Particle Flow Filters: MPI 2023 Report

Pak-Wing Fok
University of Delaware
pakwing@udel.edu

Maxim J. Goldberg
Ramapo College of NJ
mgoldber@ramapo.edu

Richard Moore
Society for Industrial and Applied Mathematics
moore@siam.org

Christopher Raymond
University of Delaware
craymond@udel.edu

Praneeth Vepakomma
MBZUAI and MIT
vepakom@mit.edu

Manuchehr Aminian
California State Polytechnic Unviersity Pomona
maminian@cpp.edu

Erik Bergland
Brown University
erik_bergland@brown.edu

Hunter Akins
UC San Diego
fakins@ucsd.edu

Pietro Zanin
Northwestern University
pietrozanin2027@u.northwestern.edu

Li Meng
University of Houston
lmeng3@cougarnet.uh.edu

Sameer Pokhrel
University of Cincinnati
pokhresr@mail.uc.edu

Mohammad Javad Khojasteh
UC San Diego
mjkhojasteh8@gmail.com

Florian Meyer
UC San Diego
flmeyer@ucsd.edu

Maxim Zyskin
University of Oxford
maximzyskin@gmail.com

June 2023

1 Introduction

One of the main uses of filters is to estimate the state of a noisy dynamical system. They have numerous applications in robotics and tracking systems. The filter estimates the new state of a system from noisy measurements. This calculation relies on using Bayes' Theorem to calculate a posterior distribution, thereby providing the statistics of the the new state.

One of the most well-known filters is the Kalman filter which assumes the noise is Gaussian. However, the posterior can be difficult to calculate when the noise is non-Gaussian. In this case, particle filters can be quite competitive. The idea behind a particle filter is to approximate posterior distributions by weights associated with a set of random sample points (particles). The weights are updated sequentially. However, a currently unresolved problem with particle filters is "particle degeneracy," where most particle weights evolve to become close to zero, the result of sampling the posterior in places where the density is also close to zero. Particle filters also suffer from the curse of dimensionality [6] in which exponentially more particles are needed to maintain a certain level of accuracy. (Question: is it true that for larger dimensions, it's easier to get particle degeneracy?)

Particle flow filters are designed to address this issue. In these kinds of filters, the particles are advected to desirable positions in the posterior in order to avoid particle degeneracy. A key ingredient is the design of the flow that advects the particles. Although particle flow filters can out-perform classical particle filters, designing the flows can be challenging. For example, the resulting differential equations to advect the particles often become stiff.

Our team was tasked with the challenge of designing a particle filter flow that reduces stiffness and is more general than the results in [2]. This report discusses the ideas and results that our team produced.

1.1 Basic Mathematical Framework

For any events A and B, Bayes' Theorem states that

$$\underbrace{P(A|B)}_{\text{posterior}} = \underbrace{P(A)}_{\text{prior}}\underbrace{P(B|A)}_{\text{likelihood}}/P(B) \tag{1}$$

We introduce the symbols g and h for the prior distribution and likelihood functions respectively. We also introduce a parameter $0 \le \lambda \le 1$ and a function $K(\lambda)$ such that $K(0) = 1$. If we define

$$p(x, \lambda) = \frac{g(x)h(x)^{\lambda}}{K(\lambda)}, \tag{2}$$

we notice that when $\lambda = 0$, $p = g$ is just the prior distribution but when $\lambda = 1$, p becomes the posterior distribution. Therefore, eq. (2) interpolates between the prior and the posterior distribution and we can think of λ as a time-like variable: as this variable increases, the prior evolves into the posterior. Eq. (2) defines a *homotopy* between the prior g and $p(x, \lambda)$.

Taking logs of both sides of (2),

$$\log p(x, \lambda) = \log g(x) + \lambda \log h(x) - \log K(\lambda) \tag{3}$$

The idea of a particle filter is that we introduce a set of particles which are used to sample the prior. We wish to advect these particles to "good" locations so that the posterior is adequately sampled. For example, there should be more particles in places where the posterior density is larger.

Suppose we have a flow $\mathbf{f}$ that advects particles according to

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda} = \mathbf{f}. \tag{4}$$

We can design $\mathbf{f}$ so that the particle filter has desirable properties. The PDE obeyed by p is Liouville's equation

$$\frac{\partial p}{\partial \lambda} + \nabla \cdot (p\mathbf{f}) = 0. \tag{5}$$

The stochastic version of (4) is

$$d\mathbf{x} = \mathbf{f}d\lambda + d\mathbf{w}, \qquad E[d\mathbf{w}d\mathbf{w}^T] = Qd\lambda, \tag{6}$$

for a matrix Q and Brownian measure $d\mathbf{w}$, and the associated PDE is not Liouville's equation, but the *Fokker-Planck* equation:

$$\frac{\partial p}{\partial \lambda} = -\nabla \cdot (\mathbf{f}p) + \frac{1}{2}\nabla \cdot (Q\nabla p) \tag{7}$$

In [2], several $\mathbf{f}$ and Q were proposed. However, it seems that with these choices, the SDE (6) often became stiff. Our team was challenged to find alternative $\mathbf{f}$ and Q that reduce stiffness in the problem.

This report is organized in the following way. In sections 2 and 3, we try to understand the origin of the stiffness by studying a 1D problem where all distributions are Gaussian. In sections 4 and 5, we propose some mathematical modifications of $\mathbf{f}$ and Q that extend the results of [3]. In section 6, we propose an inertial regularization scheme for the stochastic particle dynamics called HOMEBREW for mitigating stiffness. In section 7, a specific choice for $\mathbf{f}$ is proposed based on kernel density estimators. In section 8, we analyze the stiffness in the Crouse's 2018 paper. Finally, in section 9 we consider modifications to the homotopy (2) and in section 10 we consider an approach to solve for $\mathbf{f}$ and Q in non-Gaussian cases, either directly or via solving ODEs in λ.

2 Deterministic flow with 1D Gaussian posterior

author: Hunter Akins, `fakins@ucsd.edu`, UCSD

The idea is to derive the drift function $f(x,\lambda)$ for the flow corresponding to a 1D Gaussian prior on a state x and a direct measurement of the state $z = x+n$, where n is zero-mean Gaussian noise with mean σ_1^2. The thinking is that this simple toy problem provides some insight into the way the flow works and maybe gives a clue on the origin of stiffness. It is a simpler result than the result in [7], equations 8, 9 and 10. In that paper, they present the drift function $f(x,\lambda)$ for an n-dimensional Gaussian with prior covariance P_0 and prior mean $\bar{x}$ and a measurement of the state $z = Hx + n$, where the measurement noise n is zero-mean gaussian with covariance R.

The deterministic (zero diffusion) form of the Fokker-Planck equation is

$$\frac{\partial p}{\partial \lambda} = -\mathrm{div}(fp) = -\frac{\partial f}{\partial x}p - \frac{\partial p}{\partial x}f \ . \tag{8}$$

Divide through by p (assumes that it's nowhere equal to zero which is true for us) to get the equivalent form

$$\frac{\partial f}{\partial x} + \frac{\partial \log p}{\partial x}f = -\frac{\partial \log p}{\partial \lambda} \ . \tag{9}$$

For given variances σ_0^2, σ_1^2 and mean z, define a Gaussian prior and likelihood

$$\begin{aligned} g(x) &= \exp\left\{-\frac{x^2}{2\sigma_0^2}\right\}, & (10)\\ h(x) &= \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left(-\frac{(x-z)^2}{2\sigma_1^2}\right). & (11)\end{aligned}$$

The "tempered" posterior for a given λ is $p = g(x)h(x)^\lambda/K(\lambda)$. Note that

$$g(x)h(x)^\lambda \propto \exp\left\{-\frac{x^2}{2\sigma_0^2} - \frac{(x-z)^2}{2\sigma_1^2/\lambda}\right\} \propto \exp\left\{-\frac{\sigma_0^2+\sigma_1^2/\lambda}{2\sigma_0^2\sigma_1^2/\lambda}(x - \frac{\sigma_0^2}{\sigma_0^2+\sigma_1^2/\lambda}z)^2\right\} \ .$$

Therefore the product $p(x)$ must also be Gaussian with mean

$$\mu(\lambda) = \frac{\sigma_0^2}{\sigma_0^2+\sigma_1^2/\lambda}z \tag{12}$$

and variance

$$\Sigma(\lambda) = \frac{\sigma_0^2\sigma_1^2/\lambda}{\sigma_0^2+\sigma_1^2/\lambda} = \frac{\sigma_0^2\sigma_1^2}{\lambda\sigma_0^2+\sigma_1^2} \ . \tag{13}$$

Therefore, the normalizing factor is $K(\lambda) = \sqrt{2\pi\Sigma(\lambda)}$. Define

$$\beta(\lambda) = \frac{\sigma_0^2}{\sigma_0^2+\sigma_1^2/\lambda} = \frac{\sigma_0^2\lambda}{\lambda\sigma_0^2+\sigma_1^2}. \tag{14}$$

We can then write

$$\log p = -\frac{(x-\beta(\lambda)z)^2}{2\Sigma(\lambda)} - \frac{1}{2}\log(2\pi\Sigma(\lambda)) \ . \tag{15}$$

Meanwhile we have that $-\frac{\partial \log p}{\partial \lambda} = \frac{\partial f}{\partial x} + \frac{\partial \log p}{\partial x}f$. This has solution

$$f(x) = c_1/I(x) + \frac{1}{I(x)}\int^x -\frac{\partial \log p}{\partial \lambda}I(x')\mathrm{d}x' \ ,$$

with

$$I(x') = \exp\int^x \frac{\partial \log p}{\partial x} = p(x,\lambda) \ .$$

We have

$$\frac{\partial \log p}{\partial \lambda} = \frac{z(x-\beta z)\frac{\partial \beta}{\partial \lambda}}{\Sigma} + \frac{(x-\beta z)^2}{2\Sigma^2}\frac{\partial \Sigma}{\partial \lambda} - \frac{1}{2}\frac{\frac{\partial \Sigma}{\partial \lambda}}{\Sigma} \ . \tag{16}$$

Examining $\Sigma(\lambda)$, we have

$$\frac{\partial \Sigma}{\partial \lambda} = -\frac{\sigma_0^2\sigma_1^2}{(\lambda\sigma_0^2+\sigma_1^2)^2}\sigma_0^2 = -\frac{\sigma_0^2\Sigma(\lambda)}{(\lambda\sigma_0^2+\sigma_1^2)}. \tag{17}$$

We also have

$$\frac{\partial \beta}{\partial \lambda} = \frac{\sigma_0^2\sigma_1^2}{\lambda^2(\sigma_0^2+\sigma_1^2/\lambda)^2} = \frac{\sigma_0^2\sigma_1^2}{(\lambda\sigma_0^2+\sigma_1)^2} = \frac{\Sigma(\lambda)}{\lambda\sigma_0^2+\sigma_1^2} \ . \tag{18}$$

Plugging these in gives

$$\frac{\partial \log p}{\partial \lambda} = \frac{z(x-\beta z)}{(\lambda\sigma_0^2+\sigma_1^2)} - \frac{\sigma_0^2(x-\beta z)^2}{2\Sigma(\lambda\sigma_0^2+\sigma_1^2)} + \frac{1}{2}\frac{\sigma_0^2}{(\lambda\sigma_0^2+\sigma_1^2)} \ . \tag{19}$$

To compute f, we need to compute

$$\int^x p\frac{\partial \log p}{\partial \lambda} = \int^x \frac{e^{-(x'-\beta z)^2/2\Sigma(\lambda)}}{(\lambda\sigma_0^2+\sigma_1^2)\sqrt{2\pi\Sigma(\lambda)}}\left[z(x'-\beta z) - \frac{\sigma_0^2(x'-\beta z)^2}{2\Sigma} + \frac{1}{2}\sigma_0^2\right]\mathrm{d}x' \ . \tag{20}$$

Introduce $\mu \equiv \frac{(x-\beta z)}{\sqrt{2\Sigma(\lambda)}}$ to get

$$\int^x p\frac{\partial \log p}{\partial \lambda} = \int^{\mu(x)} \frac{1}{(\lambda\sigma_0^2+\sigma_1^2)\sqrt{2\pi\Sigma(\lambda)}}e^{-\mu^2}\left[z\sqrt{2\Sigma}\mu - \sigma_0^2\mu^2 + \frac{1}{2}\sigma_0^2\right]\sqrt{2\Sigma}\mathrm{d}\mu \ . \tag{21}$$

Use the facts

$$\int e^{-x^2}\mathrm{d}x = -\frac{1}{2}e^{-x^2},$$

$$\int e^{-x^2}x^2\mathrm{d}x = \frac{1}{4}(\sqrt{\pi}\,\mathrm{erf}(x) - 2e^{-x^2}x)$$

and

$$\int e^{-x^2}\mathrm{d}x = \frac{1}{2}\sqrt{\pi}\,\mathrm{erf}(x).$$

$$\int^x p\frac{\partial \log p}{\partial \lambda} = \frac{\sqrt{2\Sigma}}{(\lambda\sigma_0^2+\sigma_1^2)\sqrt{2\pi\Sigma(\lambda)}}\left[-\frac{1}{2}z\sqrt{2\Sigma}\exp^{-\mu^2} - \sigma_0^2\frac{1}{4}(\sqrt{\pi}\,\mathrm{erf}(\mu) - 2e^{-\mu^2}\mu) + \frac{1}{2}\sigma_0^2\frac{1}{2}\sqrt{\pi}\,\mathrm{erf}(\mu)\right] . \tag{22}$$

Cancel some stuff

$$\int^x p\frac{\partial \log p}{\partial \lambda} = \frac{\sqrt{2\Sigma}}{(\lambda\sigma_0^2+\sigma_1^2)\sqrt{2\pi\Sigma(\lambda)}}e^{-\mu^2}\left[-\frac{z}{2}\sqrt{2\Sigma} + \frac{\sigma_0^2}{2}\mu\right] . \tag{23}$$

Note that

$$p(x,\lambda) = \exp\bigl(-\mu(x)^2\bigr)/\sqrt{2\pi\Sigma}.$$

Therefore

$$1/p\int^x p\frac{\partial \log p}{\partial \lambda} = \frac{1}{(\lambda\sigma_0^2+\sigma_1^2)}\left[-z\Sigma + \frac{\sigma_0^2}{2}(x-\beta z)\right] \tag{24}$$

We can also express this as

$$1/p\int^x p\frac{\partial \log p}{\partial \lambda} = \frac{\sigma_0^2}{2(\lambda\sigma_0^2+\sigma_1^2)}x - \left(\beta\sigma_0^2/2 + \frac{\sigma_0^2\sigma_1^2}{(\lambda\sigma_0^2+\sigma_1^2)^2}\right)z \tag{25}$$

This gives

$$f(x,\lambda) = c_1/p - \frac{\sigma_0^2}{2(\lambda\sigma_0^2+\sigma_1^2)}x + \left(\beta\sigma_0^2/2 + \frac{\sigma_0^2\sigma_1^2}{(\lambda\sigma_0^2+\sigma_1^2)^2}\right)z \ . \tag{26}$$

Expanding out β gives

$$f(x,\lambda) = c_1/p - \frac{\sigma_0^2}{2(\lambda\sigma_0^2+\sigma_1^2)}x + \frac{\sigma_0^2}{2(\lambda\sigma_0^2+\sigma_1^2)^2}\left(\lambda^2\sigma_0^2 + \lambda\sigma_0^2\sigma_1^2 + 2\sigma_1^2\right)z \; . \tag{27}$$

To compare with Daum and Huang (2012), their equation 8 is

$$\frac{\mathrm{d}x}{\mathrm{d}\lambda} = A(\lambda)x + b(\lambda) \; , \tag{28}$$

where

$$A(\lambda) = -\frac{1}{2}PH^T(\lambda HPH^T + R)^{-1}H \tag{29}$$

and

$$b(\lambda) = (I + 2\lambda A)[(I+\lambda A)PH^T R^{-1}z + A\bar{x}] \; . \tag{30}$$

Evaluating their expression for $P = \sigma_0^2$, $H = 1$, $R = \sigma_1^2$, and $\bar{x} = 0$, we get that

$$A(\lambda) = -\frac{\sigma_0^2}{2(\lambda\sigma_0^2+\sigma_1^2)} \tag{31}$$

in agreement with what I have obtained. However, my $b(\lambda)$ does not seem to agree. Theirs is correct (and confirmed by simulations). Where's the error?

3 Stochastic flow with 1D Gaussian posterior

author: Richard Moore, moore@siam.org, SIAM

With nonzero diffusion, the Fokker-Planck equation takes the form

$$\frac{\partial p}{\partial \lambda} = -\mathrm{div}(fp) + \frac{1}{2}\mathrm{div}\left(Q\nabla p\right) \tag{32}$$

where Q is a symmetric positive semidefinite matrix. Evaluating this at the λ-homotopy of the one-dimensional posterior above gives

$$\frac{\partial \log h}{\partial x} = -f\frac{\partial^2 \log p}{\partial x^2} - \frac{\partial^2 f}{\partial x^2} - \frac{\partial}{\partial x}\log p\frac{\partial f}{\partial x} + \frac{\partial}{\partial x}\left[\frac{1}{2p}\frac{\partial}{\partial x}\left(Q\frac{\partial p}{\partial x}\right)\right] . \tag{33}$$

Following [8], we can define the drift using the first two terms of Eqn. (33) to find

$$f = -\Sigma(x-z)/\sigma_1^2. \tag{34}$$

where Σ is defined in eq. (13) and z and σ_1 are the mean and standard deviation of the likelihood in eq. (11). Suppose $\epsilon = \sigma_1^2/\sigma_0^2 \ll 1$. Then the variance of the likelihood is much smaller than the prior and the distribution of the likelihood is also much more sharply-peaked. Without diffusion ($Q = 0$), the solution of

$$\frac{dx}{d\lambda} = f = -\frac{\Sigma(x-z)}{\sigma_1^2} = -\frac{(x-z)}{\lambda+\epsilon} \tag{35}$$

with $x(\lambda = 0) = x_0$ is

$$x = z + \frac{\epsilon(x_0 - z)}{\lambda+\epsilon}, \tag{36}$$

and we see that the evolution of a particle from x_0 to the stable fixed point z is *algebraic* rather than exponential because of λ in the denominator of (35). With diffusion ($Q \neq 0$) and f given by (34), the Fokker-Planck equation is satisfied when

$$Q = \Sigma^2/\sigma_1^2. \tag{37}$$

Figure 1 shows the evolution of the prior into the posterior when $\epsilon = 0.1$. We observe that most of the dynamics occurs over an initial time $0 \leq \lambda \leq O(\epsilon)$.

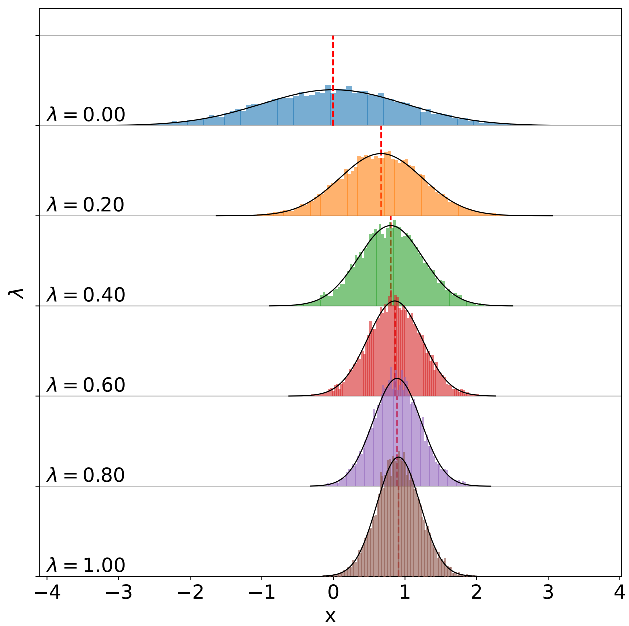

Figure 1: Simulation of ensemble of $n = 10^4$ particles evolved according to stochastic dynamics [equation ref here]. The two-timescale choice $\varepsilon = 0.1$ leads to most of the evolution of the distribution from $\lambda = 0$ to $\lambda = 1$ occurring during $(0, \varepsilon)$. Shaded colors show the empirical probability density via simple histogram of the particle ensemble, using 100 uniform bins. The black curves (expected distribution $p(x, \lambda;\, z, \varepsilon)$) agrees with results of the simulation.

4 Generalizations of Q

author: Erik Bergland, erik_bergland@brown.edu, Brown University

We look to extend the class of solutions from [3]. The next simplest choice of Q that includes some dependence on space while guaranteeing symmetric positive-definiteness is

$$Q(x, \lambda) = ||x - \mu||^2 R(\lambda) \tag{38}$$

where μ is some parameter which could be chosen and R is symmetric positive definite. Essentially, the idea is to reduce the influence of the noise term once we are in the vicinity of some pre-selected point, since larger noise terms are more difficult to compute accurately. One could also allow μ to depend on λ. Keeping the notation of the paper, we note that if we have a deterministic drift f^d and we make a choice such as the one above for Q, we find a stochastic drift f^s given by

$$f^s = f^d + \frac{1}{2} Q \nabla \log(p) + \frac{1}{2} f^Q \tag{39}$$

where f^Q is given by

$$f_i^Q = \sum_j \frac{\partial Q_{i,j}}{\partial \alpha_j} \tag{40}$$

For the remainder of this section, we will make use of the Einstein summation convention, so that any repeated index is summed over. For example, we can write the equation above as

$$f_i^Q = \frac{\partial Q_{i,j}}{\partial \alpha_j}$$

or

$$f_i^Q = \partial_j Q_{i,j}$$

We find that in our case, the corrective term f^Q is given by

$$\begin{aligned} f_i^Q &= \partial_j Q_{i,j} \\ &= R_{ij}\partial_j ||x-\mu||^2 \\ &= 2R_{ij}(x_j-\mu_j) \end{aligned}$$

As in the paper, let us now compute the Jacobian of the stochastic drift in an effort to understand stiffness. We have that

$$f_i^s = f_i^d + \frac{||x-\mu||^2}{2} R_{ij}\partial_j \log(p) + R_{ij}(x_j-\mu_j) \tag{41}$$

Differentiating by α_l, we find that

$$\begin{aligned} \partial_l f_i^s &= \partial_l f_i^d + \frac{\partial_l ||x-\mu||^2}{2} R_{ij}\partial_j \log(p) + \frac{||x-\mu||^2}{2} R_{ij}\partial_l\partial_j \log(p) + R_{ij}\partial_l(x_j-\mu_j) \\ &= \partial_l f_i^d + R_{ij}(x_l-\mu_l)\partial_j \log(p) + \frac{||x-\mu||^2}{2} R_{ij}\partial_l\partial_j \log(p) + R_{ij}\delta_{jl} \\ &= \partial_l f_i^d + R_{ij}(x_l-\mu_l)\partial_j \log(p) + \frac{||x-\mu||^2}{2} R_{ij}\partial_l\partial_j \log(p) + R_{il} \end{aligned}$$

Let us suppose that we evaluate the Jacobian at our control location μ in order to evaluate stiffness. We find that

$$A(\lambda) = A_d(\lambda) + R(\lambda) \tag{42}$$

where

$$A(\lambda) = \nabla f^s|_{x=\mu} \tag{43}$$

thus, we see that we can use R to control stiffness at the target location μ.

5 Possible modifications of previous solutions

Author: Pietro Zanin, pietrozanin2027@u.northwestern.edu, Northwestern University.

5.1 Summary

In one of the most important papers of Fred Daum in particle flows, the following SDE is solved for f and Q:

$$\frac{\partial \log(h)}{\partial x} + \frac{\partial \nabla\cdot f}{\partial x} = -f^T\frac{\partial^2 \log(p)}{\partial x^2} - \frac{\partial \log(p)}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}\left(\frac{\nabla\cdot(Q\frac{\partial p}{\partial x})}{p}\right). \tag{44}$$

We exploited the fact that the PDE is highly undetermined to create new solutions from the known one.

Many modifications are possible, leading to different results that could decrease the stiffness. Numerical simulations are necessary to verify the usefulness of the new solutions, we believe that this is a worthwhile effort to take in the future.

Some modifications require simpler equations to be solved, we did not manage to solve all of them. More work in this part could be fruitful.

5.2 Introduction

The aim here is to start from the solution for the flow derived on [12], and then modify it in such a way that stiffness is attenuated but the equations are still solved.

The intuition is that, without any ansatz or simplifying assumptions, solving the necessary PDE is very hard. Using the already known solution as a basis makes this task easier, while avoiding too many assumptions such the ones used in [3].

To avoid confusions, we stick to the notation of the paper, so

$$\frac{\partial f}{\partial x} = \text{gradient of f}, \qquad \frac{\partial^2 f}{\partial x^2} = \text{hessian of f}, \qquad \nabla \cdot f = \text{divergence of f}. \tag{45}$$

Also, we do not distinguish explicitly vectors or matrices from scalars.
The PDE to be solved is

$$\frac{\partial \log(h)}{\partial x} + \frac{\partial \nabla \cdot f}{\partial x} = -f^T \frac{\partial^2 \log(p)}{\partial x^2} - \frac{\partial \log(p)}{\partial x}\frac{\partial f}{\partial x} + \frac{1}{2}\frac{\partial}{\partial x}(\frac{\nabla \cdot (Q\frac{\partial p}{\partial x})}{p}). \tag{46}$$

In the paper, f is obtained by matching the first term of the LHS and the first term of the RHS:

$$f = -(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}(\frac{\partial \log(h)}{\partial x})^T, \tag{47}$$

and Q is obtained by matching the other 3 terms:

$$Q = -(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}\frac{\partial^2 \log(h)}{\partial x^2}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}. \tag{48}$$

5.3 Part 1: modifications independent of x

This idea comes from the fact that the PDE to be solved is obtained originally by taking the gradient in relation to x of another PDE. This suggests that there is a certain freedom on introducing terms dependent on λ.

5.3.1 First try: $\frac{\partial^2 \log(p)}{\partial x^2} \to \frac{\partial^2 \log(p)}{\partial x^2} + A(\lambda)$

This is an attempt to avoid getting a very high condition number for $\frac{\partial^2 \log(p)}{\partial x^2}$.

$$\frac{\partial^2 \log(p)}{\partial x^2} \to \frac{\partial^2 \log(p)}{\partial x^2} + A(\lambda), \qquad \text{rank}(A) = 1. \tag{49}$$

We will use the formula

$$(A+B)^{-1} = A^{-1} - \frac{1}{1+g}A^{-1}BA^{-1}, \qquad \text{if rank}(B) = 1. \tag{50}$$

With these modifications, we get that

$$\begin{aligned} Q \to &-((\frac{\partial^2 \log(p)}{\partial x^2})^{-1} - \frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}A(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}) \\ &\cdot\frac{\partial^2 \log(h)}{\partial x^2}((\frac{\partial^2 \log(p)}{\partial x^2})^{-1} - \frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}A(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}) \\ &= Q + \frac{1}{(1+g)^2}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}AQA(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} - \frac{1}{1+g}((\frac{\partial^2 \log(p)}{\partial x^2})^{-1}AQ + QA(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}), \end{aligned} \tag{51}$$

and f is modified such that

$$\begin{aligned} f &\to -((\frac{\partial^2 \log(p)}{\partial x^2})^{-1} - \frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}A(\frac{\partial^2 \log(p)}{\partial x^2})^{-1})(\frac{\partial^2 \log(h)}{\partial x^2})^T \\ &= (1 - \frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1}A)f. \end{aligned} \tag{52}$$

Then, the PDE becomes

$$\frac{\partial \log(h)}{\partial x} + \frac{\partial \nabla \cdot f}{\partial x} - \frac{\partial \nabla \cdot (\frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A f)}{\partial x} = -f^T (1 - \frac{1}{1+g} A^T (\frac{\partial^2 \log(p)}{\partial x^2})^{-1}) \frac{\partial^2 \log(p)}{\partial x^2}$$
$$- \frac{\partial \log(p)}{\partial x} \frac{\partial}{\partial x}((1 - \frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A) f) + \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (Q' \frac{\partial p}{\partial x})}{p}), \tag{53}$$

where Q' represents the new value of Q.
We can use the original equation to cancel some terms:

$$- \frac{\partial \nabla \cdot (\frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A f)}{\partial x} = \frac{f^T A^T}{1+g} + \frac{\partial \log(p)}{\partial x} \frac{\partial}{\partial x}(\frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A f) \tag{54}$$
$$+ \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (Q' \frac{\partial p}{\partial x})}{p}) - \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (Q \frac{\partial p}{\partial x})}{p}).$$

It is not clear how to simply the last 2 terms.
Alternatively, we can stick to only modifying f, so the equation for A is

$$- \frac{\partial \nabla \cdot (\frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A f)}{\partial x} = \frac{f^T A^T}{1+g} + \frac{\partial \log(p)}{\partial x} \frac{\partial}{\partial x}(\frac{1}{1+g}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} A f). \tag{55}$$

It is not clear if this equation can be solved.

5.3.2 Second try: $f \to f + \phi(\lambda), \qquad Q \to Q + A(\lambda)$

Here, we introduce a vector $\phi(\lambda)$ and a matrix $A(\lambda)$ to the original equations:

$$f \to f + \phi(\lambda), \qquad Q \to Q + A(\lambda). \tag{56}$$

The original PDE becomes

$$\frac{\partial \log(h)}{\partial x} + \frac{\partial \nabla \cdot f}{\partial x} = -f^T \frac{\partial^2 \log(p)}{\partial x^2} - \phi^T \frac{\partial^2 \log(p)}{\partial x^2} \tag{57}$$
$$- \frac{\partial \log(p)}{\partial x} \frac{\partial f}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (Q \frac{\partial p}{\partial x})}{p}) + \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (A \frac{\partial p}{\partial x})}{p}),$$
$$0 = -\phi^T \frac{\partial^2 \log(p)}{\partial x^2} + \frac{1}{2} \frac{\partial}{\partial x}(\frac{\nabla \cdot (A \frac{\partial p}{\partial x})}{p}), \qquad \phi^T = \frac{1}{2}(\frac{\partial^2 \log(p)}{\partial x^2})^{-1} \frac{\partial}{\partial x}(\frac{\nabla \cdot (A \frac{\partial p}{\partial x})}{p}).$$

It seems that there is not a way to satisfy this equation, as the LHS does not depend on x, but the RHS will depend on it, because the first term depends on the second derivative of p in relation to x, and the second depends on the third derivatives of p in relation to x.

Incidentally, this suggests a different approach: $\phi = 0$ and $A = A(x, \lambda)$. With this choice,

$$0 = \frac{\partial}{\partial x}(\frac{\nabla \cdot (A \frac{\partial p}{\partial x})}{p}), \tag{58}$$

where we just need to pick a matrix $A(x, \lambda)$ such that the product $A \frac{\partial p}{\partial x}$ does not depend on x, this is alwaysd possible.

It is not clear if this is actually useful to lower the stiffness.

5.3.3 Third try: $f \to A(\lambda) f, \quad Q \to B(\lambda) Q$

Now we try to multiply the original solutions by matrices $A(\lambda)$ and $B(\lambda)$:

$$f \to A(\lambda) f, \qquad Q \to B(\lambda) Q. \tag{59}$$

The original PDE becomes

$$\begin{aligned}
\frac{\partial \log(h)}{\partial x} + \frac{\partial \nabla \cdot (Af)}{\partial x} &= -f^T A^T \frac{\partial^2 \log(p)}{\partial x^2} - \frac{\partial \log(p)}{\partial x} \frac{\partial A f}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x} \left(\frac{\nabla \cdot (BQ \frac{\partial p}{\partial x})}{p} \right), \\
\frac{\partial \nabla \cdot ((A-1)f)}{\partial x} &= -f^T (A^T - 1) \frac{\partial^2 \log(p)}{\partial x^2} - (A-1) \frac{\partial \log(p)}{\partial x} \frac{\partial f}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x} \left(\frac{\nabla \cdot ((B-1)Q \frac{\partial p}{\partial x})}{p} \right).
\end{aligned} \tag{60}$$

It is not clear how to proceed from here. Again, we could consider the case where $A = 1$ and $B = B(x, \lambda)$, so we would need to satisfy

$$0 = \frac{\partial}{\partial x} \left(\frac{\nabla \cdot ((B-1)Q \frac{\partial p}{\partial x})}{p} \right), \tag{61}$$

which can always be solved, just choose B such that $(B-1)Q\frac{\partial p}{\partial x}$ is not 0 but does not depend on x. Again, it is not clear if this is actually useful to lower the stiffness.

Similarly, we can multiply f and Q by scalars $a(\lambda)$ and $b(\lambda)$ respectively, but we just get very similar equations.

5.3.4 Mixing different strategies

It is worth mentioning the fact that mixing different strategies could work. For example, we could try to modify f by modifying $\frac{\partial^2 \log(p)}{\partial x^2}$, and then modifying Q by multiplying it by a properly chosen matrix $B(x, \lambda)$.

Testing all the possible choices would take time, and again, it is not clear if it effectively reduces stiffness.

5.4 Part 2: starting with a different f

Here, we try to match different terms of the equation, in order to obtain different solutions.

5.4.1 First try

$$\frac{\partial \log(h)}{\partial x} = -\frac{\partial \log(p)}{\partial x} \frac{\partial f}{\partial x}. \tag{62}$$

With this choice, it is not possible to isolate f.

5.4.2 Second try

Now, we start by first finding a solution Q, and then finding a solution f:

$$\frac{\partial (\log(h))}{\partial x} = \frac{1}{2} \frac{\partial}{\partial x} \left(\frac{\nabla \cdot (Q \frac{\partial p}{\partial x})}{p} \right), \qquad \log(h) = \frac{1}{2} \frac{\nabla \cdot (Q \frac{\partial p}{\partial x})}{p} + c(\lambda), \qquad 2p(\log(h) - c) = \nabla \cdot (Q \frac{\partial p}{\partial x}). \tag{63}$$

The only sensible way to proceed is to introduce a potential $\phi(x, \lambda)$ such that $Q\frac{\partial p}{\partial x} = \frac{\partial \phi}{\partial x}$, and solving the Poisson equation

$$2p \log(h) = \nabla^2 \phi, \tag{64}$$

where we took $c(\lambda) = 0$ for simplicity.

Using Green's function method, one of the solutions for ϕ is

$$\phi(x, \lambda) = \int \frac{2p \log(h)(x', \lambda)}{\mid x - x' \mid^{d-2}} dx', \tag{65}$$

where the integration is over all the volume, and d is the number of dimensions of x.

Therefore, we would have to first compute $\phi(x, \lambda)$, then choose a solution such that $Q\frac{\partial p}{\partial x} = \frac{\partial \phi}{\partial x}$ is satisfied, and finally solve the following equation for f:

$$\frac{\partial \nabla \cdot f}{\partial x} = -f^T \frac{\partial^2 \log(p)}{\partial x^2} - \frac{\partial \log(p)}{\partial x} \frac{\partial f}{\partial x}. \tag{66}$$

We did not figure out a solution for this equation. Even if there is one, it is not clear if the time necessary to solve the Poisson equation and to find Q is not prohibitive.

These calculations can be understood as a generalization of the ones done in [9].

5.4.3 Third try

Now, we impose

$$\frac{\partial \log(h)}{\partial x} = -\frac{\partial \nabla \cdot f}{\partial x}, \qquad \log(h) + c(\lambda) = \nabla \cdot f. \tag{67}$$

Again, the only sensible way to proceed is to introduce a potential $\phi(x, \lambda)$ such that

$$\nabla^2 \phi = -\log(h), \qquad \phi(x, \lambda) = \int \frac{\log(h(x', \lambda))}{\mid x - x' \mid} dx'. \tag{68}$$

Then, we obtain the following equation for Q:

$$2\left(-\int \frac{\log(h(x', \lambda))}{\mid x - x' \mid^{d-1}} dx' \frac{\partial^2 \log(p)}{\partial x^2} - \log(h) \frac{\partial \log(p)}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{\nabla \cdot (Q\frac{\partial p}{\partial x})}{2p}\right), \tag{69}$$

and it is not clear how we can solve it.

5.5 Variational approach

We can start by minimizing the Wasserstein distance between the new f and the old f, similarly for Q. By exploring the properties of f and Q, we can impose appropriate conditions on f and Q.

$$\text{minimize}_{x,\, \lambda} \int_{\Omega} \|f_{new} - f_{old}\| \, dx \, d\lambda \tag{70}$$

But how to proceed from here remains unclear. Some related work that can be explored for future research includes [10], in which the exact unique minimum norm solution for f is given in terms of the generalized inverse of the linear differential operator.

In a different direction, a general purpose Bayesian inference algorithm, Stein variational gradient descent [14], was proposed to iteratively transports a set of particles to match the target distribution, by applying a form of functional gradient descent that minimizes the KL divergence. We can make comparisons between the condition number of the Jacobian matrix of the differential equation for a particle flow [3] and the counterpart in Stein variational gradient descent.

6 Regularization of Particle Dynamics

Contributors: Erik Berglund, Sameer Pokhrel, Chris Raymond, Pak-Wing Fok.
Summary here: Our main contribution here is to remove the stiffness in the particle advection by adding inertia or "mass" to each particle. Instead of

$$\frac{d\mathbf{x}}{d\lambda} = \mathbf{f}, \tag{71}$$

(Brownian dynamics), we include a second equation for the velocity ("Langevin Dynamics"), and quickly and continuously remove the inertia as λ increases:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = e^{-a\lambda}\begin{pmatrix}\mathbf{v}\\ (\mathbf{f}-\mathbf{v})/m\end{pmatrix} + (1-e^{-a\lambda})\begin{pmatrix}\mathbf{f}\\ d\mathbf{f}/d\lambda\end{pmatrix} \tag{72}$$

We find that this new scheme reduces the stiffness in the ODE or SDE for one simple test case in [1], while still retaining the desirable properties of the original scheme. Specifically, when the prior is broad and the likelihood is sharply peaked (corresponding to accurate measurements), the original particle method blows up due to stiffness, while the new scheme (which we call HOMEBREW) is able to advect particles to their "proper" positions in phase space. Any scheme in the form (71) with any flow field $\mathbf{f}$ can be regularized in this way.

Full explanation: This calculation is motivated by the results in [11] and uses the same notation. Recall that the governing ODE for noiseless particle dynamics is

$$\frac{d\mathbf{x}}{d\lambda} = \mathbf{f}, \tag{73}$$

where the flow field $\mathbf{f}$ satisfies the PDE

$$\nabla\cdot(p\mathbf{f}) = -p\log h + p\frac{d}{d\lambda}\log K(\lambda) \equiv \eta(\mathbf{x},\lambda) \tag{74}$$

We anticipate issues with the dynamics associated with this system. Loosely speaking, one can "solve" eq. (74) to find

$$\begin{aligned}\mathbf{f} &= \frac{1}{p}\int \eta(\mathbf{x},\lambda)d\mathbf{x},\\ \Rightarrow \frac{d\mathbf{x}}{d\lambda} &= O\left(\frac{1}{p}\right)\end{aligned}$$

which shows that the $p\to 0$ limit is singular: particle positions evolve very quickly, over a time scale of $O(p)$, giving rise to stiffness. We have investigated two ways to relieve this stiffness: a change of spatial variable, and inertial regularization.

6.1 Inertial Regularization (HOMEBREW)

For concreteness, suppose we follow the "Coulomb" method discussed in [11], in which case we define $p\mathbf{f} = \nabla V$ so that V satisfies the Poisson equation

$$\nabla^2 V = \eta \Longrightarrow V = -c\int \frac{\eta(\mathbf{y},\lambda)}{||\mathbf{x}-\mathbf{y}||^{d-2}}d\mathbf{y} \tag{75}$$

where $c = \frac{\Gamma(\frac{d}{2}-1)}{4\pi^{d/2}}$ is a constant and d is the number of spatial dimensions. Note that the inertial regularization could potentially be applied to *any* method that generates a flow field $\mathbf{f}$.

The ODE corresponding to the Coulombic particle method is

$$p\frac{d\mathbf{x}}{d\lambda} = \nabla V. \tag{76}$$

The physical interpretation of the above equation is that a particle moves up the gradient of V, which acts as an external force to the particle. The $p(d\mathbf{x}/d\lambda)$ term is analogous to an air resistance, or a friction term if λ plays the role of time. When an external force is applied to a particle with vanishing ($p \to 0$) friction, the resulting velocity is enormous. Given that the particles start stationary, the ODE (76) is very stiff.

Suppose we think of the particles in the "particle filter" as actual point masses. Then, the problem with (76) is that the mass of the particle has been completely neglected. The key idea behind inertial regularization is to imbue the particles with mass so that when an external force is applied, their velocities increase smoothly:

$$m\frac{d^2\mathbf{x}}{d\lambda^2} + p\frac{d\mathbf{x}}{d\lambda} = \nabla V. \tag{77}$$

This can be solved as a system of coupled ODEs:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\\ (\nabla V - p\mathbf{v})/m\end{pmatrix}. \tag{78}$$

Initial conditions for the velocity must also be specified:

$$\begin{aligned} x_k(\lambda = 0) &= x_{k,0}, \qquad k = 1, \ldots, d & (79)\\ v_k(\lambda = 0) &= v_{k,0}, \qquad k = 1, \ldots, d, & (80)\end{aligned}$$

and $v_{k,0} \sim N(0, \sigma_v^2)$, so the components of the initial velocity are randomly distributed while the initial positions are given.

In the general case, we add inertia to

$$\frac{d\mathbf{x}}{d\lambda} = \mathbf{f} \tag{81}$$

to obtain the new ODE

$$m\frac{d^2\mathbf{x}}{d\lambda^2} + \frac{d\mathbf{x}}{d\lambda} = \mathbf{f}. \tag{82}$$

Introducing $\mathbf{v} = \frac{d\mathbf{x}}{d\lambda}$,

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\\ (\mathbf{f} - \mathbf{v})/m\end{pmatrix}. \tag{83}$$

This is the fully inertial regime. We found that using this ODE for advecting particles resulted in oscillations and instability. To remedy this, we tried to interpolate between (81) and (83), essentially adopting (83) when $\lambda = 0$ and (81) when $\lambda = 1$:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = (1-\lambda)\begin{pmatrix}\mathbf{v}\\ (\mathbf{f} - \mathbf{v})/m\end{pmatrix} + \lambda\begin{pmatrix}\mathbf{f}\\ d\mathbf{f}/d\lambda\end{pmatrix}. \tag{84}$$

When $\lambda = 0$, it is clear that this equation reduces to (83). When $\lambda = 1$, we have

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{f}\\ d\mathbf{f}/d\lambda\end{pmatrix}, \tag{85}$$

and the second ODE is redundant. However, this interpolation scheme was also found to give rise to inaccurate posterior distributions. We adjusted the interpolation so that inertia in the scheme would decay exponentially as λ departed from 0 with a decay constant a that is controllable by the user:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = e^{-a\lambda}\begin{pmatrix}\mathbf{v}\\ (\mathbf{f} - \mathbf{v})/m\end{pmatrix} + (1 - e^{-a\lambda})\begin{pmatrix}\mathbf{f}\\ d\mathbf{f}/d\lambda\end{pmatrix} \equiv \mathbf{F}. \tag{86}$$

The constant a is a tuning parameter that controls how quickly inertia is removed from the system. We found that $a \gg 1$ seems to give good results. We call our new inertial regularization method, along with this particular form of interpolation "HamiltOnian MontE carlo-ization of Bayesian paRticlE floW" (HOMEBREW). In the stochastic version, the external forcing $\mathbf{f}$ is replaced by $\mathbf{f} + d\mathbf{w}/d\lambda$ where $\langle d\mathbf{w}, d\mathbf{w}\rangle = Qd\lambda$ and Q is a symmetric positive definite matrix.

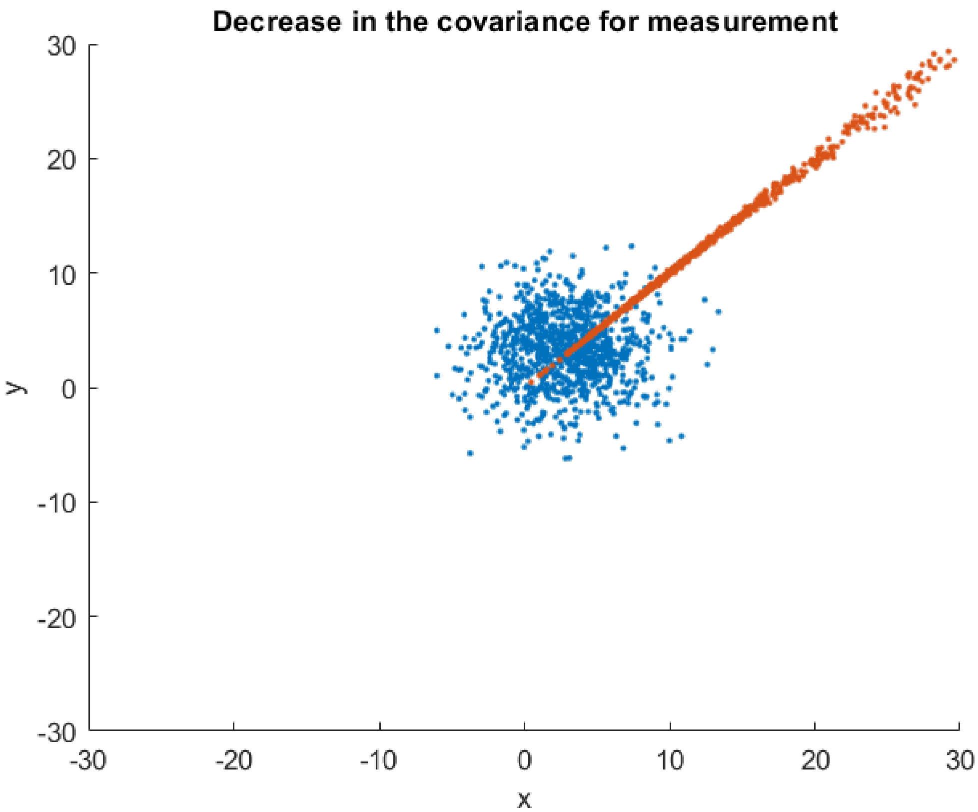

Figure 2: The particles after a polar angle-only measurement, given a Gaussian prior using the Gromov flow and explicit Euler method. Prior particles are blue circles and posterior particles are red circles. The particles are biased far away from the true value of (3,3). The number of particles is 1000 and the covariance for measurement noise is taken as 10^{-2}.

6.2 Results and Condition Numbers

In this section, we test HOMEBREW on the example in [1], section VII. This problem involves inferring the position of a point in the Cartesian plane from a noisy measurement of its angle from the x-axis. The prior is a 2D Gaussian. A more accurate measurement results in a narrower likelihood function, resulting in more stiffness. We find that HOMEBREW is able to produce good posterior distributions even when the prior is flat and the likelihood is sharply peaked.

In Figure 2 which is not stiff, and uses $\sigma = 10^{-2}$ as the measurement noise, the prior is realized with the blue points and the posterior with the red points. However, when the problem is stiffened by reducing σ to 10^{-4} in Fig. 3, almost all of the particles shoot off to infinity. The same value of σ but with the flow equations integrated using HOMEBREW results in Fig. 4: the results are much more reasonable and we obtain a posterior realized with particles lying on a ray with angle given by the measurement.

We can define condition numbers for the original ODE (81) and HOMEBREW (86):

$$\kappa_0 = \left| \frac{\lambda_{\max}(\nabla \mathbf{f})}{\lambda_{\min}(\nabla \mathbf{f})} \right|, \tag{87}$$

$$\kappa_{HB} = \left| \frac{\lambda_{\max}(\nabla \mathbf{F})}{\lambda_{\min}(\nabla \mathbf{F})} \right|, \tag{88}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues. These condition numbers quantify the stiffness of each system. As future work, we plan to track the condition number for a single particle as it

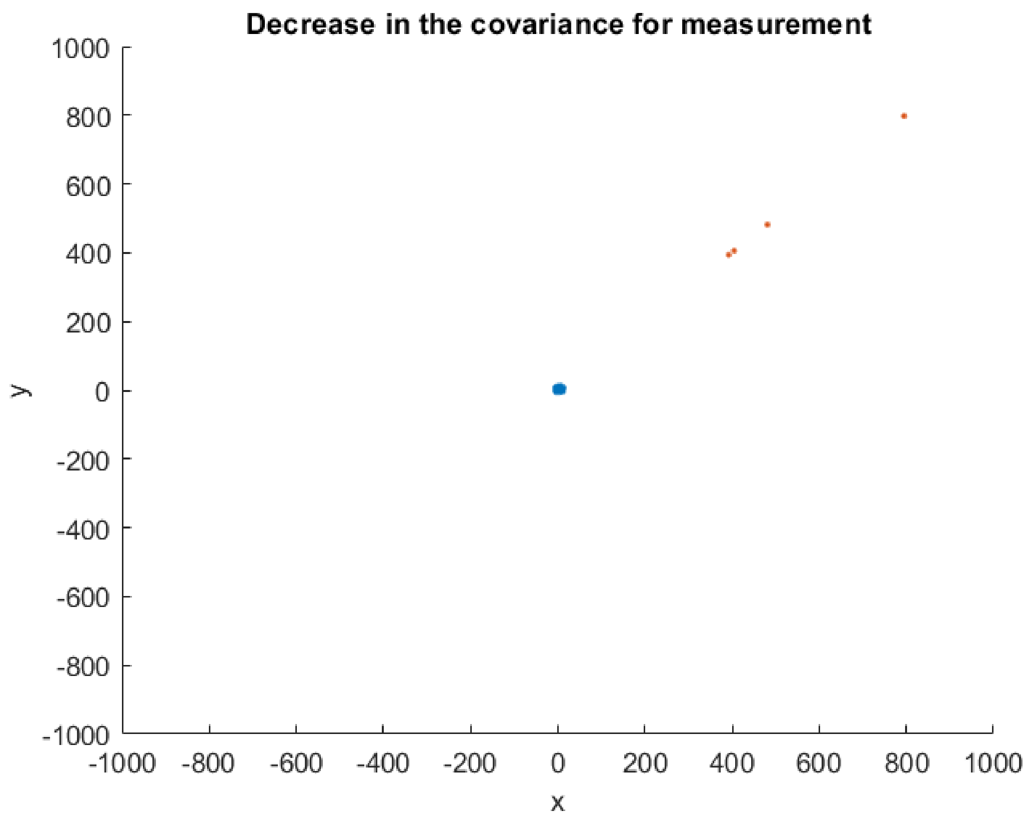

Figure 3: The measurement noise is taken as 10^{-4} and all other parameters are same as figure 2. The result shows that a decrease in covariance of measurement makes the problem stiff which results in an undesirable solution.

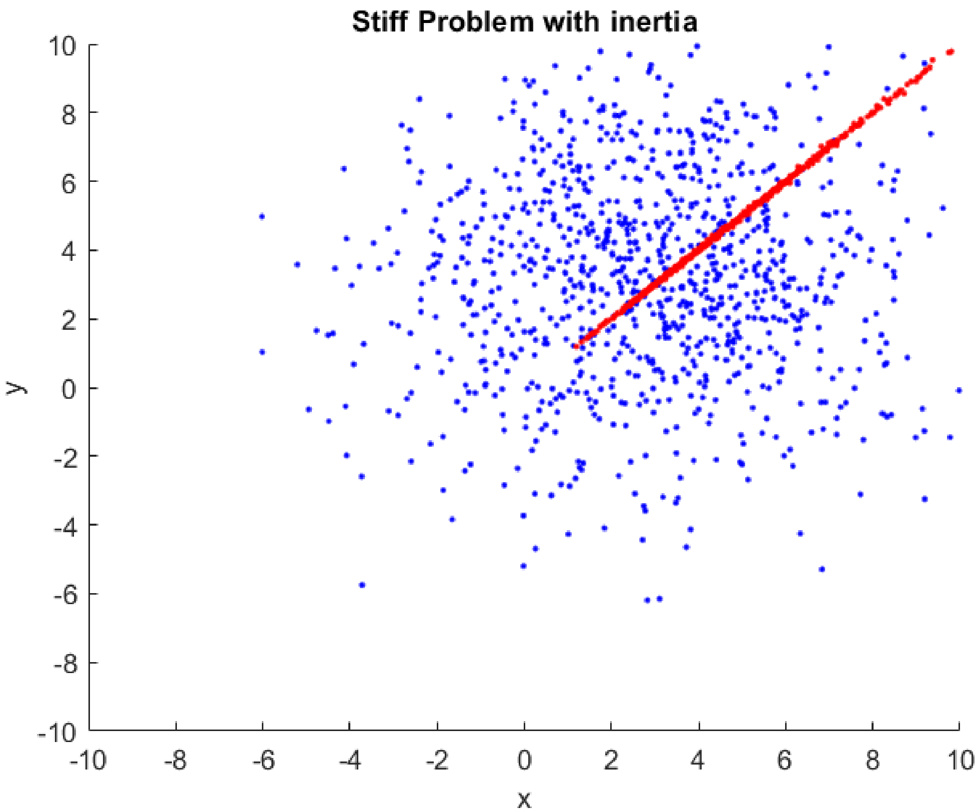

Figure 4: The measurement noise is taken as 10^{-4} and all other parameters are same as figure 3 and 2. However, the proposed inertial regularization method is used for the flow. The biased is reduced and the result is much better when compared to figure 3.

evolves in state space.

6.3 Changing spatial variables

Stiffness arises because when $p \to 0$, eq. (74) is singular. We make a change of variable in an attempt to make the argument of the divergence in (74) identical to the right hand side of (73). Define

$$\begin{aligned} \mathbf{J} &= p\mathbf{f}, && (89) \\ d\mathbf{y} &= p(\mathbf{x}, \lambda)d\mathbf{x} && (90) \end{aligned}$$

This gives rise to the system

$$\begin{aligned} \mathrm{div}\mathbf{J} &= -\frac{\partial p}{\partial \lambda}, && (91) \\ \frac{d\mathbf{y}}{d\lambda} &= \mathbf{J}. && (92) \end{aligned}$$

Superficially, this seems to have removed the singularity since $\mathbf{J} = O(1)$ rather than $O(p^{-1})$, but stiffness remains because of (90): we need to switch between $\mathbf{y}$ and $\mathbf{x}$ variables by solving a stiff ODE.

7 Stochastic mean shift dynamics

Author of this section: Praneeth Vepakomma, vepakom@mit.edu.

In the context of the continuity equation we propose to use a specific choice for the deterministic flow. Firstly, the following are kernel density estimators of pdf's, their gradients and hessians.

$$\widehat{f_{h,k}}(x) = \hat{f}(x) = c_N \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{93}$$

The gradient of pdf estimator is

$$\nabla \hat{f}(x) = c_N \sum_{i=1}^{n} \left(\frac{x_i - x}{h^2}\right) \exp\left(-\frac{1}{2}\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{94}$$

The hessian of pdf estimator is

$$H : \mathbb{R}^d \to \mathbb{R}^{d \times d} := H(x) = \frac{c_N}{h^2} \sum_{i=1}^{n} \left(-I + \frac{(x - x_i)(x - x_i)^T}{h^2}\right) \exp\left(-\frac{1}{2}\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{95}$$

Given the continuity equation below,

$$\partial x = f(x, \lambda)\partial\lambda + \sqrt{Q}\partial\omega \tag{96}$$

we use the mean-shift iterate used for clustering as the deterministic part of the flow as given below where the first term coincidentally happens to be a kernel-density estimator of log density.

$$f_d = \nabla \widehat{\log p(x)}_{\mathrm{kde}} + x = \frac{\sum_{i=1}^{n} \left(\frac{x_i - x}{b^2}\right) \exp\left\{-\frac{1}{2}\left\|\frac{x_i - x}{b}\right\|^2\right\}}{\sum_{i=1}^{n} \exp\left[\frac{-1}{2}\left\|\frac{x_i - x}{b}\right\|^2\right]} + x \tag{97}$$

In order for this flow to fit the Fred Daum et.al framework, the following condition needs to be satisfied.

$$\frac{dp}{\partial\lambda} = -\mathrm{div}\left(pf_d\right) \text{ where } p(x, 0) = g(x) \tag{98}$$

This condition can be simplified using standard property of divergence of product of scalar and vector-valued functions to get the follows.

$$-\left[p\nabla \cdot f_d + f_d \cdot \nabla P\right] \tag{99}$$

But

$$f_d = \nabla_{\text{ kde}} \log p(x) + x$$

and as divergence of gradient is the Laplacian, we get

$$\tag{100}$$

$$\nabla \cdot f_d = \Delta \log_{\text{kde}} (x) + \nabla \cdot x \tag{101}$$

$$\frac{dp}{d\lambda} = \Delta \log_{\text{kde}}\ p(x) + d \tag{102}$$

$$\tag{103}$$

Now, the generic distribution case above can also be "restricted" to the Gaussian case using the proposed f_d above to fit into the Fred Daum et.al framework by using

$$f_s = f_d + \left(\frac{\partial^2 \log p}{\partial x^2}\right)^{-1} K \frac{\partial \log p}{\partial x} \tag{104}$$

where the stability depends on $J(f_s)|_{x=x(\alpha)}$ and

$$K = \frac{1}{2}\left(\nabla_x \nabla_x^T \log p\right) Q(\lambda) + \frac{1}{2}\left(\nabla_x \nabla_x^T \log h\right)\left(\nabla_x \nabla_x^T \log p\right)^{-1} \tag{105}$$

.

7.1 Non-singularity conditions

Now as a secondary exercise, we try to place conditions on natural gradient based flow used in Fred Daum to guarantee non-singularity on the hessian term that needs to be inverted. We first introduce some notation for brevity as follows.

$$A : \mathbb{R}^d \to \mathbb{R}^{d\times d} := A(x) = \sum_{i=1}^{n} (x - x_i)(x - x_i)^T \exp\left(-\frac{1}{2}\left\|\frac{x - x_i}{h}\right\|^2\right) \tag{106}$$

and

$$M : \mathbb{R}^d \to \mathbb{R} := M(x) = \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left\|\frac{x - x_i}{h}\right\|^2\right) > 0 \tag{107}$$

Now the hessian of the pdf can be estimated based on kernel density estimates as

$$\boldsymbol{\nabla}^2 \hat{p}(x) = \frac{C_N}{b^2}\left[-M(x)I_d + \frac{A(x)}{b^2}\right] \tag{108}$$

Similarly, the hessian of the log density can be estimated as

$$\nabla^2 \widehat{\log p(x)} = \frac{C_N}{b^2}\left[\frac{-M(x)I_d + \frac{A(x)+(x)}{b^2}}{M(x)} - \frac{[\nabla \hat{p}(x)]^2}{[M(x)]^2}\right] \tag{109}$$

Now the non-singularity condition requires the product of determinants

$$|I_d - [S_x U_x^T [I + k(x)]^{-1} U_x]||I + dx| > 0$$

where

$$k(x) = \frac{[\nabla \hat{p}(x)]^2}{[M(x)]^2}$$

Note that $I + k(x)$ is diagondly dominant and positive definite if $1 + k(x) \geq (n-1)k(x)$ and therefore the non-singularity is satisfied if

$$d \geq \mathrm{Tr}\left(\frac{A(x)}{b^2 \mu(x)}\right) \text{ and } n \leq \frac{2 + k(x)}{k(x)}$$

8 Analysis of Crouse 2018 stiffness

author: Hunter Akins, `fakins@ucsd.edu`, UCSD

The goal here is to get a better understanding of the source of stiffness in the problem analyzed in "Particle Flow Filters: Biases and Bias Avoidance" (Crouse 2018). First, I show how the stochastic differential equation randomly rotates each particle in the Cartesian coordinate system (Crouse explains this in the paper). Second, I show that a simple numerical scheme will cause the norm of each particle to grow. Third, I suggest a norm-preserving numerical scheme for this specific problem. It is suspected that this is somehow equivalent to solving the problem in polar coordinates. The analysis here suggests that using numerical schemes that conserve an appropriate particle quantity (e.g. "leapfrog" method used for integrating particle paths in Hamiltonian Monte Carlo) may be a way to control stiffness.

8.1 Rotation drift and diffusion

The stochastic ODE analyzed in Crouse randomly rotates partiles in a 2-d Cartesian coordinate system. Therefore, it preserves the norm of the particles $||\mathbf{x}||$ Crouse analyzes the particle flow associated with the stochastic differential equation

$$d\mathbf{x} = f(\mathbf{x}, \lambda)\mathrm{d}\lambda + \mathbf{B}\mathrm{d}\mathbf{w} \ . \tag{110}$$

The drift function f

$$f(\mathbf{x}, \lambda) = \left(\frac{\sigma^2 \nu}{R||\mathbf{x}||^2 + \lambda\sigma^2}\right)\begin{bmatrix} y \\ -x \end{bmatrix} \tag{111}$$

and diffusion matrix

$$\mathbf{Q} = \mathbf{B}\mathbf{B}^T = \left(\frac{R\sigma^4}{(R||\mathbf{x}||^2 + \lambda\sigma^2)^2}\right)\begin{bmatrix} y^2 & -xy \\ -xy & x^2 \end{bmatrix} \tag{112}$$

where z is the angle of a target in a two-d plane, and $\nu = h(\mathbf{x}) - z$ is the innovation. R is the measurement error variance in radians squared. σ^2 is the prior covariance on the target position in a two-d plane. $\mathbf{x} = [x \ \ y]^T$ is the state vector representing the target position in the two-d plane.

We can express f

$$f(x, \lambda) = \alpha(||\mathbf{x}||, \lambda)\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{x} \tag{113}$$

where

$$\alpha(||\mathbf{x}||, \lambda) = -\frac{\sigma^2 \nu}{R||\mathbf{x}||^2 + \lambda\sigma^2} \ . \tag{114}$$

For deterministic flow (ignoring diffusion) we see that the effect of the drift at time λ is to apply an infinitesimal rotation matrix to the particle state $\mathbf{x}$.

$$\mathbf{x}(\lambda + \mathrm{d}\lambda) = \mathbf{x}(\lambda) + f d\lambda = (I + \alpha(\lambda)\mathrm{d}\lambda\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix})\mathbf{x}(\lambda) = (I + \mathrm{d}\theta(\lambda))\mathbf{x}(\lambda) \ . \tag{115}$$

The infinitesimal angle is $\mathrm{d}\theta(\lambda) = \alpha(||\mathbf{x}||, \lambda)$. The true solution to the problem therefore preserves $||\mathbf{x}||$.

The noise covariance has also been chosen to preserve $||\mathbf{x}||$. This is clear from the Cholesky decomposition of Q:

$$\mathbf{B} = \frac{R^{1/2}\sigma^2}{(R||\mathbf{x}||^2 + \lambda\sigma^2)} \begin{bmatrix} y & 0 \\ -x & 0 \end{bmatrix} . \tag{116}$$

We can rewrite the diffusion term in the drift equation in terms of a 1-d Wiener process

$$\mathbf{B}\mathrm{d}\mathbf{w} = -\frac{R^{1/2}\sigma^2}{(R||\mathbf{x}||^2 + \lambda\sigma^2)} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{x}\mathrm{d}w = \beta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{x}\mathrm{d}w , \tag{117}$$

where I have introduced $\beta = -\frac{R^{1/2}\sigma^2}{(R||\mathbf{x}||^2+\lambda\sigma^2)}$

Therefore the effect of the stochastic differential equation at time λ is to apply a random infinitesimal rotation matrix to the particle state $\mathbf{x}$:

$$\mathbf{x}(\lambda + \mathrm{d}\lambda) = \mathbf{x}(\lambda) + fd\lambda + \mathbf{B}\mathrm{d}\mathbf{w} = \left[I + (\alpha(\lambda)\mathrm{d}\lambda + \beta \mathrm{d}w) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right] \mathbf{x}(\lambda) = (I + \mathrm{d}\theta(\lambda))\mathbf{x}(\lambda) . \tag{118}$$

The infinitesimal angle $\mathrm{d}\theta$ consists of a deterministic component $\mathrm{d}\theta_{det}(\lambda) = \alpha(||\mathbf{x}||, \lambda)\mathrm{d}\lambda$ and a random component $\mathrm{d}\theta_{ran}(\lambda) = \beta\mathrm{d}w$. The true solution to the problem therefore preserves $||\mathbf{x}||$. If, in the course of numerical integration, the norm $||\mathbf{x}||$ changes, then the amount of rotation $\mathrm{d}\theta(\lambda)$ will change and the particle after integrating to $\lambda = 1$ will not converge to the correct angle.

8.2 Effect of discretization

Let's now examine the effect of discretizing the ODE. Assuming a simple Euler-Maruyama integration with stepsize h, we have

$$x_{k+1} = x_k + (\alpha h + \beta w\sqrt{h})y_k \tag{119}$$

and

$$y_{k+1} = y_k - (\alpha h + \beta w\sqrt{h})x_k , \tag{120}$$

for $w \sim \mathcal{N}(0, 1)$. In contrast to the true solution, in the numerical solution the norm of $\mathbf{x}$ will not be conserved:

$$||\mathbf{x}_{k+1}|| = \sqrt{x_{k+1}^2 + y_{k+1}^2} = \sqrt{1 + (\alpha h + \beta w\sqrt{h})^2}||\mathbf{x}_k|| . \tag{121}$$

There are two things to note here. The first is that $h\alpha$ must be small to keep the norm of $\mathbf{x}$ roughly conserved. The second is that the norm of $\mathbf{x}$ will always grow in this numerical scheme (since $h(\sqrt{h}\alpha + \beta)^2 > 0$. If $||\mathbf{x}||$ increases, it will decrease differential angle $\mathrm{d}\angle$ in the stochastic ODE. These observations explain why the stiffness manifests as a) particles ending up biased to greater range and b) particles stopping before they reach the measured angle (where $\nu = 0$).

At $\lambda = 0$, $\alpha = -\frac{\sigma^2\nu}{R||\mathbf{x}||^2}$. This number can be quite large for a precise measurement $R \ll 1$, or if $||\mathbf{x}|| \ll 1$.

If $h\alpha$ is not small (exactly how small? good question), then the norm of $\mathbf{x}$ will change. Since the dynamics of this system are designed to preserve this norm, diverging from the norm will change the final angle that the particle ends up at.

8.3 Norm-preserving numerical scheme

Let $\gamma \equiv \sqrt{1 + (\alpha h + \beta w\sqrt{h})^2}$ (the factor by which the particle state norm increases at each numerical step). Consider

$$x_{k+1} = \gamma^{-1}\Big(x_k + (\alpha h + \beta w\sqrt{h})y_k\Big) \tag{122}$$

and

$$y_{k+1} = \gamma^{-1}\Big(y_k - (\alpha h + \beta w\sqrt{h})x_k\Big) . \tag{123}$$

Since the ratio y_{k+1}/x_{k+1} is unchanged by this rescaling, this scheme will cause the same rotation as the original scheme. However, the rescaling will preserve the norm. This is probably somehow equivalent to solving the problem in polar coordinates. This may also have some link to leapfrog integration, which is necessary in Hamiltonian Monte Carlo to preserve energy of the particles. In the problem presented in Crouse (2018), the numerical error increases the norm of the particles. This changes the angular momentum of the particle at each state. Therefore, perhaps considering numerical schemes that conserve quantities will lead to better behavior in stiff particle flow.

Forward euler using the proposed norm-preserving scheme avoids stiffness.

8.4 Summary

Analysis of the problem presented in Crouse demonstrates that a simple Euler numiercal scheme will tend to grow the norm of the particle state vector, in contrast with the true flow which preserves particle norm.

9 A possible homotopy

Author: Maxim J. Goldberg, mgoldber@ramapo.edu, Ramapo College of NJ

Consider

$$p(x, \lambda) = \frac{g^{1-\lambda}(x) h^{\lambda}(x)}{K(\lambda)}, \tag{124}$$

where $K(\lambda)$ is the normalization constant. Note that $p(x, 0) = g(x)$ and $p(x, 1) = h(x)$, which are both known. We have that:

$$p(x, 1/2) = \frac{g^{1/2}(x) h^{1/2}(x)}{K(1/2)}. \tag{125}$$

Squaring $p(x, 1/2)$ gives us the goal density, except for the wrong constant. The proposed homotopy will give rise to a PDE very similar to the one considered by F. Daum (on the left hand side, $\log h$ will be replaced by $\log h - \log g$).

The proposed homotopy has the advantage to be "self-checking", in that if $p(x, 1)$ that is obtained does indeed match $h(x)$, then $p(x, 1/2)$ has a good chance to have been correctly evaluated.

10 Homotopy flows and relative entropy

Contributed by Maxim Zyskin

10.1 Homotopy of probabilities

We consider a homotopy between prior $g(x)$ and posterior $g(x)h(x)$ probability densities given by

$$\begin{aligned} &\log p(x, \lambda) = (1 - \lambda) \log g(x) + \lambda \log (g(x)h(x)) + c(\lambda) \\ &= \log g(x) + \lambda \log h(x) + c(\lambda), 0 \le \lambda \le 1. \end{aligned} \tag{126}$$

where $p(x, \lambda)$, $x \in \mathbb{R}^n$, is a probability density in $\mathbb{R}^n$ for all homotopy parameters $\lambda, 0 \le \lambda \le 1$, and $c(\lambda) \equiv -\log Z(\lambda)$ an appropriate normalization to ensure that. Such homotopies were considered by [4] in the context of computationally efficient multi-dimensional filters. We assume that all the densities are smooth. Differentiating with respect to λ,

$$\frac{\partial p(x, \lambda)}{\partial \lambda} = p(x, \lambda) \log h(x) + p(x, \lambda) c^{(1)}(\lambda). \tag{127}$$

Since $p(x, \lambda)$ is a probability density, $\int_{x \in \Omega} p(x, \lambda) dx = 1$. Integrating (127) over $x \in \Omega$ (and interchanging differentiation and integration),

$$c^{(1)}(\lambda) = -\langle \log h \rangle_{p_\lambda} := \int_{x \in \Omega} p(x, \lambda) \log h(x) dx, \tag{128}$$

thus

$$\frac{\partial p(x,\lambda)}{\partial\lambda} = p(x,\lambda)\left(\log h(x) - \langle\log h\rangle_{p_\lambda}\right), \tag{129}$$

and similarly

$$\begin{aligned}
&\frac{\partial^2 p(x,\lambda)}{\partial\lambda^2} = p(x,\lambda)\left(\log h(x) - \langle\log h\rangle_{p_\lambda}\right)^2 + p(x,\lambda)c^{(2)}(\lambda),\\
&c^{(2)}(\lambda) = -\langle\left(\log h - \langle\log h\rangle_{p_\lambda}\right)^2\rangle_{p_\lambda},\\
&\ldots\\
&\frac{\partial^n p(x,\lambda)}{\partial\lambda^n} = p(x,\lambda)\Lambda_n\left[\log h(x), c^{(i<n)}(\lambda)\right] + p(x,\lambda)c^{(n)}(\lambda),\\
&c^{(n)}(\lambda) = -\langle\Lambda_n\rangle_{p_\lambda}
\end{aligned} \tag{130}$$

10.2 Relative entropy flow

Note that

$$\begin{aligned}
\langle\log h\rangle_{p_1} &= \int_{x\in\Omega} p_1(x)\log\frac{p_1(x)}{p_0(x)}dx = D_{KL}\left(p_1|p_0\right),\\
\langle\log h\rangle_{p_0} &= -\int_{x\in\Omega} p_0(x)\log\frac{p_0(x)}{p_1(x)}dx = -D_{KL}\left(p_0|p_1\right),
\end{aligned} \tag{131}$$

where $D_{KL}\left(P|Q\right)$ is the Kullback-Leibler divergence (relative entropy), to do with how different two probability distributions are. It follows from the Gibbs inequality that $D_{KL}\left(P|Q\right) \geq 0$, and $D_{KL}\left(P|Q\right) = 0$ if $P = Q$ as measures, thus

$$\langle\log h\rangle_{p_0} \leq 0, \langle\log h\rangle_{p_1} \geq 0. \tag{132}$$

Also note that since

$$p(x,\lambda) = \frac{e^{(1-\lambda)p_0+\lambda p_1}}{Z(\lambda)}, \tag{133}$$

$$\tfrac{1}{Z(\lambda)}\tfrac{d}{d\lambda}\left(Z(\lambda)\langle\log h\rangle_{p_\lambda}\right) = \int_{x\in\Omega} p_\lambda\left(p_1 - p_0\right)\log\frac{p_1}{p_0} = \langle\left(p_1 - p_0\right)\log\frac{p_1}{p_0}\rangle_{p_\lambda} \geq 0, \tag{134}$$

(as one might expect from an entropy, up to a normalization) , since

$$\left(p_1 - p_0\right)\log\frac{p_1}{p_0} = p_0(h-1)\log h,$$

$h = \frac{p_1}{p_0}$, and $f(h) = (h-1)\log h \geq 0$ for any $h > 0$. For the same reason, all odd derivatives of $\langle\log h\rangle_{p_\lambda}$ are non-negative,

$$\frac{1}{Z(\lambda)}\frac{d^{2n+1}}{d\lambda^{2n+1}}\left(Z(\lambda)\langle\log h\rangle_{p_\lambda}\right) \geq 0, n = 0, 1, \ldots$$

Since $Z(\lambda) = \int_{x\in\Omega} e^{(1-\lambda)p_0+\lambda p_1}$, we have that

$$\tfrac{1}{Z(\lambda)}\tfrac{d}{d\lambda}Z(\lambda) = \langle p_1 - p_0\rangle_{p_\lambda} = \langle\left(h-1\right)p_0\rangle_{p_\lambda}, \tag{135}$$

thus (134) implies

$$\tfrac{d}{d\lambda}\langle\log h\rangle_{p_\lambda} \geq \langle\left(h-1\right)p_0\rangle_{p_\lambda}\langle\log h\rangle_{p_\lambda}. \tag{136}$$

Note that in statistical physics $-\log Z(\lambda)$ is free energy, and (135) has to do with the rate of the free energy flow.

10.3 Fokker-Planck homotopy flow

Our task here is actually to find F which is generating a Fokker-Planck flow such that

$$\frac{\partial p(x,\lambda)}{\partial \lambda} = \mathrm{div} F, \tag{137}$$

$$F = -fp + \tfrac{1}{2} Q \tfrac{\partial p}{\partial x} \tag{138}$$

(with some f, Q which may depend on p). Knowing F allows to realize the flow computationally efficiently, especially in multi-dimensions, via a stochastic differential equation, with the drift term f and covariance of the noise term Q.)

Note that solving (137) for F is **not** covered by $2.3.8^{(1)}$ in [13]. Indeed, for any polynomials $L_i, i = 1 \dots n$ in n variables $k_1, \dots k_n$, obviously

$$\textstyle\sum_i L_i k_i \neq 1, \tag{139}$$

as the left hand side does not contain a constant term. We might instead view (137) (or its versions obtained by differentiating (137) with respect to x and comparing with an equation obtained for $\frac{\partial^2 \log p}{\partial x \partial \lambda}$ from (126), which eliminates the normalization constant) as under-determined systems for f, Q; however such problem is not a problem with constant coefficients.

10.4 Gaussian case

As a warm-up, not considering any homotopies yet, note that if $x \in \mathbb{R}^1$, $p(x) = \frac{1}{Z} e^{-x^2}$, $Z = \sqrt{\pi}$, and $q_2(x) = \sum_{i=0}^{2} \alpha_i x^i$, is a square trinomial, then F, given by

$$\begin{aligned} -F &= \int_x^\infty p(s) q_2(s) ds - \langle q_2 \rangle_p \int_x^\infty p(s) ds \\ &= \frac{1}{2\sqrt{\pi}} (\alpha_2 x + \alpha_1) p(x) = \frac{1}{4\sqrt{\pi}} \left(-\alpha_2 \frac{\partial}{\partial x} + 2\alpha_1 \right) p(x) \end{aligned} \tag{140}$$

is simply related to p (and does not involve an error function as one would expect from integration of a Gaussian) since integration by parts

$$\textstyle\int_x^\infty e^{-s^2} ds = x e^{-x^2} + 2 \int_x^\infty s^2 e^{-s^2} ds, \tag{141}$$

is reducing the integration of s^2 to integration of a constant, while the constant and linear terms are easy to deal with directly.

The general Gaussian case can be reduced to the above. Computationally efficient formulas in this case are already known , and it is not our goal here to re-derive those; we will just would like to elaborate on why the error function need not to appear in the general Gaussian case. So let

$$\begin{aligned} &p(x,\lambda) = \tfrac{1}{Z(\lambda)} \exp\left(-Q_2(x,\lambda)\right), Q_2(x,\lambda) = x^T A_\lambda x + b_\lambda^T x, \\ &A_\lambda = (1-\lambda) A_0 + \lambda A_1, b_\lambda = (1-\lambda) b_0 + \lambda b_1, \\ &\log h(x) = -x^T \delta A x - \delta b^T x \text{ (up to a constant)}, \\ &\delta A := A_1 - A_0 = \tfrac{d}{d\lambda} A_\lambda, \delta b := b_1 - b_0 = \tfrac{d}{d\lambda} b_\lambda \end{aligned} \tag{142}$$

As the matrix A_λ is positive-definite, and both A_λ and δA are symmetric real matrices, they can be simultaneously diagonalized; while the linear term $b_\lambda^T x$ can be eliminated by a shift in x. Thus there is an affine change of coordinates

$$\begin{aligned} &x = Sy + v, \text{ such that} \\ &Q_2 = -y^2, \log h = y^T D(\lambda) y + \tilde{b}_\lambda^T y + \tilde{c}(\lambda), \end{aligned} \tag{143}$$

where $D(\lambda)$ is a diagonal matrix. We can now use separation of variables and the warm-up example above, to see that error functions can be avoided. As we are free to chose which variable to integrate in order to get F (transformed to the new y coordinates as a vector, accordingly), and we will integrate with respect to y_i, similarly to (140), the terms in $\log h$ involving y_i, for all $i = 1, \dots n$.

10.5 Non-Gaussian cases

We may not get low-complexity answers here, as in the Gaussian case, but we will accept answers in the form of special functions or solutions of ODEs with respect to homotopy parameter λ as sufficiently low complexity, while integrals over x are treated as high complexity (which they would be, in multi-dimensional x setting). See also [5].

Let us consider simplest example, homotopy between probability distribution $\frac{1}{Z_0}e^{-x^4}$ and a Gaussian $\frac{1}{Z_1}e^{-x^2}$ probability distributions on $\mathbb{R}$,

$$\begin{array}{l} p(x,\lambda) = \frac{1}{Z(\lambda)}\exp\left(-q_{2,4}(x,\lambda)\right), q_{2,4} = \lambda x^2 + (1-\lambda)x^4, 0 \le \lambda \le 1, x \in \mathbb{R}, \\ \log g(x) = -x^4, \log h(x) = x^4 - x^2. \end{array} \tag{144}$$

Some simplification here is possible by taking x^2 as a new variable, yet we will consider this example for illustration, and in a manner such that different powers of x can be treated in a similar fashion. To compute F,

$$\begin{array}{l} -F = \frac{1}{Z(\lambda)}\displaystyle\int_x^\infty \exp\left(-q_{2,4}(s,\lambda)\right)\left(s^4 - s^2\right)ds = \\ -\frac{1}{Z^2(\lambda)}\displaystyle\int_x^\infty \exp\left(-q_{2,4}(s,\lambda)\right)ds \int_{-\infty}^\infty \exp\left(-q_{2,4}(s,\lambda)\right)\left(s^4 - s^2\right)ds \end{array} \tag{145}$$

we need to deal with integrals of the form

$$\begin{array}{l} I\left(n,\lambda|x\right) := \int_x^\infty e^{-\left(\lambda s^2 + (1-\lambda)s^4\right)} s^n ds = \lambda^{\frac{n+1}{4}} Z\left(n, a_\lambda | x_\lambda\right), a_\lambda := \frac{\lambda}{(1-\lambda)^{1/4}}, x_\lambda := x(1-\lambda)^{1/4}, \\ Z(n,a|x) := \displaystyle\int_x^\infty e^{-s^4 - as^2} s^n ds, \\ Z(n,a) := Z(n,a|-\infty) = \displaystyle\int_{-\infty}^\infty e^{-s^4 - as^2} s^n ds \end{array} \tag{146}$$

Differentiating, we have that

$$\begin{array}{l} \frac{\partial}{\partial a} Z(n,a|x) = -Z(n+2,a|x), \\ \frac{\partial^2}{\partial a^2} Z(n,a|x) = Z(n+4,a|x), \end{array} \tag{147}$$

while from integrating by parts we have a recursion relation

$$\begin{array}{l} Z(n+4,a|x) = \frac{1}{4}\left((n+1)Z(n,a|x) - 2aZ(n+2,a|x) + e^{-x^4 - ax^2} x^{n+1}\right), \\ Z(n+4,a) = \frac{1}{4}\left((n+1)Z(n,a) - 2aZ(n+2,a)\right). \end{array} \tag{148}$$

As the three functions, $Z(n,a)$ and its first two derivatives with respect to a, can be expressed as a linear combination of just two functions, $Z(n,a)$ and $Z(n+2,a)$ by using the recursion relation, it follows that $Z(n,a)$ will satisfy a liner second order differential equation (with respect to the a variable), and the same is true for $Z(n,a|x)$. Those equations are

$$\begin{array}{l} \left(\frac{\partial^2}{\partial a^2} - \frac{a}{2}\frac{\partial}{\partial a} - \frac{n+1}{4}\right) Z(n,a) = 0, \\ \left(\frac{\partial^2}{\partial a^2} - \frac{a}{2}\frac{\partial}{\partial a} - \frac{n+1}{4}\right) Z(n,a|x) = e^{-x^4 - ax^2} x^{n+1}. \end{array} \tag{149}$$

The initial conditions,

$$\begin{array}{l} Z(n,0) = \frac{1}{4}\left(1 + (-1)^n\right)\Gamma\left[\frac{1+2n}{4}\right], \frac{\partial}{\partial a} Z(n,a)\big|_{a=0} = -\frac{1}{4}\left(1 + (-1)^n\right)\Gamma\left[\frac{3+2n}{4}\right], \\ Z(n,0|x) = \frac{1}{4}\Gamma\left[\frac{1+2n}{4}, x^4\right], \frac{\partial}{\partial a} Z(n,a|x)\big|_{a=0} = -\frac{1}{4}\Gamma\left[\frac{3+2n}{4}, x^4\right], \end{array} \tag{150}$$

where $\Gamma\left[m,x\right] := \int_x^\infty e^{-s} s^{n-1} ds, \Gamma\left[m\right] = \Gamma\left[m,0\right]$. Now, write

$$Z(n,a) := w_n\left(\frac{a^2}{4}\right), Z(n,a|x) = w_{n,x}\left(\frac{a^2}{4}\right) \tag{151}$$

thus

$$
\begin{aligned}
\tfrac{a}{2}\partial_a Z(n,a) &= \tfrac{a^2}{4} w_n'\left(\tfrac{a^2}{4}\right), \\
\partial_a^2 Z(n,a) &= \tfrac{a^2}{4} w_n''\left(\tfrac{a^2}{4}\right) + \tfrac{1}{2} w_n'\left(\tfrac{a^2}{4}\right),
\end{aligned} \tag{152}
$$

and the same relations hold for $Z(n,a|x), w_{n,x}\left(\frac{a^2}{4}\right)$. Therefore , the equations (149) in terms of the functions w_n read:

$$\left(t d_t^2 + \left(\tfrac{1}{2} - t\right) d_t - \tfrac{n+1}{4}\right) w_n(t) = 0, \tag{153}$$

$$\left(t d_t^2 + \left(\tfrac{1}{2} - t\right) d_t - \tfrac{n+1}{4}\right) w_{n,x}(t) = e^{-x^4 - 2\sqrt{t}x^2} x^{n+1}. \tag{154}$$

Equation (153) is the *Kummer's equation*, solved by the *confluent hypergeometric functions*, and (153) is an inhomogeneous version of the same equation, defining a sort of "incomplete" version of confluent hypergeometric functions, by analogy error functions or incomplete Γ functions. The solution may be expressed as a series, by expanding $Z(n,a)$ in powers of a; or by solving ODE; or by computing corresponding integrals The "complete" versions of those are coded in Matlab and Mathematica, quite efficiently. In view of the boundary conditions, it follows that

$$Z(n,a) = 2^{-\frac{1}{2}-n}\Gamma\left[\tfrac{1}{2}+n\right] KummerU\left[\tfrac{n+1}{4}, \tfrac{1}{2}, \tfrac{a^2}{4}\right]. \tag{155}$$

Note that the above considerations describe a flow from quartic ($a = 0$) to Gaussian ($a \to \infty$), as it is easier to establish connection with already known special functions. An alternative is to differentiate the integrals $I(n, \lambda|x)$ with respect to λ, and to use recursion relation emerging from integrating by parts to arrive at a second order differential equation for those integrals. All the steps are similar, but formulas are more cumbersome and the relationship with hypergeometric functions is more obscure in that way.

References

[1] David F Crouse and Codie Lewis. Consideration of particle flow filter implementations and biases. Technical report, NAVAL RESEARCH LAB WASHINGTON DC, 2020.

[2] Liyi Dai and Fred Daum. A new parameterized family of stochastic particle flow filters. arXiv preprint arXiv:2103.09676, 2021.

[3] Liyi Dai and Fred Daum. On the design of stochastic particle flow filters. IEEE Transactions on Aerospace and Electronic Systems, 2022.

[4] Fred Daum. work on particle flows methods in filters problems, 1983.

[5] Fred Daum. Solution of the zakai equation by separation of variables. IEEE Transactions on automatic control, 32(10):941–943, 1987.

[6] Fred Daum and Jim Huang. Particle degeneracy: root cause and solution. In Signal Processing, Sensor Fusion, and Target Recognition XX, volume 8050, pages 367–377. SPIE, 2011.

[7] Fred Daum and Jim Huang. Particle flow and monge-kantorovich transport. In 2012 15th International Conference on Information Fusion, pages 135–142. IEEE, 2012.

[8] Fred Daum and Jim Huang. Particle flow with non-zero diffusion for nonlinear filters. In Signal Processing, Sensor Fusion, and Target Recognition XXII, volume 8745, pages 226–238. SPIE, 2013.

[9] Fred Daum, Jim Huang, and Arjang Noushin. Exact particle flow for nonlinear filters. In Signal processing, sensor fusion, and target recognition XIX, volume 7697, pages 92–110. SPIE, 2010.

[10] Fred Daum, Jim Huang, and Arjang Noushin. Exact particle flow for nonlinear filters. In Signal Processing, Sensor Fusion, and Target Recognition XIX, volume 7697, page 769704. SPIE, 2010.

[11] Fred Daum, Jim Huang, and Arjang Noushin. Coulomb's law particle flow for nonlinear filters. In Signal and Data Processing of Small Targets 2011, volume 8137, pages 99–108. SPIE, 2011.

[12] Fred Daum, Jim Huang, and Arjang Noushin. New theory and numerical results for gromov's method for stochastic particle flow filters. In 2018 21st International Conference on Information Fusion (FUSION), pages 108–115. IEEE, 2018.

[13] Misha Gromov. Partial differential relations, volume 9. Springer Science & Business Media, 2013.

[14] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2019.