THE DIFFUSION KERNEL FILTER

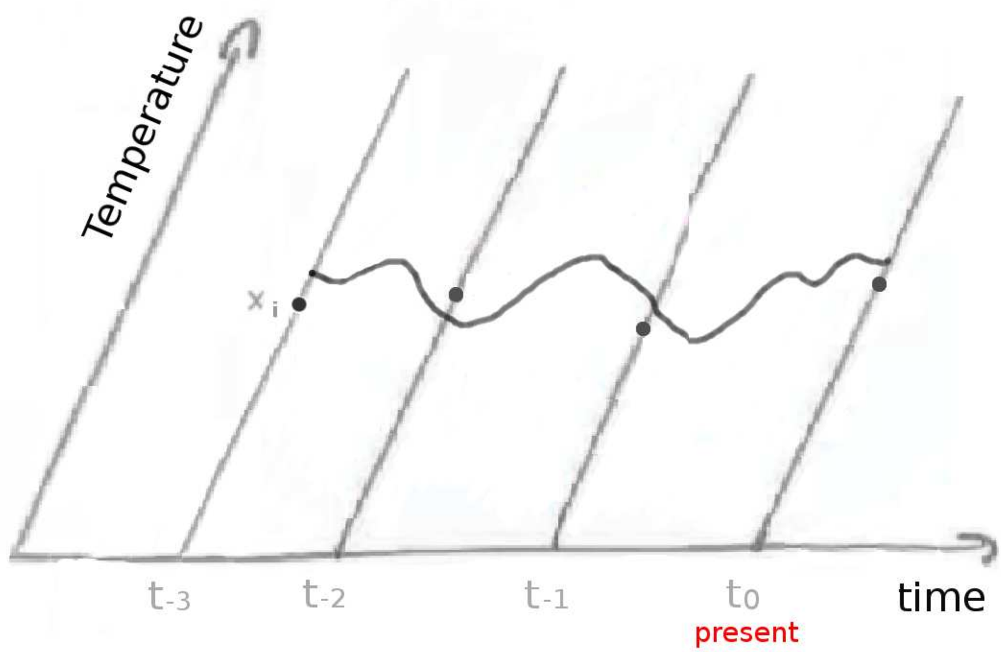
P. Krause

In collaboration with J.M. Restrepo

University of Arizona
Mathematics

Sponsored by NSF

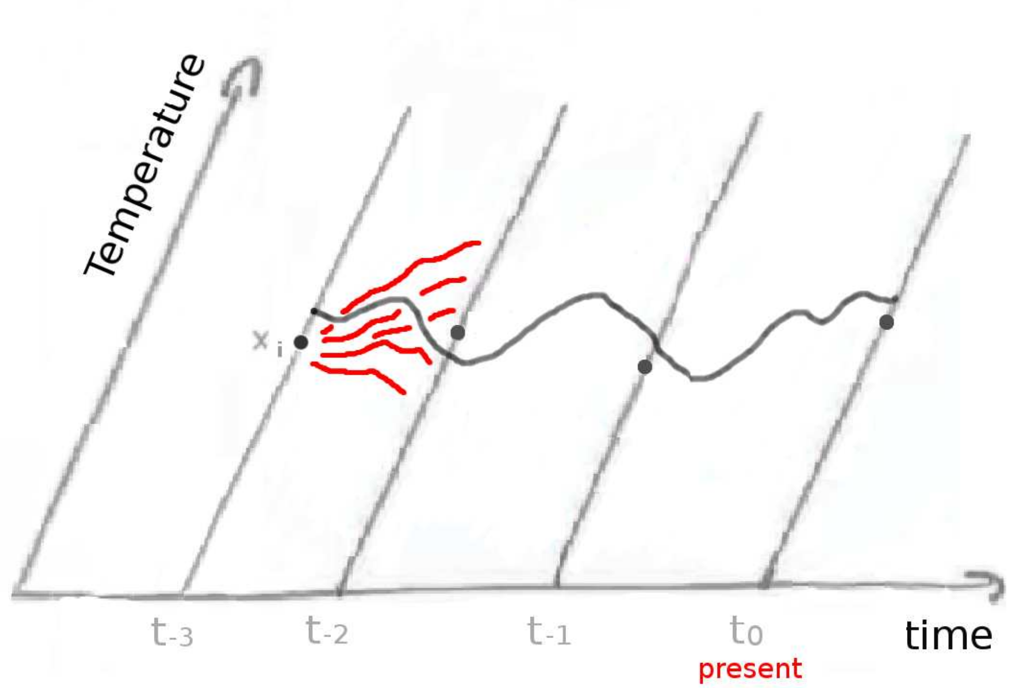
Bootstrap Filter



noisy measurements

Wish to sample $\Phi(x_i, t_0)$ corrected by past data, where Φ is the dynamics of a model $dT = f(T)\,dt + g(t, w)\,dw$ for temperature changes

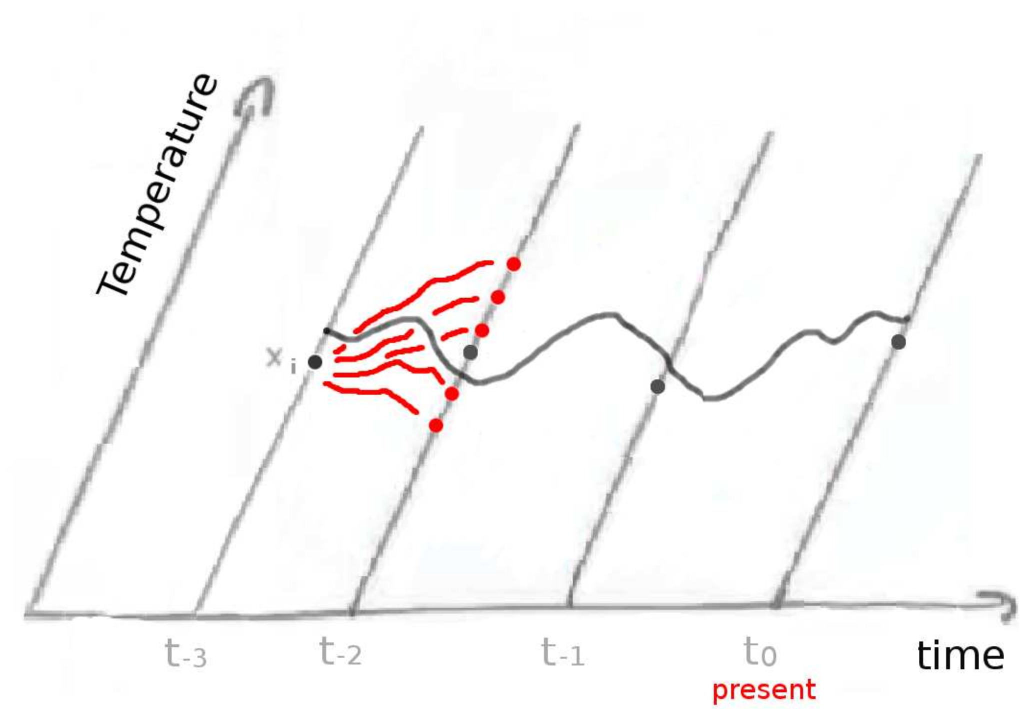
Bootstrap Filter



Prediction step:

$$dT = f(T)\,dt + g(t,w)\,dw$$

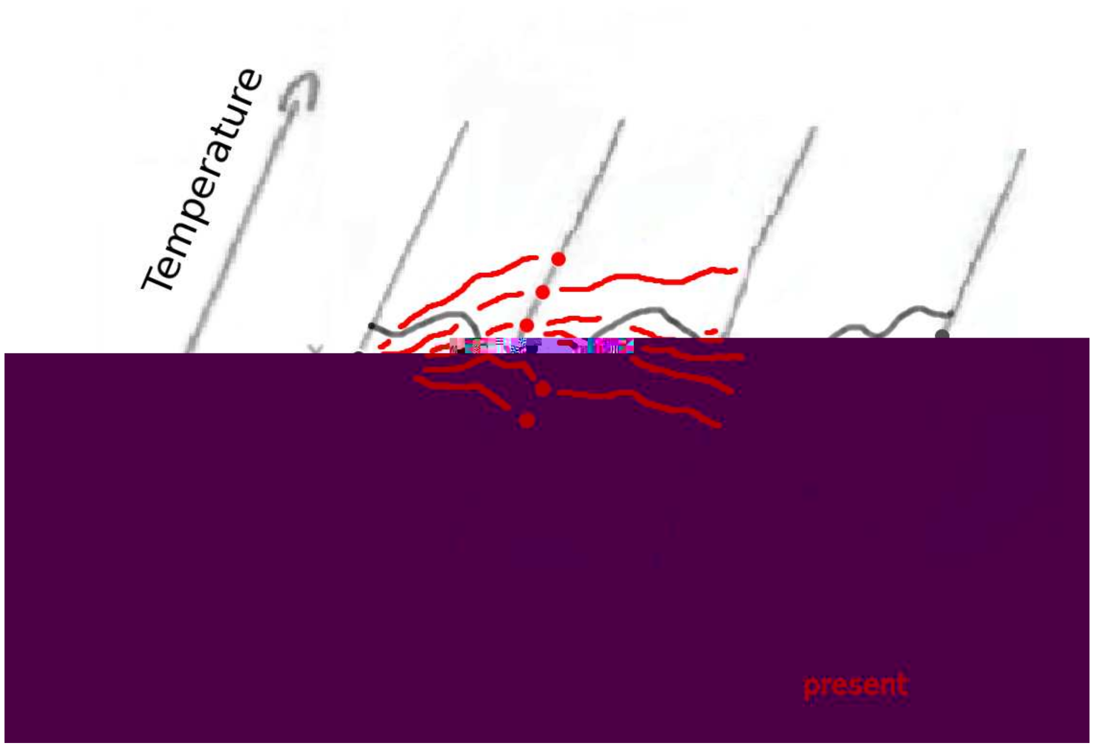
Bootstrap Filter



Filtering step:

weigh samples and define branching

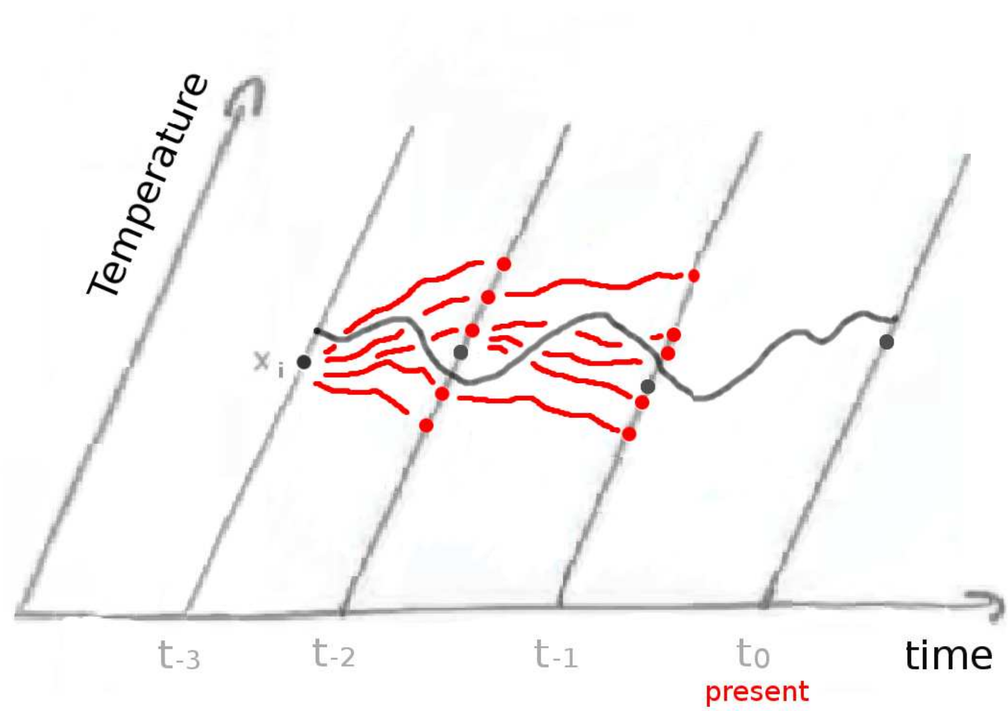
Bootstrap Filter



Prediction step:

$$dT = f(T)\,dt + g(t,w)\,dw$$

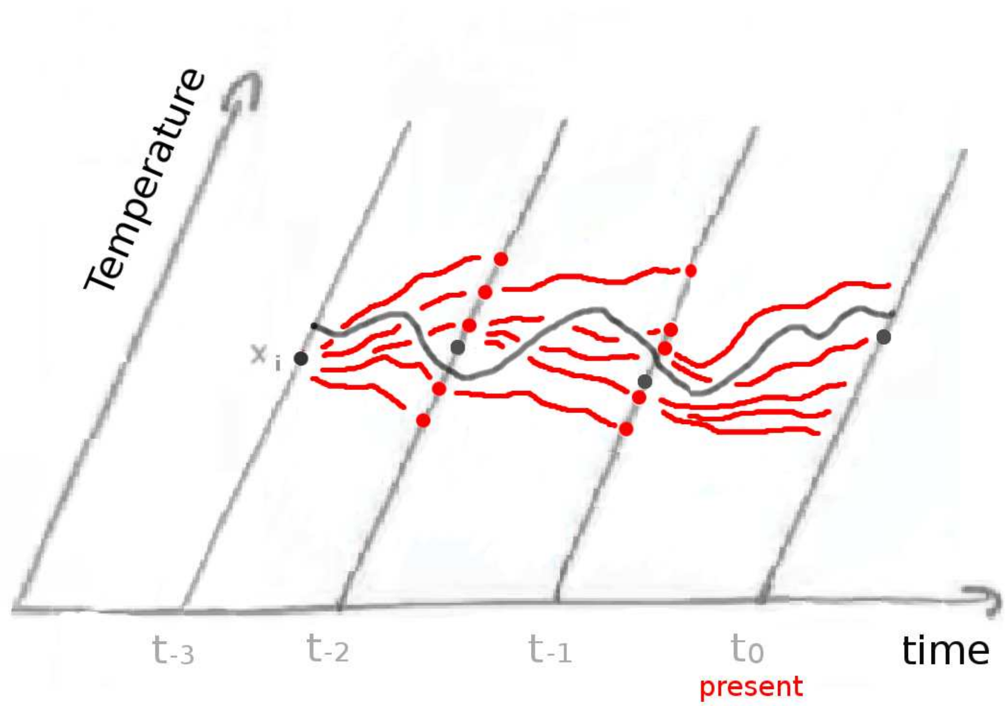
Bootstrap Filter



Filtering step:

weigh samples and define branching

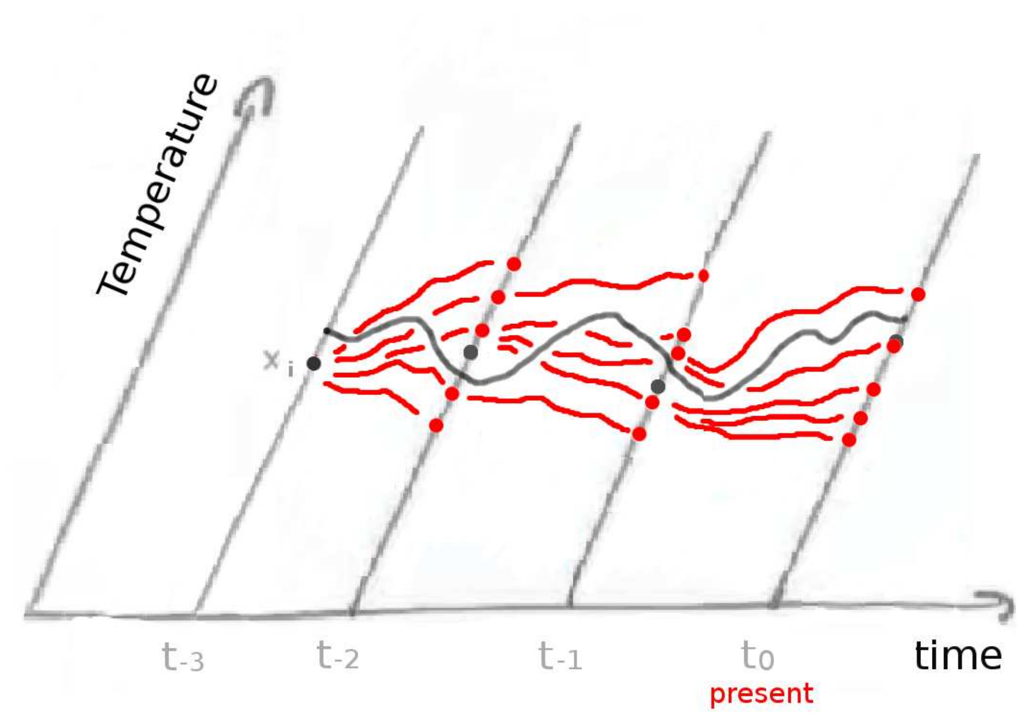
Bootstrap Filter



Prediction step:

$$dT = f(T)\,dt + g(t,w)\,dw$$

Bootstrap Filter



Filtering step:

weigh samples and define branching

Bootstrap Filter

Drawbacks:

(1) HUGE operations count: $O(Id)$

$I :=$ sampling size

$d :=$ model dimension

e.g.: $I = 500,000$ for capturing

first 3 moments of noisy-Lorenz

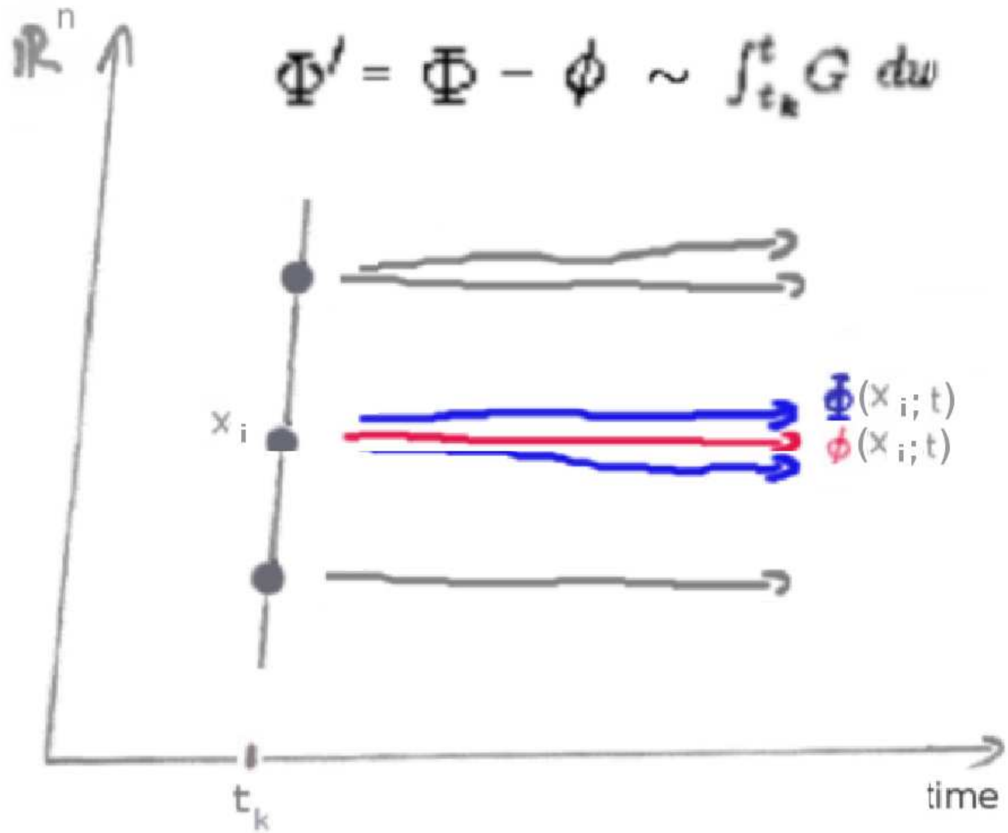
(fast increase with # moments)

(2) Troubles defining prediction

e.g.: average estimates of (p, ρ, T)

won't satisfy $p = R\rho T$ (gas eq.)

Diffusion Kernel Filter (DKF)



Parametrization:

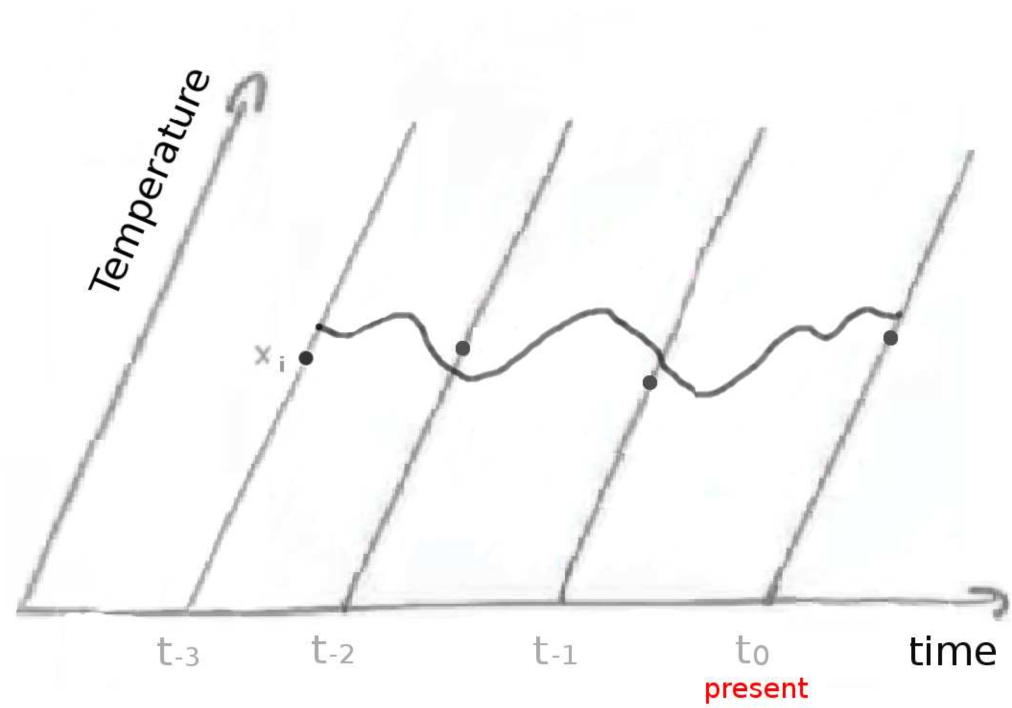
$$\Phi'(x_i, t) = \nabla\phi(x_i, t) \int_{t_k}^{t} g(s, w(s - t_k))dw(s - t_k)$$

in distribution, while small,

where $\Phi' = \Phi - \phi$, Φ stoch. dynamics, ϕ determ. dynamics

$\nabla\phi$ determ. propagator of perturbs., $\int_{t_k}^{t}()$ accumulated noise

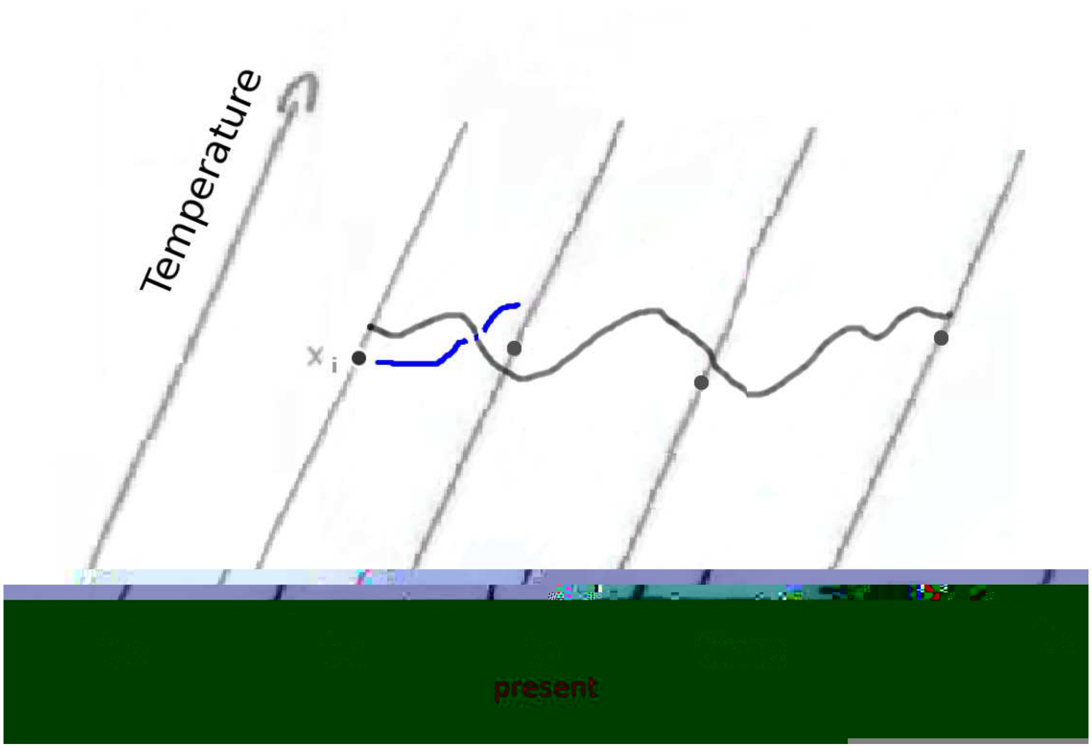
DKF



noisy measurements

Wish to sample $\Phi(x_i, t_0)$ corrected by past data, where Φ is the dynamics of a model $dT = f(T)\,dt + g(t,w)\,dw$ for temperature changes

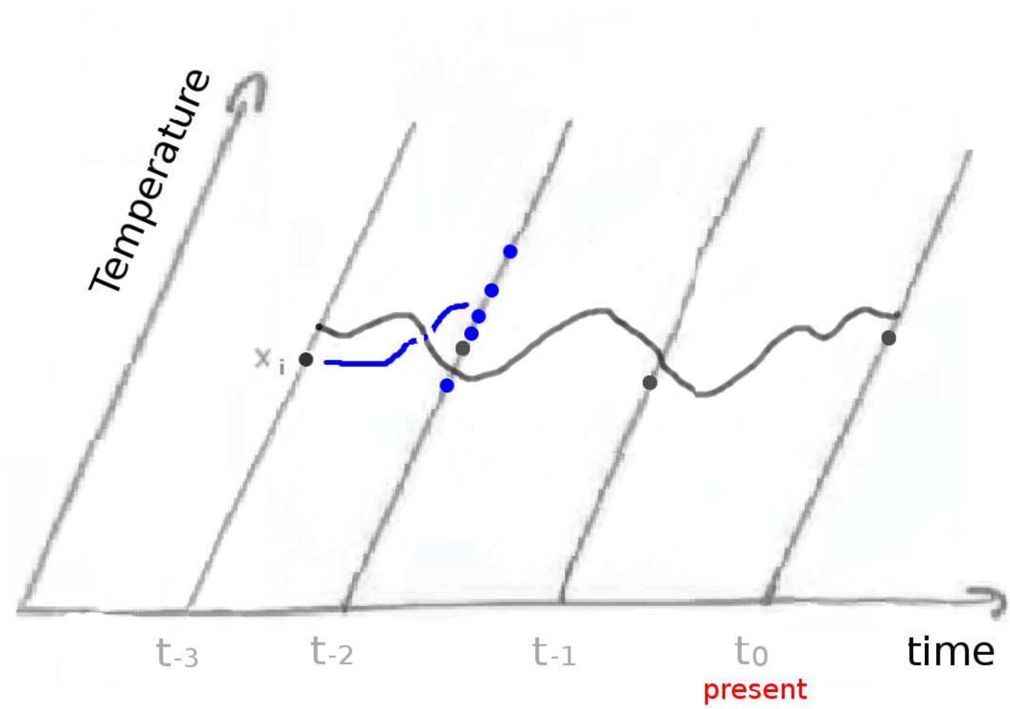
DKF



Prediction step:

$$(1)\ \frac{d}{dt}T = f(T)$$
$$(2)\ \frac{d}{dt}\nabla T = \nabla f(T)\ \nabla T$$

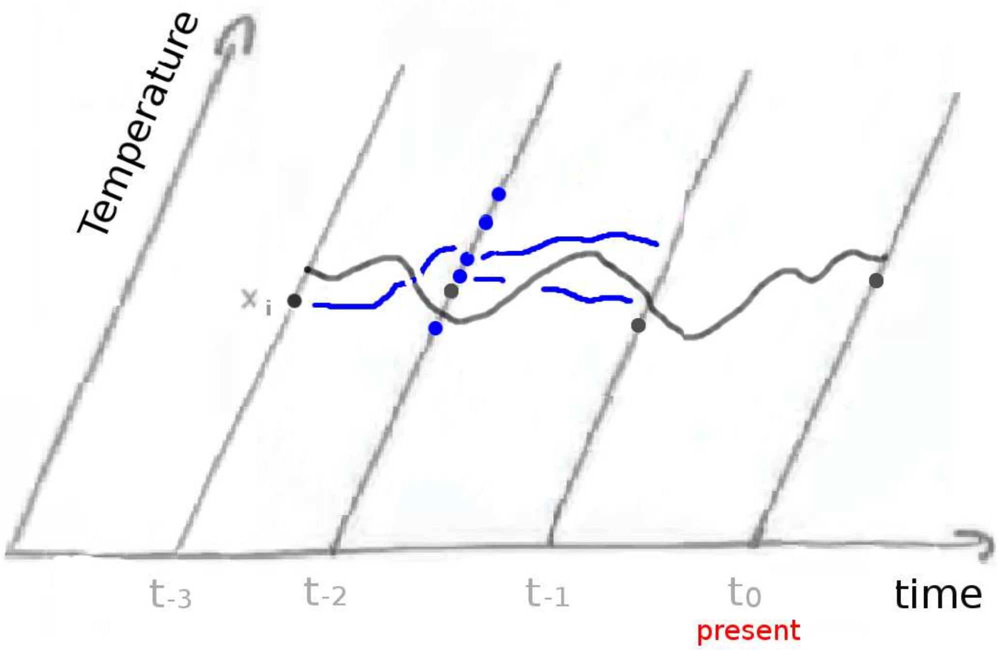
DKF



Filtering step:

(1) draw samples from parametrization
(2) weigh samples and define branching

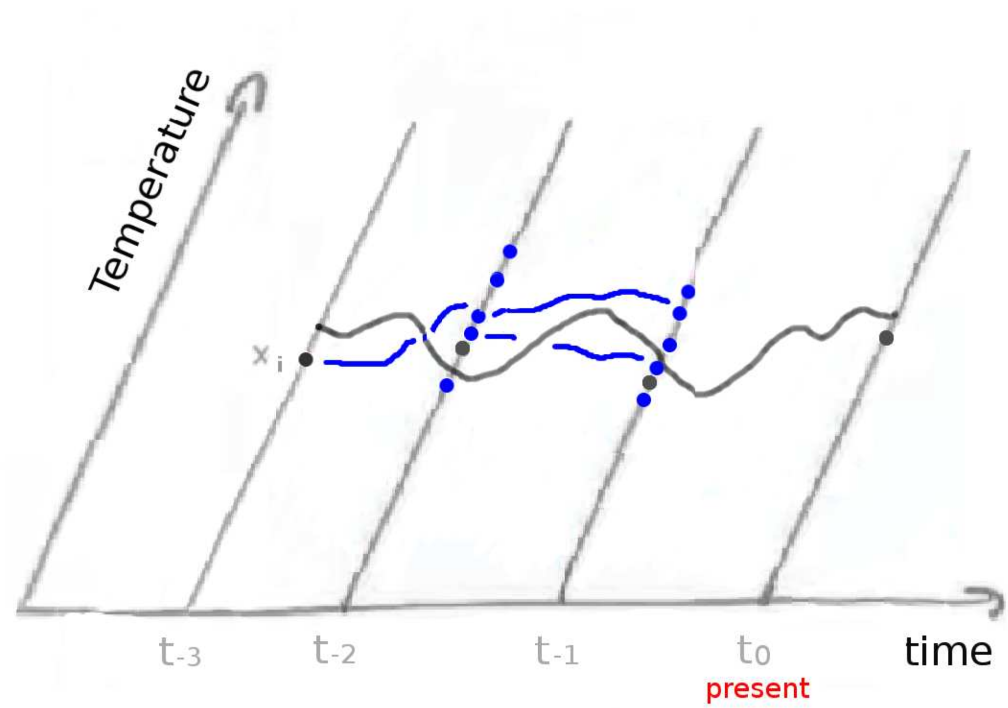
DKF



Prediction step:

$$(1)\ \frac{d}{dt}T = f(T)$$
$$(2)\ \frac{d}{dt}\nabla T = \nabla f(T)\ \nabla T$$

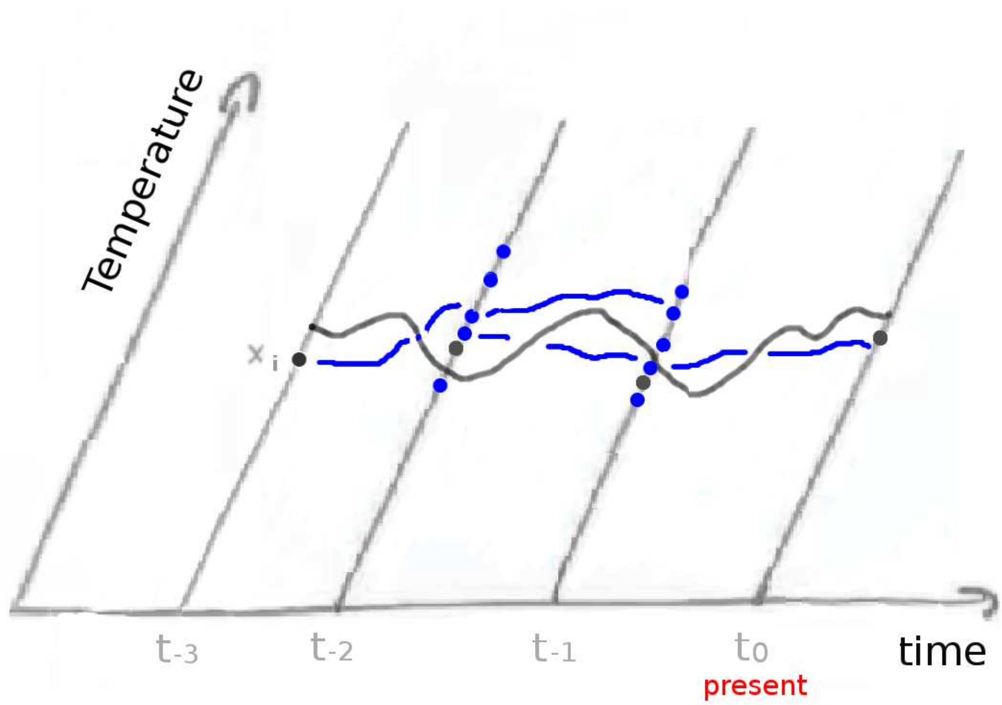
DKF



Filtering step:

(1) draw samples from parametrization
(2) weigh samples and define branching

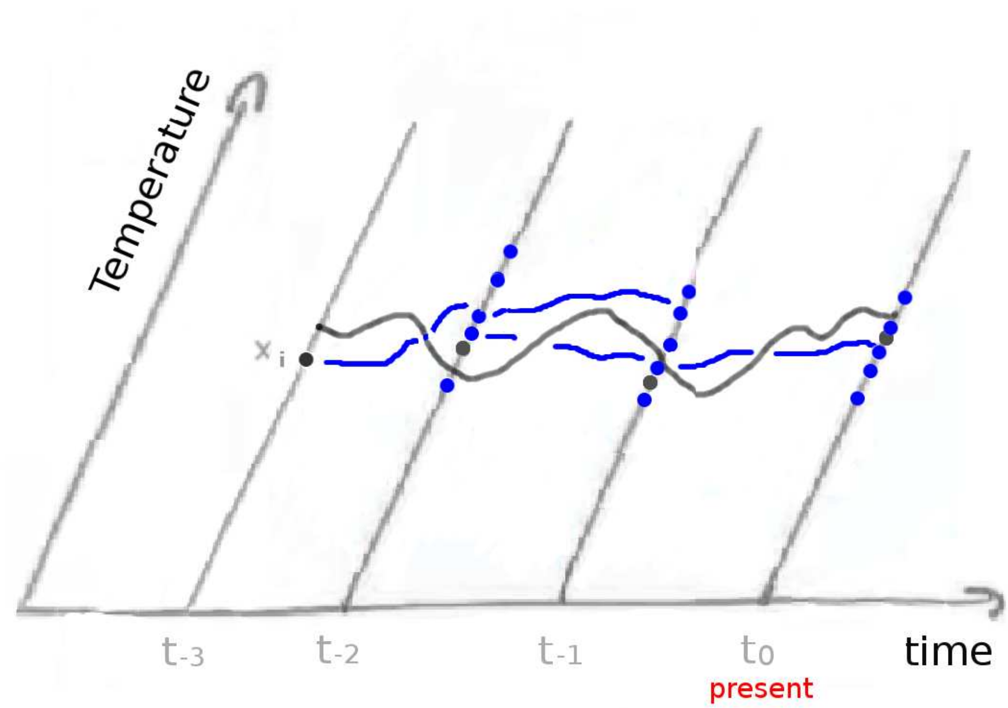
DKF



Prediction step:

$$(1)\ \frac{d}{dt}T = f(T)$$
$$(2)\ \frac{d}{dt}\nabla T = \nabla f(T)\ \nabla T$$

DKF



Filtering step:

(1) draw samples from parametrization
(2) weigh samples and define branching

DKF

Operations count:

$$O(I_k d^2) \text{ with } I_k \leq I$$

where I_k is # branches at time t_k

$$\frac{\text{DKF count}}{\text{Bootstrap count}} = O(\frac{I_k d}{I})$$

Good job! for:

$$d < const\, \frac{I}{I_k}$$

(non-Gaussianity increases I_k)

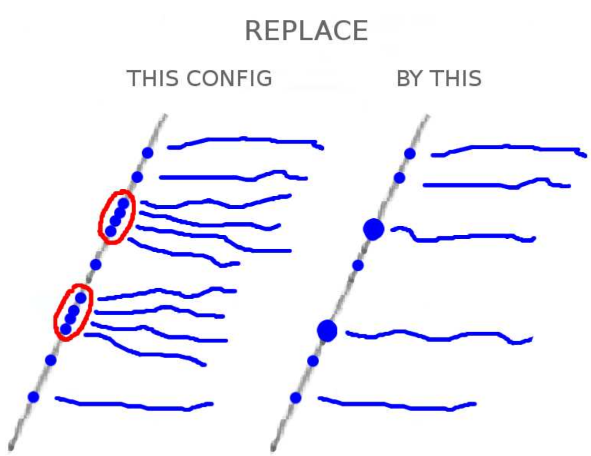
Clustered DKF (cDKF)

Filtering step:

(1) draw samples from parametrization

(2) weigh samples

(3) stick to samples that are cluster representatives: the weights associated with them are taken to be the sum of the weights of their cluster members



(4) define branching from the weighed cluster representatives

cDKF

Operations count:

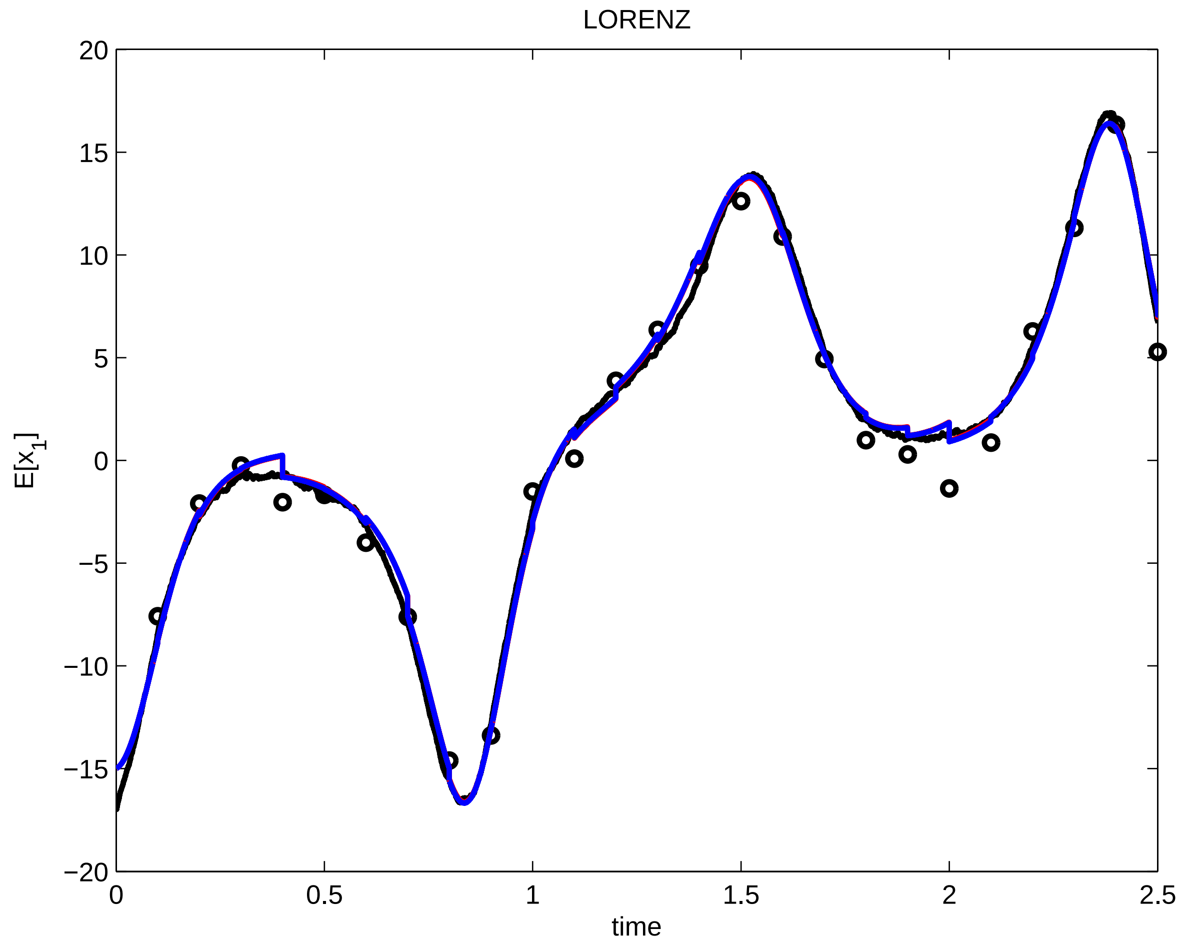
$$O(I_k d^2) \text{ with } I_k \le 10^{-r} I$$

where I_k is # branches at time t_k

$$\frac{\text{cDKF count}}{\text{Bootstrap count}} = O(\frac{I_k d}{I}) \le O(10^{-r} d)$$

Good job! for:

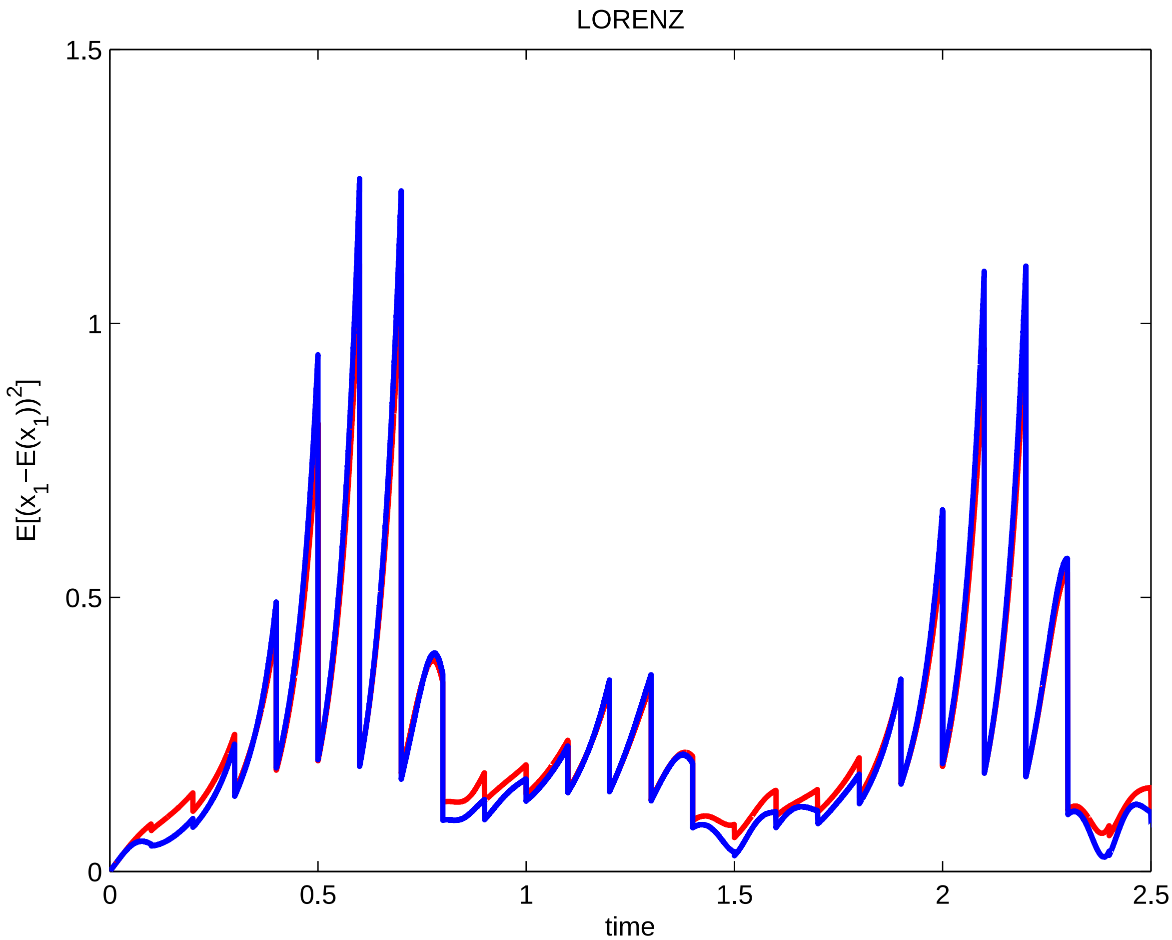
$$d < const\ 10^r$$



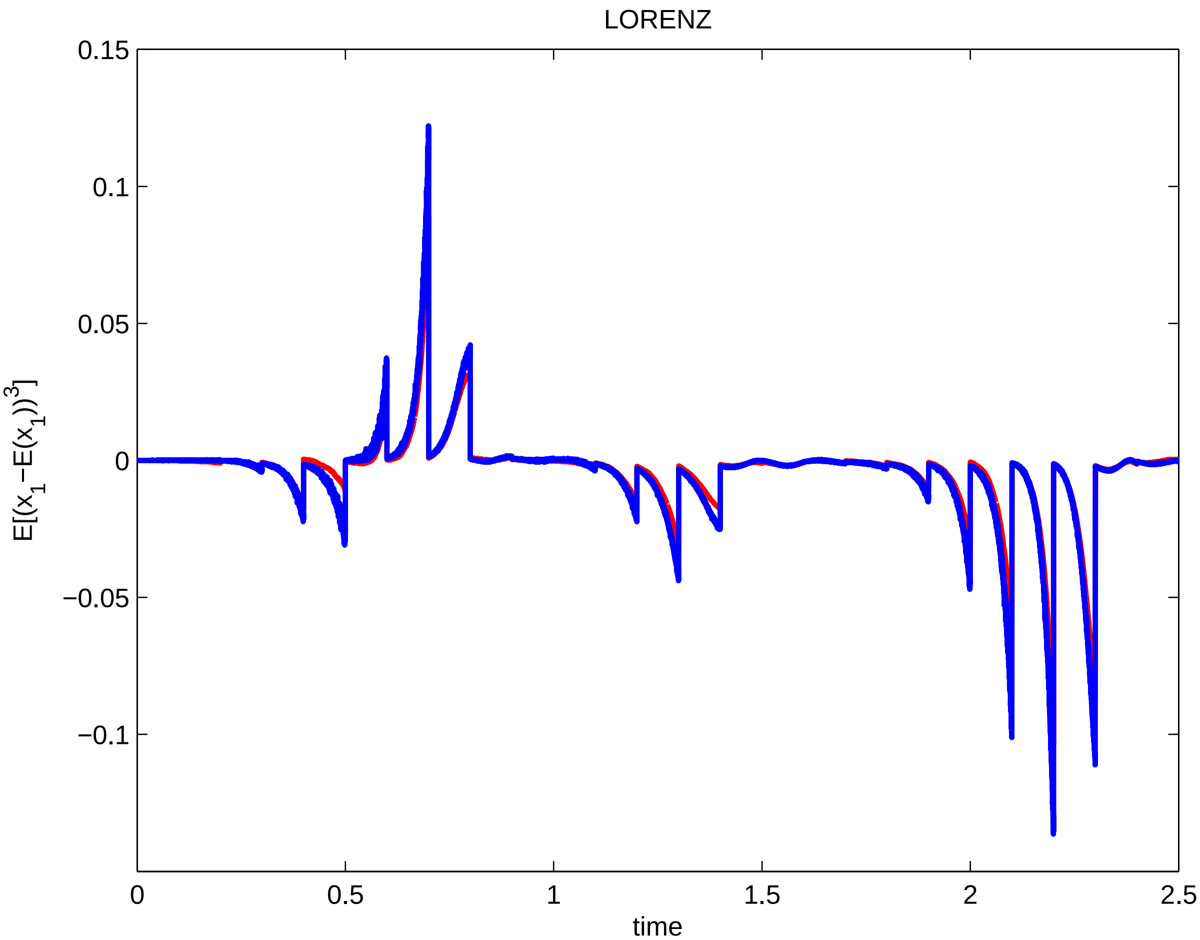
RED = Bootstrap

BLUE = DKF

BLACK = real path



RED = Bootstrap
BLUE = DKF



RED = Bootstrap
BLUE = DKF

Average-entropy prediction

Diffusion Kernel:

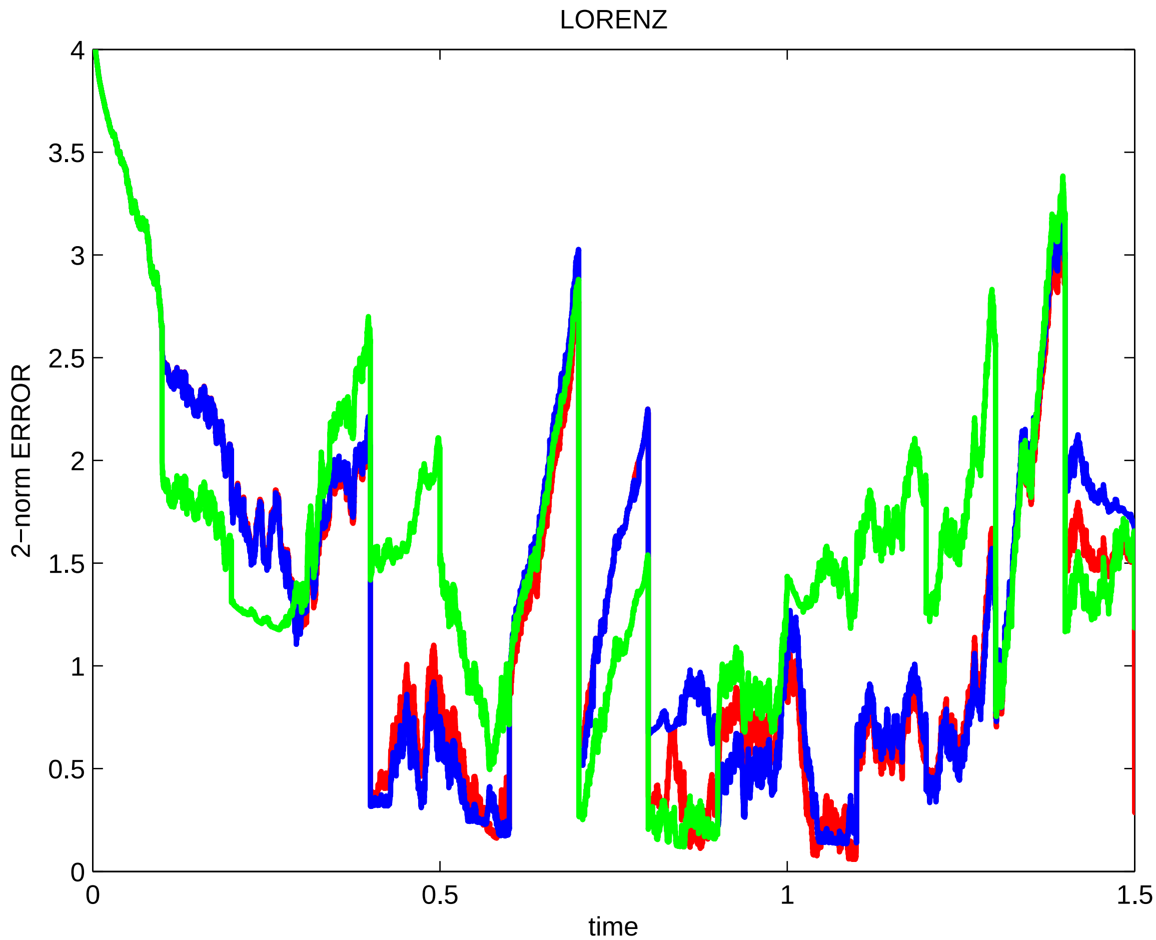
$$G(x_i; t, s) := \nabla\phi(x_i, t)\, g(s, w(s - t_k))$$

Uncertainty norm:

entropy $\sim ||\mathrm{Cov}(\Phi'(x_i; t))||_\infty \leq ||G(x_i; t, \,\cdot\,)||^2$

Average-entropy prediction: deterministic path within the branch of prediction whose $||G||$ (or entropy) at the end of the prediction time-interval is closest to the average $||G||$ (or entropy) over all branches

Max-likelihood prediction: deterministic path emanating from most likely initial



RED = average estimate
BLUE = average-entropy prediction
GREEN = max-likelihood prediction

Derivation

1) Reformulation of problem into Liouville SPDE
2) Use of Duhamel to project onto OPEN SODE
3) Closure