particle flow for nonlinear filters with Gromov's method

Fred Daum, Jim Huang
& Arjang Noushin

4 June 2019

	ensemble Kalman filter	particle filter (bootstrap with resampling)	stochastic particle flow filter
1. prediction of ensemble of particles in time (forecast)	Monte Carlo (particle flow in time)	Monte Carlo (particle flow in time)	Monte Carlo (particle flow in time)
2. measurement update of the conditional probability density (analysis)	Kalman filter formulas for mean & covariance matrix updates	Bayes' rule by multiplication (prior times likelihood)	Bayes' rule by particle flow from the prior to the posteriori
3. suffers from particle degeneracy?	no	**yes**	no
4. suffers from curse of dimensionality?	no	**yes** (even for linear Gaussian problems)	**no** for certain smooth nowhere vanishing densities
5. resample particles to mitigate particle degeneracy?	no	yes	no
6. optimal accuracy (with large enough N) for nonlinear & non-Gaussian problems?	**no**	yes	**yes** for certain smooth nowhere vanishing densities

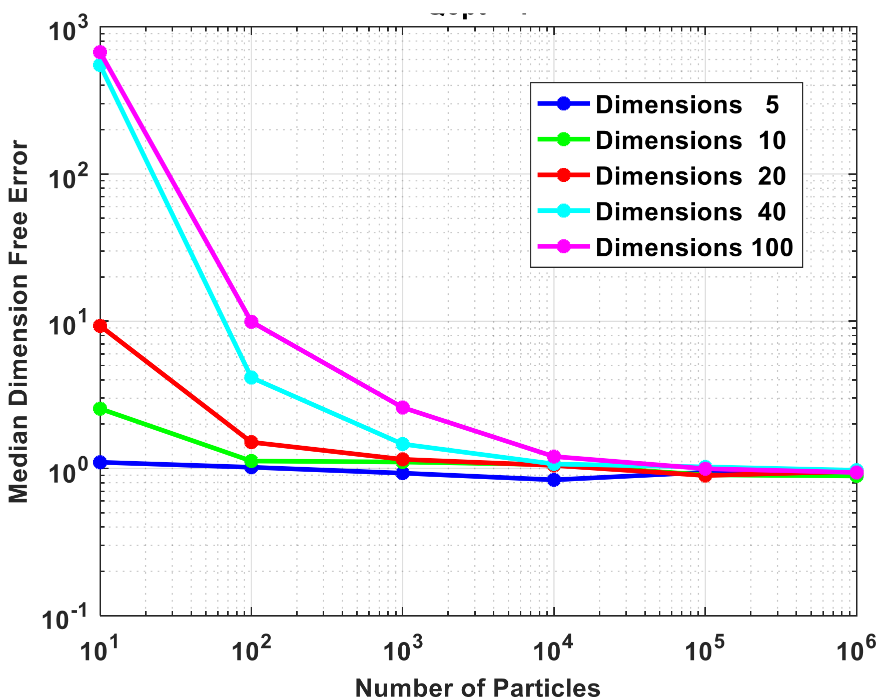
Bayes' rule using stochastic particle flow:

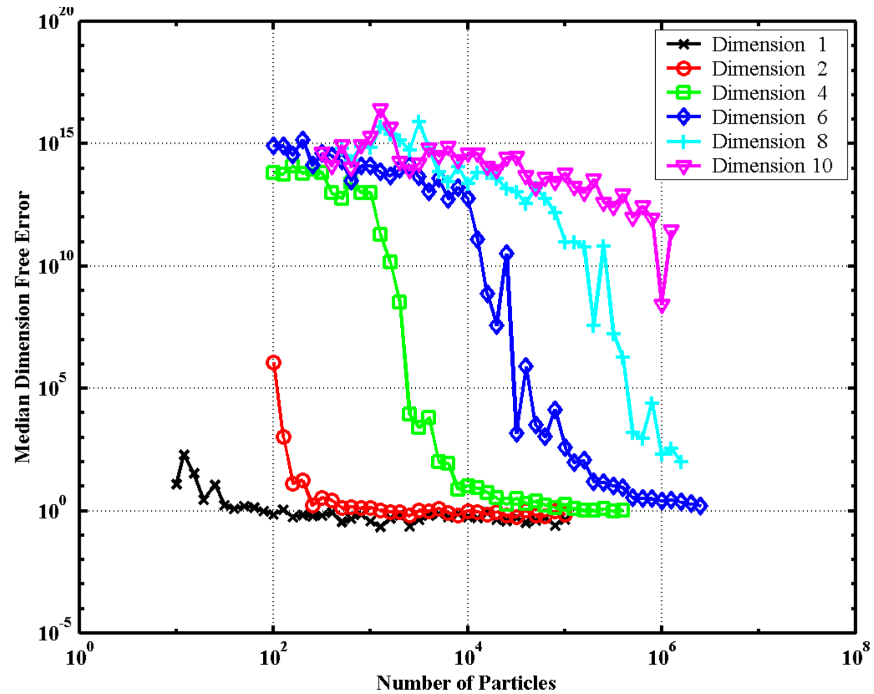
$$\frac{dx}{d\lambda} = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1} \left(\frac{\partial \log h}{\partial x}\right)^T + \frac{dw}{d\lambda}$$

approximation for Gaussian prior and likelihood (similar to Ensemble Kalman filter but for continuous time measurements for each particle x):

$$\frac{dx}{d\lambda} \approx P \left(\frac{\partial \theta}{\partial x}\right)^T R^{-1}(z - \theta(x)) + \frac{dw}{d\lambda}$$

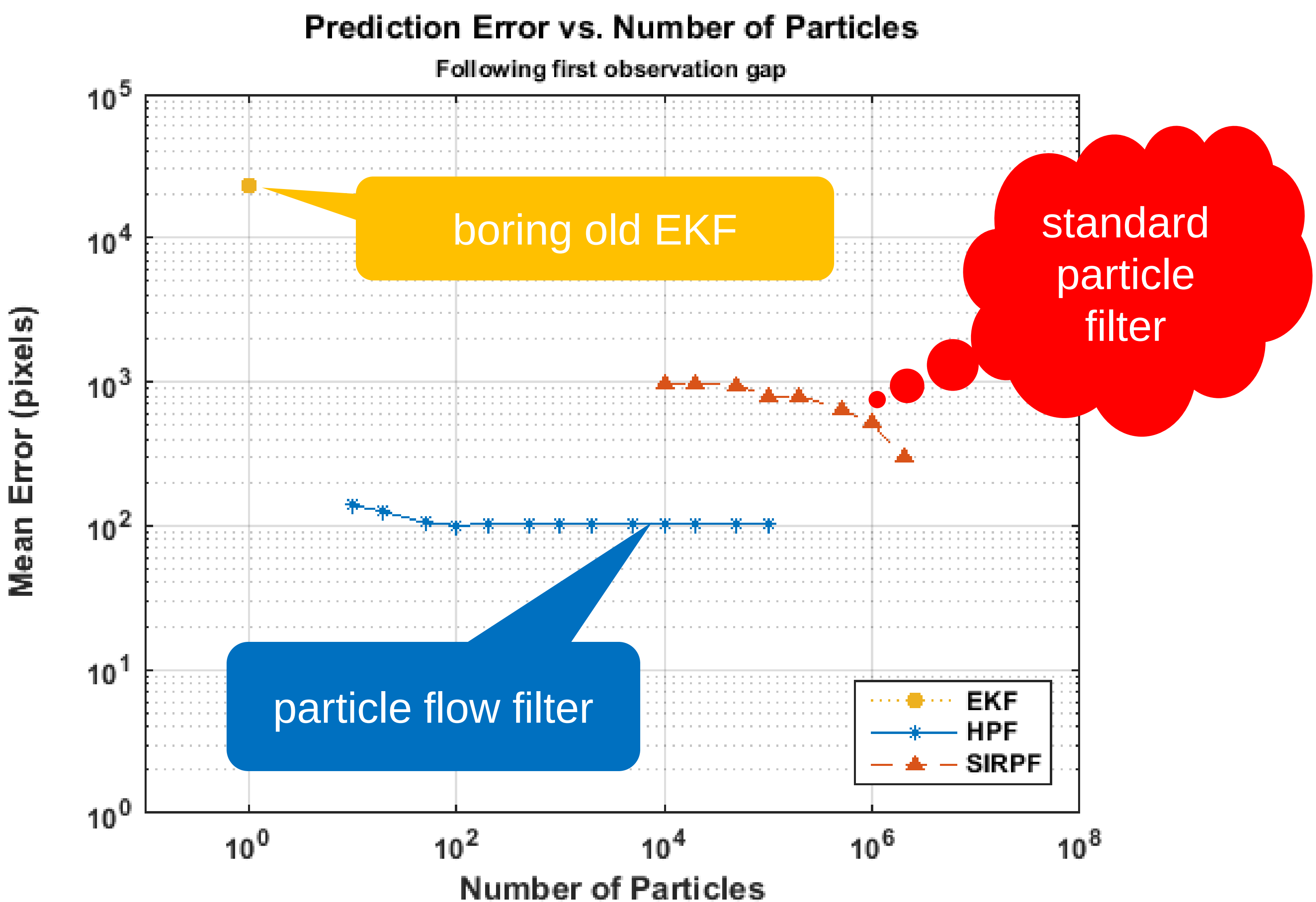
P = sample covariance matrix from set of particles





stochastic particle flow for Bayes' rule mitigates the curse of dimensionality

standard particle filter (with resampling from proposal density) suffers from the curse of dimensionality



Nima Moshtagh, Jonathan Chan, Moses Chan, "Homotopy Particle Filter for Ground-Based Tracking of Satellites at GEO," AMOS Conference, Hawaii September 2016.

Nima Moshtagh, Jonathan Chan, Moses Chan, "Homotopy Particle Filter for Ground-Based Tracking of Satellites at GEO," AMOS Conference, Hawaii September 2016.

nonlinear filter problem*

dynamical model of state:

$$\begin{cases} dx = F(x,t)dt + G(x,t)dw \quad \text{or} \\ \mathrm{x}(\mathrm{t}_{k+1}) = F(x(t_k), t_k, w(t_k)) \end{cases}$$

$x(t) =$ state vector at time t

estimate x given noisy measurements

$w(t) =$ process noise vector at time t

$z_k =$ measurement vector at time t_k

$$z_k = H(x(t_k), t_k, v_k)$$

$v_k =$ measurement noise vector at time t_k

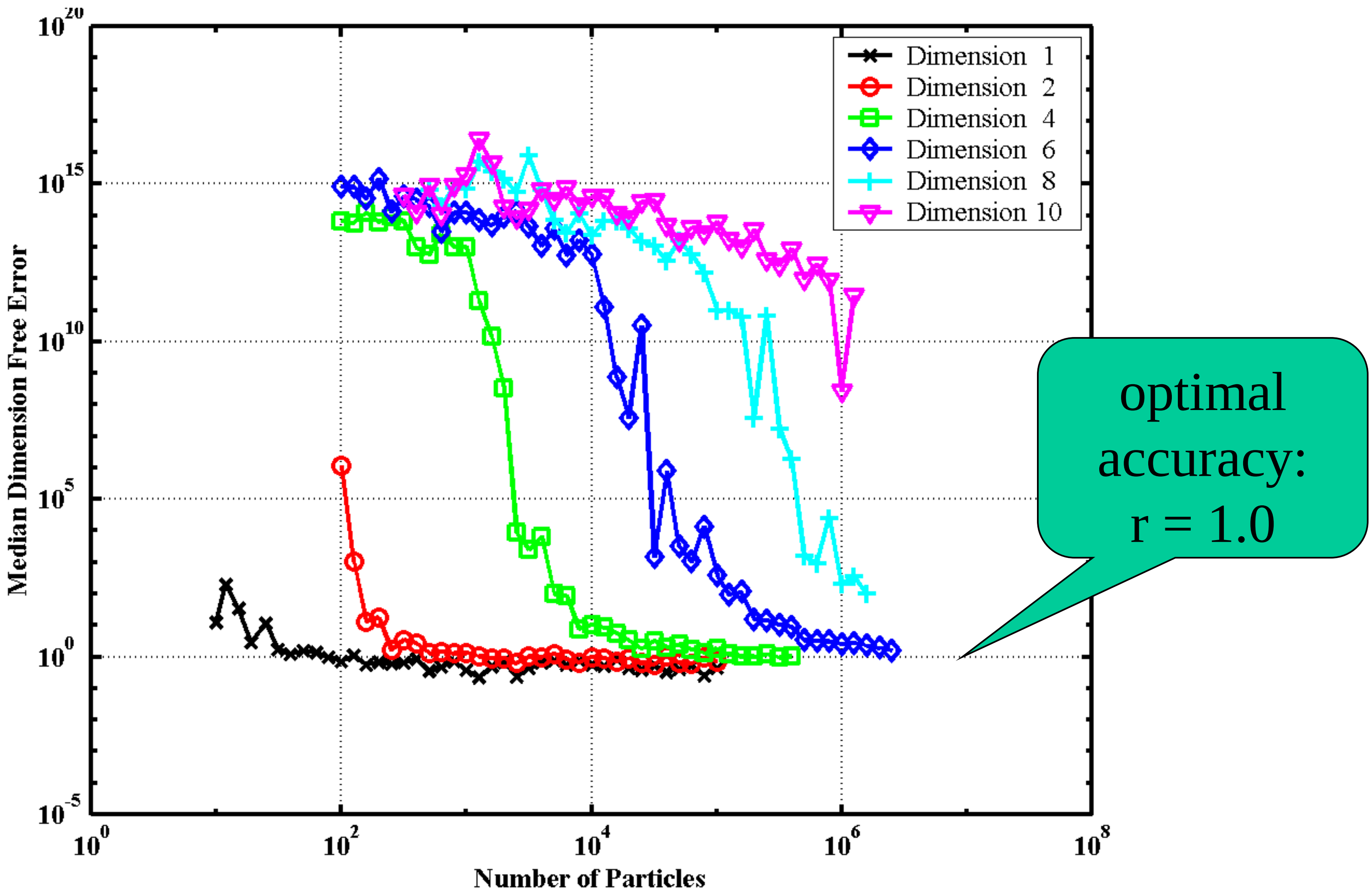
$p(x, t_k | Z_k) =$ probability density of x at time t_k given Z_k

$Z_k =$ set of all measurements

$$\mathrm{Z}_\mathrm{k} = \{z_1, z_2, \ldots, z_k\}$$

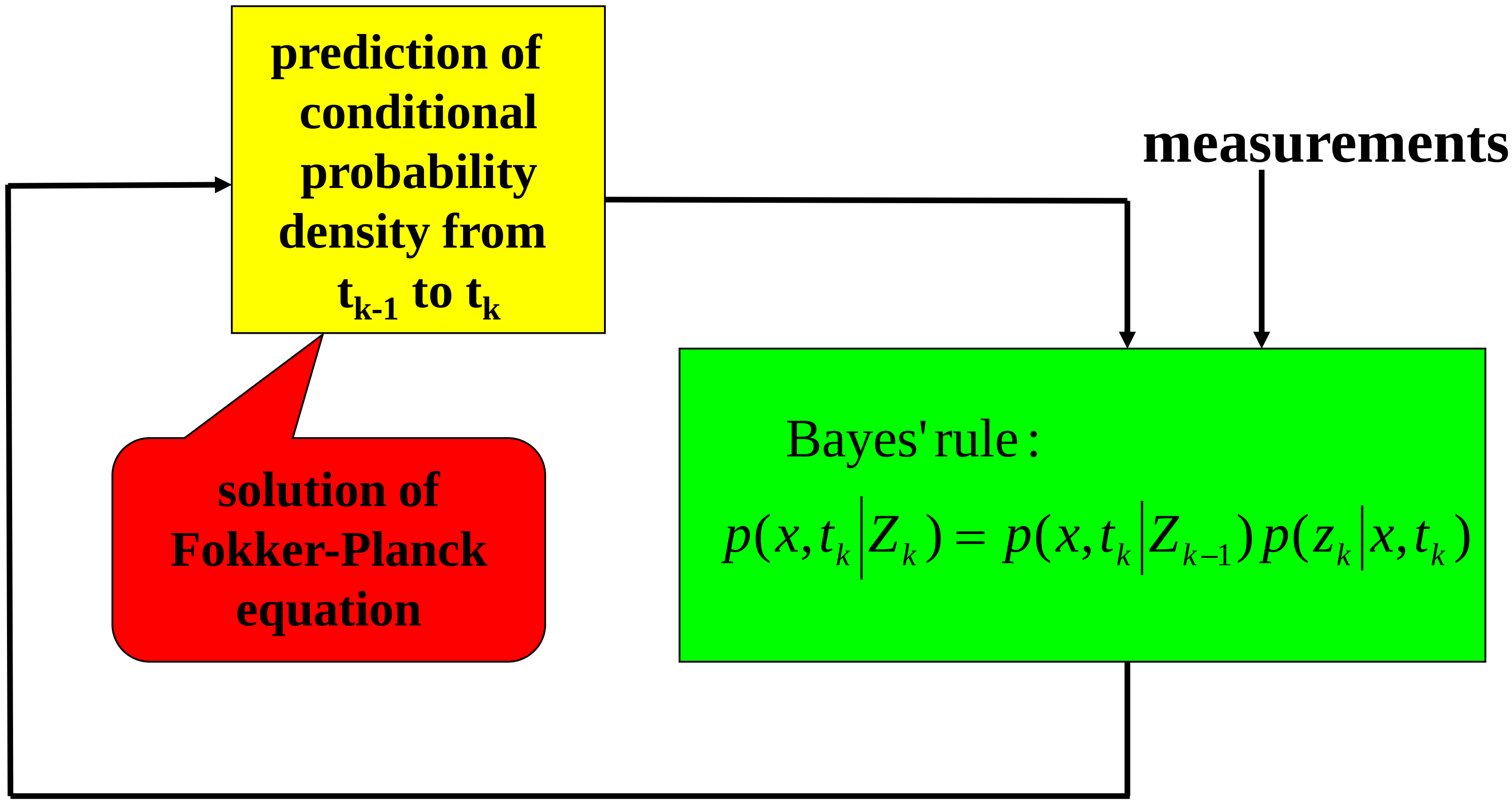
*"The Oxford handbook of nonlinear filtering," edited by Dan Crisan and Boris Rozovskii, Oxford University Press, 2011.

curse of dimensionality for classic particle filter*



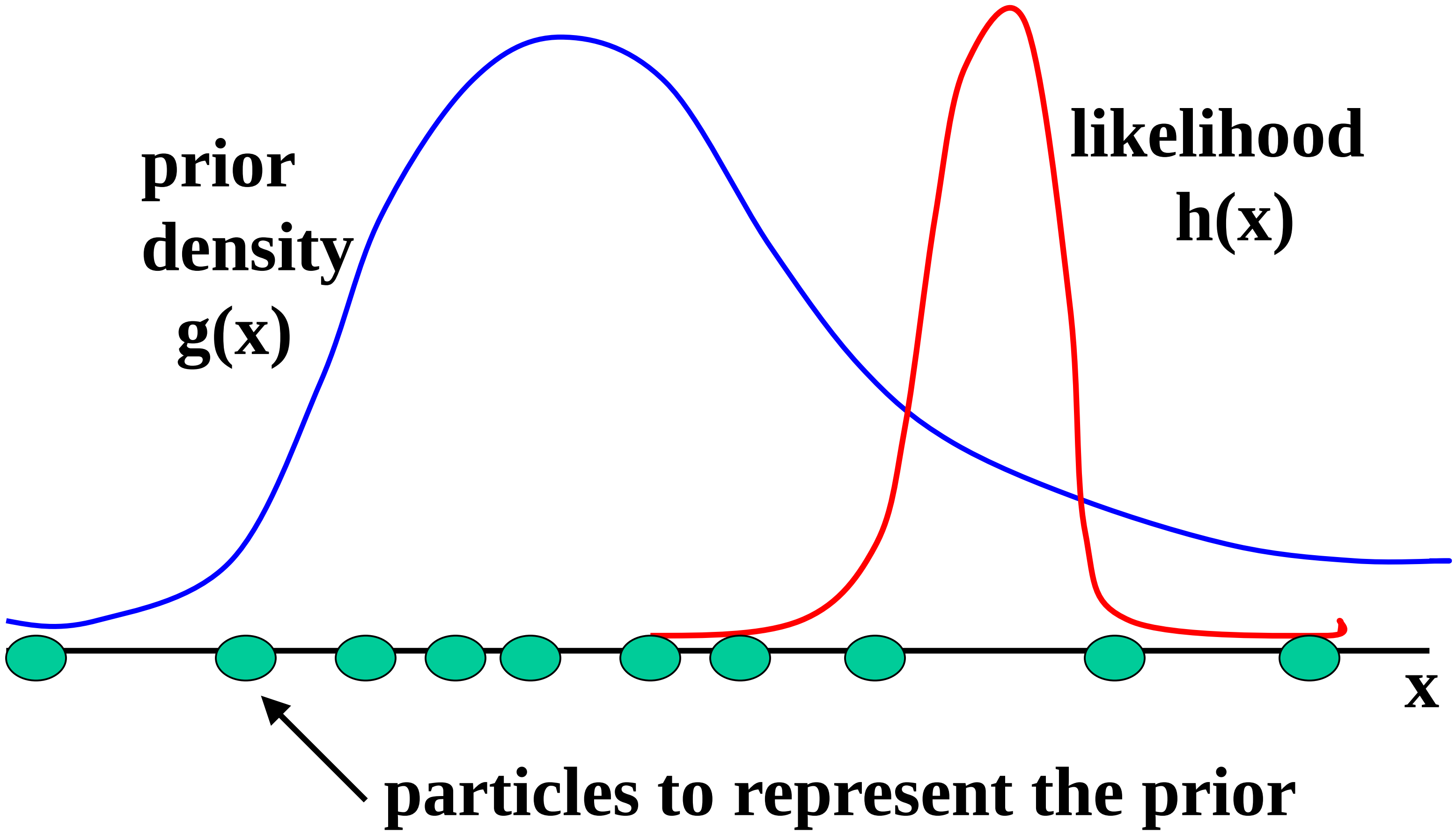
*Daum & Huang, IEEE AES Big Ski Conference, March 2003.

nonlinear filter*



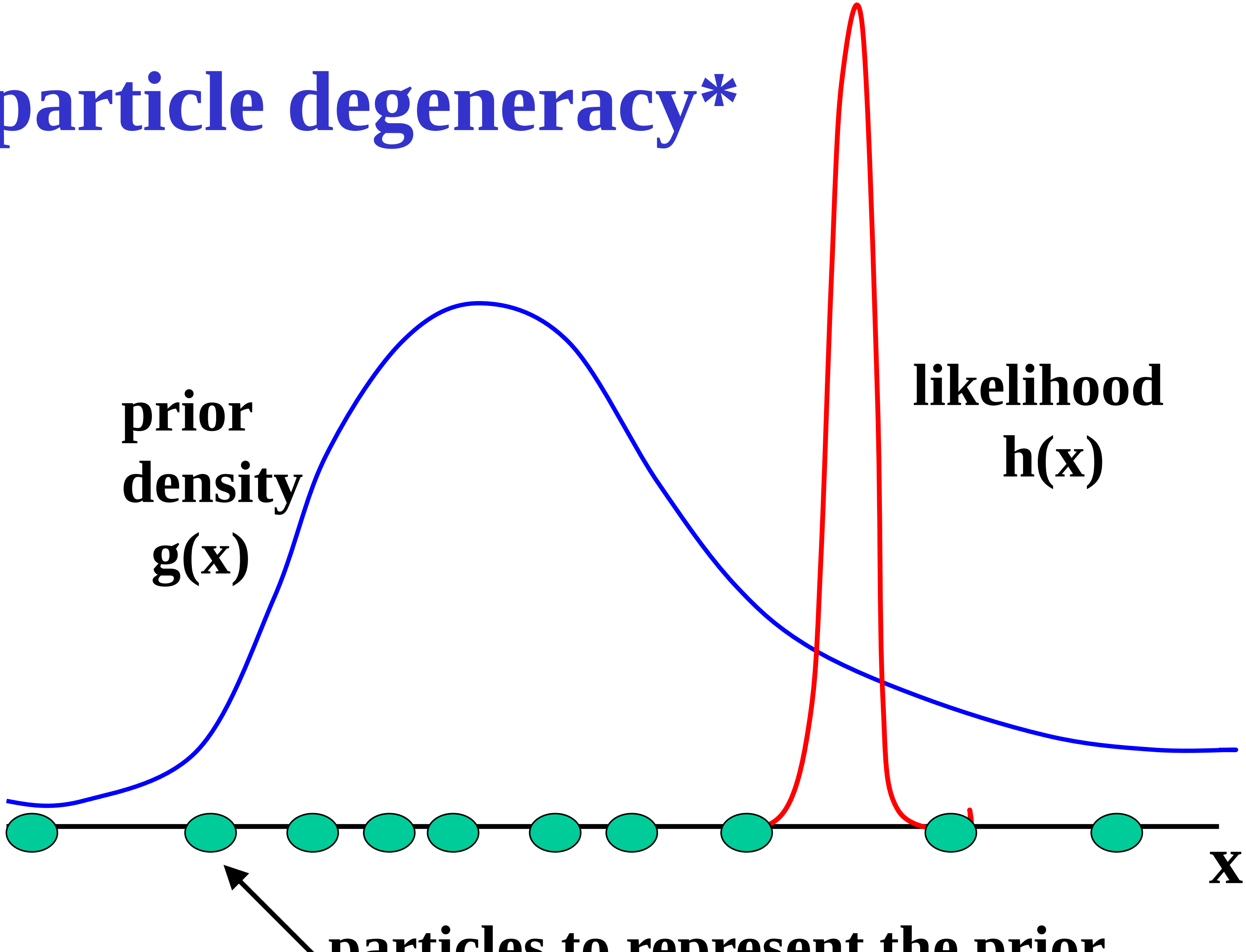
*Yu-Chi Ho & R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," IEEE Transactions on automatic control, pages 333-339, October 1964.

particle degeneracy*



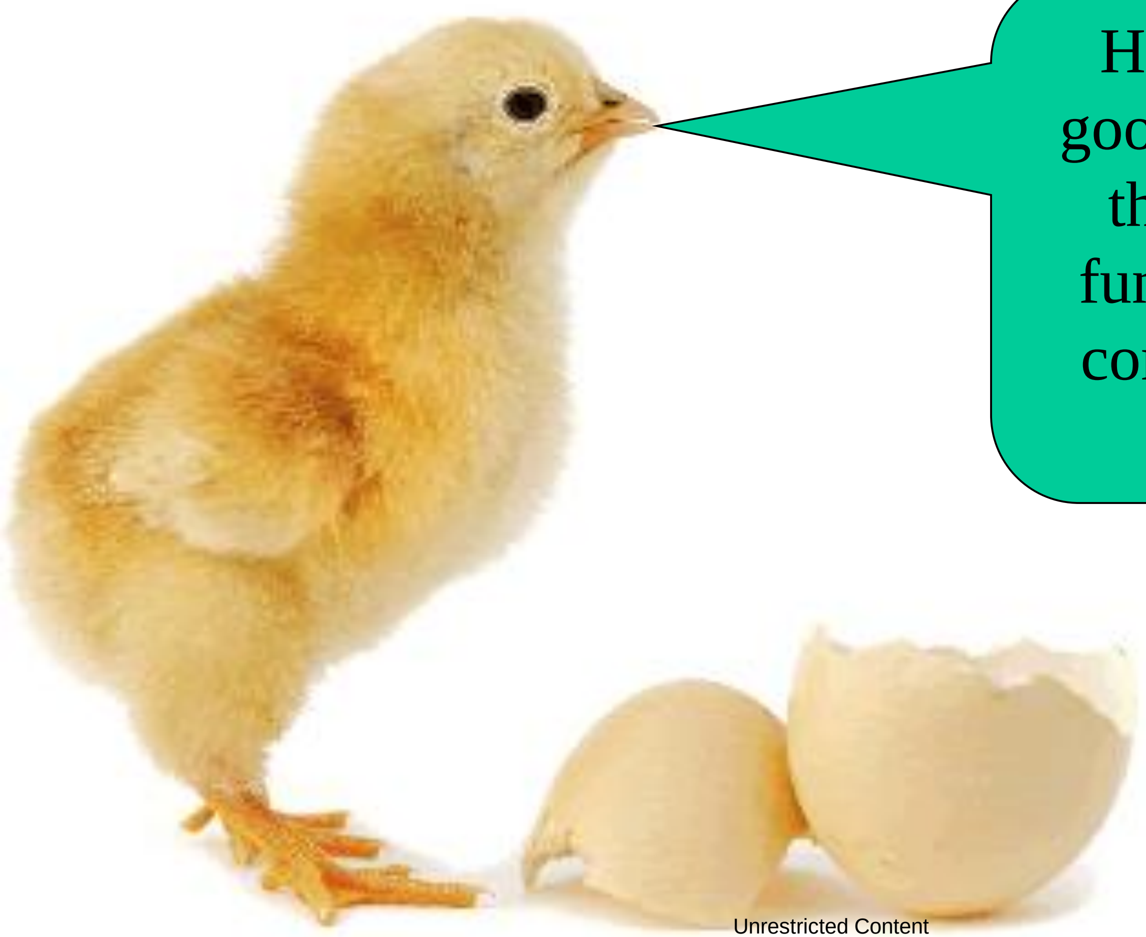
*Daum & Huang, "Particle degeneracy: root cause & solution," SPIE Proceedings 2011.

particle degeneracy*



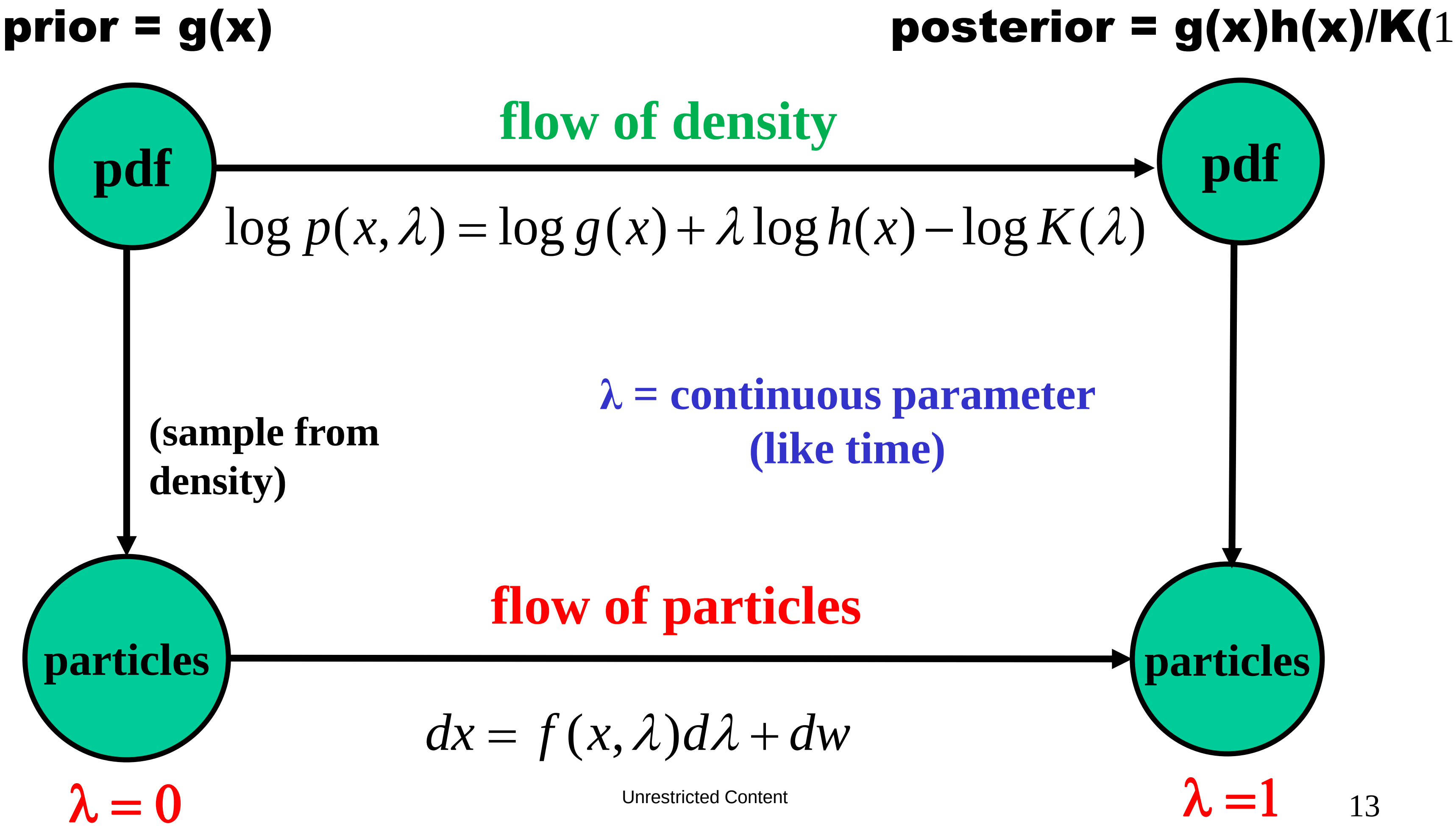
*Daum & Huang, "Particle degeneracy: root cause & solution," SPIE Proceedings 2011.

chicken & egg problem

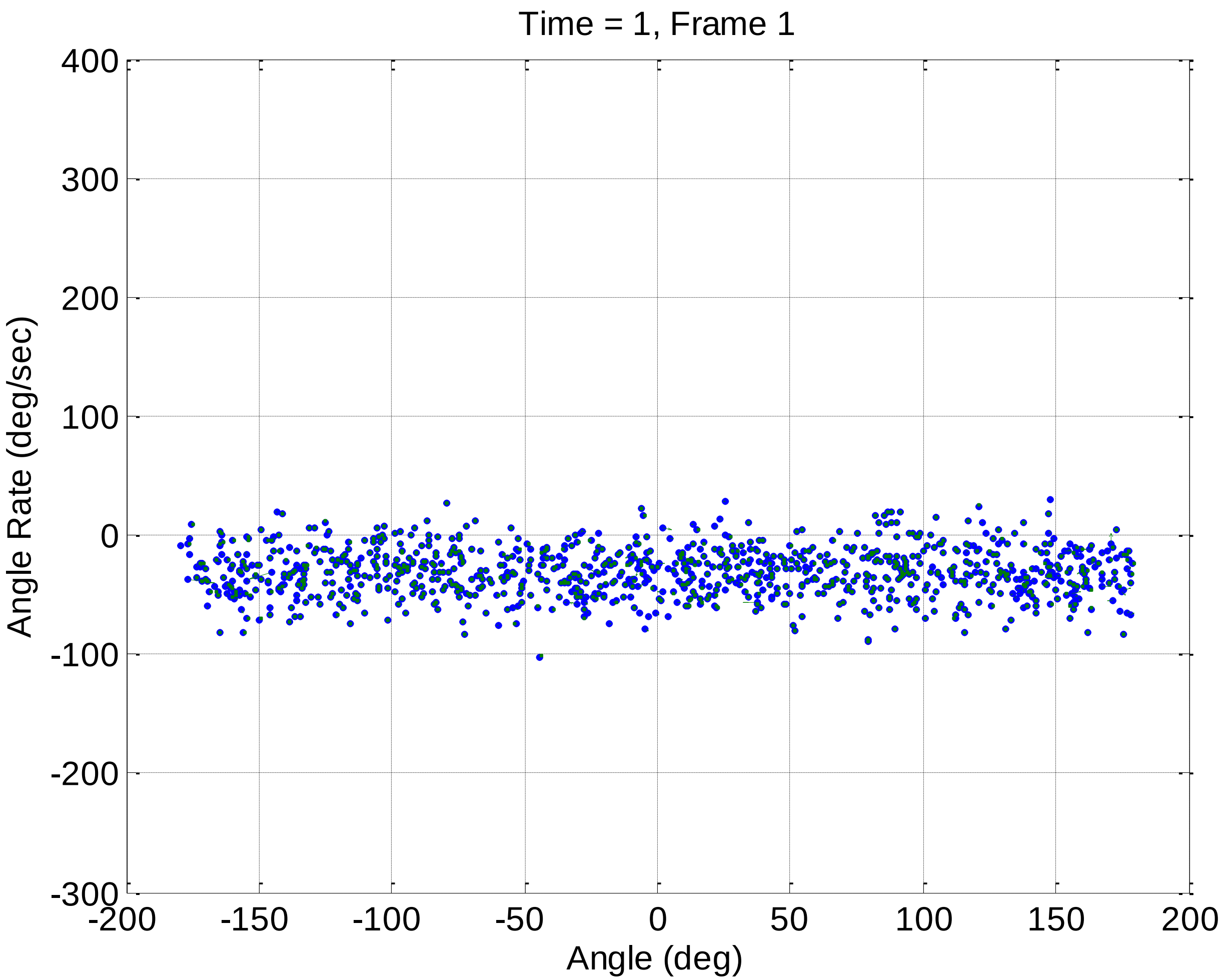


How do you pick a good way to represent the product of two functions before you compute the product itself?

induced flow of particles for Bayes' rule

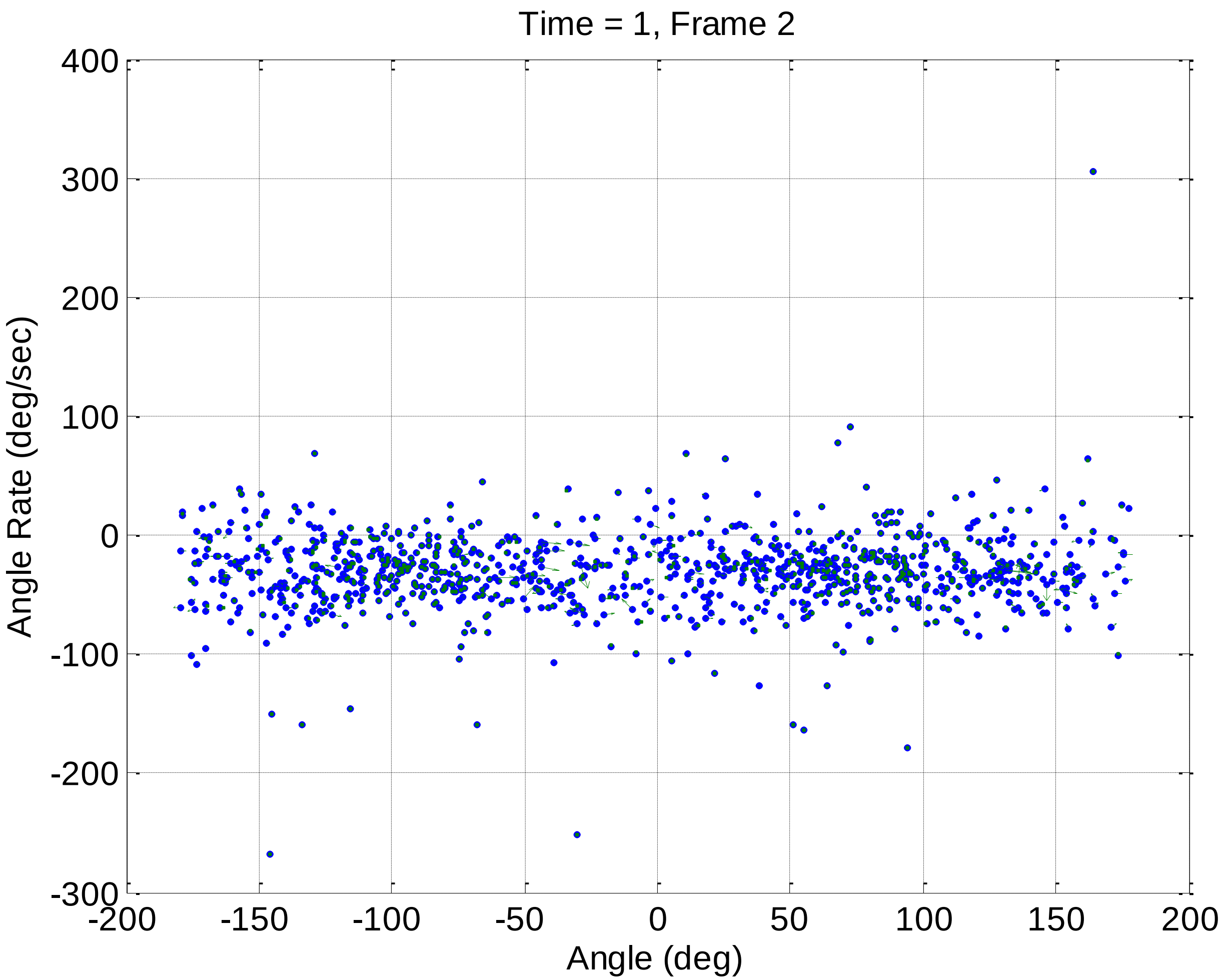


initial probability distribution of particles:



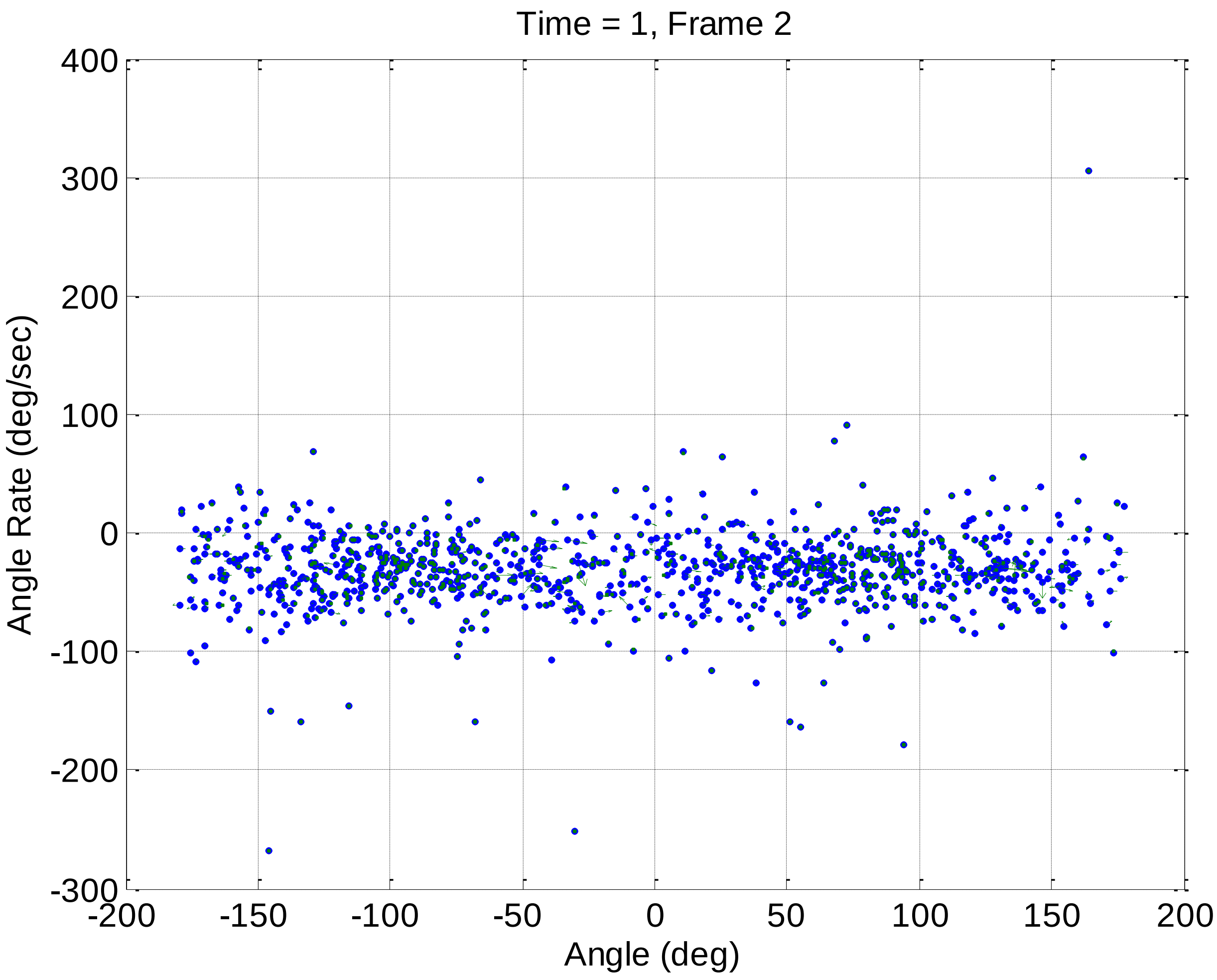
λ = 0.0

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



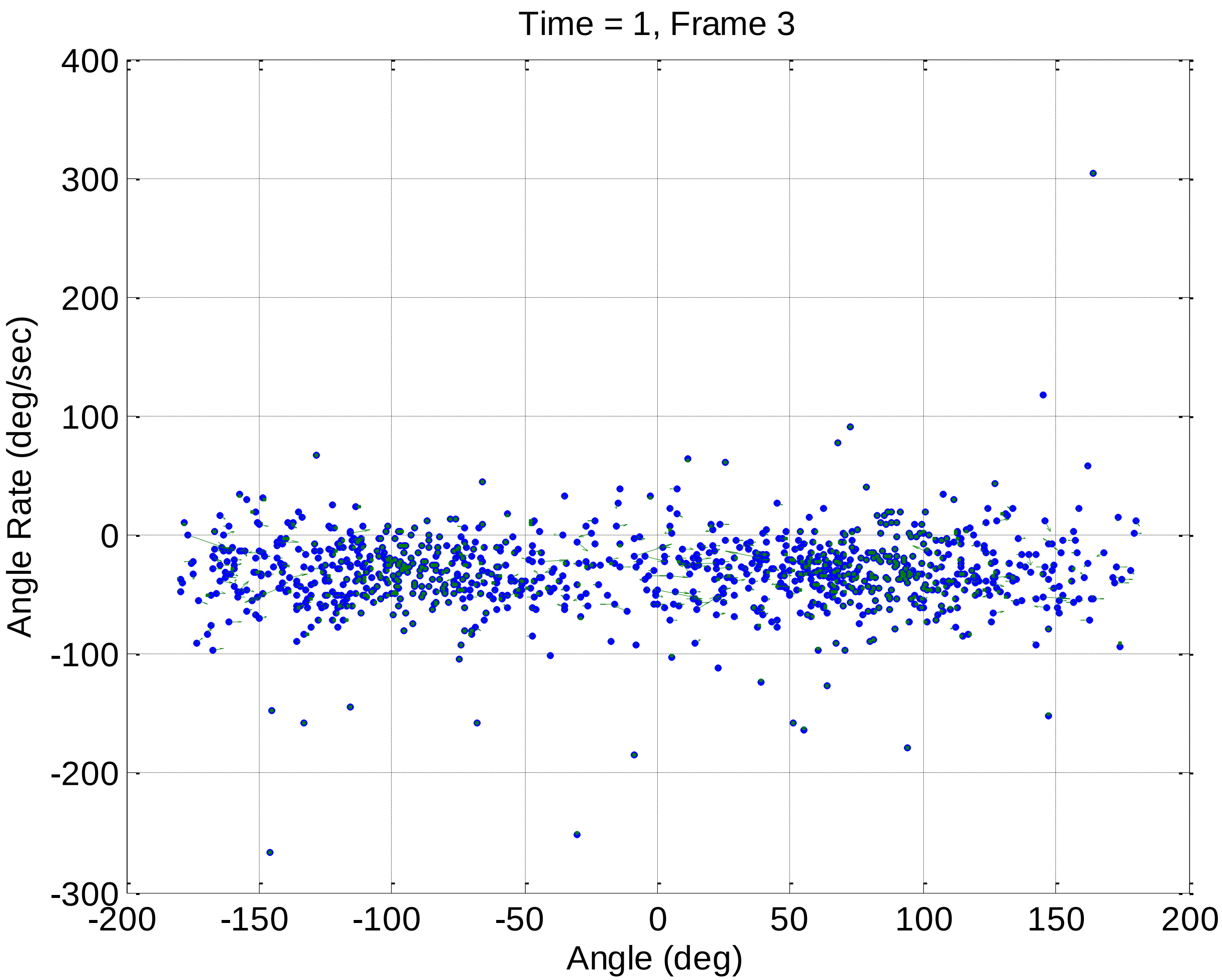
$\lambda = 0.1$

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



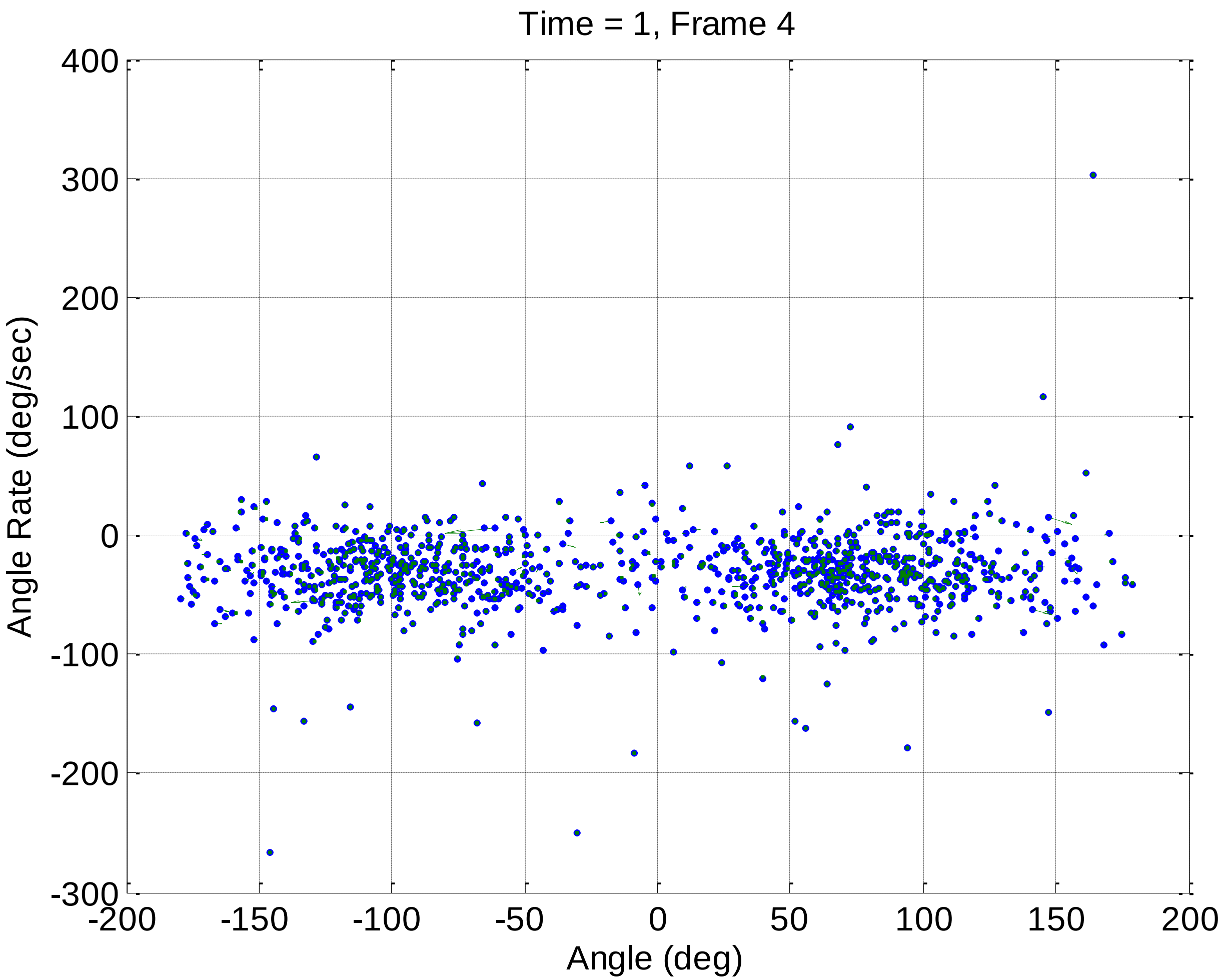
λ = 0.2

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



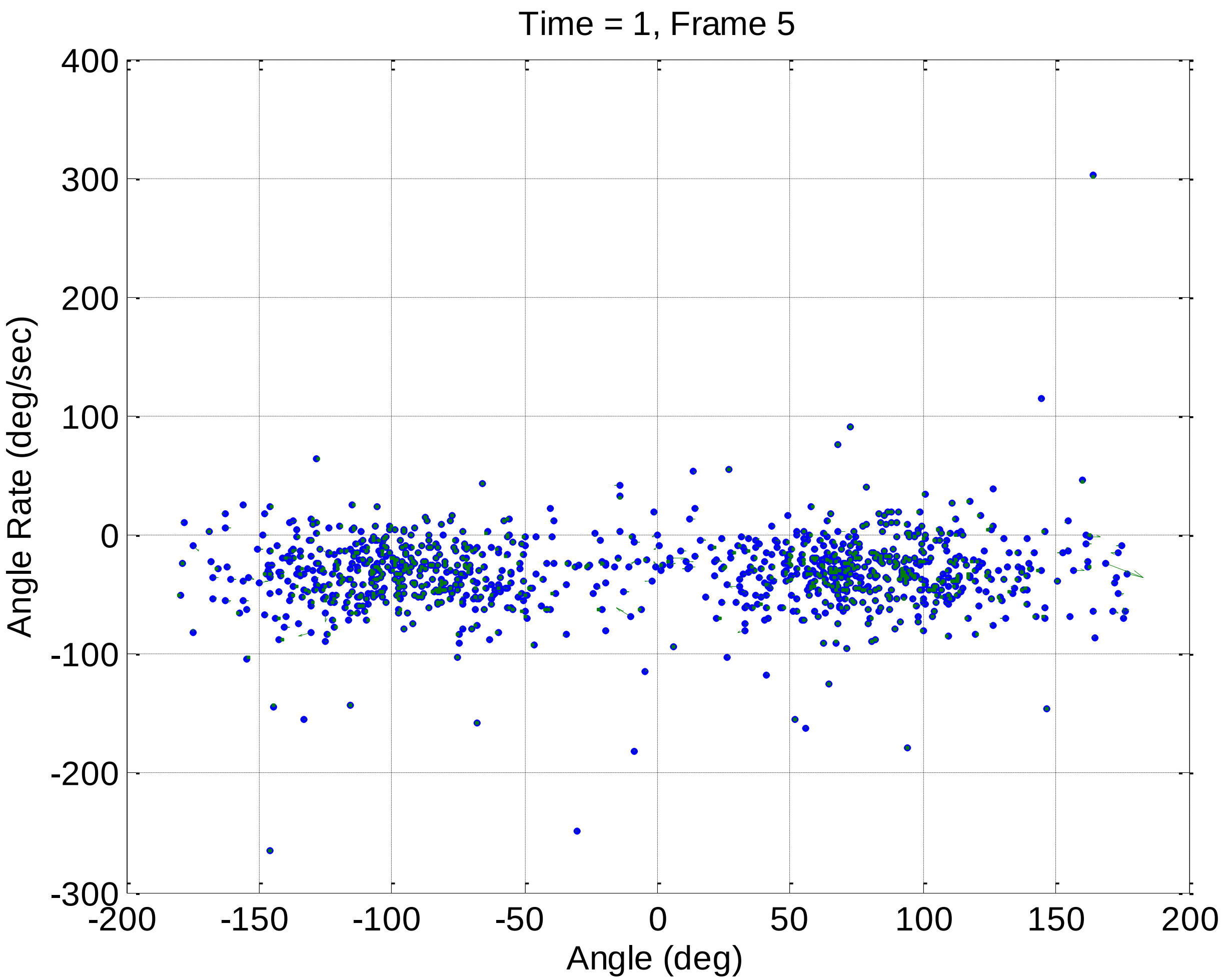
λ = 0.3

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



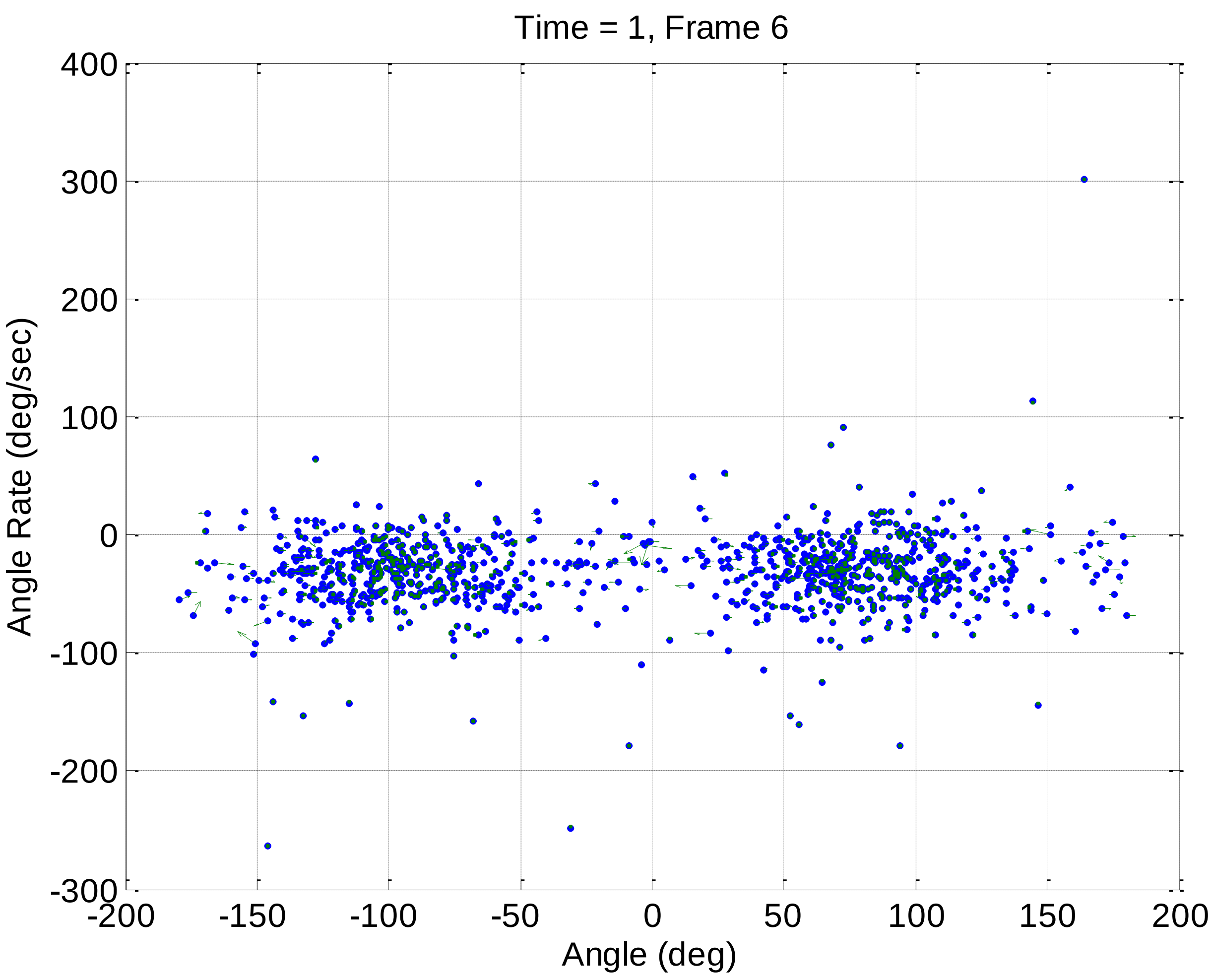
λ = 0.4

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



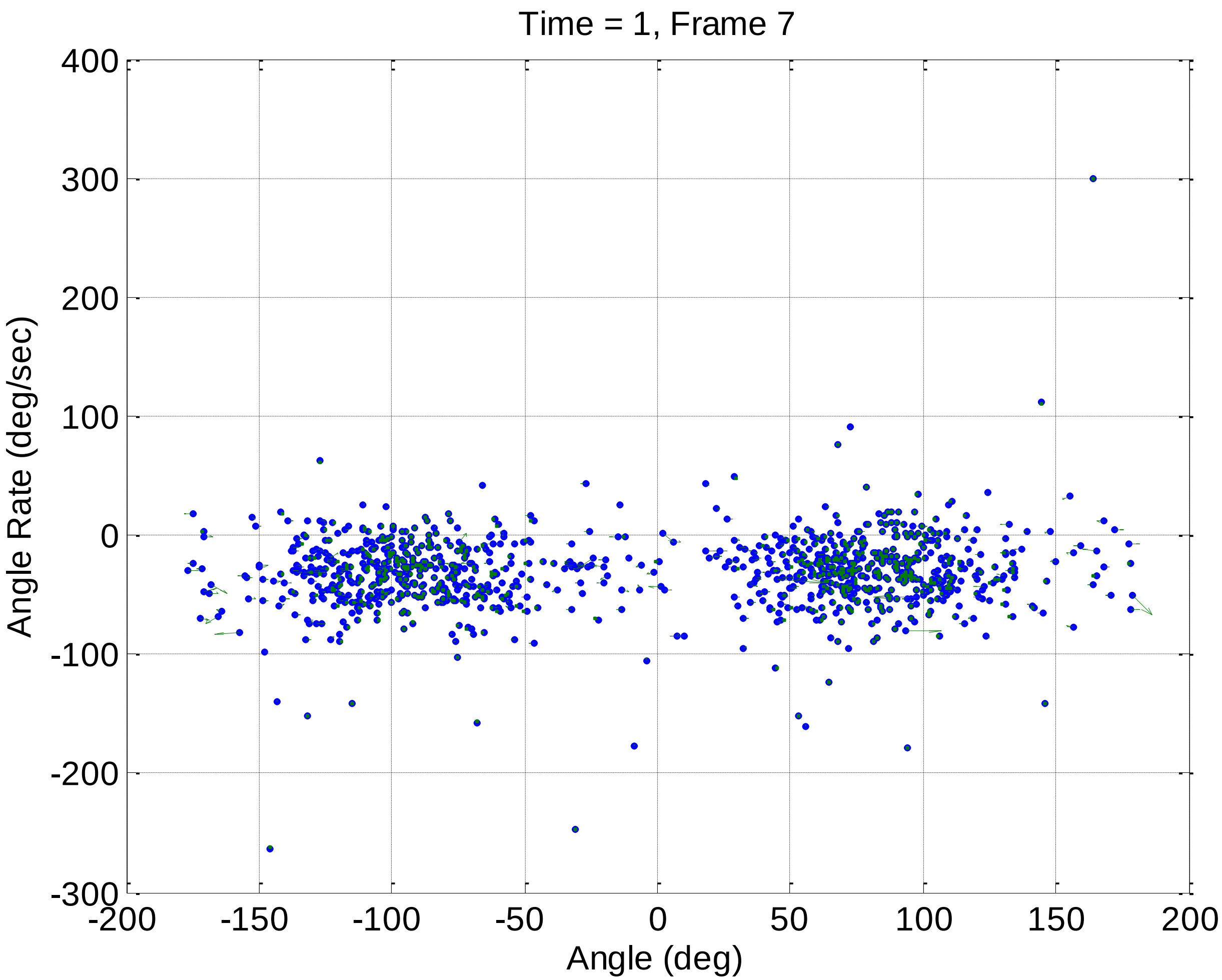
λ = 0.5

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



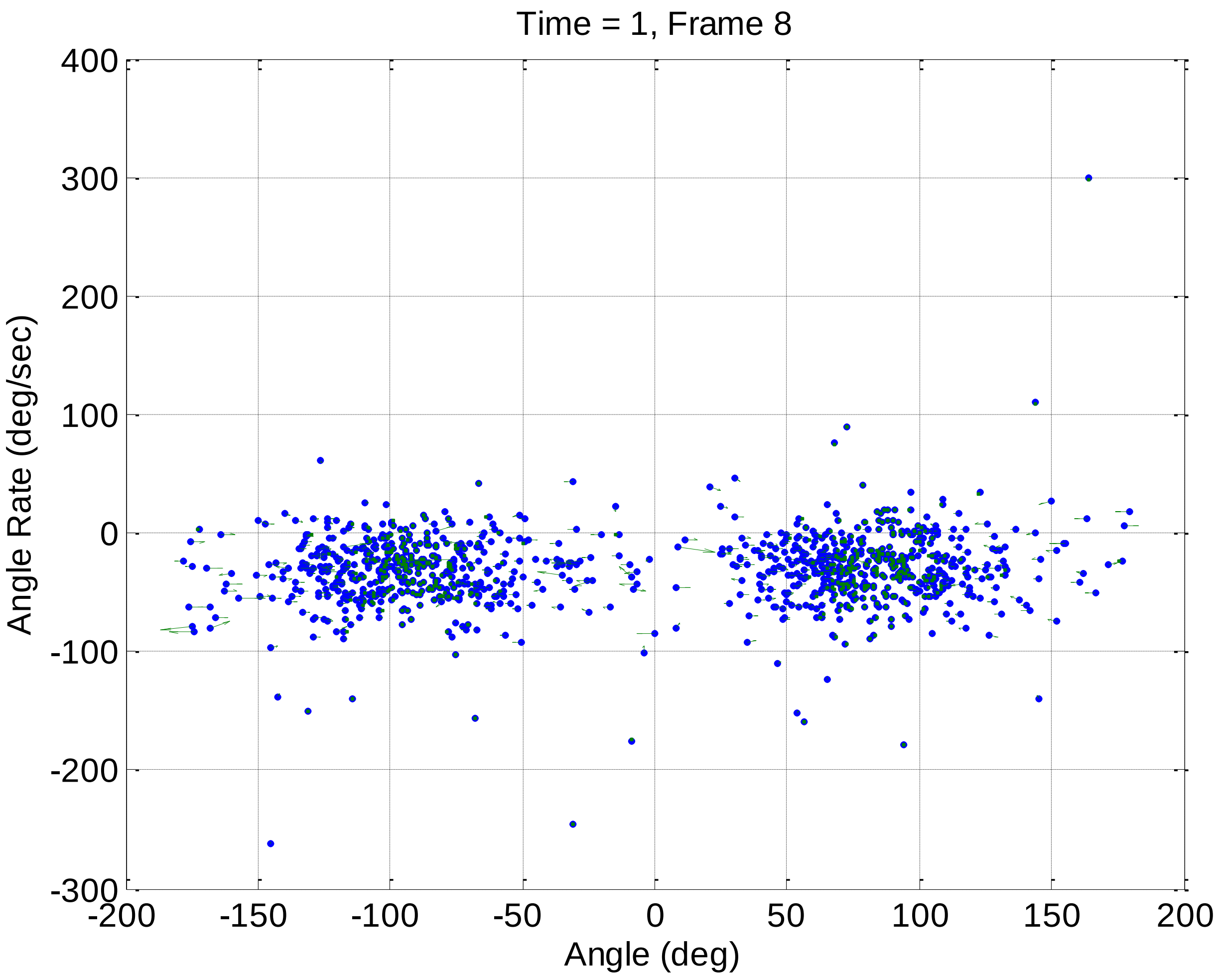
$\lambda = 0.6$

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



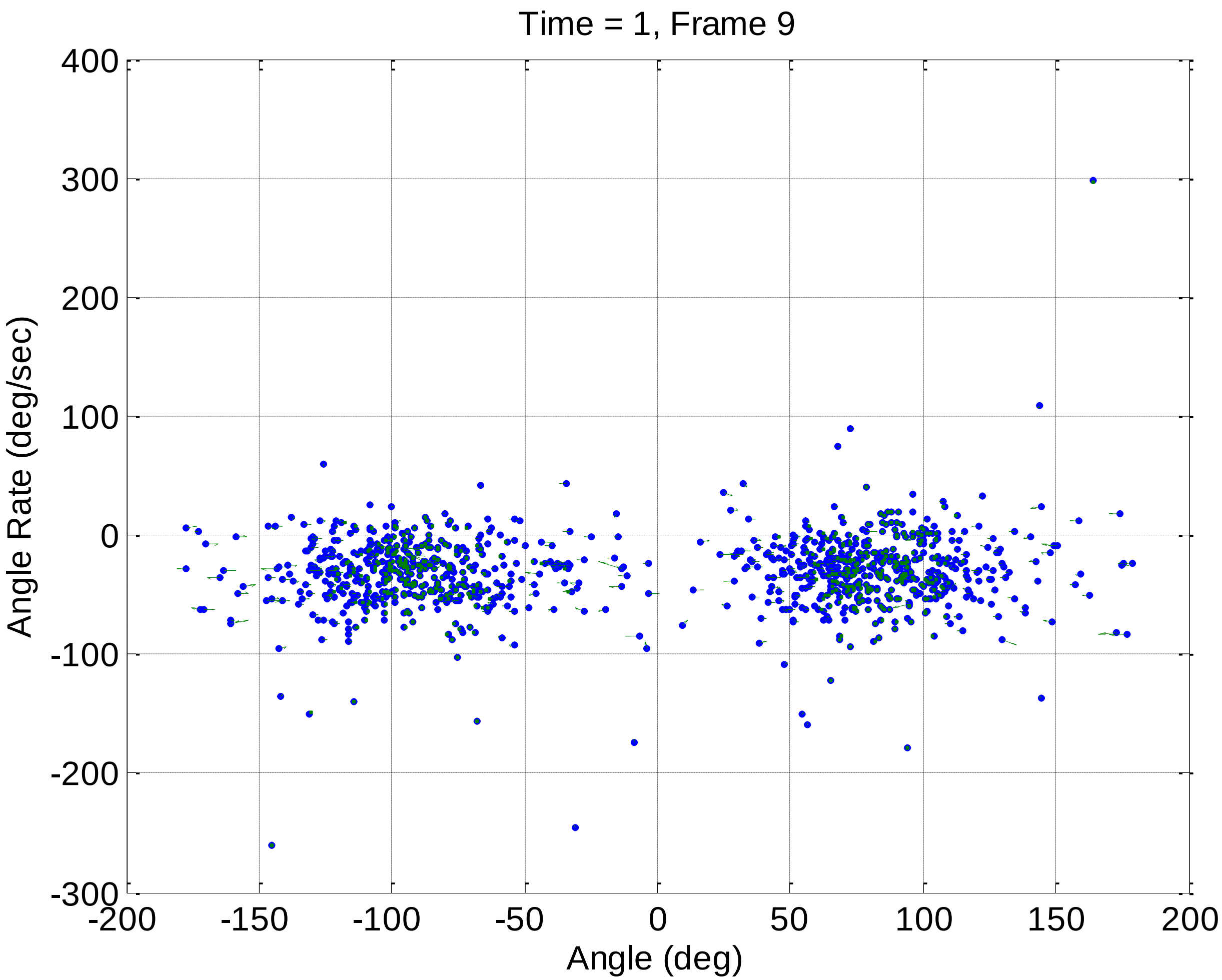
λ = 0.7

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



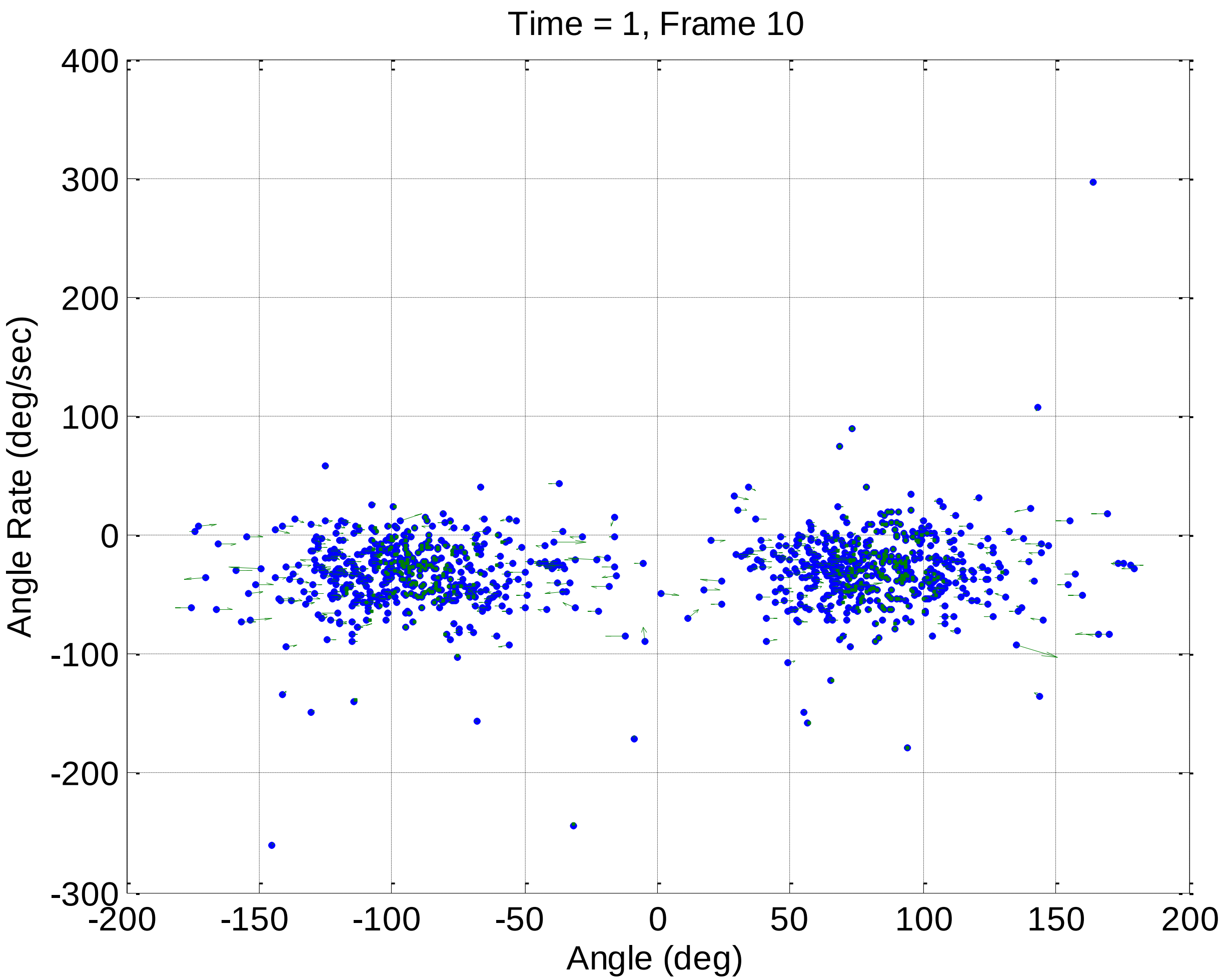
λ = 0.8

flow of particles (for one noisy measurement of sin(θ) with Bayes' rule):



λ = 0.9

final probability distribution of particles (resulting from one noisy measurement of sin(θ) with Bayes' rule):



λ = 1

incompressible flow	irrotational flow	Coulomb’s law flow	small curvature flow	constant curvature & constant speed flows
Gaussian densities	exponential family	differential Knothe-Rosenblatt flow	stochastic flows	geodesic flows
Fourier transform flow	direct integration	stabilized flows	finite dimensional flow	optimal Monge-Kantorovich transports
method of characteristics	separation of variables	renormalization group flow inspired by QFT	hybrid particle-parameter flow	non-optimal transport
Gibbs sampler like flow	Gromov’s method	Moser coupling flow	Monge-Ampère with N-principle	Monge-Ampère flow

exact particle flow for Gaussian densities:

$$\frac{dx}{d\lambda} = f(x,\lambda)$$

$$\log(h) - \frac{d\log K(\lambda)}{d\lambda} = -div(f) - \frac{\partial \log p}{\partial x} f$$

for g & h Gaussian, we can solve for f exactly:

$$f = Ax + b$$

$$A = -\frac{1}{2} PH^T \left[\lambda HPH^T + R\right]^{-1} H$$

$$b = (I + 2\lambda A)\left[(I + \lambda A)PH^T R^{-1} z + A\bar{x}\right]$$

dx/dλ does not depend on K(λ), despite the fact that the PDE does!

incompressible particle flow

$$\frac{\mathrm{dx}}{\mathrm{d}\lambda} = \begin{cases} -\log(h(x)) \dfrac{\left[\dfrac{\partial \log p(x,\lambda)}{\partial x}\right]^T}{\left\|\dfrac{\partial \log p(x,\lambda)}{\partial x}\right\|^2} & \text{for non - zero gradient} \\ 0 & \text{otherwise} \end{cases}$$

for $\mathrm{d} \geq 2$

dx/dλ does not depend on K(λ), despite the fact that the PDE does!

geodesic particle flow* :

$$\frac{dx}{d\lambda} = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1}\left(\frac{\partial \log h}{\partial x}\right)^T$$

dx/dλ does not depend on K(λ), despite the fact that the PDE does!

If we approximate the density p as Gaussian, then the observed Fisher information matrix can be computed using the sample covariance matrix (C) over the set of particles:

$$\frac{dx}{d\lambda} \approx C\left(\frac{\partial \log h}{\partial x}\right)^T$$

for Gaussian densities we get the EKF for each particle:

$$\frac{dx}{d\lambda} \approx C\left(\frac{\partial \theta(x)}{\partial x}\right)^T R^{-1}\left(z - \theta(x)\right)$$

*Daum & Huang, "particle flow with non-zero diffusion for nonlinear filters," SPIE conference proceedings, San Diego, August 2013.

derivation of PDE for particle flow with Q ≠ 0:

$$dx = f(x,\lambda)d\lambda + \sqrt{Q(x,\lambda)}dw$$

$$\frac{\partial p(x,\lambda)}{\partial \lambda} = -div(pf) + \frac{1}{2}div\left[Q(x,\lambda)\frac{\partial p}{\partial x}\right]$$

$$\frac{\partial \log p(x,\lambda)}{\partial \lambda}p(x,\lambda) = -div(pf) + \frac{1}{2}div\left[Q\frac{\partial p}{\partial x}\right]$$

$$\log p(x,\lambda) = \log g(x) + \lambda \log h(x) - \log K(\lambda)$$

$$\left[\log h(x) - \frac{d\log K(\lambda)}{d\lambda}\right]p(x,\lambda) = -div(pf) + \frac{1}{2}div\left[Q\frac{\partial p}{\partial x}\right]$$

$$\left[\log h - \frac{d\log K}{d\lambda}\right]p = -p\,div(f) - \frac{\partial p}{\partial x}f + \frac{1}{2}div\left[Q\frac{\partial p}{\partial x}\right]$$

$$\left[\log h - \frac{d\log K}{d\lambda}\right] = -div(f) - \frac{\partial \log p}{\partial x}f + \frac{1}{2p}div\left[Q\frac{\partial p}{\partial x}\right]$$

$$\frac{\partial \log h}{\partial x} = -f^T\frac{\partial^2 \log p}{\partial x^2} - \frac{\partial div(f)}{\partial x} - \frac{\partial \log p}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left\{div\left[Q\frac{\partial p}{\partial x}\right]/p\right\}$$

stochastic particle flow:

$$\frac{\partial \log h}{\partial x} = -f^T \frac{\partial^2 \log p}{\partial x^2} - \frac{\partial div(f)}{\partial x} - \frac{\partial \log p}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left\{ div\left[Q\frac{\partial p}{\partial x}\right] / p \right\}$$

$$dx = f(x,\lambda)d\lambda + \sqrt{Q(\lambda)}dw$$

$$f = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1}\left(\frac{\partial \log h}{\partial x}\right)^T$$

$$Q = \left[P^{-1} + \lambda H^T R^{-1} H\right]^{-1} H^T R^{-1} H \left[P^{-1} + \lambda H^T R^{-1} H\right]^{-1}$$

$$Q = -\left[\frac{\partial^2}{\partial x^2} logp\right]^{-1} \frac{\partial^2 logh}{\partial x^2} \left[\frac{\partial^2}{\partial x^2} logp\right]^{-1}$$

$$\frac{dx}{d\lambda} = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1} \left(\frac{\partial \log h}{\partial x}\right)^T + \frac{dw}{d\lambda}$$

$$\frac{\partial^2 \log p}{\partial x^2} = \frac{\partial^2 \log g}{\partial x^2} + \lambda \frac{\partial^2 \log h}{\partial x^2}$$

$$\frac{\partial^2 \log p}{\partial x^2} = -C^{-1} + \lambda \frac{\partial^2 \log h}{\partial x^2}$$

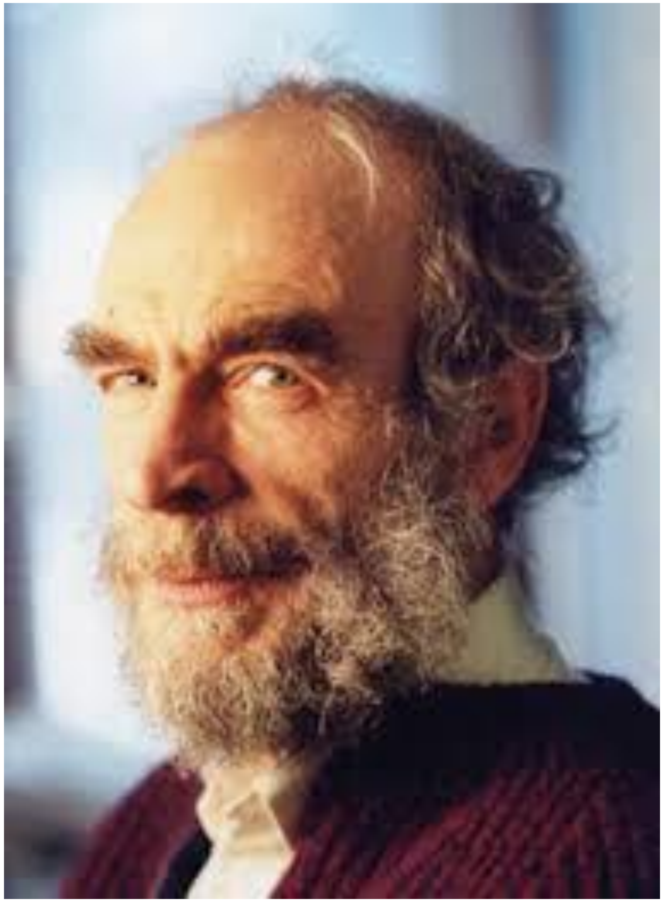
$$\frac{dx}{d\lambda} \approx P \left(\frac{\partial \theta}{\partial x}\right)^T R^{-1}(z - \theta(x)) + \frac{dw}{d\lambda}$$

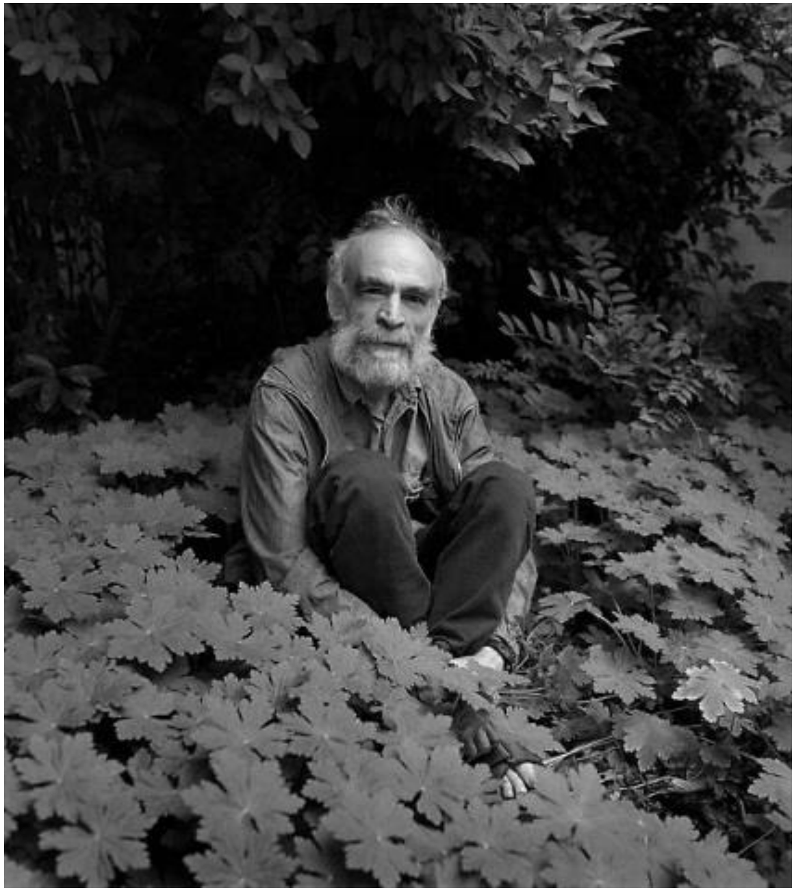
$$\frac{\partial \log h}{\partial x} = -f^T \frac{\partial^2 \log p}{\partial x^2} - \frac{\partial div(f)}{\partial x}$$

$$- \frac{\partial \log p}{\partial x} \frac{\partial f}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x} \left\{ div\left[Q \frac{\partial p}{\partial x} \right] / p \right\}$$

$$f = -\left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \left(\frac{\partial \log h}{\partial x} \right)^T$$

$$Q = -\left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \frac{\partial^2 \log h}{\partial x^2} \left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1}$$





There exists a “nice” solution (i.e., no integration required) to a linear constant coefficient PDE for smooth functions if and only if the number of unknowns is sufficiently large (at least the number of linearly independent equations plus the dimension of x).

simplest non-trivial example of Gromov's method:

$$\frac{\partial q_1}{\partial x_1} + \frac{\partial q_2}{\partial x_2} + q_3 = \eta$$

$$q = M\eta$$

$$q = \left[-\frac{\partial \eta}{\partial x_2}, \frac{\partial \eta}{\partial x_1}, \eta \right]$$

check solution :

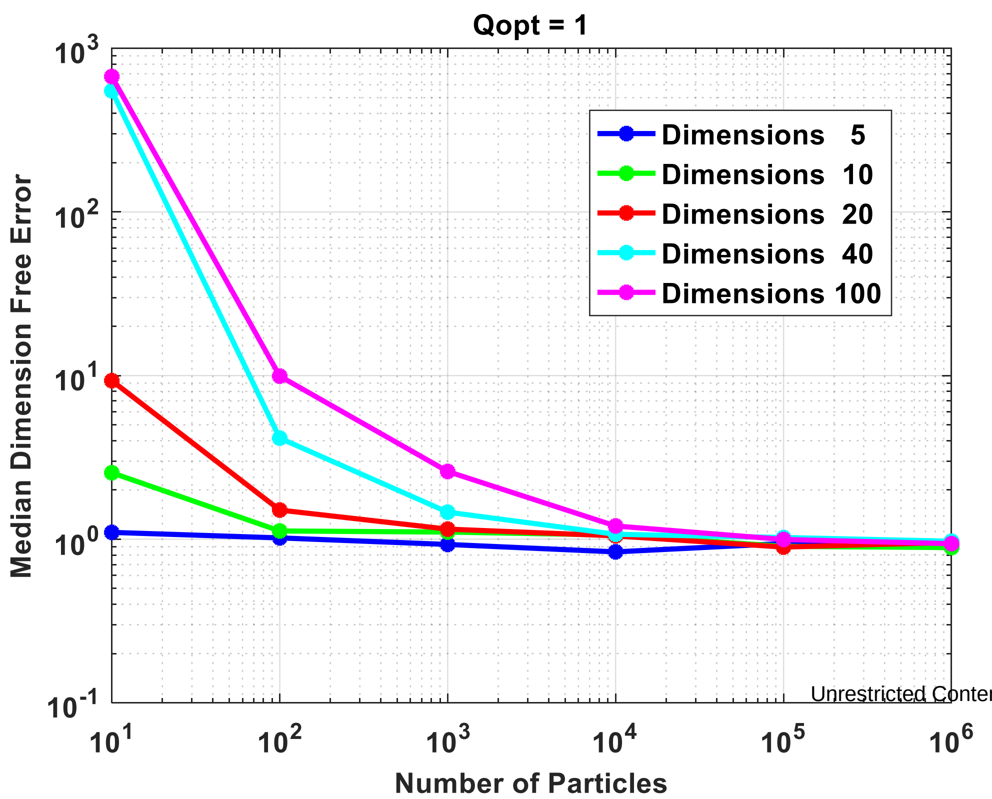
$$-\frac{\partial^2 \eta}{\partial x_1 \partial x_2} + \frac{\partial^2 \eta}{\partial x_2 \partial x_1} + \eta = \eta$$

more details:

VIDEO of recent talk at Stony Brook University (24 April 2018)

https://youtu.be/vqJGB47XoeY

Daum, Huang & Noushin, "new theory & numerical experiments for Gromov's method," IEEE FUSION Conference, Cambridge England July 2018



$$\frac{\partial \log h}{\partial x} = -f^{T}\frac{\partial^{2}\log p}{\partial x^{2}} - \frac{\partial div(f)}{\partial x} - \frac{\partial \log p}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left\{div\left[Q\frac{\partial p}{\partial x}\right]/p\right\}$$

$$f = -\left[\frac{\partial^{2}\log p}{\partial x^{2}}\right]^{-1}\left(\frac{\partial \log h}{\partial x}\right)^{T}$$

$$Q = -\left[\frac{\partial^{2}\log p}{\partial x^{2}}\right]^{-1}\frac{\partial^{2}\log h}{\partial x^{2}}\left[\frac{\partial^{2}\log p}{\partial x^{2}}\right]^{-1}$$

further research:

(1) compute f & Q assuming that g & h are in the exponential family or Gaussian mixture or exponential family mixture

(2) compute f & Q without "splitting the PDE", i.e., without assuming that the last 3 terms in the PDE sum to zero (but rather something else "nice" of our design, similar to Beneš filter or Daum exact filters)

(3) use Dirac approximation to solution of Fokker-Planck equation

(4) geometric solutions of PDE using involution or other EDS ideas (Deane Yang, Robert Bryant, Shirley Yap)

(5) compute SQRT(Q) rather than Q

(6) use quasi Monte Carlo (QMC) with Hilbert space filling curve rather than boring old Monte Carlo samples (Gerber & Chopin 2015)

(7) invent better methods to mitigate stiffness of the flow (Crouse 2019)

(8) numerical experiments & practical applications

(9) many more open problems; state & prove theorems, bounds,….

$$If \quad N \geq \frac{52\kappa^2}{\epsilon}\sqrt{\frac{d}{m}+D^2}\ \log\left[\frac{24}{\epsilon}\left(\frac{d}{m}+D^2\right)\right]$$

$$d = dimension\ of\ x$$
$$\kappa = condition\ number\ of\ Hessian\ of\ \log p$$
$$\kappa = \frac{L}{m}$$
$$\|x_0 - x^*\| \leq D$$

assuming that p is strictly log concave, positive and C^2

Then $W_2(p_0, \mathrm{p}) \leq \epsilon$

BACKUP

generalization of Gromov's method:

$$div(r) + b^T s = \eta$$

in which

$$q = \begin{bmatrix} r \\ s \end{bmatrix}$$

$$r = A\left[\frac{\partial \eta}{\partial x}\right]^T + B\left[\frac{\partial \theta}{\partial x}\right]^T + \beta(x)$$

in which A and B are arbitrary skew - symmetric matrices,

and the ith component of β is not a function of the ith element of x.

$$s = \frac{b}{\|b\|^2}\eta + \left[I - \frac{bb^T}{\|b\|^2}\right]y$$

where y is an arbitrary vector (with the same dimension as s)

further generalization of Gromov's method:

$$div(r) + b^T s = \eta$$

suppose that we know an exact solution to the PDE :

$$div(\tilde{r}) = \tilde{\eta}$$

add them to get :

$$div(r + \tilde{r}) + b^T s = \eta + \tilde{\eta}$$

we can solve this similar to the previous chart :

$$r = A\left[\frac{\partial \eta}{\partial x}\right]^T + B\left[\frac{\partial \theta}{\partial x}\right]^T + \beta(x) - \tilde{r}$$

$$\mathrm{s} = \frac{\mathrm{b}}{\|\mathrm{b}\|^2}(\eta + \tilde{\eta}) + \left[I - \frac{bb^T}{\|b\|^2}\right]y$$

even further generalization of Gromov’s method:

$div(r) + b^T s = \eta$

suppose that we know a family of exact solutions to the PDE :

$div(\tilde{r}) = \tilde{\eta}$

we can use this family to solve our PDE similar to the previous chart :

$$r = A\left[\frac{\partial \eta}{\partial x}\right]^T + B\left[\frac{\partial \theta}{\partial x}\right]^T + \beta(x) - \sum_{\Omega} \tilde{r}_k$$

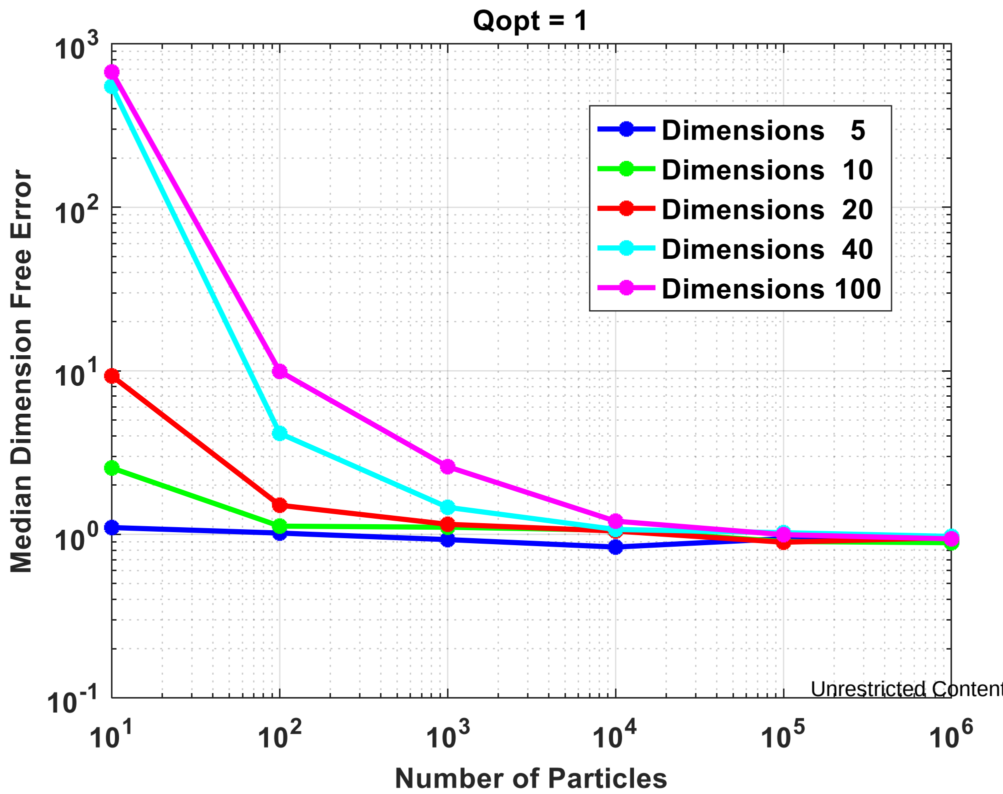
$$s = \frac{b}{\|b\|^2}(\eta + \sum_{\Omega} \tilde{\eta}_k) + \left[I - \frac{bb^T}{\|b\|^2}\right] y$$

more details:

VIDEO of recent talk at Stony Brook University (24 April 2018)

https://youtu.be/vqJGB47XoeY

Daum, Huang & Noushin, "new theory & numerical experiments for Gromov's method," IEEE FUSION Conference, Cambridge England July 2018



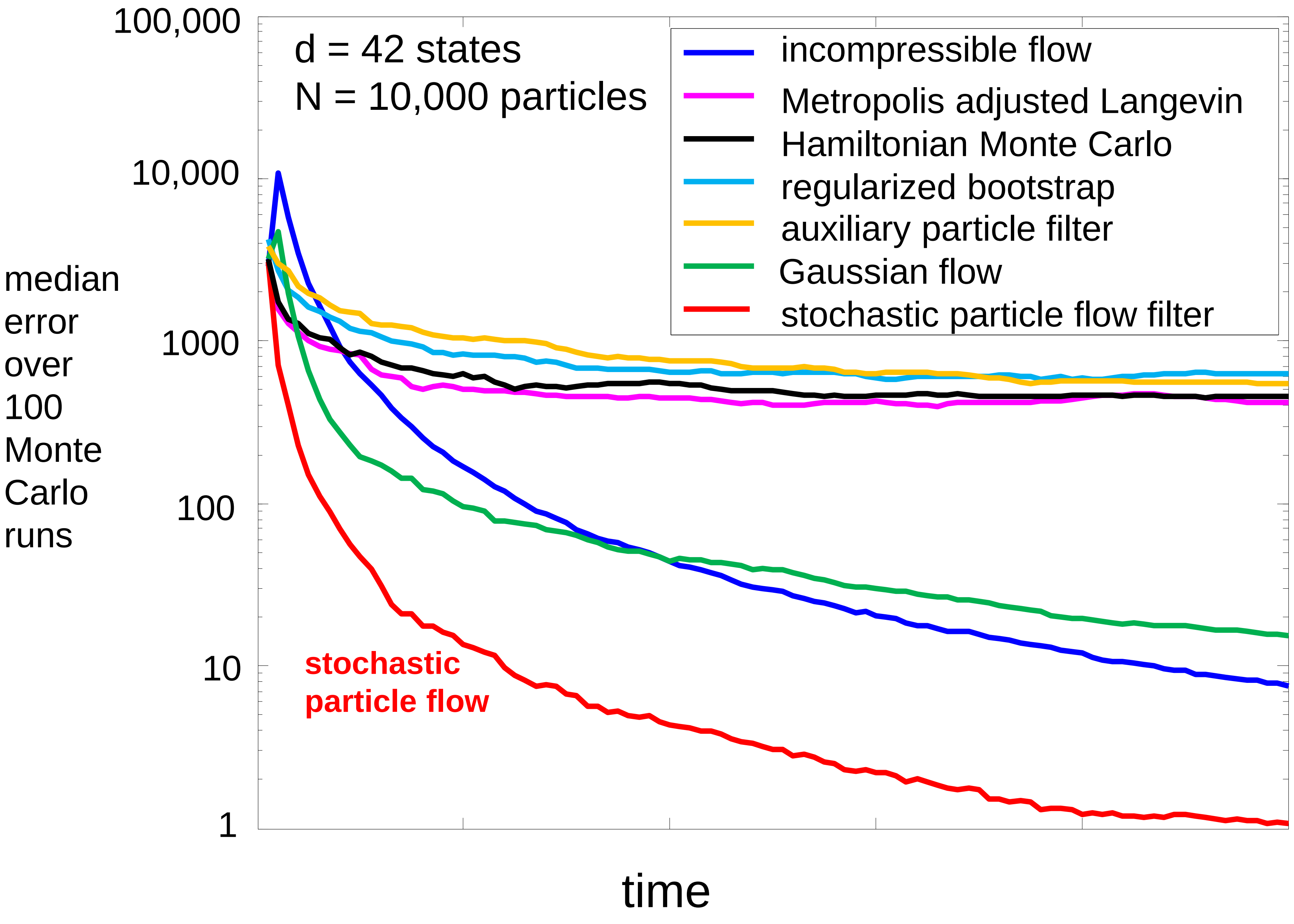
$$\frac{\partial \log h}{\partial x} = -f^T \frac{\partial^2 \log p}{\partial x^2} - \frac{\partial div(f)}{\partial x} - \frac{\partial \log p}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left\{ div\left[Q \frac{\partial p}{\partial x} \right] / p \right\}$$

$$f = -\left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \left(\frac{\partial \log h}{\partial x} \right)^T$$

$$Q = -\left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \frac{\partial^2 \log h}{\partial x^2} \left[\frac{\partial^2 \log p}{\partial x^2} \right]^{-1}$$

new nonlinear filter: particle flow

new particle flow filter	standard particle filters
many orders of magnitude faster than standard particle filters for difficult high dimensional problems	suffers from curse of dimensionality due to particle degeneracy
Bayes' rule is computed using particle flow (like physics)	Bayes' rule is computed using a pointwise multiplication of two functions
no proposal density	depends on proposal density (e.g., Gaussian from EKF or UKF or other)
no resampling of particles	resampling is needed to repair the damage done by Bayes' rule
embarrassingly parallelizable	suffers from bottleneck due to resampling
computes log of unnormalized density	suffers from severe numerical problems due to computation of normalized density
avoid normalization of conditional density & mitigate stiffness of flow	pick good proposal density for resampling (e.g., bootstrap or EKF or UKF)
stochastic particle flow	ad hoc roughening or rejuvenation of particles (covariance inflation)
assumes smooth nowhere vanishing densities (and exploits such regularity)	does not exploit any smoothness or other regularity of densities or functions


100,000
10,000
1000
100
10
1
median error over 100 Monte Carlo runs
d = 42 states
N = 10,000 particles
incompressible flow
Metropolis adjusted Langevin
Hamiltonian Monte Carlo
regularized bootstrap
auxiliary particle filter
Gaussian flow
stochastic particle flow filter
stochastic particle flow
time

how to mitigate stiffness in ODEs for certain particle flows*

method	computational complexity	filter accuracy	comments
1. use a stiff ODE solver (e.g., implicit integration rather than explicit)	large to extremely large	uncertain	standard textbook advice
2. use very small integration steps everywhere	extremely large	good	brute force solution
3. use very small integration steps only where needed (adaptively determined)	small to medium	2^{nd} best	Shozo Mori & Daum (2016)
4. use very small integration steps only where needed (determined non-adaptively)	small	3rd best	easy to do with particle flow
5. transform to principal coordinates or approximately principal coordinates	small	best	easy for certain applications
6. Battin's trick (i.e., sequential scalar measurement updates)	small	very bad	destroys particle flow
7. Tychonov regularization of the Hessian of log p	very small	often helps	
8. shrinkage of the Hessian of log p	very small	often helps	Khan & Ulmke (2015)

*Daum & Huang, "seven dubious methods to mitigate stiffness in particle flow for nonlinear filters," Proceedings of SPIE Conference, May 2014.

algorithm	randomness	comment
bootstrap particle filter (1993)	roughening & resampling	ad hoc randomness
other standard particle filters	roughening & resampling	ad hoc randomness
optimal transport	none	rigorous math theorems
Reich’s optimal transport particle filters	rejuvenation	ad hoc randomness
early ensemble Kalman filters (1994)	none	did not work well for many problems
mature ensemble Kalman filters (1998)	artificial measurement noise	fixes problems of early ensemble Kalman filters
early particle flow filters	none	covariance optimistic for many problems
stochastic particle flow filters (2016)	non-zero diffusion for Bayes’ rule	principled math derivation of stochastic flow & improved accuracy & covariance consistency

item	deep learning	particle flow
purpose	learning & decisions	learning & estimation & decisions
interesting wrinkle (which annoys many people)	lack of uniqueness of solution for highly non-convex loss functions	lack of uniqueness for solution of highly underdertermined transport PDE
architecture	many layers	many steps in log-homotopy
fundamental issues	curse of dimensionality & ill-conditioning & singularity of Hessian	curse of dimensionality & ill-conditioning & singularity of Hessian
tools	stochastic gradient or natural gradient	stochastic natural gradient
representation of geometry	Hessian of loss function (log p)	Hessian of log p
useful theory to explain performance	none	none
performance evaluation	numerical experiments	numerical experiments
theory of design	ersatz Bayesian	echt Bayesian
computers of choice today	GPUs	GPUs
regularization	random dropout & sparsity of coupling between layers and within layers	Tychonov regularization or shrinkage or preferred coordinate system
key adaptive method	adaptive learning rate	adaptive step size in λ
dynamics of learning	backpropagation (i.e., chain rule)	Fokker-Planck equation (i.e., chain rule)

BIG DIG (17 million cubic yards of dirt, one million truckloads & $24 billion)*

*Daum & Huang, “particle flow & Monge-Kantorovich transport,” proceedings of FUSION conference, Singapore, July 2012.

superb books on transport theory

Cédric Villani, "Topics in optimal transportation," AMS Press 2003.

Very clear & accessible introduction; wonderful book!

Cédric Villani, "Optimal transport: old & new," Springer-Verlag 2009. More detailed & rigorous math; free on internet!

history of mathematics

1. creation of the integers

2. invention of counting

3. invention of addition as a fast method of counting

4. invention of multiplication as a fast method of addition

5. invention of particle flow as a fast method of multiplication*

(1) Fred Daum, Jim Huang & Arjang Noushin, “Gromov’s method for Bayesian stochastic particle flow: a simple exact formula for Q,” Proceedings of IEEE Conference on Multisensor Data Fusion, Baden-Baden, September 2016.

(2) Fred Daum, “nonlinear filters: beyond the Kalman filter," IEEE Aerospace & Electronic Systems Magazine special tutorial, pages 57-69, August 2005.

(3) Fred Daum and Jim Huang, “particle flow with non-zero diffusion for nonlinear filters," Proceedings of SPIE conference, San Diego, August 2013.

(4) Fred Daum & Jim Huang, “seven dubious methods to mitigate stiffness in particle flow with non-zero diffusion for nonlinear filters, Bayesian decisions and transport” Proceedings of SPIE Conference, Baltimore, April 2014.

(5) Fred Daum and Jim Huang, “particle flow and Monge-Kantorovich transport," Proceedings of IEEE FUSION Conference, Singapore, July 2012.

(6) Fred Daum & Jim Huang, “how to avoid normalization of particle flow for nonlinear filters, Bayesian decisions and transport,” Proceedings of SPIE conference, Baltimore, May 2014.

(7) Muhammad Khan & Martin Ulmke, “improvements in the implementation of log-homotopy based particle flow filters,” Proceedings of IEEE FUSION Conference, Washington DC, July 2015.

(8) Yunpeng Li & Mark Coates, “particle filtering with invertible particle flow,” IEEE Transactions on Signal Processing, preprint, February 2017.

(9) Pete Bunch & Simon Godsill, “approximations of the optimal importance density using Gaussian particle flow importance sampling,” Journal of American Statistical Association, 2015.

(10) Jeremy Heng, Arnaud Doucet & Yvo Pokern, “Gibbs flow for approximate transport with applications to Bayesian computation,” Journal of Statistics, Royal Society, December 2015.

(11) Peter Bickel, Bo Li & Thomas Bengsston, “sharp failure rates for the bootstrap particle filter in high dimensions,” in “pushing the limits of contemporary statistics,” edited by J. K. Ghosh, IMS Press, 2008.

(12) Paul Bui Quang, Christian Musso & Francois Le Gland, “an insight into the issue of dimensionality in particle filtering,” IEEE Proceedings of FUSION Conference, 2010.

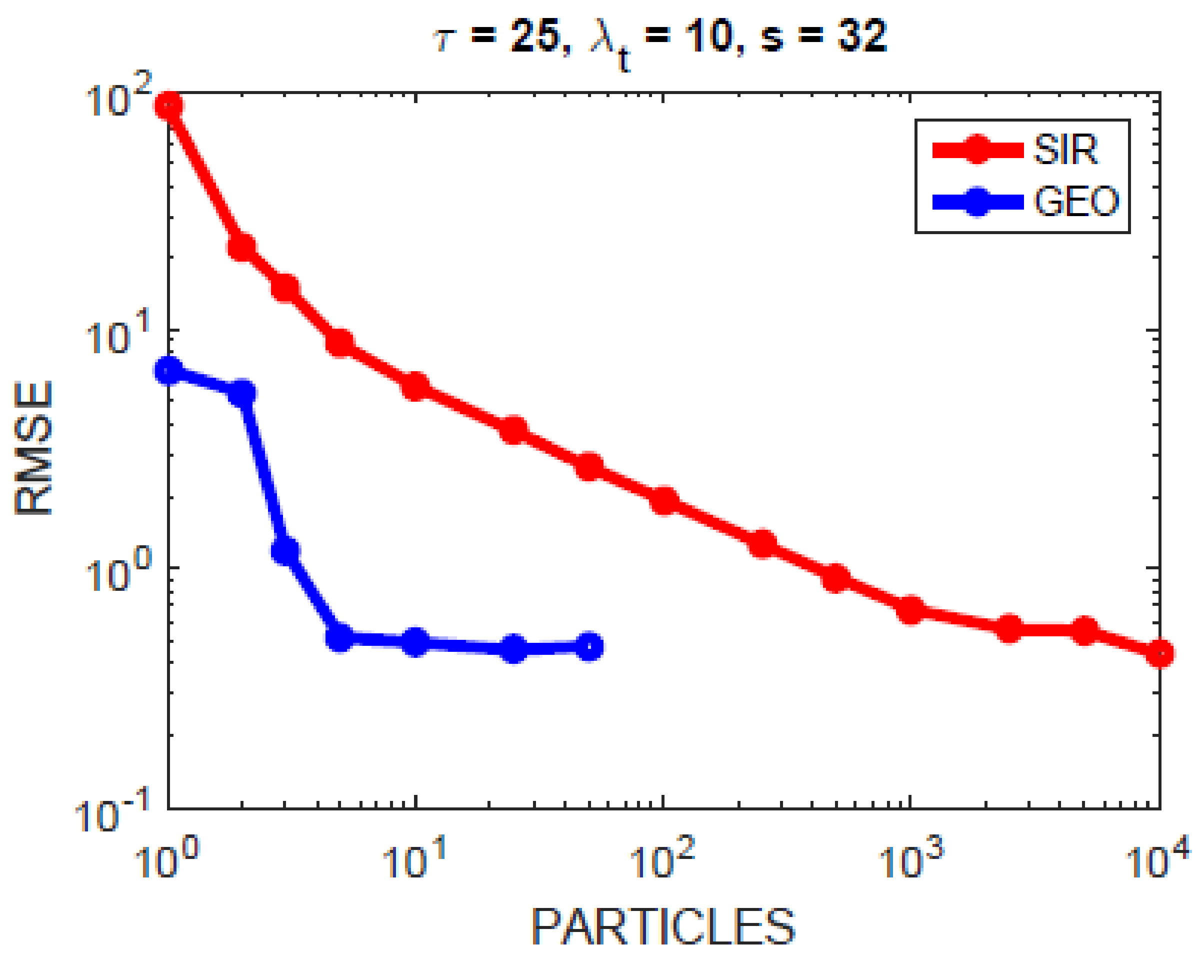
derivation of particle flow with Q ≠ 0:

$$\left[\log h - \frac{d\log K}{d\lambda}\right] = -div(f) - \frac{\partial \log p}{\partial x} f + \frac{1}{2p} div\left[Q(x)\frac{\partial p}{\partial x}\right]$$

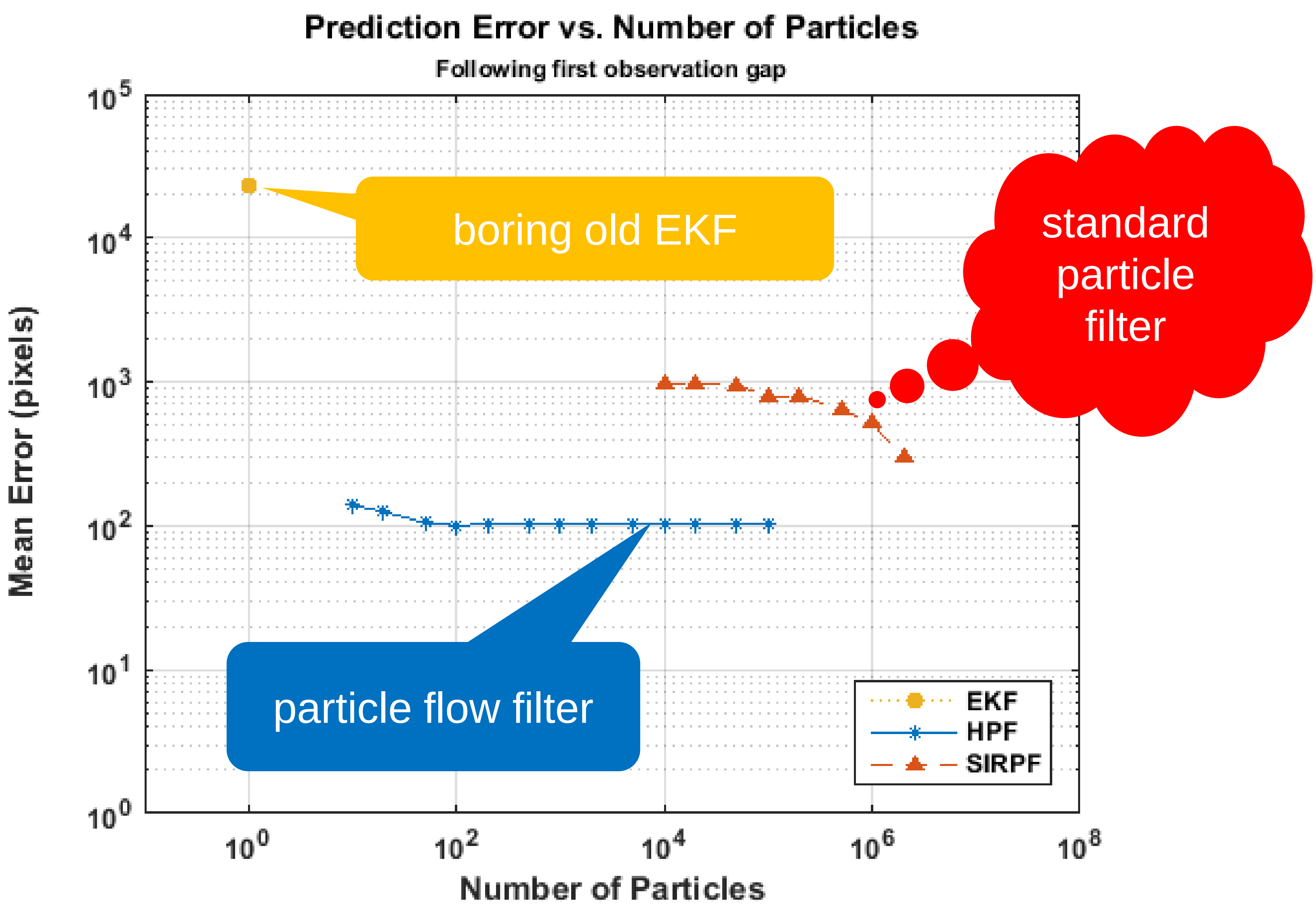
$$\frac{\partial \log h}{\partial x} = -f^T \frac{\partial^2 \log p}{\partial x^2} - \frac{\partial div(f)}{\partial x} - \frac{\partial \log p}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left\{div\left[Q(x)\frac{\partial p}{\partial x}\right]/p\right\}$$

assuming that Q is a constant positive multiple of the identity matrix, i.e., Q = αI, and approximating f using small curvature flow & natural gradient flow, we get:

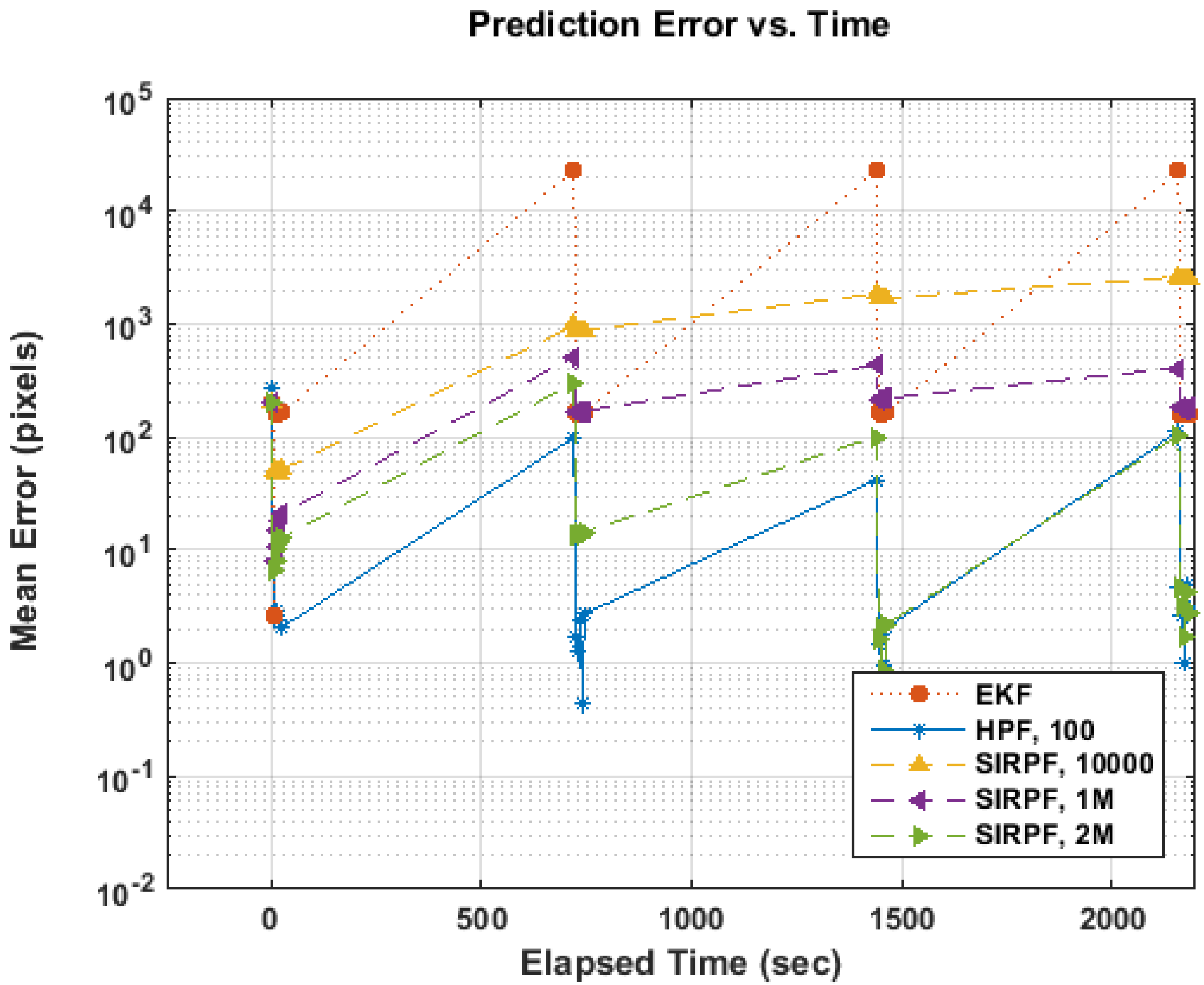
$$\alpha \approx \frac{2\left\|\frac{\partial \log p}{\partial x}\frac{\partial \hat{f}}{\partial x}\right\|}{\left\|\frac{\partial}{\partial x}\left\{div\left[\frac{\partial p}{\partial x}\right]/p\right\}\right\|}$$



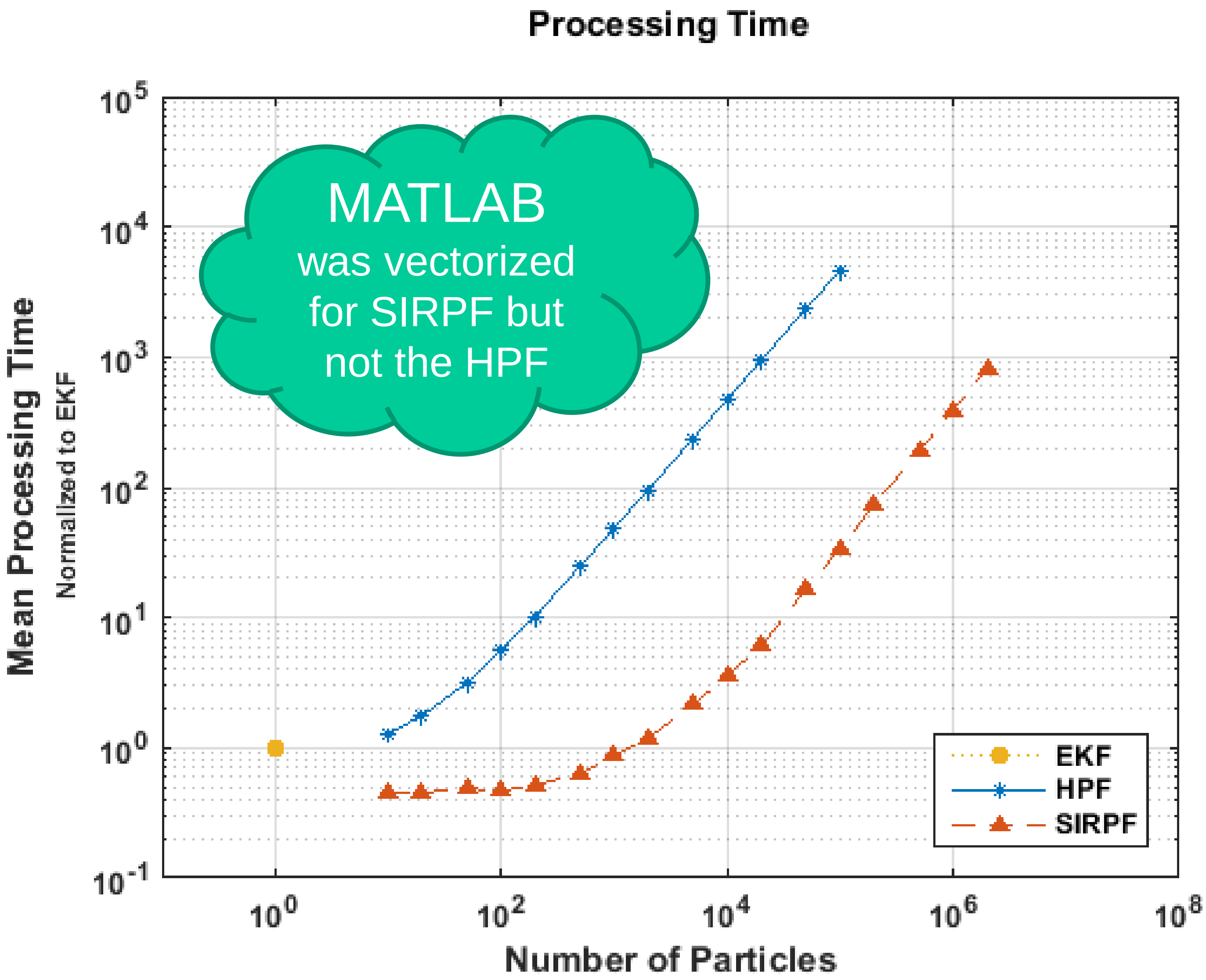
Chris Kreucher, “A Geodesic Flow Particle Filter for Non-Thresholded Measurements,” 14 October 2016.



Nima Moshtagh, Jonathan Chan, Moses Chan, "Homotopy Particle Filter for Ground-Based Tracking of Satellites at GEO," AMOS Conference, Hawaii September 2016.



Nima Moshtagh, Jonathan Chan, Moses Chan, "Homotopy Particle Filter for Ground-Based Tracking of Satellites at GEO," AMOS Conference, Hawaii September 2016.

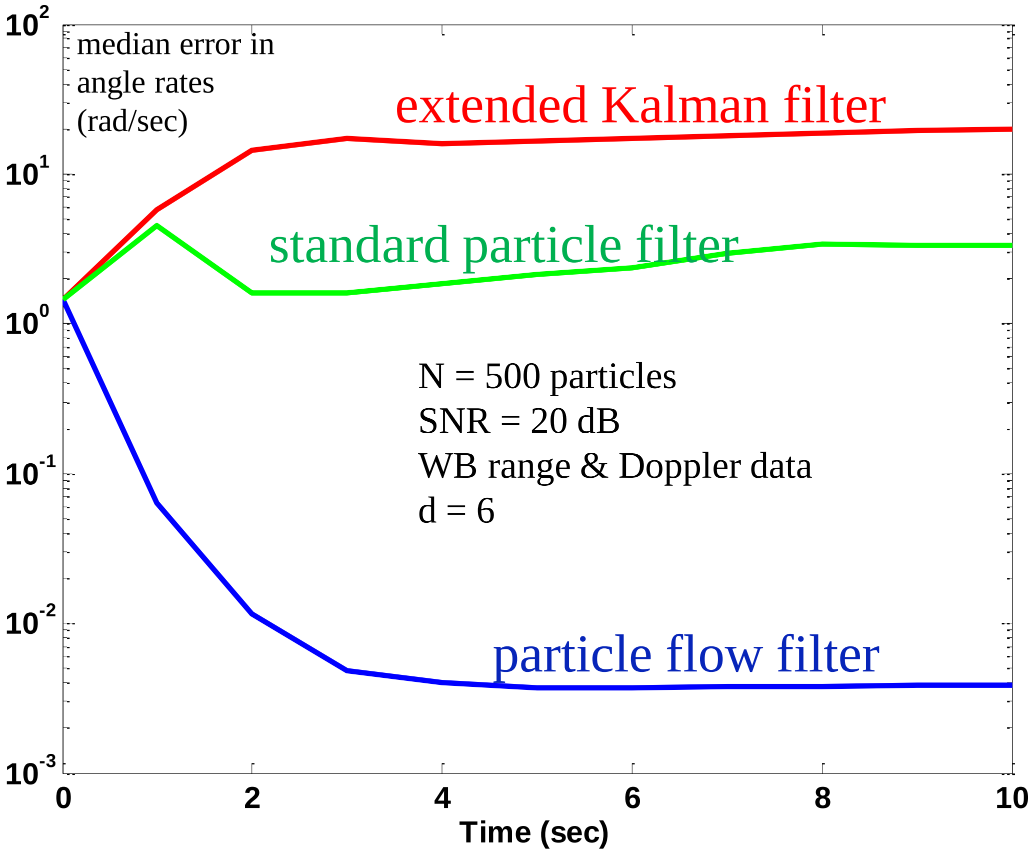


Nima Moshtagh, Jonathan Chan, Moses Chan, "Homotopy Particle Filter for Ground-Based Tracking of Satellites at GEO," AMOS Conference, Hawaii September 2016.

new filter improves angle rate estimation accuracy by two or three orders of magnitude

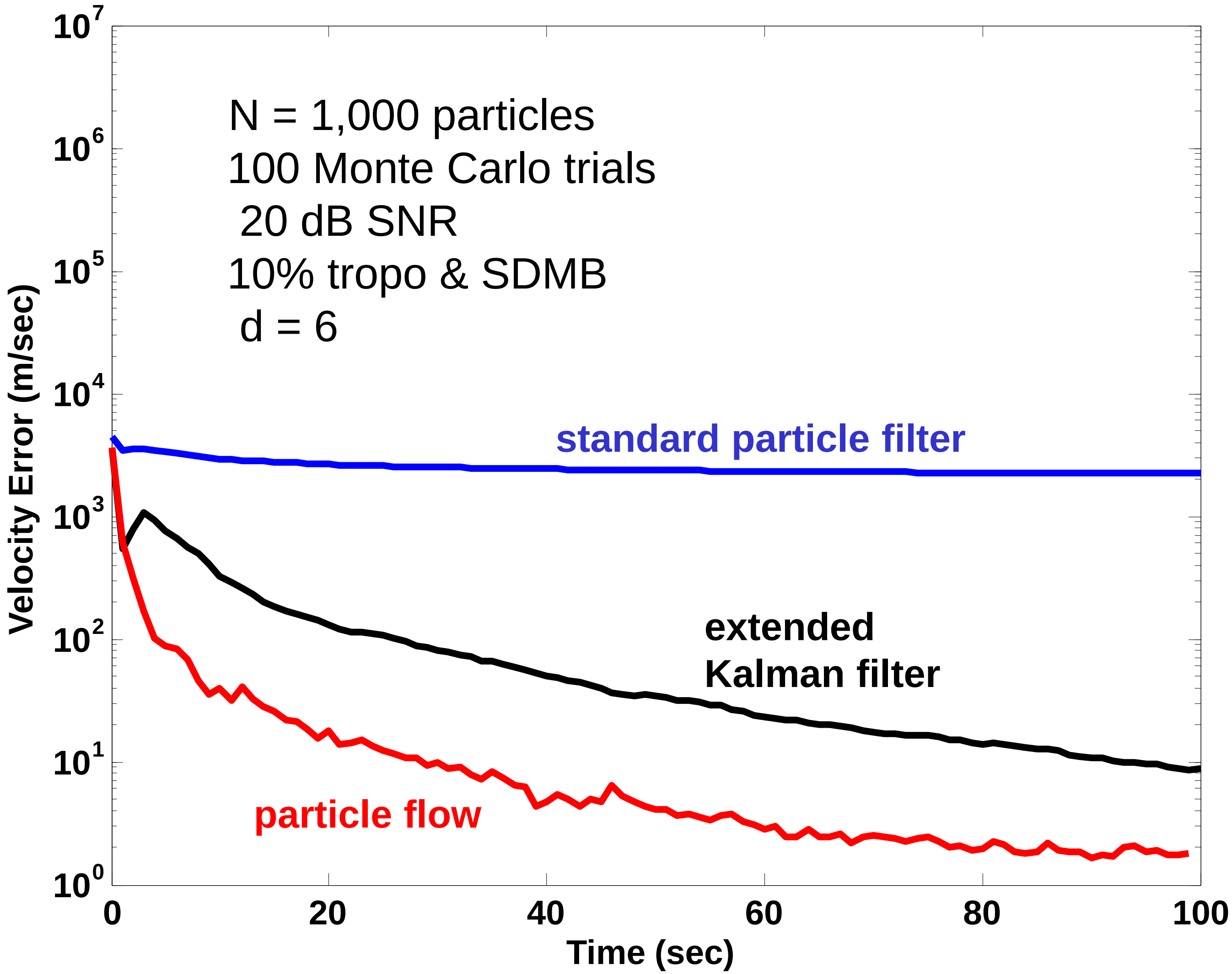
highly nonlinear dynamics:

$$I_1\dot{\omega}_1 + (I_3 - I_2)\omega_3\omega_2 = M_1$$
$$I_2\dot{\omega}_2 + (I_1 - I_3)\omega_1\omega_3 = M_2$$
$$I_3\dot{\omega}_3 + (I_2 - I_1)\omega_2\omega_1 = M_3$$

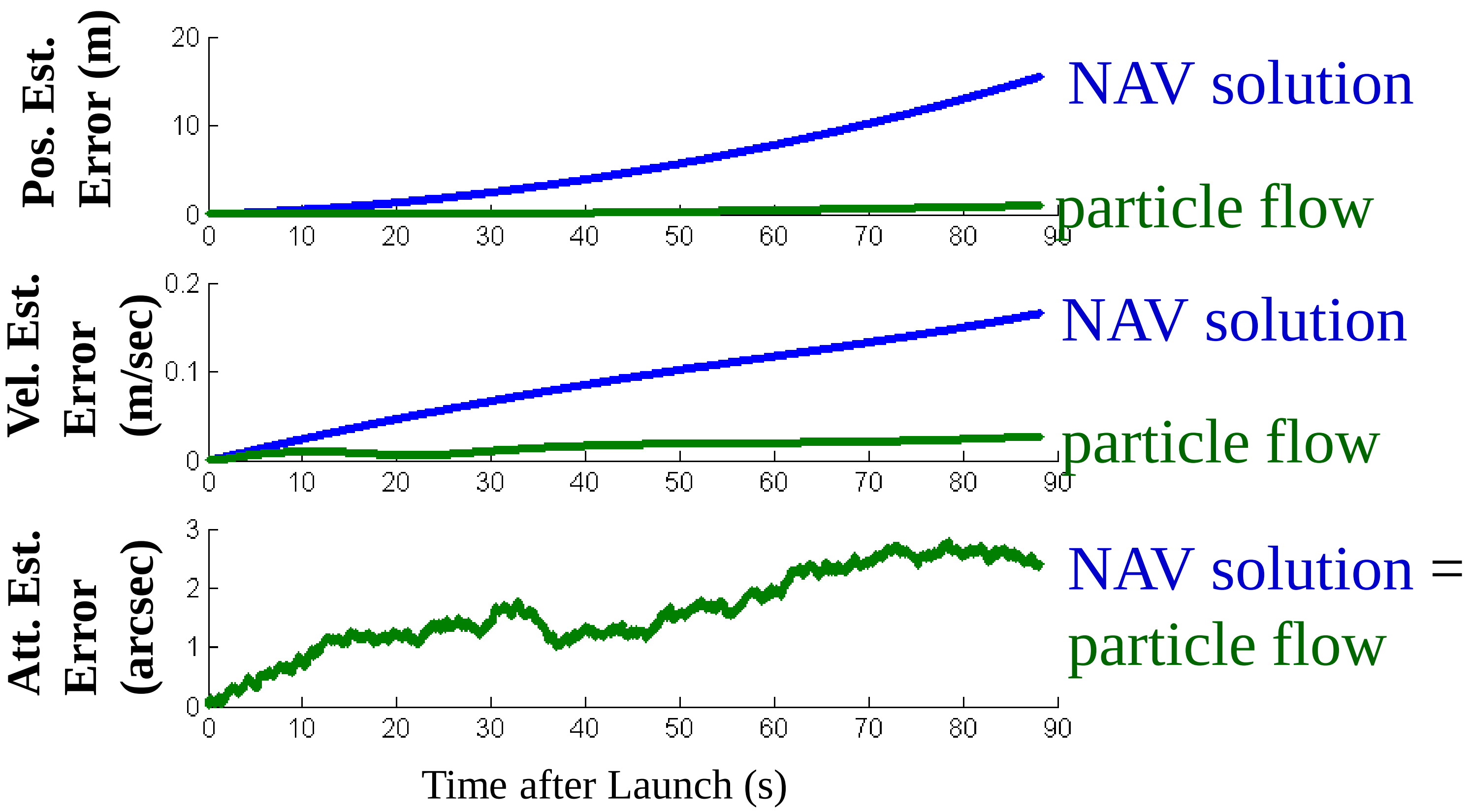


extended Kalman filter diverges because it cannot model multimodal conditional probability densities accurately

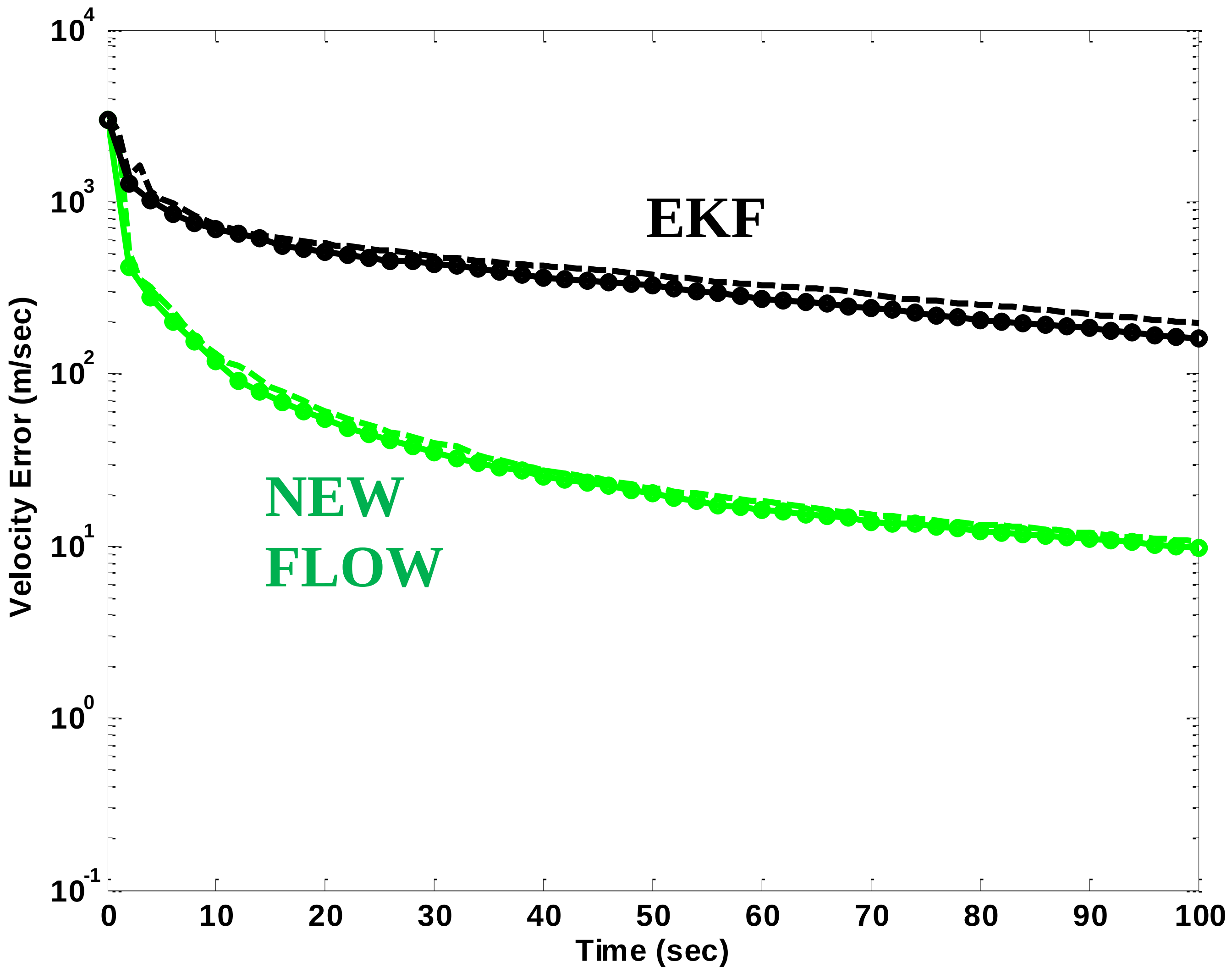
comparison of estimation accuracy for three filters:



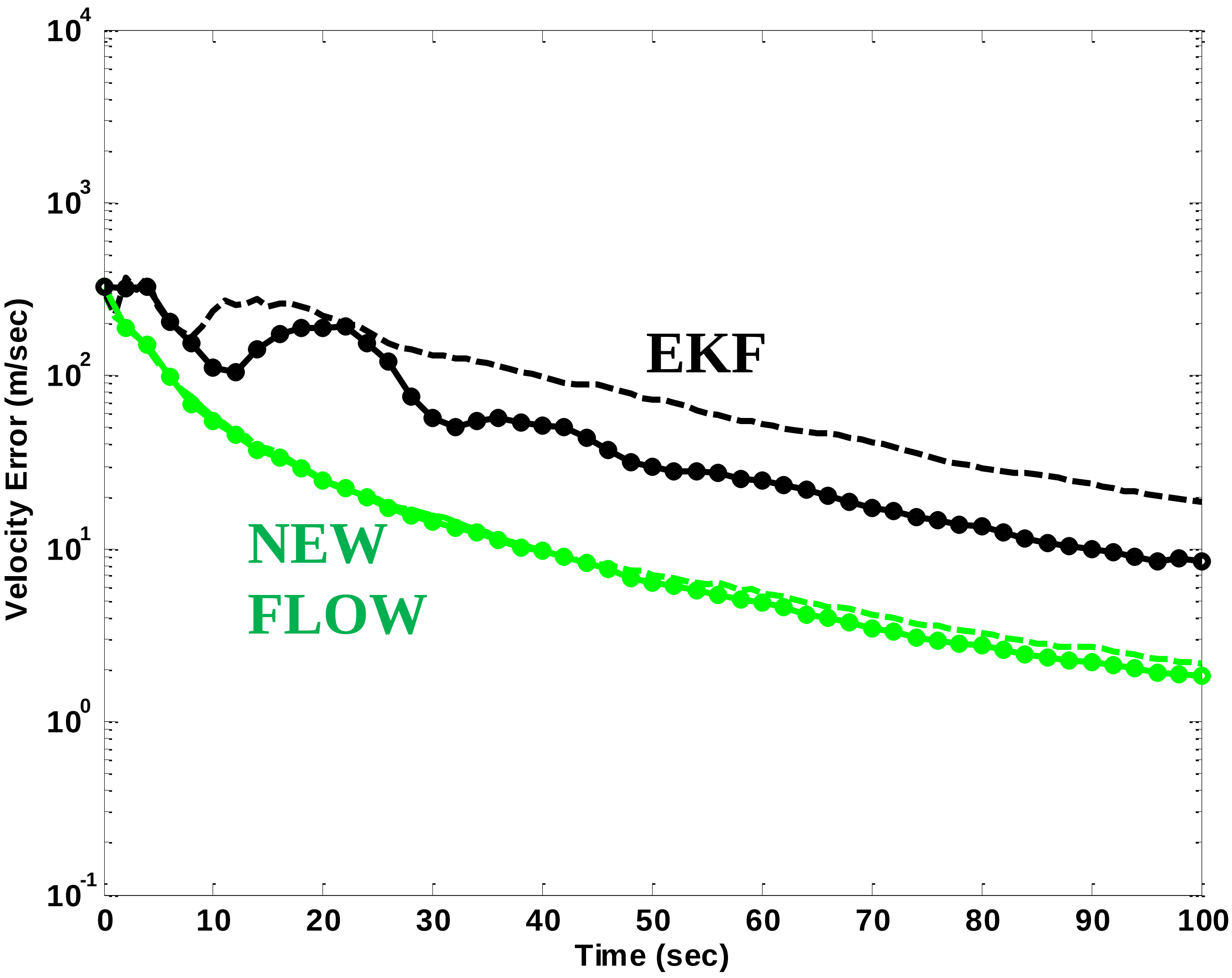
IMU-only Navigation problem (no GPS)



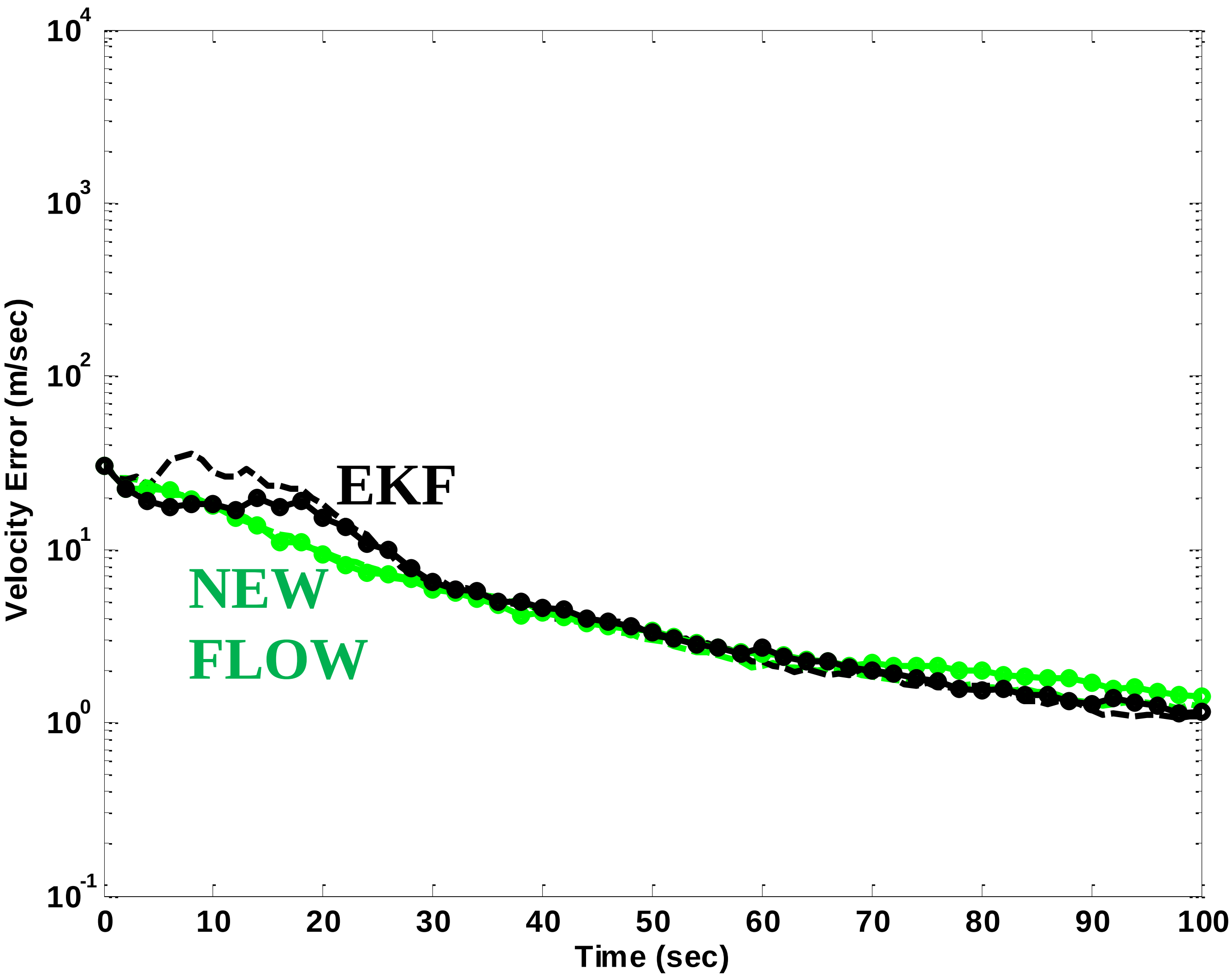
EKF diverges (not shown); d = 15, N = 1000



Velocity Error (m/sec)
Time (sec)
10^4
10^3
10^2
10^1
10^0
10^{-1}
0
10
20
30
40
50
60
70
80
90
100
EKF
NEW
FLOW



EKF
NEW
FLOW
Velocity Error (m/sec)
Time (sec)
10^4
10^3
10^2
10^1
10^0
10^{-1}
0
10
20
30
40
50
60
70
80
90
100



Unrestricted Content

$$div(pf) = p\left[-\log h + \frac{d \log K}{d\lambda}\right]$$

$$\text{let } \mathrm{q} = \mathrm{pf}$$

$$\frac{\partial \mathrm{q}_1}{\partial \mathrm{x}_1} + \frac{\partial q_2}{\partial x_2} + \ldots + \frac{\partial q_d}{\partial x_d} = \eta$$

(1) linear PDE in unknown f or q
(2) constant coefficient PDE in q
(3) first order PDE
(4) highly underdetermined PDE
(5) same as the Gauss law in Maxwell's equations
(6) same as Euler's equation in fluid dynamics
(7) existence of solution if and only if volume integral of η is zero (i.e., neutral charge density for plasma; satisfied automatically)

the N-principle*

*Emily Walsh & Chris Budd, “moving mesh methods for problems in meteorology,” talk at ICIAM Vancouver 2011.

STIFFNESS

What is "stiffness" (in the context of ODEs)?

various definitions of "stiff" ODE:

1. An ODE is "stiff" if certain numerical integration methods are unstable unless we use an extremely small step size.
2. An ODE is "stiff" if explicit methods for numerical integration do not work well.
3. An ODE is "stiff" if the Jacobian matrix of the flow is ill-conditioned.
4. An ODE is "stiff" if the solution changes rapidly over a time scale that is short compared with the time interval of interest.
5. Stiff ODEs are evil.

geodesic particle flow :

$$\frac{dx}{d\lambda} = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1}\left(\frac{\partial \log h}{\partial x}\right)^T + \frac{dw}{d\lambda}$$

If we approximate the density p as Gaussian, then the observed Fisher information matrix can be computed using the sample covariance matrix (C) over the set of particles:

$$\frac{dx}{d\lambda} \approx C\left(\frac{\partial \log h}{\partial x}\right)^T + \frac{dw}{d\lambda}$$

for Gaussian likelihoods (h) we get the **EKF for each particle**:

$$\frac{dx}{d\lambda} \approx C\left(\frac{\partial \theta(x)}{\partial x}\right)^T R^{-1}\left(z - \theta(x)\right) + \frac{dw}{d\lambda}$$

how to mitigate stiffness in ODEs for certain particle flows*

method	computational complexity	filter accuracy	comments
1. use a stiff ODE solver (e.g., implicit integration rather than explicit)	large to extremely large	uncertain	standard textbook advice
2. use very small integration steps everywhere	extremely large	good	brute force solution
3. use very small integration steps only where needed (adaptively determined)	small to medium	2nd best	Shozo Mori & Daum (2016)
4. use very small integration steps only where needed (determined non-adaptively)	small	3rd best	easy to do with particle flow
5. transform to principal coordinates or approximately principal coordinates	small	best	easy for certain applications
6. Battin's trick (i.e., sequential scalar measurement updates)	small	very bad	destroys particle flow
7. Tychonov regularization of the Hessian of log p	very small	often helps	
8. shrinkage of the Hessian of log p	very small	often helps	Khan & Ulmke (2015)

*Daum & Huang, "seven dubious methods to mitigate stiffness in particle flow for nonlinear filters," Proceedings of SPIE Conference, May 2014.

particle flow with non-uniform non-adaptive integration to mitigate stiffness of the flow

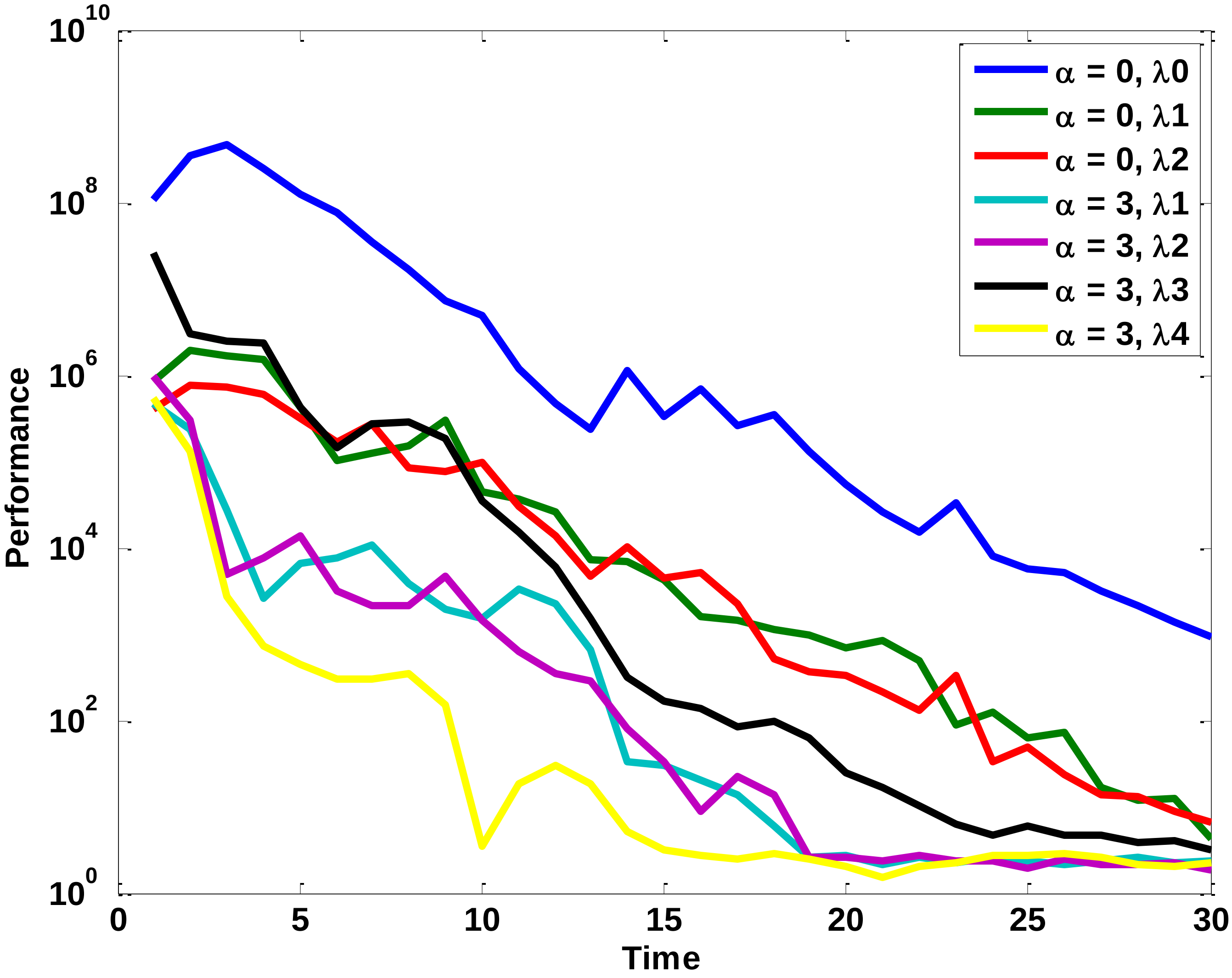
$\lambda 0 = \text{fixed} \quad 100 \text{ steps}$

$\lambda 1 = \text{fixed} \quad 1000 \text{ steps}$

$\lambda 2 = \text{fixed} \quad 370 \text{ steps}$

$\lambda 3 = \text{fixed} \; 10 \text{ steps}$

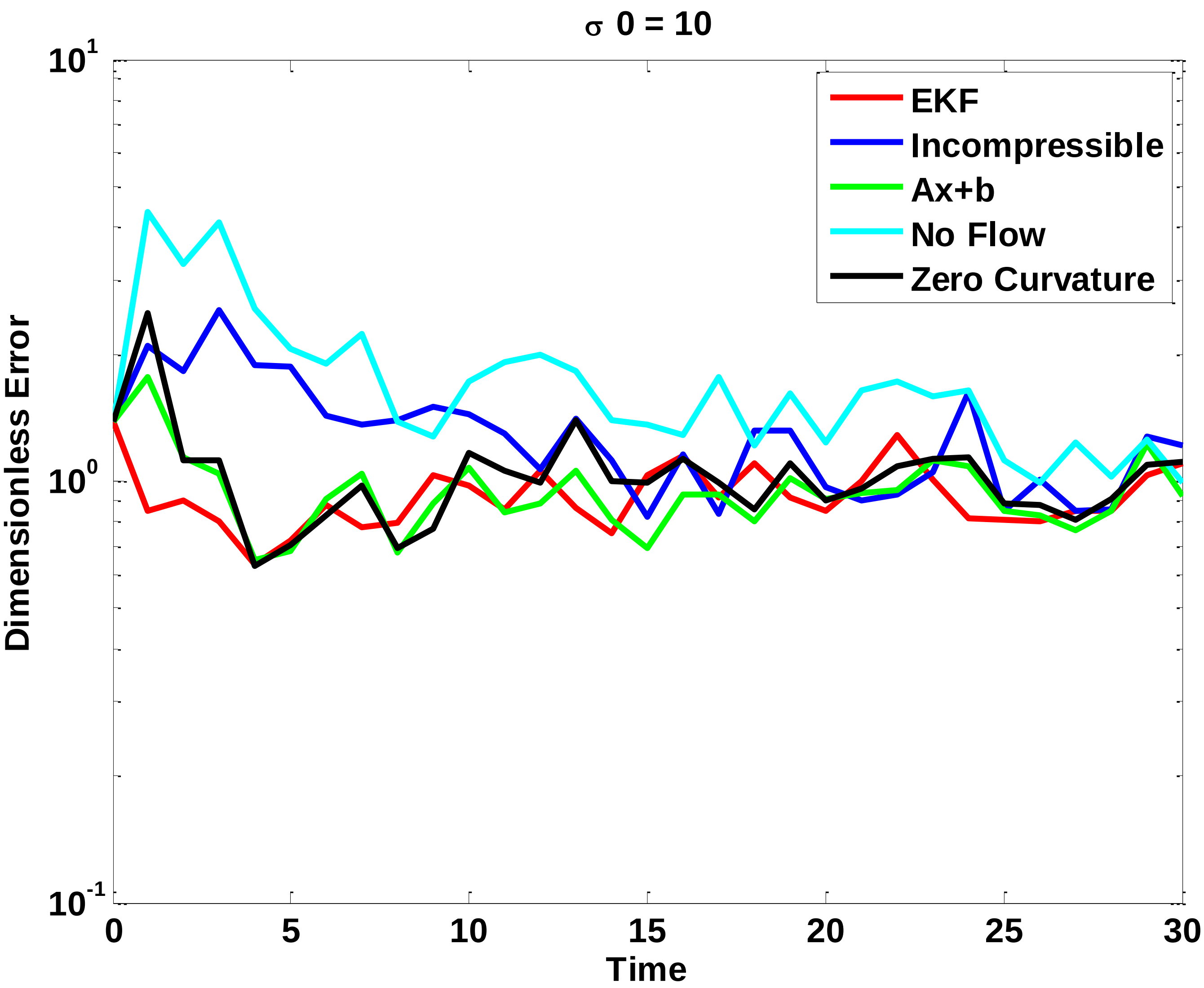
$\lambda 4 = variable \; 29 \text{ steps}$



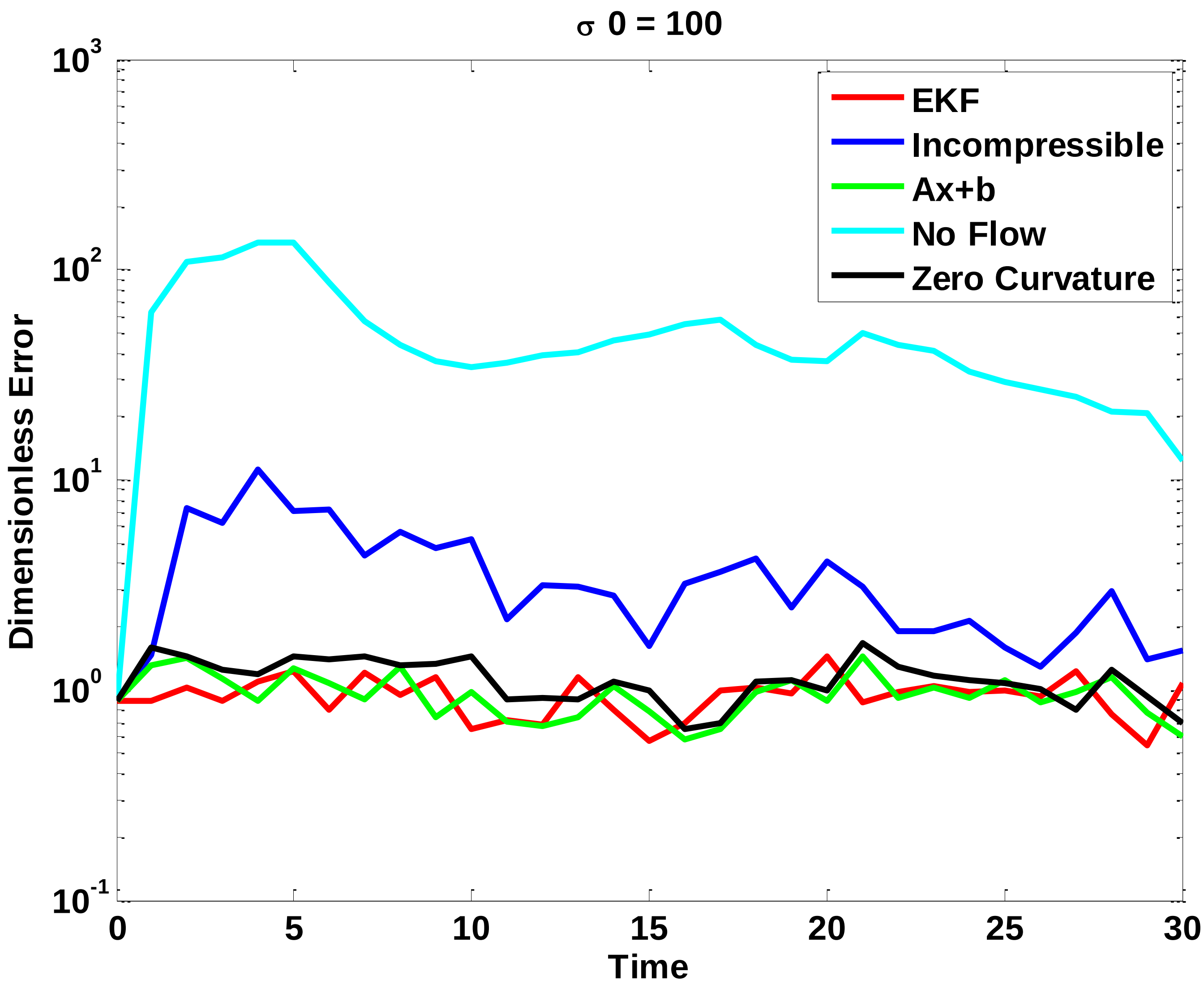
d = 9, N = 500, λ = 0.9, σ0 = 100

lambda =
0
0.001000000000000
0.001279802213998
0.001637893706954
0.002096179992453
0.002682695795280
0.003433320018282
0.004393970560761
0.005623413251903
0.007196856730012
0.009210553176895
0.011787686347936
0.015085907086002
0.019306977288833
0.024709112279856
0.031622776601684
0.040470899507598
0.051794746792312
0.066287031618264
0.084834289824407
0.108571111940220
0.138949549437314
0.177827941003892
0.227584592607479
0.291263265490874
0.372759372031494
0.477058269614393
0.610540229658533
0.781370737651809
1.000000000000000

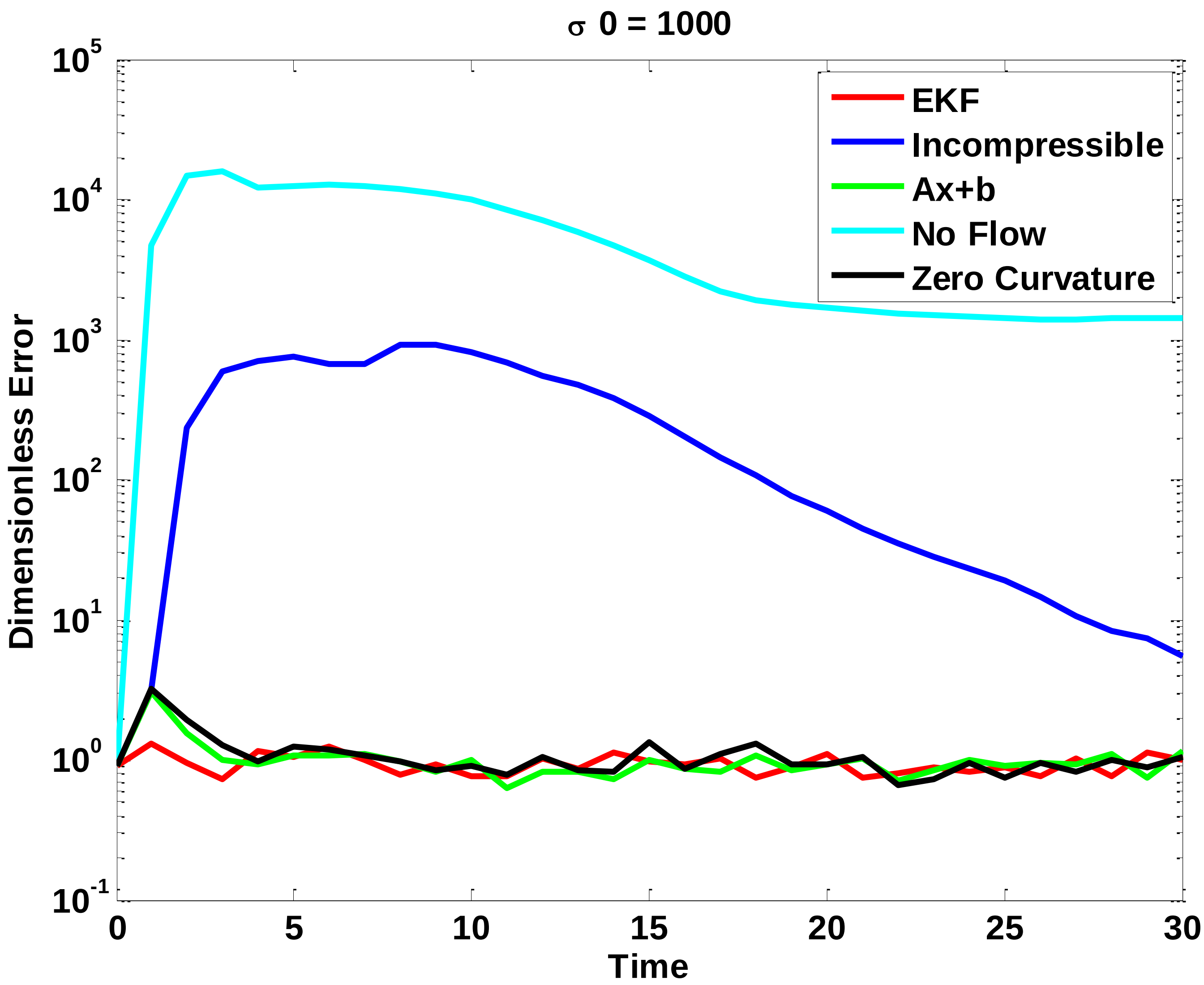
example of non-adaptive
non-uniform step size in lambda



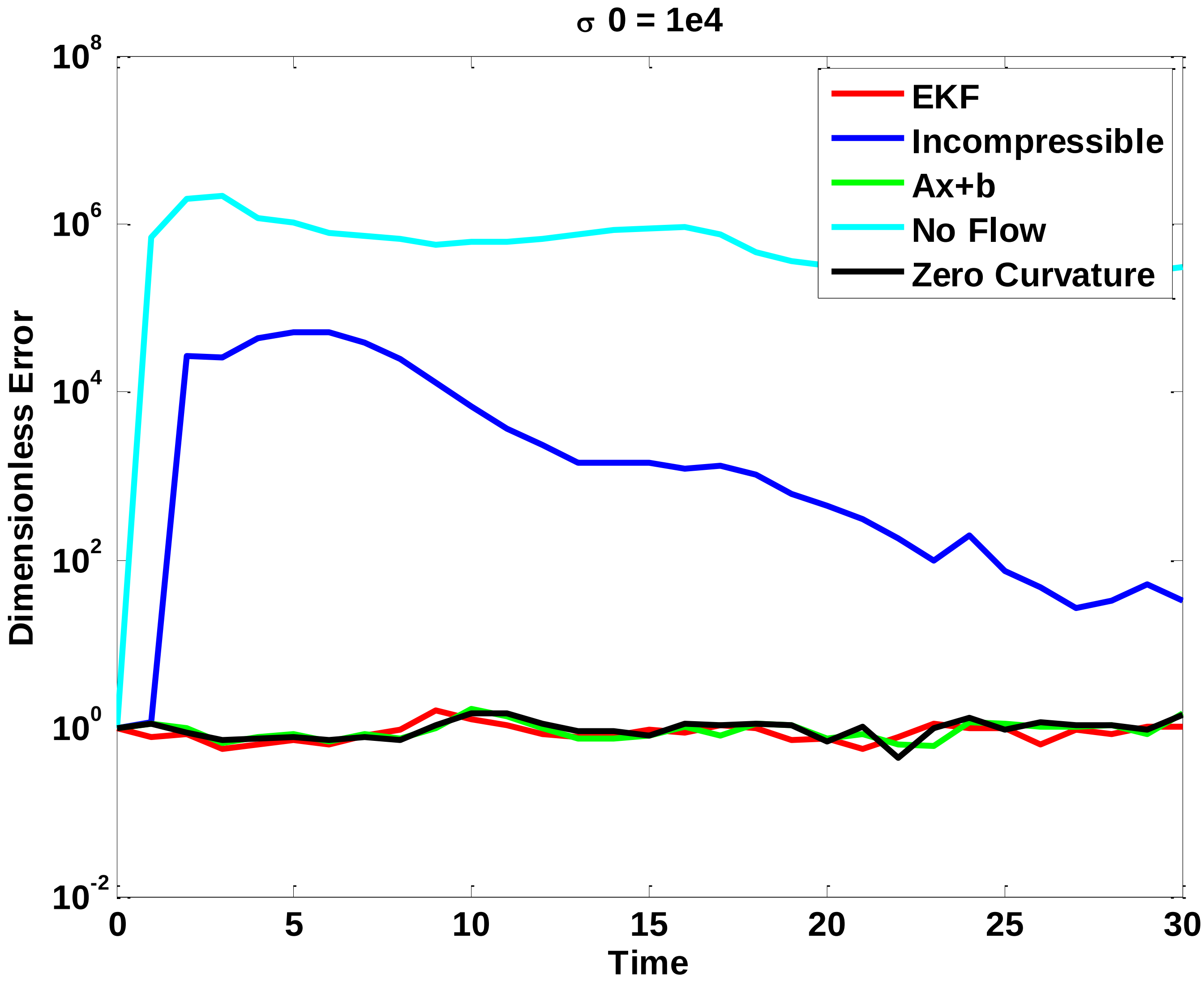
$\Delta\lambda_0$ = 1e-3



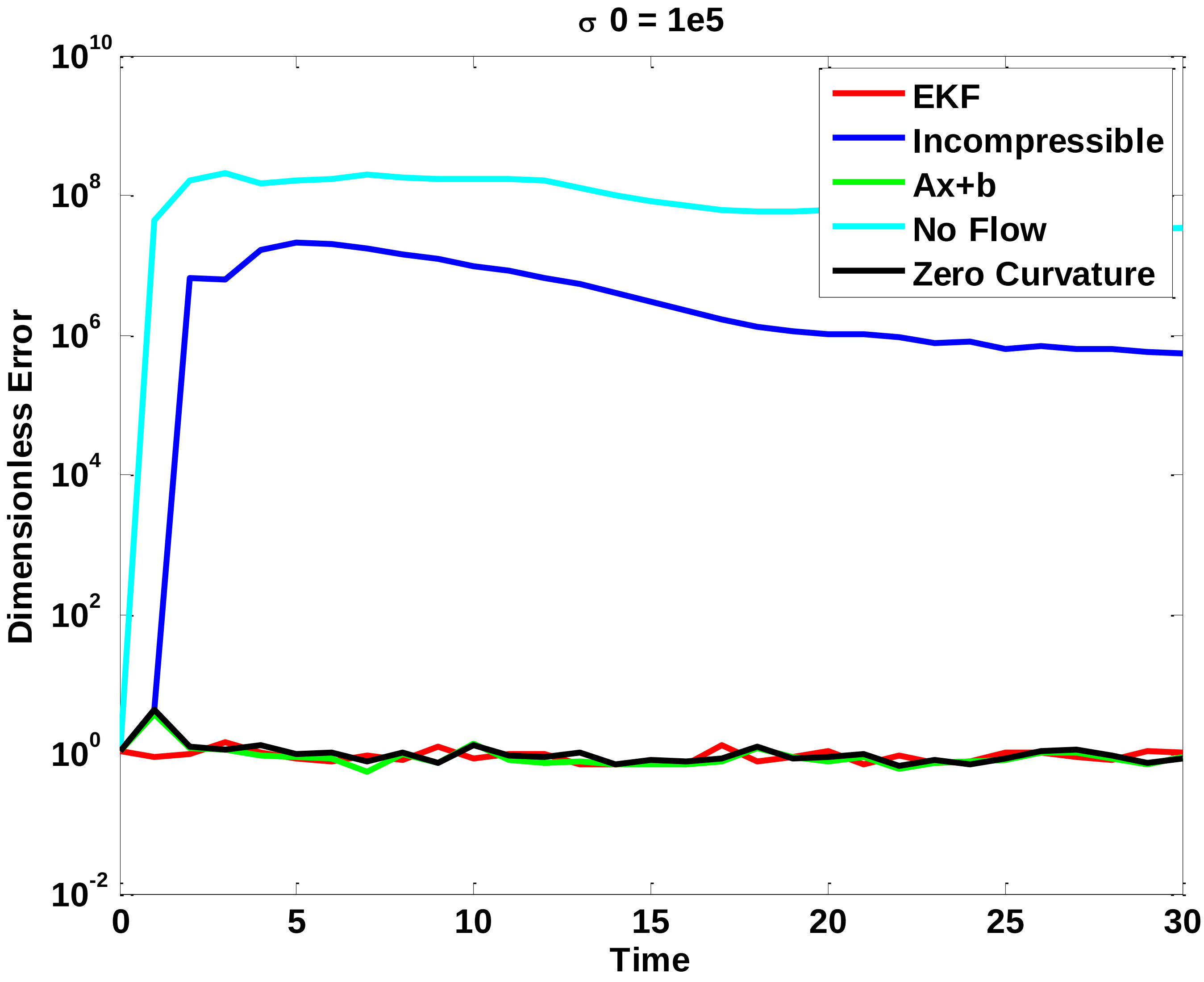
$\Delta\lambda_0 = 1e\text{-}5$



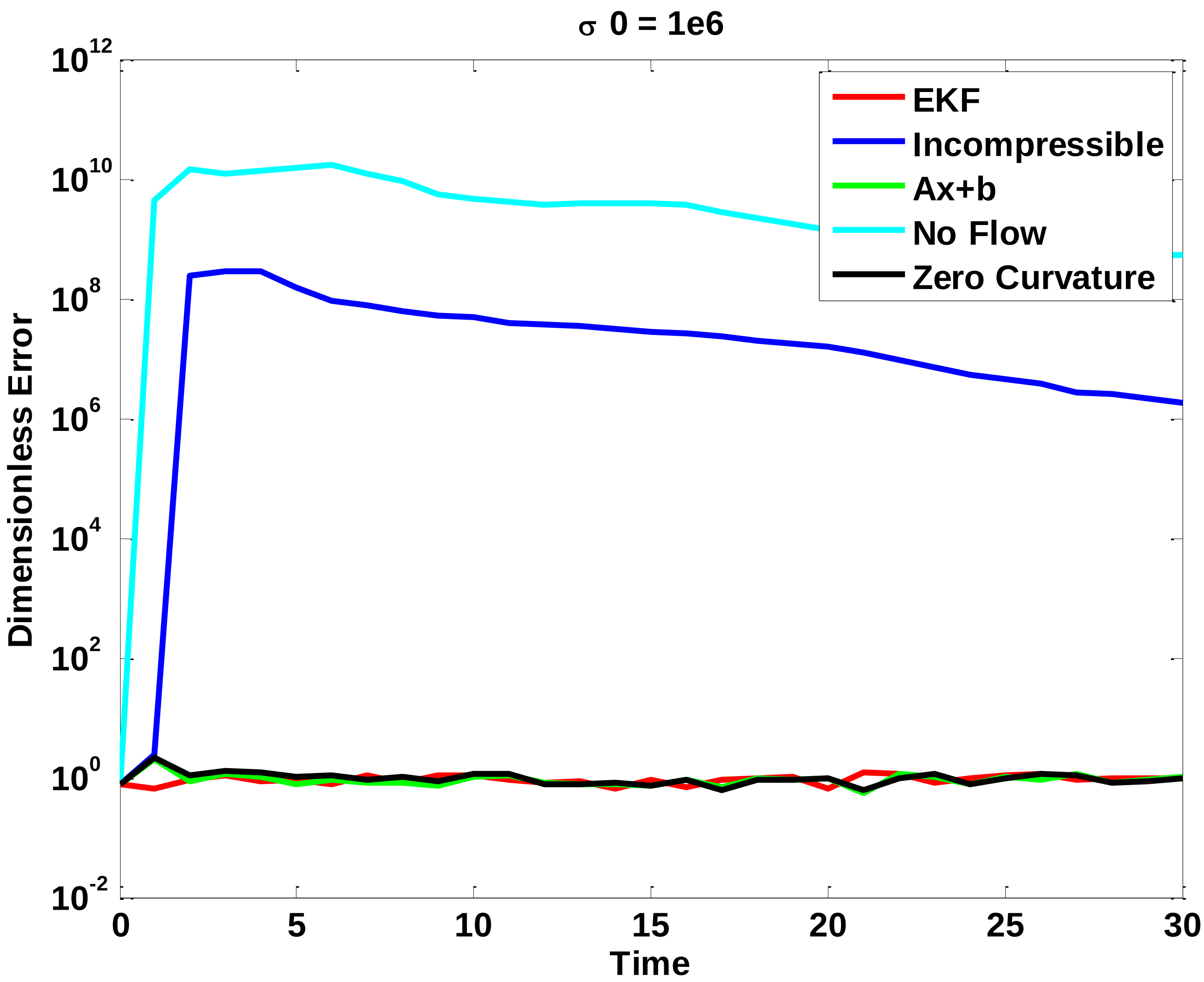
$\Delta\lambda_0 = 1e\text{-}7$



$\Delta\lambda_0$ = 1e-9

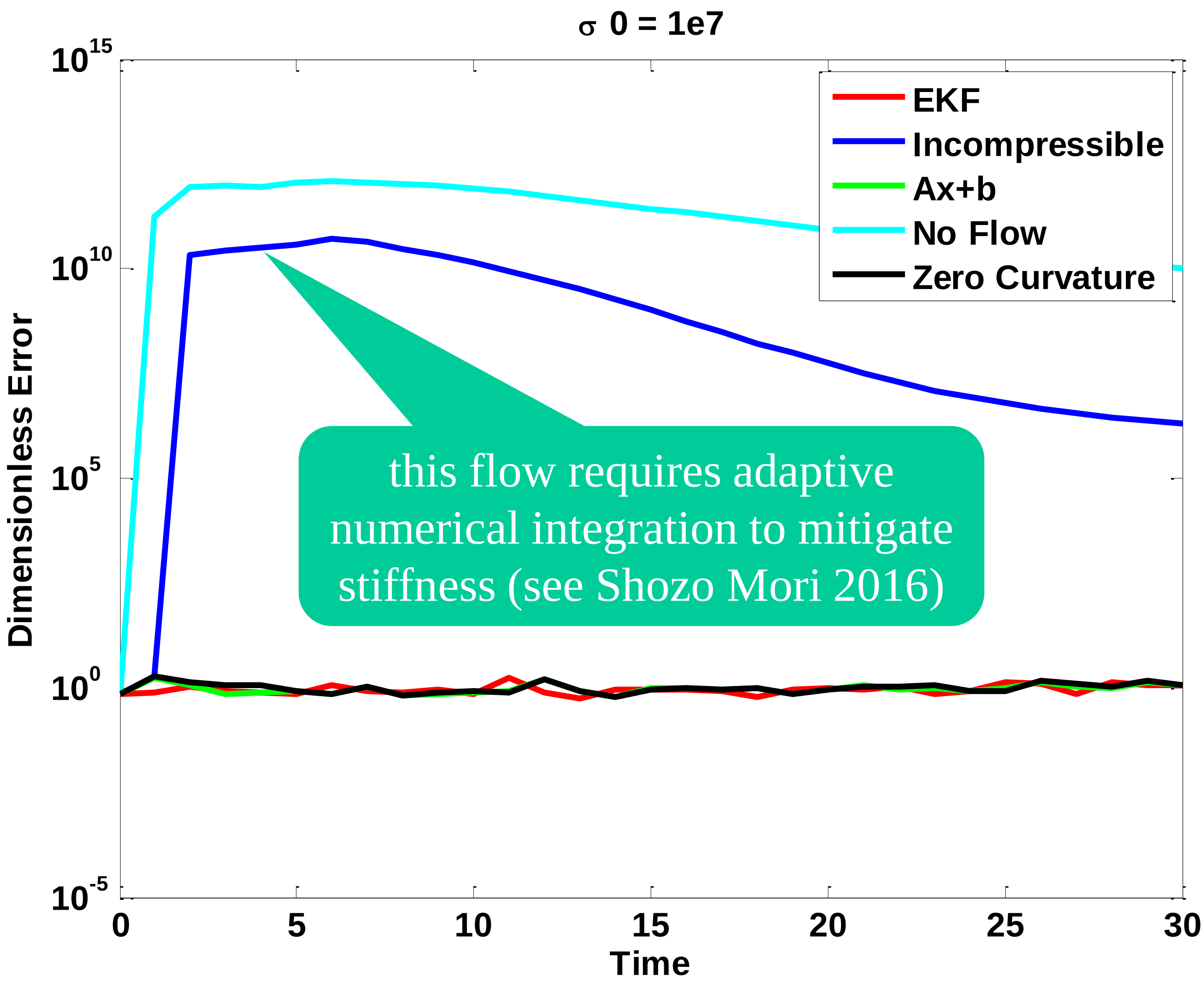


$\Delta\lambda_0 = 1e\text{-}11$



Unrestricted Content

$\Delta\lambda_0$ = 1e-13



$\Delta\lambda_0$ = 1e-15

REFERENCES ON STIFFNESS:

1. Fred Daum & Jim Huang, “seven dubious methods to mitigate stiffness in particle flow for nonlinear filters,” Proceedings of SPIE Conference on signal processing, edited by Oliver Drummond, Baltimore, May 2014.

2. Muhammad Khan & Martin Ulmke, “improvements in the implementation of log-homotopy based particle flow filters,” Proceedings of IEEE FUSION Conference, Washington DC, July 2015.

3. Shozo Mori, Fred Daum & Joel Douglas, “adaptive step size approach to homotopy-based particle filtering Bayesian update,” Proceedings of IEEE FUSION Conference, Heidelberg, July 2016.

4. Shozo Mori, Fred Daum & Joel Douglas, “Algorithm-Based Inertial Measurement Unit Only Navigation Filtering,” MSS National Symposium on Sensor & Data Fusion, October 2017.

5. Kenneth Eriksson, Claes Johnson and Anders Logg, “On explicit time stepping for stiff ODEs,” SIAM Journal of Scientific Computing, volume 25 No. 4, pages 1142-1157, 2003.

6. Dan Kushnir and Vladimir Rokhlin, “a highly accurate solver for stiff ODEs,” Yale math dept. preprint, 2011.

stan v2.10.0

Daniel Lee; Bob Carpenter; Peter Li; Michael Betancourt; maverickg; Marcus Brubaker; Rob Trangucci; Marco Inacio; Alp Kucukelbir; Mitzi Morris; bgoodri; Jeffrey Arnold; Dustin Tran; Matt Hoffman; Stan buildbot; Avraham Adler; Alexey Stukalov; Allen Riddell; Rob J Goedman; Kevin S. Van Horn; Juan Sebastián Casallas; Mike Lawrence; Amos Waterland; Jonah Gabry; Daniel Mitchell; tosh1ki; wds15; Krzysztof Sakrejda; Guido Biele; Damjan Vukcevic

v2.10.0 (17 June 2016) New Team Members

- Aki Vehtari (Aalto Uni) --- GPs, LOO, statistical modeling, MATLAB
- Rayleigh Lei (U. Michigan) --- vectorizing functions
- Sebastian Weber (Novartis) --- diff eq models

- **stiff diff eq solver** CVODES from Sundials
- add control parameters (tolerance, max iteraitons) to ODE solvers
- rename ODE solvers based on algorithm, integrate_ode_rk45 for existing **non-stiff Runge-Kutta solver** and integrate_ode_bdf for the **stiff backward differentiation form**; deprecate the unmarked integrate_ode function (#1886)
- limiting diff eq iterations in solvers (Boost/CVODES)
- unit_vector as parameter (#1713) [it never worked in the past]
- rename multiply_log and log_binomial_coefficient to lmultiply and lchoose (also part of #1811)
- incomplete beta function as inc_beta (#1540)

New Internal Features

- exhaustive HMC (XHMC)
- multinomial variant of NUTS (#1846)
- simplified NUTS criterion (#1852)
- uniform static HMC (#1849)
- **Riemannian HMC** with SoftAbs (#304)

14.7 MB Preview Download

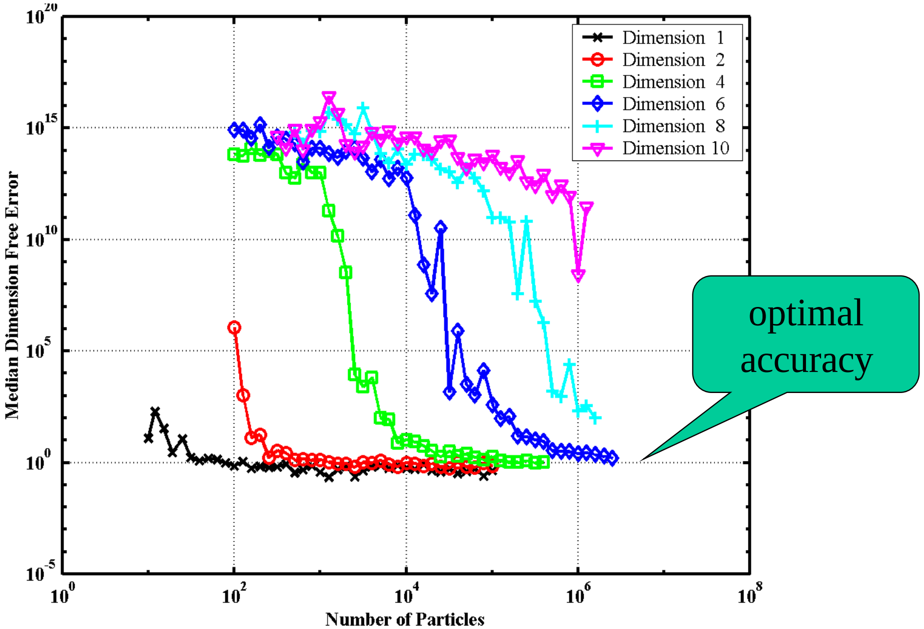
What is Stan?

•"Probabilistic programming language implementing full Bayesian statistical inference"
–MCMC sampling (Hamiltonian MC, NUTS)
–Maximum likelihood estimation (BFGS)
•Coded in C++ and runs on all major platforms
•Open-source software (+ maintained): http://mc-stan.org/
•Standalone software, or interfaces with R, Python, Matlab, Julia
•HMC uses gradient information → less affected by correlations between parameters than random walk MC

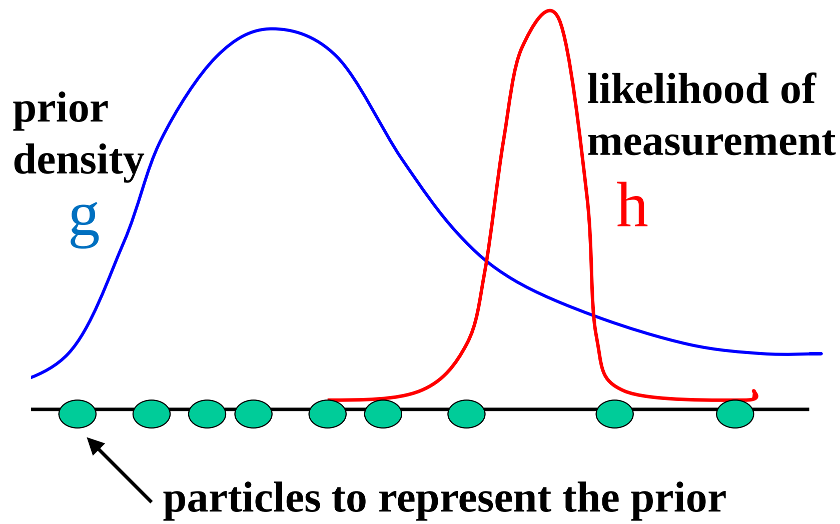
Contributions from ExaScience Lab

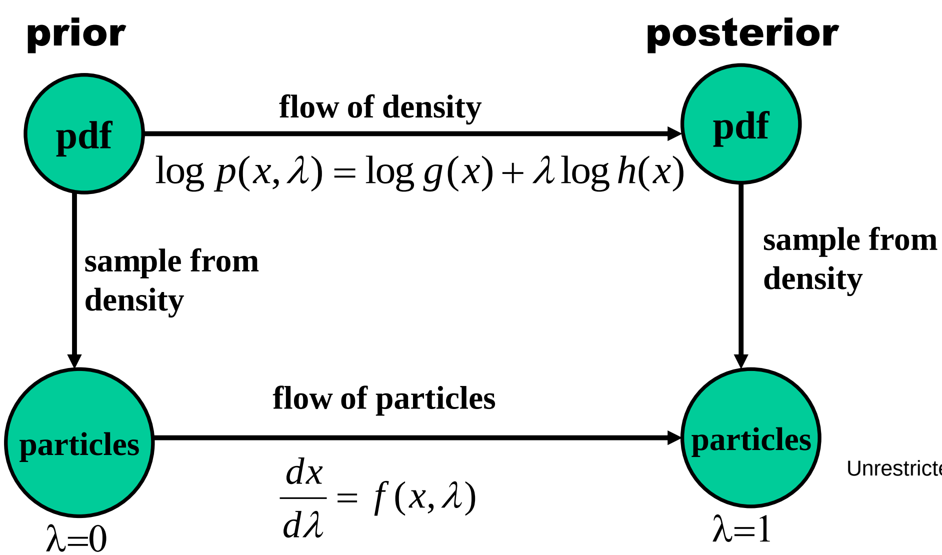
- More complex models
- Bug fixes:
  - Memory leak (later incorporated into Stan 2.6)
  - Initial condition ODE (t0): removed restriction (timepoints $\neq$ t0)
- Implemented better ODE solver: **CVODE** (Sundials)
  - Currently in Stan: only Runge-Kutta (simple/non-stiff)
  - CVODE: can deal **with difficult (stiff/unstable) models**
  - Jacobian: built using the auto-diff system of Stan
- Stan development team (Daniel Lee) is currently looking at Stan-CVODE implementation

curse of dimensionality:



root cause of curse of dimensionality:



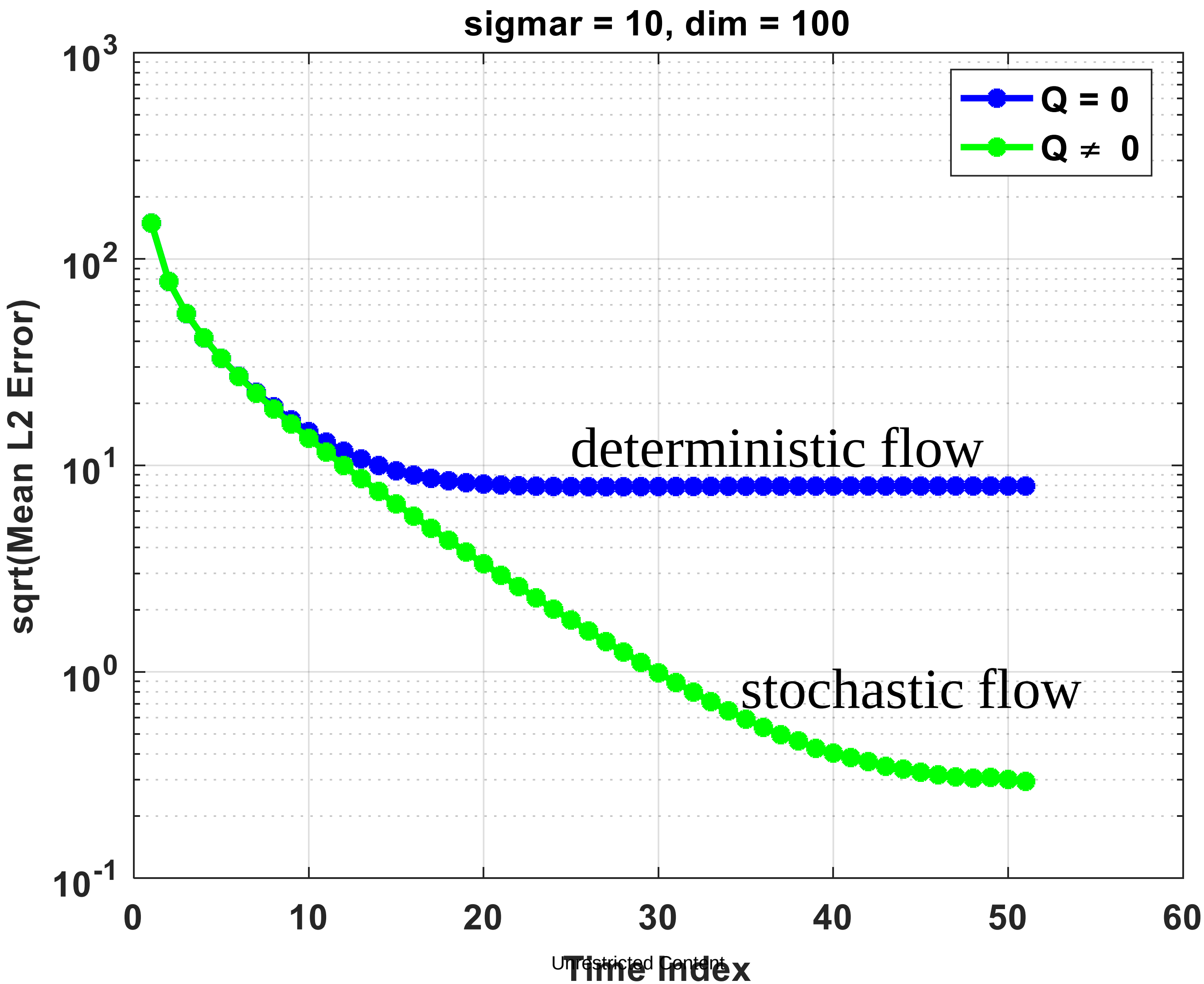


$$div(pf) = p\left[-\log h + \frac{d\log K}{d\lambda}\right]$$

We design the particle flow by solving the above PDE for f.

WHY STOCHASTIC ?

benefit of stochastic vs. deterministic flow



WHY STOCHASTIC?

1. **it works better (see plots)**
2. all practical particle filters that actually work robustly use stochastic methods; e.g., “roughening” in bootstrap filter, “rejuvenation” in optimal transport, “pseudo-noise” in second generation ensemble Kalman filter, Metropolis-Hastings, Hamiltonian Monte Carlo, Metropolis adjusted Langevin (MALA)
3. correction for bias of fixed (random) initial distribution of particles
4. **the solution of our PDE using Gromov’s method requires a stochastic term (to make the PDE sufficiently underdetermined)**
5. stochastic term is required in order to give correct uncertainty quantification (e.g., covariance consistency); theory & MC
6. simple intuition from real world: how well would your car work at absolute zero temperature?