particle flow for nonlinear filters, Bayesian decisions & transport

Fred Daum & Jim Huang
7 April 2014
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) = state vector at time t
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
Z_k = \{z_1, z_2, ..., z_k\}

estimate x given noisy measurements
curse of dimensionality for classic particle filter

optimal accuracy: \( r = 1.0 \)
nonlinear filter

Bayes' rule:

\[ p(x, t_k | Z_k) = p(x, t_k | Z_{k-1}) p(z_k | x, t_k) \]
particle degeneracy*

prior density

likelihood

particles to represent the prior

induced flow of particles
for Bayes’ rule

prior = g(x)

posterior = g(x)h(x)

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

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

\(\lambda = 0\)

\(\lambda = 1\)
curse of dimensionality:

- Optimal accuracy

prior posterior

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

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

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

root cause of curse of dimensionality:

likelihood of measurement

prior density

\( g \)

\( h \)

particles to represent the prior
\[ \text{div}(pf) = p \left[ - \log h + \frac{d \log K}{d\lambda} \right] \]

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 divergence 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)
<table>
<thead>
<tr>
<th>incompressible flow</th>
<th>irrotational flow</th>
<th>Coulomb’s law flow</th>
<th>small curvature flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian densities</td>
<td>exponential family</td>
<td>Fourier transform flow</td>
<td>constant curvature flow (e.g. zero curvature)</td>
</tr>
<tr>
<td>Knothe-Rosenblatt flow</td>
<td>non-zero diffusion flow</td>
<td>method of characteristics</td>
<td>geodesic flow</td>
</tr>
<tr>
<td>stabilized flows</td>
<td>finite dimensional flow</td>
<td>direct integration</td>
<td>Monge-Kantorovich transport</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:

\[
f = Ax + b
\]

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

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

automatically stable under very mild conditions & extremely fast
incompressible particle flow

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

\[ \lambda = 0.0 \]
Time = 1, Frame 2

\[ \lambda = 0.1 \]
\[ \lambda = 0.2 \]
\[ \lambda = 0.3 \]
Time = 1, Frame 4

\[ \lambda = 0.4 \]
Time = 1, Frame 5

\[ \lambda = 0.5 \]
Time = 1, Frame 6

λ = 0.6
Time = 1, Frame 7

\[ \lambda = 0.7 \]
Time = 1, Frame 8

\[ \lambda = 0.8 \]
\[ \lambda = 0.9 \]
final probability distribution of particles (resulting from one noisy measurement of $\sin(\theta)$ with Bayes’ rule):

\[ \lambda = 1 \]
new 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 \( P \) over the set of particles:

\[
\frac{dx}{d\lambda} \approx P \left( \frac{\partial \log h}{\partial x} \right)^T
\]

for Gaussian densities we get the EKF for each particle:

\[
\frac{dx}{d\lambda} \approx P \left( \frac{\partial \theta(x)}{\partial x} \right)^T R^{-1}(z - \theta(x))
\]
N = 1,000 particles nonlinear dynamics & nonlinear measurements dimension of state vector = 17
100 Monte Carlo trials, SNR = 20 dB

d = 42 states
SNR = 20 dB
N = 1,000 particles

median error over 100 Monte Carlo runs

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

<table>
<thead>
<tr>
<th>item</th>
<th>exact particle flow</th>
<th>Monge-Kantorovich transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. purpose</td>
<td>fix particle degeneracy due to Bayes’ rule</td>
<td>move physical objects with minimal effort from one probability density to another</td>
</tr>
<tr>
<td>2. conservation of probability mass along flow</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>3. deterministic</td>
<td>yes*</td>
<td>yes</td>
</tr>
<tr>
<td>4. homotopy of density</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>5. log-homotopy of density</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6. optimality criteria</td>
<td>none</td>
<td>Wasserstein metric (dirt mover’s metric) or other functional</td>
</tr>
<tr>
<td>7. how to pick a solution</td>
<td>19 distinct methods</td>
<td>minimize convex functional</td>
</tr>
<tr>
<td>8. stability of flow considered</td>
<td>yes</td>
<td>rarely</td>
</tr>
<tr>
<td>9. high dimensional applications</td>
<td>yes (d ≤ 42)</td>
<td>no (d = 1, 2 or 3)</td>
</tr>
<tr>
<td>10. computational complexity</td>
<td>numerical integration of ODE for each particle</td>
<td>solution of Monge-Ampere highly nonlinear PDE or other</td>
</tr>
<tr>
<td>11. solution of PDE for nice special cases</td>
<td>many (incompressible, irrotational, Gaussian, etc.)</td>
<td>Knothe-Rosenblatt</td>
</tr>
<tr>
<td>12. math theory for existence of incompressible flow</td>
<td>borrow Shnirelman’s theorem</td>
<td>Shnirelman’s theorem for d ≥ 3</td>
</tr>
</tbody>
</table>
best books on transport theory

Very clear & accessible introduction; wonderful book!

idea inspired by renormalization group flow:

\[
\begin{align*}
\frac{dx}{d\lambda} &= -C^\# [\log h - \frac{d \log K(\lambda)}{d\lambda}] + (I - C^\# C)y \\
f &= \Gamma + \Pi y \\
\Pi &= \text{projection into null-space of } C
\end{align*}
\]

\[
\begin{align*}
\frac{\partial f}{\partial L} &= \frac{\partial \Gamma}{\partial L} + \frac{\partial \Pi}{\partial L} y = 0 \\
y &= -\left[\frac{\partial \Pi}{\partial L}\right]^\# \left(\frac{\partial \Gamma}{\partial L}\right) \\
L &= \{\log g(x), \log K(\lambda)\}^T \\
(\cdot)^\# &= \text{generalized inverse of } (\cdot)
\end{align*}
\]

L rather than g or K (just like QFT): linear in L but not g; result does not depend on g itself; avoids singularity at g = 0; slightly different & it works
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</td>
<td>suffers from curse of dimensionality due to particle degeneracy</td>
</tr>
<tr>
<td>3 to 4 orders of magnitude faster code per particle for any ( d \geq 3 ) problems</td>
<td>requires resampling using a proposal density</td>
</tr>
<tr>
<td>3 to 4 orders of magnitude fewer particles required to achieve optimal accuracy for ( d \geq 6 ) problems</td>
<td>requires millions or billions of particles for high dimensional problems</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>
</tbody>
</table>
BACKUP
convergence with $N$ for particle filters:

$$\sigma^2 \approx c / N$$

$N =$ number of particles
$c =$ so-called “constant” which depends on:

1. dimension of the state vector ($x$)
2. initial uncertainty in the state vector
3. measurement accuracy
4. shape of probability densities (e.g., log-concave or multimodal etc.)
5. Lipschitz constants of log densities
6. stability of the plant
7. curvature of nonlinear dynamics & measurements
8. ill-conditioning of Fisher information matrix
9. smoothness of densities & dynamics & measurements
Oh’s Formula for Monte Carlo errors

\[ \sigma^2 \approx \left\{ \left[ \frac{1 + k}{\sqrt{1 + 2k}} \right] \exp \left[ \frac{\varepsilon^2}{1 + 2k} \right] \right\}^d / N \]

Assumptions:

1. Gaussian density (zero mean & unit covariance matrix)
2. d-dimensional random variable
3. Proposal density is also Gaussian with mean \( \varepsilon \) and covariance matrix \( kI \), but it is not exact for \( k \neq 1 \) or \( \varepsilon \neq 0 \)
4. \( N = \) number of Monte Carlo trials
<table>
<thead>
<tr>
<th>university or company</th>
<th>researchers</th>
<th>topic</th>
<th>papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecticut</td>
<td>Willett &amp; Choi</td>
<td>numerical experiments</td>
<td>2011</td>
</tr>
<tr>
<td>McGill</td>
<td>Coates &amp; Ding</td>
<td>invented new flow &amp; numerical experiments</td>
<td>2012</td>
</tr>
<tr>
<td>New Orleans</td>
<td>Jilkov &amp; Wu &amp; Chen</td>
<td>GPUs numerical experiments</td>
<td>2013</td>
</tr>
<tr>
<td>Melbourne</td>
<td>Morelande</td>
<td>generalization of theory</td>
<td>2011</td>
</tr>
<tr>
<td>Goteborg</td>
<td>Svensson, et al.</td>
<td>generalization of theory</td>
<td>2011</td>
</tr>
<tr>
<td>Scientific Systems</td>
<td>Chen &amp; Mehra</td>
<td>analysis of singularities in incompressible flow for ( d = 1 )</td>
<td>2010</td>
</tr>
<tr>
<td>Liverpool</td>
<td>Maskell</td>
<td>relation to MCMC</td>
<td></td>
</tr>
<tr>
<td>London</td>
<td>Julier</td>
<td>numerical experiments</td>
<td></td>
</tr>
<tr>
<td>METRON</td>
<td>Bell</td>
<td>numerical experiments</td>
<td>2014</td>
</tr>
<tr>
<td>BU</td>
<td>Castanon, et al.</td>
<td>numerical experiments</td>
<td></td>
</tr>
<tr>
<td>Toulouse, MIT, Harvard, Technion</td>
<td>Pereyra, et al.</td>
<td>numerical experiments &amp; theory</td>
<td></td>
</tr>
<tr>
<td>Tufts</td>
<td>Umarov</td>
<td>fractional Brownian motion</td>
<td>2014</td>
</tr>
<tr>
<td>Raytheon Boston</td>
<td>Daum &amp; Huang &amp; Noushin</td>
<td>theory &amp; numerical experiments</td>
<td>2007-2014</td>
</tr>
<tr>
<td>Raytheon Arizona</td>
<td>Frankot &amp; Reid &amp; Kyle</td>
<td>theory &amp; numerical experiments</td>
<td></td>
</tr>
<tr>
<td>Raytheon California</td>
<td>Ploplys &amp; Casey</td>
<td>theory &amp; numerical experiments</td>
<td></td>
</tr>
</tbody>
</table>
## 7 dubious methods to mitigate stiffness in ODEs

<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>textbook advice &amp; many papers</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>large</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. use very small integration steps only where needed (determined non-adaptively)</td>
<td>small</td>
<td>2nd 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 to do 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 the benefit of particle flow</td>
</tr>
<tr>
<td>7. Tychonov regularization of the Hessian of log p</td>
<td>very small</td>
<td>uncertain</td>
<td></td>
</tr>
<tr>
<td>8. hybrid combinations of above</td>
<td></td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>item</td>
<td>renormalization group flow in quantum field theory</td>
<td>particle flow for Bayes’ rule</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------------------------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>1. purpose</td>
<td>avoid infinite integrals at all energy scales (µ)</td>
<td>fix particle degeneracy in particle filters</td>
<td></td>
</tr>
<tr>
<td>2. PDE</td>
<td>linear first order PDE</td>
<td>linear first order PDE</td>
<td></td>
</tr>
<tr>
<td>3. method</td>
<td>“the trick of doing an integral a little bit at a time” (Tony Zee QFTNS p. 346)</td>
<td>homotopy of log-density</td>
<td></td>
</tr>
<tr>
<td>4. efficacy</td>
<td>“the most important conceptual advance in QFT over the last 3 or 4 decades” (Tony Zee QFTNS p. 337)</td>
<td>reduces computational complexity by many orders of magnitude for high dimensional problems</td>
<td></td>
</tr>
<tr>
<td>5. algorithm</td>
<td>ODE for motion of particles in N-dimensional space (Tony Zee QFTNS p. 340)</td>
<td>( f = \frac{dx}{d\lambda} ) ( x = ) particle in d-dimensions</td>
<td></td>
</tr>
<tr>
<td>6. derivation of PDE</td>
<td>( \frac{dH(x)}{d\mu} = 0 ) ( H(x) = ) Hamiltonian</td>
<td>Fokker-Planck equation &amp; definition of log p</td>
<td></td>
</tr>
<tr>
<td>7. new idea for particle flow inspired by RNGF</td>
<td>( \frac{dH(x)}{d\mu} = 0 ) (we want H to be scale invariant)</td>
<td>( \frac{\partial f}{\partial g} = 0 ) ( g = ) prior density</td>
<td></td>
</tr>
</tbody>
</table>
two steps in renormalization group flow & particle flow

<table>
<thead>
<tr>
<th>step</th>
<th>physics</th>
<th>nonlinear filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. regularization</td>
<td>regularization (e.g., cut-off of integral or dimensional regularization d-ε)</td>
<td>homotopy of log-density</td>
</tr>
<tr>
<td>2. renormalization</td>
<td>modify effective charge &amp; mass as the energy scale varies from high to low (integrate out degrees of freedom to maintain symmetries &amp; finite number of parameters); scale invariant flow of parameters:</td>
<td>compute flow of particles that is invariant to errors in the prior density &amp; the normalization constant: [ \partial f / \partial g = 0 ]</td>
</tr>
</tbody>
</table>

\[ dH/d\mu = 0 \]
long range wideband radar tracking of ballistic missile:

- $N = 1,000$ particles
- 100 Monte Carlo trials
- 20 dB SNR
- 10% tropo & SDMB
- $d = 6$
many applications of particle flow

<table>
<thead>
<tr>
<th>Bayesian decisions &amp; learning</th>
<th>robotics</th>
<th>communications</th>
<th>control of chemical, mechanical, electrical &amp; nuclear plants</th>
</tr>
</thead>
<tbody>
<tr>
<td>guidance &amp; navigation</td>
<td>tracking</td>
<td>weather &amp; climate prediction</td>
<td>predicting ionosphere, thermosphere, troposphere</td>
</tr>
<tr>
<td>science</td>
<td>imaging</td>
<td>medicine (e.g., MRI, surgical planning, drug design, diagnosis)</td>
<td>transport</td>
</tr>
<tr>
<td>oil &amp; mineral exploration</td>
<td>financial engineering</td>
<td>adaptive antennas</td>
<td>audio &amp; video signal processing</td>
</tr>
<tr>
<td>nonlinear filtering &amp; smoothing</td>
<td>multi-sensor data fusion</td>
<td>compressive sensing</td>
<td>CRYPTO</td>
</tr>
</tbody>
</table>
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*
all roads lead to new flow:

- zero curvature & solution of vector Riccati equation rather than PDE
- maximum likelihood estimation with Newton’s method
- maximum likelihood estimation with homotopy
- non-zero diffusion & clever choice of Q to avoid PDE
- Svensson & Morelande et al., Hanebeck et al., Daum & Huang, Girolami & Calderhead, etc.
Monge-Ampere highly nonlinear PDE

\[ y = T(x) \]
\[ p(x)dx = p(y)dy \]
\[ p(x) = p(y) \det \left[ \frac{\partial y}{\partial x} \right] \]

Let \( T(x) = \frac{\partial V}{\partial x} \)

Hence,

\[ p(x) = p(y) \det \left[ \frac{\partial^2 V}{\partial x^2} \right] \]

one shot transport requires **nonlinear** PDE (and we cannot evaluate the functions at good points!), whereas particle flow only needs an extremely simple **linear** PDE
computing the Hessian of log p:

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

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

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

\(C = \) sample covariance matrix of particles for prior (\(\lambda = 0\))

with Tychonov regularization; or EKF or UKF covariance matrix

\[
\frac{\partial^2 \log p}{\partial x^2} \approx -P^{-1}
\]

\(P = \) sample covariance matrix of particles for \(p(x, \lambda)\)

with Tychonov regularization; or EKF or UKF covariance matrix

\[\text{can compute Hessians using calculus or 2
}^\text{nd}
\] differences
formula that avoids inverse of sample covariance matrix:

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

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

\[
\frac{\partial^2 \log p}{\partial x^2} \approx -C^{-1} + \lambda \frac{\partial^2 \log h(x)}{\partial x^2} \quad \text{for } g(x) \approx \text{Gaussian}
\]

but Woodbury's matrix inversion lemma gives us:

\[
(A + B)^{-1} = A^{-1} - A^{-1}B(I + A^{-1}B)^{-1}A^{-1} \quad \text{for arbitrary B and non-singular A}
\]

hence

\[
\left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \approx -C - CB(I - CB)^{-1}C
\]

in which

\[
B = \lambda \frac{\partial^2 \log h(x)}{\partial x^2}
\]
formula that avoids computing Hessian of $g(x)$:

$$\frac{\partial^2 \log g(x)}{\partial x^2} \approx -\frac{1}{N} \sum_{j=1}^{N} \left( \frac{\partial \log g(x_j)}{\partial x} \right)^T \frac{\partial \log g(x_j)}{\partial x}$$

derivation of the above:

$$E \left[ \frac{\partial^2 \log g(x)}{\partial x^2} \right] = -E \left[ \left( \frac{\partial \log g(x)}{\partial x} \right)^T \frac{\partial \log g(x)}{\partial x} \right]$$

$$\frac{\partial^2 \log g(x)}{\partial x^2} \approx E \left[ \frac{\partial^2 \log g(x)}{\partial x^2} \right]$$

$$E \left[ \left( \frac{\partial \log g(x)}{\partial x} \right)^T \frac{\partial \log g(x)}{\partial x} \right] \approx \frac{1}{N} \sum_{j=1}^{N} \left( \frac{\partial \log g(x_j)}{\partial x} \right)^T \frac{\partial \log g(x_j)}{\partial x}$$
<table>
<thead>
<tr>
<th>method to solve PDE</th>
<th>how to pick unique solution</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. generalized inverse of linear differential operator</td>
<td>minimum $L^2$ norm</td>
<td>Coulomb’s law or fast Poisson solver</td>
</tr>
<tr>
<td>2. Poisson’s equation</td>
<td>irrotational flow</td>
<td>Coulomb’s law or fast Poisson solver</td>
</tr>
<tr>
<td>3. generalized inverse of gradient of log-homotopy</td>
<td>incompressible flow</td>
<td>workhorse for multimodal densities</td>
</tr>
<tr>
<td>4. stabilized version of method #3</td>
<td>most robustly stable filter</td>
<td>workhorse for multimodal densities</td>
</tr>
<tr>
<td>5. separation of variables (Gaussian)</td>
<td>pick solution of specific form</td>
<td>extremely fast &amp; hard to beat in accuracy</td>
</tr>
<tr>
<td>6. separation of variables (exponential family)</td>
<td>pick solution of specific form</td>
<td>generalization of Gaussian flow</td>
</tr>
<tr>
<td>7. variational formulation (Gauss &amp; Hertz)</td>
<td>convex function minimization</td>
<td>generalization of minimum $L^2$ norm</td>
</tr>
<tr>
<td>8. optimal transport formulation (Monge-Kantorovich)</td>
<td>convex functional minimization (e.g., least action or Wasserstein metric, etc.)</td>
<td>very high computational complexity (e.g. Monge-Ampere fully nonlinear PDE)</td>
</tr>
<tr>
<td>9. direct integration (of first order linear PDE in divergence form)</td>
<td>choice of $d-1$ arbitrary functions</td>
<td>should work with enforcement of neutral charge density &amp; importance sampling</td>
</tr>
<tr>
<td>10. method of characteristics (or generalized method of characteristics)</td>
<td>more conditions (e.g., small curvature or specify curl, or use Lorentz invariance)</td>
<td>can solve any first order linear PDE except for the one of interest to us!</td>
</tr>
<tr>
<td>11. another homotopy of the PDE (inspired by Gromov’s h-principle)</td>
<td>initial condition of ODE &amp; uniqueness of solution to ODE</td>
<td>like Feynman’s perturbation for QED</td>
</tr>
<tr>
<td>12. finite dimensional parametric flow (e.g., $f = Ax + b$ with $A$ &amp; $b$ parameters)</td>
<td>non-singular matrix to invert</td>
<td>avoids PDE completely</td>
</tr>
<tr>
<td>13. Fourier transform of PDE (divergence form of linear PDE has constant coefficients!)</td>
<td>minimum $L^2$ norm or most stable flow</td>
<td>generalized inverse &amp; Monte Carlo integration avoids inverse Fourier transform at random points in $d$ dimensions</td>
</tr>
<tr>
<td>14. small “curvature” flow</td>
<td>set certain 2$^{nd}$ derivatives of flow to zero</td>
<td>solve $d \times d$ system of linear equations or numerically integrate ODE (like Feynman)</td>
</tr>
<tr>
<td>15. zero “curvature” flow</td>
<td>set acceleration of particles to zero</td>
<td>solve vector Riccati equation exactly in closed form (rather than solve PDE)!</td>
</tr>
<tr>
<td>16. constant “curvature” flow etc. etc.</td>
<td>set acceleration of particle to constant</td>
<td>solve polynomial multivariate equations (rather than PDE); maybe use homotopy</td>
</tr>
<tr>
<td>17. upper triangular Jacobian flow</td>
<td>set certain lower triangular terms in Jacobian to zero (but not all terms to zero)</td>
<td>inspired by Knothe-Rosenblatt rearrangement in transport theory</td>
</tr>
<tr>
<td>18. non-zero process noise in flow for Bayes’ rule, with clever choice of $f$ &amp; $Q$ to avoid PDE</td>
<td>compute gradient of PDE to obtain $d$ equations in $d$ unknowns</td>
<td>$Q = \text{covariance matrix of diffusion in flow}$: $dx = f(x, \lambda)d\lambda + \sqrt{Q}, dw$</td>
</tr>
</tbody>
</table>
$d = 12, n_y = 3, y = x^2, \text{SNR} = 20\text{dB}$

**quadratic measurement nonlinearity**

Particle flow filter beats EKF by orders of magnitude
variation in initial uncertainty of $x$

N = 1000, Stable, $d = 10$, Quadratic, $\lambda = 0.6$

- Huge Initial Uncertainty
- Large Initial Uncertainty
- Medium Initial Uncertainty
- Small Initial Uncertainty

25 Monte Carlo Trials
variation in eigenvalues of the plant ($\lambda$)

N = 1000, d = 10, Cubic
variation in dimension of $x$
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
derivation of PDE for particle flow with $Q \neq 0$:

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

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

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

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

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

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

\[
\left[ \log h - \frac{d \log K}{d\lambda} \right] = -\text{div}(f) - \frac{\partial \log p}{\partial x} f + \frac{1}{2} \frac{1}{p} \text{div} \left[ Q \frac{\partial p}{\partial x} \right]
\]
derivation of first new particle flow with $Q \neq 0$:

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

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

pick $Q$ such that the three last terms sum to zero, and solve for $f$:

\[
f = -\left[\frac{\partial^2 \log p}{\partial x^2}\right]^{-1} \left(\frac{\partial \log h}{\partial x}\right)^T
\]
why does the new flow work so well?

<table>
<thead>
<tr>
<th>item</th>
<th>new flow</th>
<th>old flows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. solution for flow</td>
<td>d x d matrix inverse</td>
<td>Moore-Penrose inverse of 1 equation in d unknowns</td>
</tr>
<tr>
<td>2. normalization of probability density</td>
<td>we killed the normalization</td>
<td>explicitly computed</td>
</tr>
<tr>
<td>3. exploits smoothness of density functions</td>
<td>smoother (2\textsuperscript{nd} derivatives wrt x)</td>
<td>less smooth (only first derivatives wrt x)</td>
</tr>
<tr>
<td>4. exploits calculus to compute Hessian &amp; gradient of likelihood</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>5. exploits greater freedom with non-zero diffusion in flow</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>6. depends on Monte Carlo approximation</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>7. generality</td>
<td>more</td>
<td>less</td>
</tr>
</tbody>
</table>
small curvature flow:

Gaussian flow:
\[ f = A(\lambda)x + b(\lambda) \]
\[ \text{div}(f) = \text{Tr}(A) \]

incompressible flow:
\[ \text{div}(f) = 0 \]

\[ \frac{\partial \text{div}(f)}{\partial x} = 0 \]
irrotational particle flow:

\[
\frac{dx}{d\lambda} = f(x, \lambda) = \left[ \frac{\partial V(x, \lambda)}{\partial x} \right]^T / p(x, \lambda)
\]

\[
Tr \left[ \frac{\partial^2 V(x, \lambda)}{\partial x^2} \right] = \eta(x, \lambda)
\]

\[
V(x, \lambda) = -\int \eta(y, \lambda) \frac{c}{\|x-y\|^{d-2}} dy \quad \text{for } d \geq 3
\]

\[
V(x, \lambda) = \int p(y, \lambda) \left[ \log h(y) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right] \frac{c}{\|x-y\|^{d-2}} dy
\]

\[
\frac{\partial V(x, \lambda)}{\partial x} = \int p(y, \lambda) \left[ \log h(y) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right] \frac{c(2-d)(x-y)^T}{\|x-y\|^d} dy
\]

\[
\frac{\partial V(x, \lambda)}{\partial x} = E \left[ \left( \log h(y) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right) \frac{c(2-d)(x-y)^T}{\|x-y\|^d} \right]
\]

\[
\frac{\partial V(x_i, \lambda)}{\partial x} \approx \frac{1}{M} \sum_{j \in S_i} \left[ \left( \log h(x_j) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right) \frac{c(2-d)(x_i-x_j)^T}{\|x_i-x_j\|^d} \right]
\]

Poisson’s equation

like Coulomb’s law
derivation of Fourier transform particle flow:

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

take the Fourier transform:

\[ i\omega^T \mathcal{F}(pf) = -\mathcal{F}\left\{ p[\log h - \frac{d \log K(\lambda)}{d\lambda}] \right\} \]

\[ i\omega^T \int p(x, \lambda) f(x, \lambda) \exp(-i\omega^T x) dx = -\int p(x, \lambda)[\log h(x) - E(\log h)] \exp(-i\omega^T x) dx \]

approximate the integrals using the Monte Carlo sum over particles:

\[ i\omega^T \left\{ \frac{1}{N} \sum_{j=1}^{N} f(x_j, \lambda) \exp(-i\omega^T x_j) \right\} \approx -\frac{1}{N} \sum_{j=1}^{N} \left[ \log h(x_j) - E(\log h) \right] \exp(-i\omega^T x_j) \]

evaluate the above at k points in \( \omega \) (e.g., \( k = d \) or \( 2d \)) and write this as a linear operator on the unknown function \( f \):

\[ L(\omega)f = y(\omega) \]

\[ f(x) = L^# y \quad \text{in which } L^\# = \text{generalized inverse of } L \]

in which \( L^\# = L^T \left( LL^T \right)^{-1} \)
Lf = y written out explicitly:

\[
\begin{bmatrix}
\omega_1^T \sin(\omega_1^T x_1) & \omega_1^T \sin(\omega_1^T x_2) & \cdots & \omega_1^T \sin(\omega_1^T x_N) \\
\omega_1^T \cos(\omega_1^T x_1) & \omega_1^T \cos(\omega_1^T x_2) & \cdots & \omega_1^T \cos(\omega_1^T x_N) \\
\omega_2^T \sin(\omega_2^T x_1) & \omega_2^T \sin(\omega_2^T x_2) & \cdots & \cdots \\
\omega_2^T \cos(\omega_2^T x_1) & \omega_2^T \cos(\omega_2^T x_2) & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots \\
\omega_k^T \sin(\omega_k^T x_1) & \omega_k^T \sin(\omega_k^T x_2) & \cdots & \omega_k^T \sin(\omega_k^T x_N) \\
\omega_k^T \cos(\omega_k^T x_1) & \omega_k^T \cos(\omega_k^T x_2) & \cdots & \omega_k^T \cos(\omega_k^T x_N)
\end{bmatrix}
\begin{bmatrix}
f(x_1) \\
f(x_2) \\
f(x_N)
\end{bmatrix}_{dN \times 1}
\]

\[
= \begin{bmatrix}
- \sum_j [\log h(x_j) - E(\log h)] \cos(\omega_1^T x_j) \\
\sum_j [\log h(x_j) - E(\log h)] \sin(\omega_1^T x_j) \\
- \sum_j [\log h(x_j) - E(\log h)] \cos(\omega_2^T x_j) \\
\sum_j [\log h(x_j) - E(\log h)] \sin(\omega_2^T x_j) \\
\vdots \\
- \sum_j [\log h(x_j) - E(\log h)] \cos(\omega_k^T x_j) \\
\sum_j [\log h(x_j) - E(\log h)] \sin(\omega_k^T x_j)
\end{bmatrix}_{2k \times 1}
\]
optimization of points in k-space for Fourier transform
exact flow filter is many orders of magnitude faster per particle than standard particle filters

* Intel Corel 2 CPU, 1.86GHz, 0.98GB of RAM, PC-MATLAB version 7.7
particle flow filter is many orders of magnitude faster for real time computation (for the same or better estimation accuracy)

- 3 or 4 orders of magnitude faster per particle
- many orders of magnitude faster
- 3 or 4 orders of magnitude fewer particles
- avoids bottleneck in parallel processing due to resampling
exact flow: performance vs. number of particles

$\lambda = 1.2$, Linear, Large Initial Uncertainty

extremely unstable plant