Bayesian particle flow for estimation, decisions & transport

Fred Daum & Jim Huang

23 September 2014
nonlinear filter problem*

dynamical model of state:

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

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

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

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

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

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

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

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

curse of dimensionality for classic particle filter*

nonlinear filter*

prediction of conditional probability density from $t_{k-1}$ to $t_k$

solution of Fokker-Planck equation

Bayes' rule:

$$p(x, t_k | Z_k) = p(x, t_k | Z_{k-1}) p(z_k | x, t_k)$$

measurements

particle degeneracy*

prior density $g(x)$

likelihood $h(x)$

particles to represent the prior

particle degeneracy*

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)

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

(flow of density)

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

(flow of particles)

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

\( \lambda = 0 \)

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

\[
\lambda = 0 \quad \text{or} \quad \lambda = 1
\]

prior density

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

flow of density

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

sample from density

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

We design the particle flow by solving the above PDE for \( f \).
$\text{div}(pf) = p \left[ -\log h + \frac{d \log K}{d\lambda} \right]$

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>
<th>constant curvature flows (e.g. zero)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian densities</td>
<td>exponential family</td>
<td>Knothe-Rosenblatt flow</td>
<td>non-zero diffusion flow</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>renormalization group flow for log $K(\lambda)$ inspired by QFT</td>
<td>renormalization group flow for log $g(x)$ inspired by QFT</td>
<td>renormalization group flow for log$K(\lambda)$ and log $g(x)$</td>
<td>exponential family with non-zero diffusion</td>
</tr>
<tr>
<td>Gibbs sampler like flow (inspired by direct integration)</td>
<td>non-singular Jacobian flow (inspired by proof)</td>
<td>maximum entropy flow</td>
<td>Moser coupling flow</td>
<td>suboptimal Monge-Kantorovich</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:

\[
f = Ax + b
\]

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

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

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

for \( d \geq 2 \)

dx/d\lambda\ does not depend on K(\lambda), despite the fact that the PDE does!
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):

\[ \lambda = 1 \]
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 \( (C) \) 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:

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

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
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 \(g(x)\) from the exponential family, then the flow is:

\[
\frac{dx}{d\lambda} \approx -\left[\frac{\partial^2 \log q(x)}{\partial x^2} + \psi^T \frac{\partial^2 \theta(x)}{\partial x^2} + \lambda \frac{\partial^2 \log h}{\partial x^2}\right]^{-1}\left(\frac{\partial \log h}{\partial x}\right)^T
\]

in which the unnormalized prior density is:

\[
g(x) \approx q(x) \exp[\psi^T \theta(x)]
\]
BIG DIG (17 million cubic yards of dirt, one million truckloads & $24 billion)*

<table>
<thead>
<tr>
<th>item</th>
<th>particle flow</th>
<th>transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. purpose</td>
<td>fix particle degeneracy caused by Bayes’ rule</td>
<td>move physical objects from one probability density to another*</td>
</tr>
<tr>
<td>2. optimality criterion</td>
<td>NONE!</td>
<td>minimize convex functional</td>
</tr>
<tr>
<td>3. how to pick a unique solution</td>
<td>many different methods (e.g. min L² norm, given form, most stable flow, min convex function, more smoothness, max entropy, arbitrary, random, etc.)</td>
<td>minimize convex functional (e.g., dirt mover’s metric or Wasserstein metric)</td>
</tr>
<tr>
<td>4. computational complexity</td>
<td>numerical integration of ODE for each particle</td>
<td>solution of PDE (Poisson’s, Monge-Ampere, HJB, etc.)</td>
</tr>
<tr>
<td>5. high dimensional problems</td>
<td>d = 1 to 42</td>
<td>d = 1, 2, 3</td>
</tr>
<tr>
<td>6. solution of nice special cases</td>
<td>Gaussian, incompressible, irrotational, geodesic, exponential family, etc.</td>
<td>Moser coupling (1990), Brenier (1991-2011), Knothe-Rosenblatt (1952), etc.</td>
</tr>
<tr>
<td>7. homotopy of densities</td>
<td>log-homotopy</td>
<td>homotopy</td>
</tr>
<tr>
<td>8. stability of flow considered (e.g., Lyapunov stability)</td>
<td>often explicitly designed into algorithm</td>
<td>rarely</td>
</tr>
<tr>
<td>9. existence of flows proved</td>
<td>adapt proofs from transport theory</td>
<td>Shnirelman irrotational flow d &gt; 2, Moser &amp; Dacorogna d &gt;1, Brenier any d, et al.</td>
</tr>
<tr>
<td>10. conservation of probability mass along the flow</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>11. avoiding normalization of probability density is crucial for practical algorithms</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>12. explicitly compute normalized densities</td>
<td>no, log of unnormalized densities</td>
<td>yes</td>
</tr>
<tr>
<td>13. stiff ODEs mitigated</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>14. stochastic flows</td>
<td>rarely</td>
<td>rarely</td>
</tr>
</tbody>
</table>
superb books on transport theory

Very clear & accessible introduction; wonderful book!

# 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>
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*
BACKUP
REFERENCES:


<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>Peter Willett &amp; Sora Choi</td>
<td>numerical experiments</td>
<td>2011</td>
</tr>
<tr>
<td>McGill</td>
<td>Mark Coates &amp; Ding</td>
<td>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>Mark 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>Lingji Chen &amp; Raman Mehra</td>
<td>analysis of singularities in incompressible flow for $d = 1$</td>
<td>2010</td>
</tr>
<tr>
<td>Mitsubishi</td>
<td>Grover &amp; Sato</td>
<td>theory &amp; numerical experiments</td>
<td>2012</td>
</tr>
<tr>
<td>Cambridge</td>
<td>Peter Bunch &amp; Simon Godsill</td>
<td>theory &amp; numerical experiments</td>
<td>2014</td>
</tr>
<tr>
<td>Liverpool</td>
<td>Simon Maskell &amp; Flávio De Melo</td>
<td>relation to MCMC &amp; numerical experiments</td>
<td>2014</td>
</tr>
<tr>
<td>London</td>
<td>Simon Julier</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>METRON</td>
<td>Kristine Bell &amp; Larry Stone</td>
<td>numerical experiments &amp; theory</td>
<td>2014</td>
</tr>
<tr>
<td>Lockheed Martin</td>
<td>Nima Moshtagh &amp; Moses Chan</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>STR</td>
<td>Shozo Mori</td>
<td>theory &amp; numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>BU</td>
<td>Castanon, et al.</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>Toulouse</td>
<td>Marcelo Pereyra</td>
<td>numerical experiments &amp; theory</td>
<td>---</td>
</tr>
<tr>
<td>Bonn</td>
<td>Martin Ulmke &amp; Ahmed</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>Istanbul</td>
<td>Serdar Aslan</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>Wuhan</td>
<td>Lanlan Pang</td>
<td>numerical experiments</td>
<td>---</td>
</tr>
<tr>
<td>Tufts &amp; Hartford</td>
<td>Umarov &amp; Nelson</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>
The figure illustrates the relationship between the dimension of the state vector and nonlinearity or non-Gaussianity.

- **Dimension of the State Vector**: The y-axis represents the dimension of the state vector, ranging from 1 to 1000.
- **Nonlinearity or Non-Gaussianity**: The x-axis represents nonlinearity or non-Gaussianity.

Two types of filters are depicted:

- **Extended Kalman Filters**: Located in the lower-left quadrant, these filters are effective in systems with moderate to high nonlinearity.
- **Standard Particle Filters**: Located in the upper-right quadrant, these filters are more suitable for highly nonlinear systems.

A central region labeled "Particle Flow Filters" indicates the general area where both types of filters can be applied, depending on the specific characteristics of the system being modeled.
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></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>
exact flow filter is many orders of magnitude faster per particle than standard particle filters

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

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

\[
\frac{\partial p(x, \lambda)}{\partial \lambda} = -Tr\left[ \frac{\partial (pf)}{\partial x} \right]
\]

\[
\frac{\partial \log p(x, \lambda)}{\partial \lambda} p(x, \lambda) = -Tr\left[ \frac{\partial (pf)}{\partial x} \right]
\]

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

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

\[
div(q) = \eta
\]

\[
\eta = -p(x, \lambda) \left[ \log h(x) - \frac{\partial \log K(\lambda)}{\partial \lambda} \right]
\]
Fokker-Planck equation*

\[ \frac{\partial p}{\partial t} = -\frac{\partial p}{\partial x} f - Tr\left( \frac{\partial f}{\partial x} \right)p + \frac{1}{2} Tr\left( \frac{\partial^2 p}{\partial x^2} Q \right) \]

\[ p = p(x, t) \]

\[ p(x, t) = \text{probability density of } x \text{ at time } t \]

\[ \frac{dx}{dt} = f(x, t) + \frac{dw}{dt} \]

\[ Q = \text{covariance matrix of white noise } w(t) \]

Bayes’ rule:

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

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

\[x = \text{state vector}\]

\[t_k = \text{time of the } k^{th} \text{ measurement}\]

\[z_k = k^{th} \text{ measurement vector}\]

\[Z_k = \text{set of all measurements up to } & \text{ including time } t_k\]

\[Z_k = \{ z_1, z_2, z_3, \ldots, z_k \} \]
induced flow of particles for Bayes’ rule

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

\begin{align*}
\log p(x, \lambda) &= \log g(x) + \lambda \log h(x) - \log K(\lambda) \\
\frac{dx}{d\lambda} &= f(x, \lambda)
\end{align*}

why log?
convergence with $N$ for particle filters:

$$\sigma^2 \approx \frac{c}{N}$$

$N = \text{number of particles}$

$c = \text{so-called “constant” which depends on:}$

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
nonlinear filter performance (accuracy wrt optimal & computational complexity)
variation in initial uncertainty of $x$

$N = 1000$, Stable, $d = 10$, Quadratic, $\lambda = 0.6$
variation in eigenvalues of the plant ($\lambda$)

$N = 1000, d = 10, \text{Cubic}$

- $\lambda = 0.1$
- $0.5$
- $1.0$
- $1.1$
- $1.2$

25 Monte Carlo Trials
variation in dimension of x

N = 1000, \( \lambda = 1.0 \), Cubic

Dimensionless Error

Time

Dimension = 5
10
15
20
25
30

25 Monte Carlo Trials
\[ d = 12, n_y = 3, y = x^2, \text{SNR} = 20\text{dB} \]

**quadratic measurement nonlinearity**

- Dimensionless Error
- Number of Particles
- EKF
- PF

particle flow filter beats EKF by orders of magnitude
exact flow: performance vs. number of particles

![Graph showing the dimensionless error after 30 updates vs. number of particles for different dimensions and initial uncertainties. The graph indicates that as the number of particles increases, the error decreases, with different colors representing different dimensions and data points showing the effect of initial uncertainty.

extremely unstable plant

25 Monte Carlo Trials
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.
computing the Hessian of log $p$:

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

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

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

$C = \text{sample covariance matrix of particles for prior } (\lambda = 0)$

with Tychonov regularization; or EKF or UKF covariance matrix

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

$P = \text{sample covariance matrix of particles for } p(x, \lambda)$

with Tychonov regularization; or EKF or UKF covariance matrix
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:

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

hence

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

in which

\[ B = \lambda \frac{\partial^2 \log h(x)}{\partial x^2} \]
### how to mitigate stiffness in ODEs for certain particle flows*

<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>
</tbody>
</table>

**RED flows are extremely stiff**

<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)</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>

*non-stiff flows work well with Euler integration $\Delta\lambda = 0.1$*
new particle flow:

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

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:

\[
\frac{dx}{d\lambda} \approx P \left( \frac{\partial \theta(x)}{\partial x} \right)^T R^{-1} (z - \theta(x))
\]
importance of avoiding explicit computation of normalization $K(\lambda)^*$

(1) our 3 best flows do not explicitly compute the normalization of the conditional density

(2) small errors in computing $K(\lambda)$ can ruin the filter accuracy (e.g., Coulomb’s law flow & Fourier transform flow)

(3) similar effect in numerical weather prediction using transport theory (Bath University); must compute $K(\lambda)$ to machine precision!

(4) exploit this effect in designing flows

Red flows do not explicitly compute normalization $K(\lambda)$

<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>

RED flows do not explicitly compute normalization $K(\lambda)$.
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:

\[ f = Ax + b \]

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

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

\( f \) does not depend on \( K(\lambda) \), despite the fact that the PDE does!
incompressible particle flow

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

for \( d \geq 2 \)

\( f \) does not depend on \( K(\lambda) \), despite the fact that the PDE does!
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:

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

f does not depend on \( K(\lambda) \), despite the fact that the PDE does!
most general solution for particle flow:

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

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

the most general solution is:

\[
f = -C^# \left[ \log h - \frac{d \log K(\lambda)}{d\lambda} \right] + (I - C^# C)y
\]

in which \( C \) is a linear differentiable operator:

\[
C = \frac{\partial \log p}{\partial x} + \text{div}
\]

\( C^# = \text{generalized inverse of} \ C \)

\( y = \text{arbitrary} \ d - \text{dimensional vector} \)
idea #1 inspired by renormalization group flow:

\[
\frac{dx}{d\lambda} = -C^\# [\log h - \frac{d \log K(\lambda)}{d\lambda}] + (I - C^\# C) y
\]

\( f = \Gamma + \Pi y \)

\( \Pi = \text{projection into null-space of } C \)

\[
\frac{\partial f}{\partial L} = \frac{\partial \Gamma}{\partial L} + \frac{\partial \Pi}{\partial L} y = 0
\]

\[
y = -\left[ \frac{\partial \Pi}{\partial L} \right]^\# \left( \frac{\partial \Gamma}{\partial L} \right)
\]

\( L = \frac{d \log K(\lambda)}{d\lambda} \)

\( (.)^\# = \text{generalized inverse of } (.) \)

L rather than K (just like QFT): linear in L but not K; result does not depend on K itself; avoids singularity; slightly different & it works
idea #2 inspired by renormalization group flow:

\[
\frac{dx}{d\lambda} = -C^+[\log h - \frac{d \log K(\lambda)}{d\lambda}] + (I - C^#C)y
\]

\[f = \Gamma + \Pi y\]

\[\Pi = \text{projection into null-space of } C\]

\[
\frac{\partial f}{\partial L} = \frac{\partial \Gamma}{\partial L} + \frac{\partial \Pi}{\partial L} y = 0
\]

\[y = -\left[\frac{\partial \Pi}{\partial L}\right]^# \left(\frac{\partial \Gamma}{\partial L}\right)\]

\[L = \log g(x)\]

\[(.)^# = \text{generalized inverse of } (.)\]

L rather than g (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
idea #3 inspired by renormalization group flow*

$$\frac{dx}{d\lambda} = -C^\# [\log h - \frac{d \log K(\lambda)}{d\lambda}] + (I - C^\# C) y$$

$$f = \Gamma + \Pi y$$

$$\Pi = \text{projection into null-space of } C$$

$$\frac{\partial f}{\partial L} = \frac{\partial \Gamma}{\partial L} + \frac{\partial \Pi}{\partial L} y = 0$$

$$y = - \left[ \frac{\partial \Pi}{\partial L} \right]^\# \left( \frac{\partial \Gamma}{\partial L} \right)$$

$$L = \{ \log g(x), d \log K(\lambda) / d\lambda \}^T$$

$$(.)^\# = \text{generalized inverse of } (.)$$

computation of normalization using Fourier transform:

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

take the Fourier transform:

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

\[ i \omega^T \int p(x, \lambda) f(x, \lambda) \exp(-i \omega^T x) dx = - \int p(x, \lambda) [\log h(x) - \frac{d \log K(\lambda)}{d \lambda}] \exp(-i \omega^T x) dx \]

evaluate above at \( \omega = 0 \) (assuming that \( E(f) \) is finite)

\[ 0 = \int p(x, \lambda) \left[ \log h(x) - \frac{d \log K(\lambda)}{d \lambda} \right] dx \]

\[ \frac{d \log K(\lambda)}{d \lambda} = E[\log h(x)] \]

approximate the integral using the Monte Carlo sum over particles:

\[ \frac{d \log K(\lambda)}{d \lambda} \approx \frac{1}{N} \sum_{j=1}^{N} \log h(x_j) \]
MOVIES
exact particle flow for Gaussian densities:

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

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

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

\[
f = Ax + b
\]

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

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

automatically stable under very mild conditions & extremely fast
Inside = 10.6 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 13.8 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 16.6 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 17.6 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 21 percent, Magenta: truth, Green: PF estimate, Black: KF

\[ Ax + b \]
Inside = 21 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 23.8 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 26.4 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 27.6 percent, Magenta: truth, Green: PF estimate, Black: KF

$Ax + b$
incompressible particle flow

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

\[ \frac{dx}{d\lambda} = 0 \quad \text{for zero gradient} \]
Inside = 5.8 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 8.2 percent, Magenta: truth, Green: PF estimate, Black: KF

Hessian
Inside = 9.2 percent, Magenta: truth, Green: PF estimate, Black: KF

Hessian

0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
Inside = 11.2 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 11.8 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 12.8 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 12 percent, Magenta: truth, Green: PF estimate, Black: KF
Inside = 11.6 percent, Magenta: truth, Green: PF estimate, Black: KF
QUADRATIC MEASUREMENTS
d = 12, n_y = 3, y = x^2, SNR = 20dB

quadratic measurement nonlinearity
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF

1 x 10^5
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0
0.5
1
1.5

99
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
CUBIC
MEASUREMENTS
$d = 12$, $n_y = 3$, $y = x^3$, SNR = 20dB

Dimensionless Error vs. Number of Particles

Cubic measurement nonlinearity

Graph showing the comparison of EKF and PF methods for different numbers of particles.
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
Time = 1, Magenta: truth, Green: PF estimate, Black: KF
STABILITY
Stability of nonlinear filters

<table>
<thead>
<tr>
<th>Stability Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability of Kalman filter (1963 paper by Kalman)</td>
<td>Kalman filter is stable under very mild conditions (e.g., controllability &amp; observability); Kalman filter is stable for unstable plants*.</td>
</tr>
<tr>
<td>Stability of extended Kalman filters</td>
<td>EKF is unstable for unstable plants for typical nonlinear examples**</td>
</tr>
<tr>
<td>Convergence of particle filters</td>
<td>Assumes ergodicity, which implies stability of plant</td>
</tr>
<tr>
<td>Papers by Ramon van Handel &amp; Dan Crisan (2009)</td>
<td>Makes extremely strong assumptions (the complete state vector is measured)</td>
</tr>
<tr>
<td>Incompressible flow of particles</td>
<td>Explicitly stabilize filter &amp; flow for unstable plants without measuring complete state vector</td>
</tr>
<tr>
<td>Exact (compressible) particle flow</td>
<td>Automatically stable filter &amp; flow</td>
</tr>
</tbody>
</table>
most general solution for flow:

\[
\frac{dx}{d\lambda} = 0
\]

by the chain rule:

\[
\frac{d \log p(x, \lambda)}{d\lambda} = \frac{\partial \log p(x, \lambda)}{\partial x} \frac{dx}{d\lambda} + \frac{\partial \log p(x, \lambda)}{\partial \lambda} = 0
\]

\[
\frac{dx}{d\lambda} = -A^# \frac{\partial \log p(x, \lambda)}{\partial \lambda} + [I - A^T A/\text{Tr}(A^T A)]y
\]

in which \( A = \frac{\partial \log p(x, \lambda)}{\partial x} \)

\( y = \) arbitrary d-vector

\( A^# = A^T/(AA^T) \) for \( AA^T > 0 \),

and \( A^# = 0 \) otherwise.

pick \( y \) to maximize stability of filter
Pick y to maximize stability of filter:

$$\frac{dx}{d\lambda} = -A^# \frac{\partial \log p(x, \lambda)}{\partial \lambda} + [I - A^T A / Tr(A^T A)] y$$

For example, linearize about each particle

$$\frac{dx}{d\lambda} \approx Bx + Dy$$ with $$y = Kx$$

$$\Phi_{\text{filter}} \approx \Phi_{\text{plant}} \Phi_{\text{Bayes}}$$ where $$\Phi_{\text{Bayes}} \approx \exp(B + DK)$$

pick K to minimize the following stability measure

$$\sum_{j=1}^{d} |\lambda_j (\Phi_{\text{filter}})|^2 \leq Tr(\Phi_{\text{plant}} \Phi_{\text{plant}}^T \Phi_{\text{Bayes}}^T \Phi_{\text{Bayes}})$$

optimal $$K \approx -(I + B)$$ and thus optimal $$y \approx -x - Bx$$

but we can "delinearize" $$Bx$$, resulting in $$y \approx -x - \{Bx\}_{\text{delinearized}}$$

Hence, optimal $$y \approx -x + A^# \frac{\partial \log p(x, \lambda)}{\partial \lambda}$$ but $$DA^# = 0$$, and thus

Optimal y results in

$$\frac{dx}{d\lambda} \approx -A^# \frac{\partial \log p(x, \lambda)}{\partial \lambda} - Dx$$
errors in Schur’s inequality

random 10x10 real non-singular matrices
Exactly the same feedback is derived using standard control theory:

\[ y = -D^T W^{-1} x \]

in which the controllability Grammian is:

\[ W = \int_0^1 \exp(-sB)DD^T \exp(-sB^T) ds \]

where B is the linearization of the particle flow:

\[ \frac{dx}{d\lambda} = -A^\# \log(h) + (I - A^T A / AA^T)y \]

\[ \frac{dx}{d\lambda} \approx Bx + Dy \]

Using the facts that \( D = D^2 \) and D kills \( A^\# \) and a little algebra:

\[ y = -Dx \]
Particle filter accuracy depends on the plant stability & mixing

$$\lambda = \text{eigenvalue of plant}$$

d = 6, ny = 3, N = 500, #(MC) = 10, without -Dx
New theory (general flow) improves filter accuracy dramatically

$\lambda = \text{eigenvalue of plant}$

$d = 6, \ ny = 3, \ N = 500, \ #(MC) = 10, \ \text{with } -Dx$
<table>
<thead>
<tr>
<th>item</th>
<th>renormalization group flow in quantum field theory</th>
<th>particle flow for Bayes’ rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. purpose</td>
<td>avoid infinite integrals at all energy scales ($\mu$)</td>
<td>fix particle degeneracy in particle filters</td>
</tr>
<tr>
<td>2. PDE</td>
<td>linear first order PDE</td>
<td>linear first order PDE</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>
</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>
</tr>
</tbody>
</table>
| 5. algorithm | ODE for motion of particles in N-dimensional space (Tony Zee QFTNS p. 340) | $f = \frac{dx}{d\lambda}$  
x = particle in d-dimensions |
| 6. derivation of PDE | $dH(x)/d\mu = 0$  
$H(x) = $ Hamiltonian | Fokker-Planck equation & definition of log $p$ |
| 7. new idea for particle flow inspired by RNGF | $dH(x)/d\mu = 0$  
(we want $H$ to be scale invariant) | $\partial f/\partial g = 0$  
g = prior density |
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 $d$-$\varepsilon$)</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: $\frac{dH}{d\mu} = 0$</td>
<td>compute flow of particles that is invariant to errors in the prior density &amp; the normalization constant: $\frac{\partial f}{\partial g} = 0$</td>
</tr>
</tbody>
</table>
# exact recursive filters*

<table>
<thead>
<tr>
<th>Filter</th>
<th>Conditional density</th>
<th>Special condition on dynamics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman (1960)</td>
<td>$\eta = \text{Gaussian}$</td>
<td>$\partial f/\partial x = A(t)$</td>
</tr>
<tr>
<td>Beneš (1983)</td>
<td>$\eta \exp(\int f(x)dx)$</td>
<td>$f(x) = \partial V/\partial x$ and $\text{div}(f) + | f |^2 = x^*Ax + bx + c$</td>
</tr>
<tr>
<td>Daum (1986)</td>
<td>exponential family</td>
<td>$\frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial x} \left[ Qr^{-1} f \right] + \frac{1}{2} \xi - A \theta$</td>
</tr>
<tr>
<td></td>
<td>$p(x</td>
<td>Z) = p(x) \exp[\theta(x)\Psi(Z)]$</td>
</tr>
</tbody>
</table>


---

$\eta$ = Gaussian

$\frac{\partial}{\partial x}$

$\exp(\int f(x)dx)$

$\partial V/\partial x$

$\text{div}(f)$

$\| f \|^2$

$x^*Ax + bx + c$

$\frac{\partial \theta}{\partial t}$

$\frac{\partial \theta}{\partial x}$

$Qr^{-1} f$

$A \theta$

$r = \frac{\partial \log p(x)}{\partial x}$

$\xi_j$

$\text{Tr} \left[ Q \frac{\partial^2 \theta}{\partial x^2} \right]$
a miracle*

$$p(x,t|Z_t) = p(x,t|\Psi_t(Z_t))$$

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

\( \lambda \) can compute Hessians using calculus or 2\(^{nd}\) 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:

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

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>$Q = \text{covariance matrix of diffusion in flow}: \dot{x} = f(x, \lambda) ;d\lambda + \sqrt{Q} ;dw$</td>
</tr>
</tbody>
</table>
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 $Q \neq 0$:

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

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

\[ \frac{\partial \text{div}(f)}{\partial x} = 0 \]
linear first order highly underdetermined PDE:

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

\[
div(pf) = \eta
\]

let \( q = pf \) (p = known & f = unknown)

\[
div(q(x, \lambda)) = \eta
\]

\[
\eta(x, \lambda) = -p(x, \lambda) \left[ \log h(x) - \frac{d \log K(\lambda)}{d\lambda} \right]
\]

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

(1) we want a stable flow
(2) we want a full rank flow
(3) we want a fast algorithm to approximate f accurately (roughly 1%)
irrotational particle flow:

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

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

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

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

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

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

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

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

take the Fourier transform:

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

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

approximate the integrals using the Monte Carlo sum over particles:

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

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

\[
\begin{bmatrix}
\omega_1^T \sin(\omega_1^T x_1) & \omega_1^T \sin(\omega_1^T x_2) & \cdots & \omega_1^T \sin(\omega_1^T x_N) \\
\omega_1^T \cos(\omega_1^T x_1) & \omega_1^T \cos(\omega_1^T x_2) & \cdots & \omega_1^T \cos(\omega_1^T x_N) \\
\omega_2^T \sin(\omega_2^T x_1) & \omega_2^T \sin(\omega_2^T x_2) & \cdots & \cdots \\
\omega_2^T \cos(\omega_2^T x_1) & \omega_2^T \cos(\omega_2^T x_2) & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\omega_k^T \sin(\omega_k^T x_1) & \omega_k^T \sin(\omega_k^T x_2) & \cdots & \omega_k^T \sin(\omega_k^T x_N) \\
\omega_k^T \cos(\omega_k^T x_1) & \omega_k^T \cos(\omega_k^T x_2) & \cdots & \omega_k^T \cos(\omega_k^T x_N)
\end{bmatrix}
\begin{bmatrix}
f(x_1) \\
f(x_2) \\
f(x_N)
\end{bmatrix}_{dN \times 1}
= 
\begin{bmatrix}
- \sum_j \log h(x_j) - E(\log h) \cos(\omega_i^T x_j) \\
\sum_j \log h(x_j) - E(\log h) \sin(\omega_i^T x_j) \\
- \sum_j \log h(x_j) - E(\log h) \cos(\omega_2^T x_j) \\
\sum_j \log h(x_j) - E(\log h) \sin(\omega_2^T x_j) \\
\vdots \\
- \sum_j \log h(x_j) - E(\log h) \cos(\omega_k^T x_j) \\
\sum_j \log h(x_j) - E(\log h) \sin(\omega_k^T x_j)
\end{bmatrix}_{2k \times dN}
\]
optimization of points in k-space for Fourier transform
most general solution for incompressible flow:

\[
\frac{dx}{d\lambda} = ?
\]

by the chain rule:

\[
\frac{d \log p(x, \lambda)}{d\lambda} = \frac{\partial \log p(x, \lambda)}{\partial x} \frac{dx}{d\lambda} + \frac{\partial \log p(x, \lambda)}{\partial \lambda} = 0
\]

\[
\frac{dx}{d\lambda} = -A^\# \log h(x) + [I - A^\# A] y
\]

in which

\[
A = \frac{\partial \log p(x, \lambda)}{\partial x}
\]

\[y = \text{arbitrary direction vector}\]

\[A^\# = A^T / (AA^T) \text{ for } AA^T > 0,\]

and \[A^\# = 0 \text{ otherwise.}\]
$N = 1000$  \hspace{1cm}  \sigma_0 = 100$
$\sigma_0 = 10$

$\Delta \lambda_0 = 1e^{-3}$
\[ \sigma_0 = 100 \]

\[ \Delta \lambda_0 = 1e^{-5} \]
$\sigma_0 = 1000$

$\Delta \lambda_0 = 1e-7$
$\sigma_0 = 1e4$

$\Delta \lambda_0 = 1e-9$
$\Delta \lambda_0 = 1e-11$
\[ \Delta \lambda_0 = 1e-13 \]
\[ \Delta \lambda_0 = 1e-15 \]
Fisher information matrix

\[ J = -E \left[ \frac{\partial^2 \log p}{\partial x^2} \right] \]
zero curvature flow:

we want to solve the following PDE for the flow $f$ :

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

assume that the flow has zero curvature:

$$\frac{d^2 x}{d \lambda^2} = 0$$

but $f = \frac{dx}{d \lambda}$

hence $\frac{df}{d \lambda} = 0$

using this condition (after several pages of calculations) results in :

$$f^T \left[ \frac{\partial^2 \log p}{\partial x^2} \right] f + 2 \frac{\partial \log h}{\partial x} f = \frac{d^2 \log K}{d \lambda^2}$$
zero curvature flow:

\[ f^T \left[ \frac{\partial^2 \log p}{\partial x^2} \right] f + 2 \frac{\partial \log h}{\partial x} f = \frac{d^2 \log K}{d\lambda^2} \]

(1) vector Riccati equation for \( f \) rather than a PDE for \( f \)!
(2) highly underdetermined algebraic equation for \( f \)
(3) we can solve for \( f \) exactly in closed form!
(4) Hessian of \( \log p \) is similar to the Fisher information matrix (we can exploit this to solve for \( f \) exactly in closed form)
(5) for nonlinear measurements with Gaussian noise, it is easy & fun to solve for \( f \) explicitly!
exploit non-singular symmetric pre-Fisher information matrix:

\[
\begin{align*}
    f^T \left[ \frac{\partial^2 \log p}{\partial x^2} \right] f + 2 \frac{\partial \log h}{\partial x} f &= \frac{d^2 \log K}{d \lambda^2} \\
    f^T \left[ \frac{\partial^2 \log p}{\partial x^2} \right] f + 2 \frac{\partial \log h}{\partial x} f &= \frac{d^2 \log K}{d \lambda^2}
\end{align*}
\]

we can always write the above in the following canonical form :

\[
\|\tilde{f}\|^2 - 2 \frac{\partial \log h}{\partial x} \sqrt{H^{-1}} \tilde{f} = - \frac{d^2 \log K}{d \lambda^2}
\]

in which :

\[
H = - \frac{\partial^2 \log p}{\partial x^2}
\]

\[
\tilde{f} = \sqrt{H} f
\]
solution of general vector Riccati equation:

\[ \| \tilde{f} - b \|^2 = 0 \]

this is a single scalar-valued equation in \( d \) unknowns, but it nevertheless has obvious the unique solution: \( \tilde{f} = b \)

\[ (\tilde{f} - b)^T (\tilde{f} - b) = 0 \]

\[ \| \tilde{f} \|^2 - 2b^T \tilde{f} + \| b \|^2 = 0 \]

obviously this equation also has the unique solution: \( \tilde{f} = b \)

Encouraged by the above simple example, now consider our equation:

\[ \| \tilde{f} \|^2 + b^T \tilde{f} + c = 0 \]

let \( \tilde{f} = kb \) in which \( k \) is a scalar

\[ k^2 \| b \|^2 + k \| b \|^2 + c = 0 \]

\[ k^2 + k + c / \| b \|^2 = 0 \] which has the solution: \( k = \frac{-1 \pm \sqrt{1 - 4c / \| b \|^2}}{2} \)
geometrical interpretation of solution:

\[ \| \tilde{f} \|^2 + b^T \tilde{f} + c = 0 \]

let \( \tilde{f} = kb \) in which \( k \) is a scalar

\[ k^2 \| b \|^2 + k \| b \|^2 + c = 0 \]

\[ k^2 + k + c / \| b \|^2 = 0 \] which has the solution:

\[ k = \frac{-1 \pm \sqrt{1 - 4c / \| b \|^2}}{2} \]
solution of our vector Riccati equation:

\[ \tilde{f} = \sqrt{H} f \]

\[ f = \sqrt{H^{-1}} \tilde{f} \]

\[ \tilde{f} = kb \]

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

\[ k = \frac{-1 \pm \sqrt{1 - 4c / \|b\|^2}}{2} \]

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

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

\[ f = 2k \left[ \frac{\partial^2 \log p}{\partial x^2} \right]^{-1} \left( \frac{\partial \log h}{\partial x} \right)^T \]
fast Ewald’s method vs. Coulomb’s law

<table>
<thead>
<tr>
<th>item</th>
<th>fast Ewald method in physics &amp; chemistry*</th>
<th>Coulomb’s law with fast approximate k-NN</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. dimension of x</td>
<td>d = 3</td>
<td>d = 3 to 30</td>
<td>rapid decay of Coulomb kernel in higher d helps!</td>
</tr>
<tr>
<td>2. relative error desired</td>
<td>0.0001 or better</td>
<td>1% to 10%</td>
<td>all Ewald methods the same for 1% accuracy</td>
</tr>
<tr>
<td>3. cut-off in x space</td>
<td>fixed distance</td>
<td>random per k-NN</td>
<td>automatic space-taper to weight convolution</td>
</tr>
<tr>
<td>4. desired force</td>
<td>on mesh</td>
<td>at particles</td>
<td>big difference!</td>
</tr>
<tr>
<td>5. neutral charge</td>
<td>locally enforced</td>
<td>locally enforced</td>
<td>crucial!!!</td>
</tr>
<tr>
<td>6. smoothing charge</td>
<td>Gaussian</td>
<td>no explicit smoothing</td>
<td>Debye kernel</td>
</tr>
<tr>
<td>7. k-space or real-space</td>
<td>both</td>
<td>real space (x)</td>
<td>no FFT needed for Coulomb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>item</th>
<th>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 minimum action, etc.</td>
</tr>
<tr>
<td>7. how to pick a solution</td>
<td>24 distinct methods</td>
<td>minimize convex functional</td>
</tr>
<tr>
<td>8. stability of flow explicitly 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>Poisson’s PDE or HJB PDE or Monge-Ampere PDE etc.</td>
</tr>
<tr>
<td>11. solution of PDE for nice special cases</td>
<td>incompressible, irrotational, Gaussian, geodesic, etc.</td>
<td>Moser (1965 &amp; 1990), Brenier (1991), Knothe-Rosenblatt (1952)</td>
</tr>
<tr>
<td>12. math theory for existence of incompressible flow, etc.</td>
<td>borrow Shnirelman’s theorem, and Moser &amp; Dacorogna (1990)</td>
<td>Shnirelman’s theorem  for d ≥ 3, Moser &amp; Dacorogna</td>
</tr>
</tbody>
</table>