particle flow for nonlinear filters with Gromov’s method

Fred Daum, Jim Huang & Arjang Noushin
4 June 2019
<table>
<thead>
<tr>
<th></th>
<th>ensemble Kalman filter</th>
<th>particle filter (bootstrap with resampling)</th>
<th>stochastic particle flow filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. prediction of ensemble of particles in time (forecast)</td>
<td>Monte Carlo (particle flow in time)</td>
<td>Monte Carlo (particle flow in time)</td>
<td>Monte Carlo (particle flow in time)</td>
</tr>
<tr>
<td>2. measurement update of the conditional probability density (analysis)</td>
<td>Kalman filter formulas for mean &amp; covariance matrix updates</td>
<td>Bayes’ rule by multiplication (prior times likelihood)</td>
<td>Bayes’ rule by particle flow from the prior to the posteriori</td>
</tr>
<tr>
<td>3. suffers from particle degeneracy?</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>4. suffers from curse of dimensionality?</td>
<td>no</td>
<td>yes (even for linear Gaussian problems)</td>
<td>no for certain smooth nowhere vanishing densities</td>
</tr>
<tr>
<td>5. resample particles to mitigate particle degeneracy?</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6. optimal accuracy (with large enough N) for nonlinear &amp; non-Gaussian problems?</td>
<td>no</td>
<td>yes</td>
<td>yes for certain smooth nowhere vanishing densities</td>
</tr>
</tbody>
</table>
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):

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

\( P = \text{sample covariance matrix from set of particles} \)
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.
nonlinear filter problem*

dynamical model of state:

\[
\begin{align*}
  dx &= F(x, t)dt + G(x, t)dw \
  x(t_{k+1}) &= F(x(t_k), t_k, w(t_k))
\end{align*}
\]

\[x(t) = \text{state vector at time } t\]

\[w(t) = \text{process noise vector at time } t\]

\[z_k = \text{measurement vector at time } t_k\]

\[z_k = H(x(t_k), t_k, v_k)\]

\[v_k = \text{measurement noise vector at time } t_k\]

\[p(x, t_k | Z_k) = \text{probability density of } x \text{ at time } t_k \text{ given } Z_k\]

\[Z_k = \text{set of all measurements}\]

\[Z_k = \{z_1, z_2, \ldots, z_k\}\]

curse of dimensionality for classic particle filter*

optimal accuracy: 
\[ r = 1.0 \]

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

prediction of conditional probability density from \( t_{k-1} \) to \( t_k \)

solution of Fokker-Planck equation

Bayes' rule:
\[
p(x,t_k \mid Z_k) = p(x,t_k \mid Z_{k-1}) p(z_k \mid x,t_k)
\]

particle degeneracy*

prior density
\( g(x) \)

likelihood
\( h(x) \)

particles to represent the prior

particle degeneracy*

prior density $g(x)$

likelihood $h(x)$

particles to represent the prior

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

prior = g(x)

posterior = g(x)h(x)/K(1)

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

\( \lambda = \text{continuous parameter} \)
\( \text{(like time)} \)

\( \lambda = 0 \)

\( \lambda = 1 \)
initial probability distribution of particles:

\[ \lambda = 0.0 \]
flow of particles (for one noisy measurement of \(\sin(\theta)\) with Bayes’ rule):

\[
\lambda = 0.1
\]
flow of particles (for one noisy measurement of $\sin(\theta)$ with Bayes’ rule):

$$\lambda = 0.2$$
flow of particles (for one noisy measurement of \(\sin(\theta)\) with Bayes’ rule):

\[ \lambda = 0.3 \]
flow of particles (for one noisy measurement of \(\sin(\theta)\) with Bayes’ rule):

\[
\lambda = 0.4
\]
flow of particles (for one noisy measurement of \( \sin(\theta) \) with Bayes’ rule):

\[ \lambda = 0.5 \]
flow of particles (for one noisy measurement of \( \sin(\theta) \) with Bayes’ rule):

\[ \lambda = 0.6 \]
flow of particles (for one noisy measurement of $\sin(\theta)$ with Bayes’ rule):

\[ \lambda = 0.7 \]
flow of particles (for one noisy measurement of \(\sin(\theta)\) with Bayes’ rule):

\[ \lambda = 0.8 \]
flow of particles (for one noisy measurement of \(\sin(\theta)\) with Bayes’ rule):

\[ \lambda = 0.9 \]
final probability distribution of particles (resulting from one noisy measurement of $\sin(\theta)$ with Bayes’ rule):

$$\lambda = 1$$
<table>
<thead>
<tr>
<th>Incompressible flow</th>
<th>Irrotational flow</th>
<th>Coulomb’s law flow</th>
<th>Small curvature flow</th>
<th>Constant curvature &amp; constant speed flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian densities</td>
<td>Exponential family</td>
<td>Differential Knothe-Rosenblatt flow</td>
<td>Stochastic flows</td>
<td>Geodesic flows</td>
</tr>
<tr>
<td>Fourier transform flow</td>
<td>Direct integration</td>
<td>Stabilized flows</td>
<td>Finite dimensional flow</td>
<td>Optimal Monge-Kantorovich transports</td>
</tr>
<tr>
<td>Method of characteristics</td>
<td>Separation of variables</td>
<td>Renormalization group flow inspired by QFT</td>
<td>Hybrid particle-parameter flow</td>
<td>Non-optimal transport</td>
</tr>
<tr>
<td>Gibbs sampler like flow</td>
<td>Gromov’s method</td>
<td>Moser coupling flow</td>
<td>Monge-Ampère with N-principle</td>
<td>Monge-Ampère flow</td>
</tr>
</tbody>
</table>
exact particle flow for Gaussian densities:

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

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

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

\[ f = Ax + b \]

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

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

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

for \( d \geq 2 \)

\( \frac{dx}{d\lambda} \) does not depend on \( K(\lambda) \), despite the fact that the PDE does!
geodesic particle flow*:

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

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:

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


dx/d\lambda does not depend on K(\lambda), despite the fact that the PDE does!
derivation of PDE for particle flow with $Q \neq 0$:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

\[
\frac{dx}{d\lambda} \approx P \left( \frac{\partial \theta}{\partial x} \right)^T R^{-1} (z - \theta(x)) + \frac{dw}{d\lambda}
\]
\[
\frac{\partial \log h}{\partial x} = -f^T \frac{\partial^2 \log p}{\partial x^2} - \frac{\partial \text{div}(f)}{\partial x}
\]
\[
= -\frac{\partial \log p}{\partial x} \frac{\partial f}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x} \left\{ \text{div} \left[ Q \frac{\partial p}{\partial x} \right] / p \right\}
\]
\[
f = -\left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \left( \frac{\partial \log h}{\partial x} \right)^T
\]
\[
Q = -\left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \frac{\partial^2 \log h}{\partial x^2} \left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1}
\]
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


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

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

$$Q = -\left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \frac{\partial^2 \log h}{\partial x^2} \left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1}$$
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 \( N \geq \frac{52\kappa^2}{\epsilon} \sqrt{\frac{d}{m}} + D^2 \log\left[\frac{24}{\epsilon} \left(\frac{d}{m} + D^2\right)\right] \)

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

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

Then \( W_2(p_0, p) \leq \epsilon \)
BACKUP
generalization of Gromov’s method:

\[ \text{div}(r) + b^T s = \eta \]

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 \( \beta \) 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:

\[ \text{div}(r) + b^T s = \eta \]

suppose that we know an exact solution to the PDE:

\[ \text{div}(\tilde{r}) = \tilde{\eta} \]

add them to get:

\[ \text{div}(r + \tilde{r}) + b^T s = \eta + \tilde{\eta} \]

we can solve this similar to the previous chart:

\[
\begin{align*}
    r &= A \left[ \frac{\partial \eta}{\partial x} \right]^T + B \left[ \frac{\partial \theta}{\partial x} \right]^T + \beta(x) - \tilde{r} \\
    s &= \frac{b}{\|b\|^2} (\eta + \tilde{\eta}) + \left[ I - \frac{bb^T}{\|b\|^2} \right] y
\end{align*}
\]
even further generalization of Gromov’s method:

\[ \text{div}(r) + b^T s = \eta \]

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

\[ \text{div}(\tilde{r}) = \tilde{\eta} \]

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

\[
\begin{align*}
    r &= A \left[ \frac{\partial \eta}{\partial x} \right]^T + B \left[ \frac{\partial \theta}{\partial x} \right]^T + \beta(x) - \sum_{\Omega} \tilde{r}_k \\
    s &= \frac{b}{\|b\|^2} (\eta + \sum_{\Omega} \tilde{\eta}_k) + \left[ I - \frac{bb^T}{\|b\|^2} \right] y
\end{align*}
\]
more details:

VIDEO of recent talk at Stony Brook University (24 April 2018)
https://youtu.be/vqJGB47XoeY


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

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

\[ Q = -\left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \frac{\partial^2 \log h}{\partial x^2} \left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \]
## new nonlinear filter: particle flow

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

flow

incompressible flow

Gaussian flow

MALA, HMC, auxiliary & bootstrap

N = 1,000 particles

nonlinear dynamics & nonlinear measurements
dimension of state vector = 17

100 Monte Carlo trials, SNR = 20 dB

d = 42 states

N = 10,000 particles

median error over 100 Monte Carlo runs

100,000

10,000

1,000

100

10

1

stochastic particle flow

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

<table>
<thead>
<tr>
<th>algorithm</th>
<th>randomness</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstrap particle filter (1993)</td>
<td>roughening &amp; resampling</td>
<td>ad hoc randomness</td>
</tr>
<tr>
<td>other standard particle filters</td>
<td>roughening &amp; resampling</td>
<td>ad hoc randomness</td>
</tr>
<tr>
<td>optimal transport</td>
<td>none</td>
<td>rigorous math theorems</td>
</tr>
<tr>
<td>Reich’s optimal transport particle filters</td>
<td>rejuvenation</td>
<td>ad hoc randomness</td>
</tr>
<tr>
<td>early ensemble Kalman filters (1994)</td>
<td>none</td>
<td>did not work well for many problems</td>
</tr>
<tr>
<td>mature ensemble Kalman filters (1998)</td>
<td>artificial measurement noise</td>
<td>fixes problems of early ensemble Kalman filters</td>
</tr>
<tr>
<td>early particle flow filters</td>
<td>none</td>
<td>covariance optimistic for many problems</td>
</tr>
<tr>
<td>stochastic particle flow filters (2016)</td>
<td>non-zero diffusion for Bayes’ rule</td>
<td>principled math derivation of stochastic flow &amp; improved accuracy &amp; covariance consistency</td>
</tr>
<tr>
<td>item</td>
<td>deep learning</td>
<td>particle flow</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>purpose</td>
<td>learning &amp; decisions</td>
<td>learning &amp; estimation &amp; decisions</td>
</tr>
<tr>
<td>interesting wrinkle (which annoys many people)</td>
<td>lack of uniqueness of solution for highly non-convex loss functions</td>
<td>lack of uniqueness for solution of highly underdetermined transport PDE</td>
</tr>
<tr>
<td>architecture</td>
<td>many layers</td>
<td>many steps in log-homotopy</td>
</tr>
<tr>
<td>fundamental issues</td>
<td>curse of dimensionality &amp; ill-conditioning &amp; singularity of Hessian</td>
<td>curse of dimensionality &amp; ill-conditioning &amp; singularity of Hessian</td>
</tr>
<tr>
<td>tools</td>
<td>stochastic gradient or natural gradient</td>
<td>stochastic natural gradient</td>
</tr>
<tr>
<td>representation of geometry</td>
<td>Hessian of loss function (log p)</td>
<td>Hessian of log p</td>
</tr>
<tr>
<td>useful theory to explain performance</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>performance evaluation</td>
<td>numerical experiments</td>
<td>numerical experiments</td>
</tr>
<tr>
<td>theory of design</td>
<td>ersatz Bayesian</td>
<td>echt Bayesian</td>
</tr>
<tr>
<td>computers of choice today</td>
<td>GPUs</td>
<td>GPUs</td>
</tr>
<tr>
<td>regularization</td>
<td>random dropout &amp; sparsity of coupling between layers and within layers</td>
<td>Tychonov regularization or shrinkage or preferred coordinate system</td>
</tr>
<tr>
<td>key adaptive method</td>
<td>adaptive learning rate</td>
<td>adaptive step size in $\lambda$</td>
</tr>
<tr>
<td>dynamics of learning</td>
<td>backpropagation (i.e., chain rule)</td>
<td>Fokker-Planck equation (i.e., chain rule)</td>
</tr>
</tbody>
</table>
BIG DIG (17 million cubic yards of dirt, one million truckloads & $24 billion)*

superb books on transport theory

Very clear & accessible introduction; wonderful book!

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*


derivation of particle flow with $Q \neq 0$:

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

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

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

$$
\alpha \approx 2 \left\| \frac{\partial \log p}{\partial x} \frac{\partial \hat{f}}{\partial x} \right\| \left\| \frac{\partial}{\partial x} \left\{ \text{div} \left[ \frac{\partial p}{\partial x} \right] / p \right\} \right\|
$$
MATLAB was vectorized for SIRPF but not the HPF.

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:

- extended Kalman filter
- standard particle filter
- particle flow

N = 1,000 particles
100 Monte Carlo trials
20 dB SNR
10% tropo & SDMB
d = 6

Time (sec)
Velocity Error (m/sec)
IMU-only Navigation problem (no GPS)

EKF diverges (not shown); $d = 15$, $N = 1000$
Unrestricted Content
The graph shows the comparison of EKF and NEW FLOW methods over time.

- **EKF** method is represented by a solid black line.
- **NEW FLOW** method is represented by a green dotted line.

The y-axis represents the **Velocity Error (m/sec)** on a logarithmic scale, while the x-axis represents **Time (sec)** from 0 to 100.

The graph illustrates how both methods reduce velocity error over time, with **NEW FLOW** showing a slightly faster reduction compared to **EKF**.
\[
div(pf) = p \left[ -\log h + \frac{d \log K}{d\lambda} \right]
\]

let \( q = pf \)

\[
\frac{\partial q_1}{\partial x_1} + \frac{\partial q_2}{\partial x_2} + \ldots + \frac{\partial q_d}{\partial x_d} = \eta
\]

(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 \( \eta \) 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:

\[
\frac{dx}{d\lambda} \approx C \left( \frac{\partial \theta(x)}{\partial x} \right)^T R^{-1} (z - \theta(x)) + \frac{dw}{d\lambda}
\]
<table>
<thead>
<tr>
<th>method</th>
<th>computational complexity</th>
<th>filter accuracy</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. use a stiff ODE solver (e.g., implicit integration rather than explicit)</td>
<td>large to extremely large</td>
<td>uncertain</td>
<td>standard textbook advice</td>
</tr>
<tr>
<td>2. use very small integration steps everywhere</td>
<td>extremely large</td>
<td>good</td>
<td>brute force solution</td>
</tr>
<tr>
<td>3. use very small integration steps only where needed (adaptively determined)</td>
<td>small to medium</td>
<td>2nd best</td>
<td>Shozo Mori &amp; Daum (2016)</td>
</tr>
<tr>
<td>4. use very small integration steps only where needed (determined non-adaptively)</td>
<td>small</td>
<td>3rd best</td>
<td>easy to do with particle flow</td>
</tr>
<tr>
<td>5. transform to principal coordinates or approximately principal coordinates</td>
<td>small</td>
<td>best</td>
<td>easy for certain applications</td>
</tr>
<tr>
<td>6. Battin’s trick (i.e., sequential scalar measurement updates)</td>
<td>small</td>
<td>very bad</td>
<td>destroys particle flow</td>
</tr>
<tr>
<td>7. Tychonov regularization of the Hessian of log p</td>
<td>very small</td>
<td>often helps</td>
<td></td>
</tr>
</tbody>
</table>

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

\[ \lambda_0 = \text{fixed} \quad 100 \text{ steps} \]
\[ \lambda_1 = \text{fixed} \quad 1000 \text{ steps} \]
\[ \lambda_2 = \text{fixed} \quad 370 \text{ steps} \]

\[ \lambda_3 = \text{fixed} \quad 10 \text{ steps} \]
\[ \lambda_4 = \text{variable} \quad 29 \text{ steps} \]

\[ d = 9, \; N = 500, \; \lambda = 0.9, \; \sigma_0 = 100 \]
$
\begin{array}{c}
\text{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 \\
0.108571119402220 \\
0.138949549437314 \\
0.177827941003892 \\
0.227584592607479 \\
0.291263265490874 \\
0.372759372031494 \\
0.477058269614393 \\
0.610540229658533 \\
0.781370737651809 \\
1.000000000000000
\end{array}$

example of non-adaptive non-uniform step size in lambda
\[ \Delta \lambda_0 = 1 \times 10^{-3} \]
\[ \Delta \lambda_0 = 1e-5 \]
$\sigma_0 = 1000$

- EKF
- Incompressible
- $Ax+b$
- No Flow
- Zero Curvature

$\Delta \lambda_0 = 1e-7$
\[ \Delta \lambda_0 = 1e^{-9} \]

\[ \sigma_0 = 1e4 \]

- EKF
- Incompressible
- Ax+b
- No Flow
- Zero Curvature
$\sigma_0 = 1e5$

- **EKF**
- **Incompressible**
- **Ax+b**
- **No Flow**
- **Zero Curvature**

$\Delta \lambda_0 = 1e-11$
\[ \sigma_0 = 1e6 \]

\[ \Delta \lambda_0 = 1e-13 \]
this flow requires adaptive numerical integration to mitigate stiffness (see Shozo Mori 2016)
REFERENCES ON STIFFNESS:


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 itertions) 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)
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 ≠ 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:

prior density

likelihood of measurement

particles to represent the prior

prior

posterior

flow of density

\[
\log p(x, \lambda) = \log g(x) + \lambda \log h(x)
\]

sample from density

pdf

sample from density

pdf

We design the particle flow by solving the above PDE for \( f \).
WHY
STOCHASTIC?

Unrestricted Content
benefit of stochastic vs. deterministic flow

sigmar = 10, dim = 100

- Deterministic flow
- Stochastic flow

sqrt(L2 Error) vs. Time Index
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?