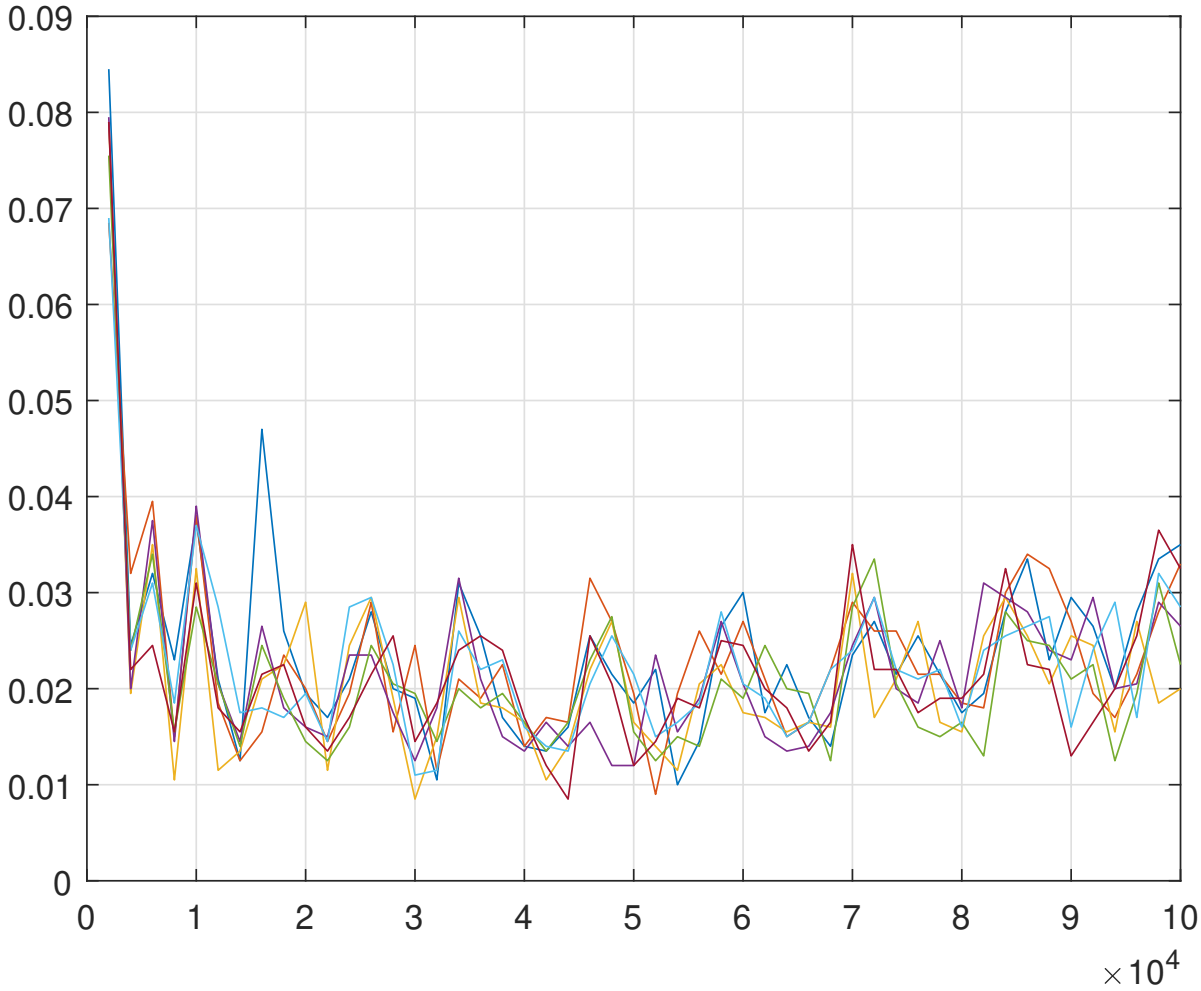


Figure: Ratio between the # of expensive cost function evaluations every 2000 elements of the chain (LA-MCMC/AM).

Kuramoto-Shivashinsky

Find the parameters that produce the 'same' attractor approximation for the system

$$u_t = -uu_x - \eta u_{xx} - \gamma u_{xxxx}. \tag{3}$$

An example result:

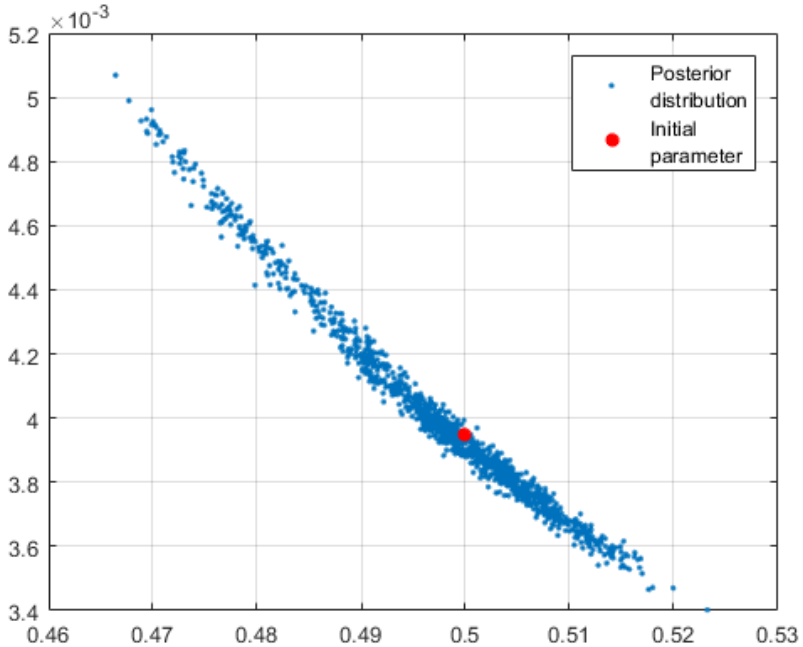


Figure: The posterior of the KS parameters

Kuramoto-Shivashinsky

The solution can be represented by Fourier series:

$$u(x,t) = \sum_{j=0}^{+\infty} \left[A_j(t) sin\left(\frac{2\pi}{L} jx\right) + B_j(t) cos\left(\frac{2\pi}{L} jx\right) \right]. \tag{4}$$

which leads to

$$\dot{A}_j(t) = \alpha_1 j^2 A(t) + \alpha_2 j^4 A(t) + F_1(A_j(t)), \tag{5}$$

$$\dot{B}_j(t) = \beta_1 j^2 B(t) + \beta_2 j^4 B(t) + F_2(B_j(t)), \tag{6}$$

where $F_1(\cdot)$ and $F_2(\cdot)$ are nonlinear. The right hand sides allow vectorization with respect to coefficient j for efficient simulation of, say, Runge-Kutta iterations. Also, it is possible to run thousands independent ensemble simulations at once.

Here,

- The training set consists of en ensemble of 64 trajectories of 1024 equidistant measurements in the interval $[t_0\ 150000]$ ($t_0 = 500$, to escape initial values)
- One such trajectory would take about 103 seconds.
- But we only need to take representative samples from the attractor, whatever sampling times.
- We take 8 equidistant measurements from an ensemble of 128 members in intervals $[t_i, t_i + 4500]$
- The initial states for ensemble members by 128 different samples from the training set.
- The computational time needed for one representative sample creation is reduced from 103 to 2.5 seconds.

Kuramoto-Shivashinsky

Two same, two different (!) solutions, with parameters inside/outside the sampled posterior

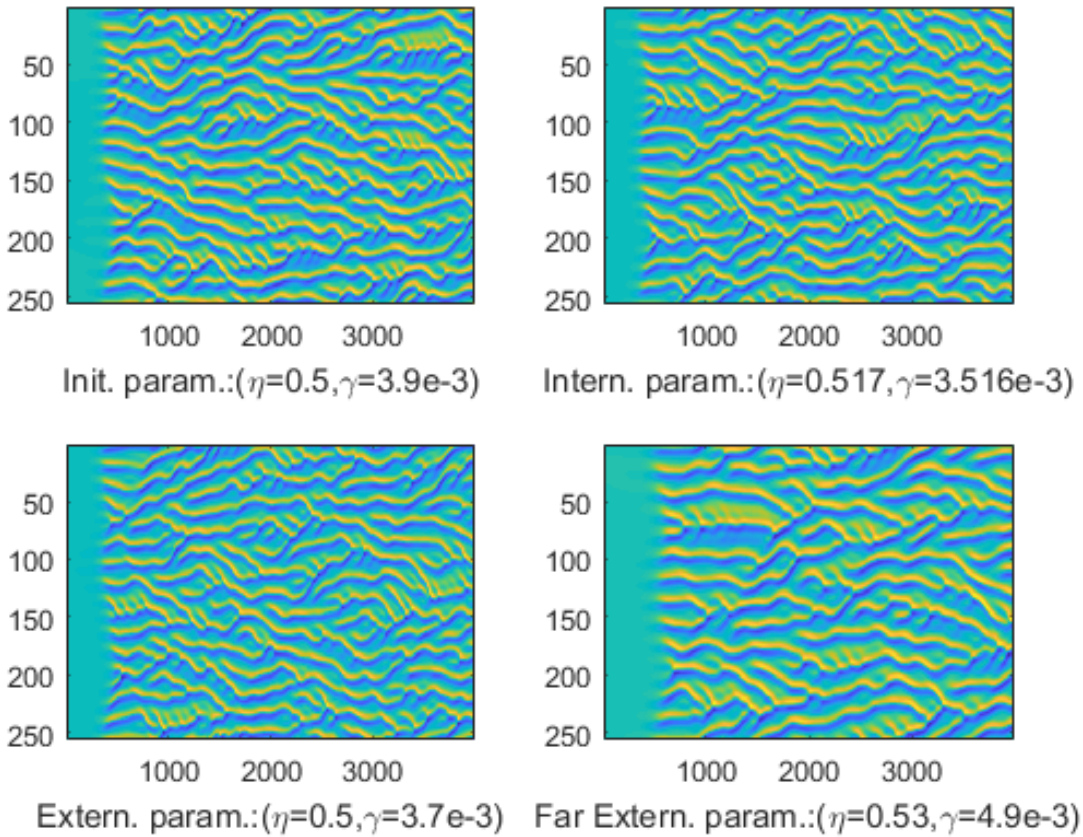


Figure: Top/bottom row: simulations with parameters inside/outside posterior

Conclusion

- The CIL approach provides a likelihood for chaotic dynamics, extensively tested in low dim systems
- Complex, high dimensional chaotic systems a challenge, but possible by
  - Parallel numerics
  - Parallel (ensemble) simulations
  - Parallel chains
  - Surrogate sampling for the parameter posterior
- Next: back to NWP models! (Ensemble runs by the OpenIFS environment, provided by ECMWF and FMI)
- Also,
  - SDE systems, after slight modifications.
  - Standard deterministic systems, after slight modifications
  - Random Turing patterns by (deterministic) reaction-diffusion systems.