ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/252898496

# Exact particle flow for nonlinear filters

**Article** *in* Proceedings of SPIE - The International Society for Optical Engineering · April 2010

DOI: 10.1117/12.839590

CITATIONS
195

READS
493

**3 authors**, including:

Fred Daum
Raytheon Technologies

**123** PUBLICATIONS **4,533** CITATIONS

SEE PROFILE

All content following this page was uploaded by Fred Daum on 04 August 2023.

The user has requested enhancement of the downloaded file.

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

# Exact particle flow for nonlinear filters

Fred Daum, Jim Huang, Arjang Noushin

Fred Daum, Jim Huang, Arjang Noushin, "Exact particle flow for nonlinear filters," Proc. SPIE 7697, Signal Processing, Sensor Fusion, and Target Recognition XIX, 769704 (27 April 2010); doi: 10.1117/12.839590

Event: SPIE Defense, Security, and Sensing, 2010, Orlando, Florida, United States

SPIE.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023 Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

# Exact particle flow for nonlinear filters

**Fred Daum, Jim Huang & Arjang Noushin**
**Raytheon 225 Presidential Way, Woburn MA, 01801**
**frederick_e_daum@raytheon.com & jim_huang@raytheon.com**

Abstract—We have invented a new theory of exact particle flow for nonlinear filters. This generalizes our theory of particle flow that is already many orders of magnitude faster than standard particle filters and which is several orders of magnitude more accurate than the extended Kalman filter for difficult nonlinear problems. The new theory generalizes our recent log-homotopy particle flow filters in three ways: (1) the particle flow corresponds to the exact flow of the conditional probability density; (2) roughly speaking, the old theory was based on incompressible flow (like subsonic flight in air), whereas the new theory allows compressible flow (like supersonic flight in air); (3) the old theory suffers from obstruction of particle flow as well as singularities in the equations for flow, whereas the new theory has no obstructions and no singularities. Moreover, our basic filter theory is a radical departure from all other particle filters in three ways: (a) we do not use any proposal density; (b) we never resample; and (c) we compute Bayes' rule by particle flow rather than as a point wise multiplication.

## 1.0 Introduction

We derive a new theory for nonlinear filters that generalizes the particle flow filter given in [11]-[13]. As shown in [13], the basic particle flow filter is already many orders of magnitude faster than the classic particle filter, and it is several orders of magnitude more accurate than the extended Kalman filter (EKF) for difficult nonlinear problems. The EKF is the workhorse algorithm in many important real world applications (e.g., tracking, navigation, guidance, control, multisensor data fusion, automatic pattern recognition, Bayesian decisions, GPS, IMUs, communications, robotics, financial engineering, weather prediction, climate prediction, image processing, bioinformatics, etc.). However, the EKF accuracy is often surprisingly poor in important applications. Particle filters have the promise of superior accuracy, but at the very large cost of huge real time computational complexity compared with the EKF. Our approach solves this problem by using a new theory which differs radically from other particle filters in many ways: (1) we do not resample particles, thereby avoiding the major computational burden of particle filters; (2) we do not rely on a proposal density from an EKF or UKF, thereby achieving superior estimation accuracy for problems where the EKF or UKF do not work well; (3) we never compute the conditional probability density itself, but rather we represent the log of the unnormalized conditional density, thereby avoiding numerical problems; (4) we do not compute Bayes' rule as a pointwise multiplication, but rather we use particle flow, which avoids one of the major problems with such filters, called "particle degeneracy"; and (5) we never use any Markov chain Monte Carlo (MCMC) methods, such as adaptive Metropolis-Hastings sampling.

We use particle flow to implement Bayes' rule rather than with a pointwise multiplication of two functions. Particle flow allows us to smoothly migrate the particles to the desired regions in state space, thereby avoiding the well known and fundamental problem with particle filters, namely "particle degeneracy" as a result of Bayes' rule. This problem is described in detail in [3] (see pages 40-44), as well as in [4] to [8] and [18]-[19]. To quote one recent paper ([7] page 234): "Practical implementation of the sequential importance sampling method inevitably results in zero weights for all but, usually, one particle, after just a few iterations. This phenomenon is known as

Signal Processing, Sensor Fusion, and Target Recognition XIX, edited by Ivan Kadar,
Proc. of SPIE Vol. 7697, 769704 · © 2010 SPIE · CCC code: 0277-786X/10/$18 · doi: 10.1117/12.839590

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

particle degeneracy in the PF literature." Many methods have been invented to mitigate particle degeneracy, but none have been successful for high dimensional fully coupled problems (see [1] to [8]). A rigorous theoretical analysis of particle degeneracy was recently given in [18]-[19], with the conclusion that: "the particle filter is known to require large Monte Carlo ensembles and frequent resampling ….This drawback is particularly prevalent in higher dimensional systems where the filter becomes unstable and quickly collapses onto a single point mass….The results imply that to avoid collapse, the sample size must grow super-exponentially in the effective dimension" (page 319 of [19]). To quote another authoritative paper (page 256 in [5]): "In particle filtering, the main difficulty arises at the correction step." That is, the major problem is due to measurement updates using Bayes' rule, rather than the propagation of the conditional density from one time to the next. The same diagnosis is given in a very recent survey of particle filters: "The presence or absence of degeneracy is the factor which most often determines whether an SMC [sequential Monte Carlo] algorithm works in practice….some degree of degeneracy is inevitable in all but trivial cases" (page 15 in [8]). In particular, the particles may be well distributed to represent the prior probability density of the state vector, but when we get a new measurement, the likelihood of the measurement is not well represented by this set of particles (i.e., there are no particles near the peak of the likelihood function). In low dimensional problems, it is easy to sample new particles from the likelihood itself to avoid this problem, but in higher dimensions, this idea does not work for obvious reasons. Moreover, the effect of particle degeneracy is much more severe as the dimension of the state vector grows. Also, particle degeneracy is worse for more accurate measurements (i.e., at higher signal to noise ratio), as noted in [4]. This is easy to understand intuitively; imagine the likelihood function getting narrower and narrower, and as a result, the particles used to represent the prior density are getting worse and worse at representing the likelihood (i.e., there are no particles near the peak of the likelihood function).

The new theory derived in this paper generalizes our particle flow filters in three ways: (1) the particle flow corresponds to the exact flow of the conditional probability density; (2) roughly speaking, the old theory was based on incompressible flow of particles (like subsonic flight in air), whereas the new theory allows compressible flow of particles (like supersonic flight in air); (3) the old theory suffers from obstruction of particle flow as well as singularities in the equations for flow, whereas the new theory has no obstructions and no singularities.

A survey of the nonlinear filter problem along with various solutions (particle filters as well as unparticle filters) is given in [10]. An excellent and extremely lucid textbook on particle filters with many detailed examples is [3]. The seminal paper on particle filters is [1], which triggered a torrent of research. The kernel of the idea was in the air much earlier (e.g., see [16]), but computers were not fast & cheap enough to make real time Monte Carlo simulations practical in those days. Many other useful papers on particle filters are in [2] to [7]. Two refreshingly honest and very recent surveys of the limitations of particle filters are given in [8] and [17]. An eagerly awaited compendium of theoretical work on nonlinear filters is soon to be published [14], which presumably contains the latest research (except for our new work on exact particle flow reported in this paper). The classic text on nonlinear filtering (before particle filters) is [15]. Considering this plethora of literature on nonlinear filters, any further elaboration here would be redundant and superfluous.

## 2.0 Derivation of exact particle flow

In this paper we assume that the state vector (x) evolves in time according to a stochastic difference equation (in discrete time) or a stochastic differential equation (in continuous time), as explained in pages 142-143 of [15]. The purpose of filtering is to estimate x given a sequence of noisy measurements related to x, as explained in pages 142-145 of [15]. We assume that the

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

measurements are made in discrete time, for the reasons given in [10]. The modern approach to filtering is to compute or approximate the probability density of x conditioned on the set of all measurements, as explained in [1]-[10]. We start by recalling that the unnormalized conditional probability density of the d-dimensional state vector x is computed using Bayes' rule as follows:

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

in which

$$z_k = k^{th} \text{ measurement}$$
$$Z_k = \text{set of all measurements up to and including the latest}$$
$$Z_k = \{z_1, z_2, ..., z_k\}$$

The so-called likelihood is the unnormalized density of the latest measurement conditioned on the state vector at the time of the measurement, and it is denoted by:

$$p(z_k | x, t_k)$$

In particle filters, as well as essentially any other nonlinear filter, Bayes' rule is implemented as a point wise multiplication of two functions; this is the obvious way to multiply two functions, and it is the way that any normal person or computer would do it. However, there is a very serious and ubiquitous problem with this straightforward approach; namely, that the representation of this product suffers from the curse of dimensionality. In the literature of particle filters, this problem is called "particle degeneracy". In contrast, prediction of the conditional density from one time to the next does not suffer from this problem, owing to the use of particle flow, wherein the particles migrate smoothly in time according to the system dynamics, and the resulting distribution of particles in state space does not suffer from particle degeneracy. Our fundamental idea is to create a differential equation to implement Bayes' rule, and induce a flow of particles analogous to the natural flow in time induced by the system dynamics. That is, we want to find an ODE to implement Bayes' rule, rather than the point wise multiplication of two functions. We will use a homotopy to create this ODE. We cannot create a flow in time, because Bayes' rule operates at discrete points in time; however, we can introduce a scalar valued parameter, called λ, that plays the role of time, and which varies from 0 to 1. You can think of λ as a little loop of synthetic time, which we insert at each discrete measurement time. You should not be afraid of the word "homotopy" as used here; we do not need any abstract mathematics, and you do not need to know the finer points of topology, but rather we will use an elementary definition and simple high school algebra that any normal engineer can readily grasp.

We will use shorthand notation for Bayes' rule, because we are only interested in the fact that we are trying to compute the product of two functions:

$$p(x) = g(x)h(x) \qquad (2)$$

in which g(x) is the prior density of the d-dimensional state vector x, and h(x) is the likelihood of the measurement.

First, take the logarithm of both sides:

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

log(p(x)) = log(g(x)) + log(h(x)) (3)

Second, we will define a homotopy function as follows:

$$\log(p(x,\lambda)) = \log(g(x)) + \lambda \log(h(x)) \tag{4}$$

in which λ is a real valued parameter that varies between 0 and 1. For λ = 1, the homotopy function is exactly equal to log(p), which is what we want to compute, whereas for λ = 0, the homotopy function is exactly equal to log(g).

Suppose that the particle flow corresponding to Bayes' rule obeys the following ODE:

$$\frac{dx}{d\lambda} = f(x,\lambda) \tag{5}$$

Using the chain rule and the Fokker-Planck equation (with zero process noise) it is easy to show that the exact flow of the probability density (corresponding to Bayes' rule) is given by:

$$\frac{dp}{d\lambda} = -pTr(\frac{\partial f}{\partial x}) \tag{6}$$

and therefore for smooth nowhere vanishing densities :

$$\frac{d\log p}{d\lambda} = -Tr(\frac{\partial f}{\partial x}) \tag{7}$$

The chain rule results in:

$$\frac{\partial \log p}{\partial x}\frac{dx}{d\lambda} + \frac{\partial \log p}{\partial \lambda} = -Tr(\frac{\partial f}{\partial x}) \tag{8}$$

Now suppose that p(x,λ) evolves according to the following log-homotopy:

$$\log p(x,\lambda) = \log[g(x(\lambda))] + \lambda \log[h(x(\lambda))] \tag{9}$$

Combining these results gives the following equation:

$$\frac{\partial \log p}{\partial x}\frac{dx}{d\lambda} + \log h = -Tr(\frac{\partial f}{\partial x}) \tag{10}$$

and recalling the definition of f from (5) we get:

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

$$\boxed{\frac{\partial \log p}{\partial x} f + \log h = -Tr(\frac{\partial f}{\partial x}) \qquad (11)}$$

Formally, the exact unique minimum norm solution for f is given in terms of the generalized inverse:

$$\boxed{\frac{dx}{d\lambda} = -[\frac{\partial \log p}{\partial x} + Tr(\frac{\partial}{\partial x})]^{\#} \log h \qquad (12)}$$

in which A# denotes the generalized inverse of the operator A. An excellent reference on the generalized inverse of such linear differential operators is pages 311-317 in [21]. If you don't like generalized inverses of differential operators, you could think of this intuitively as roughly analogous to taking the d-dimensional Fourier transform of (11) to transform the divergence into a multiplication and then use the generalized inverse to compute a unique minimum norm solution, but in fact, the Fourier transform does not actually work so easily here because the differential operators in this case do not have constant coefficients. Notice that the theory derived in [11] & [12] corresponds to the special case in which the divergence in (12) is assumed to be zero. This does not imply, however, that the solution of (12) is divergence free. That is, the special theory derived in [11] & [12] does not necessarily result in irrotational flow, but it is clearly a special case of the new theory derived here nevertheless.

Formally, the most general solution of (11) for f(x,λ) is:

$$\boxed{\frac{dx}{d\lambda} = -C^{\#} \log h + (I - C^{\#}C)y \qquad (13)}$$

in which C# is the generalized inverse of C, where C is a linear differential operator:

$$C = [\frac{\partial \log p}{\partial x} + Tr(\frac{\partial}{\partial x})] \tag{14}$$

and y is an arbitrary d-dimensional vector. For example, one can pick y to stabilize the filter or provide robust stability (similar to our derivation in [25]), but this seems to be gratuitous, based on our numerical experiments so far. One could also pick y to minimize or avoid obstruction of particle flow (see problem #11 in [35]), although this appears to be unnecessary for our exact flow.

There are many other ways that we could solve (11) for f. In particular, (11) is a first order scalar PDE that is linear in f, and thus it can be transformed exactly into a system of ODEs using the method of characteristics (the best reference is still [22]). Obviously there is no unique solution to (11) without assuming more conditions, such as that f is minimum norm. Other assumptions might include: (1) f is a gradient vector, which is the standard trick in physics; (2) f is irrotational, which is a favorite special case in fluid dynamics; this is locally the same as assuming that f is a gradient (i.e., a gradient flow is always irrotational but not conversely); it turns out that both

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

assumptions #1 and #2 are formally the same as assuming that f is minimum norm (see [24] for a crystal clear exposition of this theory without any mention of the term “generalized inverse”); (3) f is divergence free, which is roughly like (but not exactly the same as) the implicit assumption that we made in [11] & [12]; in fluid dynamics this type of flow is called incompressible, and it is a popular simplifying assumption (e.g., subsonic aircraft flight is well approximated by the assumption of incompressible flow, but this is a very bad approximation for supersonic flight); (4) f stabilizes or robustly stabilizes the filter, which is the approach that we used in section 2.0 of [25]; (5) f is balanced or symmetrical or decoupled in the components of x; (6) f avoids or minimizes obstructions to particle flow (see problem #11 in [25]); (7) f reduces ill-conditioning of the filter; (8) hybrids of the above list, and so on. More generally, we could use the d-dimensional version of the Helmholtz projection theorem to write f exactly as the sum of two orthogonal vector fields (gradient and curl). Formally, this is essentially the same as the Hodge decomposition of f into exact, co-exact and harmonic differential forms.

When we solve (11) using the generalized inverse (or assume that f is minimum norm or that f is a gradient vector or that the flow is irrotational), it turns out that the divergence form of the Fokker-Planck equation is helpful, and we can solve the resulting Poisson’s equation for the potential function (and its gradient) using the well known exact Green’s function for the Laplacian, along with integration by parts or else without integration by parts. This results in relatively simple formulas for f that are similar to the theory derived in [11] & [12]; for details, see section 3.0

For problems in which log(h) and log(g) are polynomials in the components of x (e.g., Gaussian and more generally densities from the exponential family), we can compute f(x) by inspection; that is, we can equate like coefficients of powers of the components of x on the RHS & LHS of equation (11). See section 5.0 for the exact solution of equation (11) in this case. Moreover, we could approximate log(h) and log(g) in a Taylor series in x, analogous to the EKF. It is likely that Battin’s trick (sequential scalar updates of the filter) would be particularly useful here.

The most elementary and obvious approach to solving (11) for f is to set all but one component of f to zero, and integrate (11) with respect to the corresponding component of x. One could repeat this d times for the d components of f, and any linear combination of these d solutions is also a solution to (11), owing to the superposition principle for linear PDEs. See section 6.0 for details.

One could extend this idea even further in several directions. For example, generalize the derivation by using the Fokker-Planck equation with non-zero diffusion or allow jumps in the stochastic process. We assumed zero process noise in the above derivation only to simplify the calculations. It turns out that it is easy to include non-zero diffusion in certain cases.

## 3.0 Quasi-irrotational particle flow

We will now derive an exact particle flow using the assumption that the relevant vector field is the gradient of a scalar valued potential. This flow is exact in the sense that it moves the particles to the exact conditional unnormalized probability density, owing to the Fokker-Planck equation. We emphasize that the particle flow itself is not necessarily irrotational. That is, the vector field f(x,λ) is not irrotational, despite the fact that a closely related vector field (called q(x,λ) below) is the gradient of a potential. For this reason, we call the particle flow “quasi-irrotational”.

We start with the divergence form of the Fokker-Planck equation with zero diffusion:

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

$$\frac{\partial p(x,\lambda)}{\partial \lambda} = -Tr\left[\frac{\partial (pf)}{\partial x}\right] \qquad (15)$$

For nowhere vanishing densities we get:

$$\frac{\partial \log p(x,\lambda)}{\partial \lambda} p(x,\lambda) = -Tr\left[\frac{\partial (pf)}{\partial x}\right] \qquad (16)$$

Using the definition of the log-homotopy (9) we obtain:

$$\log h(x) p(x,\lambda) = -Tr\left[\frac{\partial (pf)}{\partial x}\right] \qquad (17)$$

Define the new vector field:

$$q(x,\lambda) = p(x,\lambda) f(x,\lambda) \qquad (18)$$

Combining (17) and (18) results in:

$$\log h(x) p(x,\lambda) = -Tr\left[\frac{\partial q(x,\lambda)}{\partial x}\right] \qquad (19)$$

There is obviously no unique solution for q(x,λ) to equation (19), because (roughly speaking) there are too few equations and too many unknowns. Therefore we must make further assumptions about the solution, as discussed in section 2.0. In particular, we assume that the vector field q is the gradient of a scalar-valued potential, called V(x,λ):

$$q(x,\lambda) = \left[\frac{\partial V(x,\lambda)}{\partial x}\right]^{\mathrm{T}} \qquad (20)$$

Therefore the potential V(x,λ) satisfies Poisson's equation:

$$\log h(x) p(x,\lambda) = -Tr\left[\frac{\partial^2 V(x,\lambda)}{\partial x^2}\right] \qquad (21)$$

The exact quasi-irrotational particle flow satisfies the following ODE:

$$\frac{dx}{d\lambda} = \left[\frac{\partial V(x,\lambda)}{\partial x}\right]^{\mathrm{T}} / \mathrm{p}(\mathrm{x},\lambda) \qquad (22)$$

There are many ways to solve Poisson's equation (discussed later), but for simplicity and concreteness here we will exploit the fact that we can compute the Green's function exactly and represent the solution as a convolution integral for d ≥ 3:

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

$$V(x,\lambda)=\int \log h(y)p(y,\lambda)\frac{c}{\|x-y\|^{d-2}}dy \tag{23}$$

in which the integral is over d-dimensional Euclidean space, and we have implicitly assumed that the probability density decays to zero sufficiently fast as the norm of x approaches infinity (this is the usual mild technical assumption to guarantee convergence of the integral). Also, in (23):

$$c=\Gamma(\frac{d}{2}-1)/4\pi^{d/2} \tag{24}$$

Now we compute the gradient of V by differentiating (23):

$$\frac{\partial V(x,\lambda)}{\partial x}=\int \log h(y)p(y,\lambda)\frac{\partial}{\partial x}\left[\frac{c}{\|x-y\|^{d-2}}\right]dy \tag{25}$$

Integration by parts (again assuming that the relevant integrands decay to zero as the norm of x approaches infinity) results in:

$$\boxed{\frac{\partial V(x,\lambda)}{\partial x}=-\int \frac{\partial \log h(y)p(y,\lambda)}{\partial y}\frac{c}{\|x-y\|^{d-2}}dy \qquad (26)}$$

One could also use (25) without integration by parts, by computing the gradient of the Poisson kernel, which results in:

$$\boxed{\frac{\partial V(x,\lambda)}{\partial x}=\int \log h(y)p(y,\lambda)\frac{c(2-d)(x-y)^T}{\|x-y\|^{d}}dy \qquad (27)}$$

There are many ways to evaluate these integrals, including: (1) Monte Carlo or quasi-Monte Carlo sampling using our particles as sample points; (2) Taylor series expansion of the integrand about the point x and integration of the individual terms; (3) asymptotic expansion of the integral; (4) approximation of the integral by Laplace's method; and so on. Method #2 is standard in electrostatics, wherein the first order term corresponds to the dipole moment of the electric charge distribution (see equation 4.13 on page 101 of [30]). Method #1 is analogous to computing the electric field strength in electrostatics. In particular, it is very easy to compute the electric field strength, E(x), at point x (in d = 3) using the gradient of the Poisson kernel for a collection of N point charges (page 22 in [30]):

$$E(x)=\frac{\partial \phi(x)}{\partial x}=Q_1\frac{(x-x_1)^T}{\|x-x_1\|^3}+\ldots+Q_N\frac{(x-x_N)^T}{\|x-x_N\|^3} \tag{28}$$

in which $\mathrm{Q}_1,\ldots,Q_N$ are the values of electric charge at the N points, and $\phi$(x) is the electric potential at the point x.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

This simple formula is analogous to our equation (27), with the summation replacing the integral and point charges replacing the continuous charge density. In electrostatics, however, one cannot evaluate the field strength at the location of any point charge, owing to the singularity of the gradient of the Poisson kernel at such points. This is analogous to the singularity of the gradient of the Poisson kernel in (26) and (27). The same fundamental problem arises in gravity (both Newton & Einstein) and fluid dynamics and any other branch of physics that uses the Poisson kernel. Such singularities have not bothered physicists for the last two hundred years, except for a small minority in the avant garde of research on string theory, where point particles are replaced by strings (i.e., objects extended in space in certain dimensions), and hence no such singularities occur in string theory (e.g., Ed Witten & Michael Green have explained the essence of string theory this way on some occasions). But maybe the notion of strings (rather than particles) would help us to mitigate such singularities in (26) and (27). Regularized particle filters use Gaussian kernels rather than particles to improve PF performance (e.g., see [5]). Moreover, the notion of weak solution of PDEs has been standard for many decades, both in theory and practical numerical algorithms (e.g., finite element methods). Such speculations may or may not be useful here.

We emphasize that we do not need an excellent approximation of the integrals in (26) or (27), but rather a very crude approximation should be good enough for most practical applications. Our intuition is based on two facts: First, using the assumption of incompressible flow (i.e., divergence free flow) we achieved excellent filter accuracy; that is, by ignoring the divergence of f(x) completely (and thereby avoiding the solution of Poisson's equation) we get a very good approximation, as shown by our numerical experiments in [13]. Second, we do not need to move the particles to exactly the correct distribution, but rather we only need to move the particles to adequately approximate the conditional density.

It is tempting to approximate the integral in (26) by abusing the mean value theorem (MVT) for integrals, because it results in a very simple and convenient formula for the flow of particles:

$$\frac{dx}{d\lambda} \approx -\left[\frac{\partial \log p(x,\lambda)}{\partial x}\log h(x) + \frac{\partial \log h(x)}{\partial x}\right]^T \frac{c}{\|x\|^{d-2}} \tag{29}$$

Unfortunately, numerical experiments show that this is an extremely bad approximation, owing to the singularity of the Poisson kernel which invalidates the application of the MVT; we display this bad approximation here as a cautionary tale, as well as for other reasons explained below. Another reason to display (29) is to shown an example of what a convenient solution looks like. In particular, we can readily compute everything in (29); in fact, there is nothing new that is required beyond our old theory [13]; furthermore, (29) uses the log of the unnormalized density and its gradient rather than the density itself; we do not like to compute the density for numerical reasons (but we have done it successfully when required). Recall that an acceptable "solution" of Poisson's equation for us is rather peculiar. We emphasize that we know h(x) as a formula that can be computed easily for any x, but p(x) is only known at random points in d-dimensional space. For example, our approximation to the gradient of the log-homotopy is carefully designed to accommodate such peculiarities of our problem (see [31] for more details).

We will now show that the assumption of a gradient vector field used above is identical to the assumption of minimum norm using the generalized inverse, as discussed in section 2.0. To simplify calculations we will use the divergence form of the Fokker-Planck equation; that is, we would like to solve (19) for q(x,λ), but as discussed before, there is no unique solution, and therefore we are forced to make additional assumptions on the solution. At (20) we assumed that q(x,λ) is a gradient vector field, whereas now we assume instead that q(x,λ) is minimum norm.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

The unique minimum norm solution is given (formally) by the generalized inverse. In particular, formally the generalized inverse of C for our linear differential operator is:

$$C^{\#} = C^{T}(CC^{T})^{-1} \tag{30}$$

in which C is defined by the divergence operator in (19), and

$$C^{T} = \text{formal adjoint of C}$$
$$\mathrm{C}^{\mathrm{T}} = -\left[\frac{\partial}{\partial x}\right]^{T}$$
$$CC^{T} = \text{Laplacian (i.e., divergence of the gradient)}$$
$$(\mathrm{CC}^{\mathrm{T}})^{-1} = \text{inverse of the Laplacian (i.e., solution of Poisson's equation)}$$

In the above, we used the fact that the formal adjoint of the divergence is minus the gradient, owing to integration by parts, and the usual assumption that the integrand decays to zero at the boundary (i.e., as the norm of x approaches infinity in this case). Using the definition of q(x,λ), the particle flow derived using the generalized inverse is:

$$\frac{dx}{d\lambda} = q(x,\lambda)/p(x,\lambda) \tag{31}$$

in which q(x,λ) is the unique minimum norm solution of (19):

$$q(x,\lambda) = -C^{\#}[\log h(x)p(x,\lambda)] \tag{32}$$

As shown above, the generalized inverse in this case means: solve Poisson's equation and compute the negative gradient of the result (see equation (30)). But this is exactly the same as the particle flow that we derived using the assumption that q(x,λ) is a gradient of a scalar-valued potential. This explains the relationship between the solution using the generalized inverse, which assumes that q is minimum norm, and the solution using Poisson's equation, which assumes that the relevant vector field (q rather than f) is a gradient vector field. As shown above these two apparently very different solutions are actually identical. Lanczos gives a crystal clear explanation of all of this (see pages 139-140 and 201-205 in [24]), without any mention of generalized inverses. Lanczos apparently invented the notion of generalized inverse before Penrose's paper was published, and Moore's paper was not well known by Penrose or others. Lanczos considers the minimum norm solution to use the minimal information given by the basic problem; hence, the gradient solution uses the minimal information given by the problem without adding any gratuitous terms to the solution. Curiously, this fundamental insight that the gradient solution is the same as the minimum norm solution (and hence the origin of potential functions) is never revealed in other books or papers on physics or math.

As mentioned above, there are many other ways to solve Poisson's equation, including: (1) Monte Carlo or quasi-Monte Carlo methods applied to the PDE [29]; (2) relate Poisson's equation to the diffusion equation and solve it using Monte Carlo methods [28]; (3) use numerical quadrature by exploiting the Poisson kernel [26] & [27]. It turns out that one can approximate the solution of Poisson's equation to excellent accuracy for very high dimensional problems with reasonable computational complexity; in particular, as reported in [26] & [27] the computational

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

complexity is linear in d. For example, one can solve Poisson's equation for d up to 300 with a fractional error of roughly one part in one million for typical problems, with linear computational complexity in d (see [26] & [27] for details). This is a relatively new technology; to quote [26]: "even a few years ago this problem encountered unsurmountable [sic] difficulties due to the so-called "curse of dimensionality".

Finally, we collect the four implicit assumptions that we made to derive equations (26) and (27) for quasi-irrotational flow: (1) the densities are twice continuously differentiable with respect to x; (2) the densities are nowhere vanishing; (3) the densities decay to zero sufficiently fast as the norm of x approaches infinity; and (4) the dimension of x is at least 3; that is, d ≥ 3.

Such assumptions contrast with the usual work on particle filters, which assume no smoothness and no regularity and no symmetry and no special structure of the problem. We have been able to exploit such smoothness and regularity of the filtering problem to reduce computational complexity. Moreover, the reason that we can solve Poisson's equation with linear computational complexity in d is that the Laplacian has so much smoothness and symmetry and special structure. This is a philosophical comment on the computational complexity of nonlinear filters, but it has important practical implications. In particular, in order to have any hope of solving such problems with reasonable computational complexity, one must exploit smoothness or symmetry or structure in the problem.

## 4.0 Variational solution

We can also obtain a unique solution for the particle flow using a variational formulation of the problem. In particular, we can compute the function f(x,λ) by minimizing the following function at each point x:

$$J = \left\| f - f_0 \right\|^2 + w \left\| \log h(x) p(x,\lambda) + Tr\left[ \frac{\partial p(x,\lambda) f(x,\lambda)}{\partial x} \right] \right\|^2 \qquad (33)$$

in which $f_0 = f_0(x,\lambda)$ is an a priori estimate of f (e.g., our incompressible flow). The rationale for this particular function is that it allows us to use a good guess for f (which we happen to have available from the incompressible flow solution [11]-[13]), and also uses the more general compressible flow defined by equation (17). In the above, w is a design parameter (e.g., w = 1 weights the incompressible and compressible flows equally).

A necessary condition for f to minimize J is obtained by differentiation of (33) with respect to the function f:

$$\frac{\partial J}{\partial f} = 2(f - f_0)^T + 2w\left[ \log h(x) p(x,\lambda) + Tr\left[ \frac{\partial (pf)}{\partial x} \right] \right] \left( \frac{\partial p}{\partial x} \right) = 0 \qquad (34)$$

and therefore

$$f = f_0 - w\left[ \log h(x) p(x,\lambda) + Tr\left[ \frac{\partial (pf)}{\partial x} \right] \right] \left( \frac{\partial p}{\partial x} \right)^T \qquad (35)$$

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

and expanding the divergence of p(x,λ)f(x,λ) we get:

$$f = f_0 - w\left[\log h(x)p(x,\lambda) + \frac{\partial p}{\partial x}f + pTr\left(\frac{\partial f}{\partial x}\right)\right]\left(\frac{\partial p}{\partial x}\right)^T \tag{36}$$

Furthermore, we can solve explicitly for the function f(x, λ) as follows:

$$\left(I + w\left(\frac{\partial p}{\partial x}\right)^T\left(\frac{\partial p}{\partial x}\right)\right)f = f_0 - w\left[\log h(x)p(x,\lambda) + pTr\left(\frac{\partial f}{\partial x}\right)\right]\left(\frac{\partial p}{\partial x}\right)^T \tag{37}$$

$$f = \left(I + w\left(\frac{\partial p}{\partial x}\right)^T\left(\frac{\partial p}{\partial x}\right)\right)^{-1}\left\{f_0 - wp(x,\lambda)\left[\log h(x) + Tr\left(\frac{\partial f}{\partial x}\right)\right]\left(\frac{\partial p}{\partial x}\right)^T\right\} \tag{38}$$

The fly in the ointment is that the divergence of f is on the RHS of (38), but we can nevertheless solve for f by iteration, using an a priori estimate of the divergence of f to prime the pump, and then continuing the iterations in a bootstrap fashion. In particular, div(f) = 0 for our incompressible flow.

Moreover, in the limit as w approaches infinity, we get the following very interesting formula, which is remarkably similar to our original generalized inverse of the log-homotopy derived in [11]-[12]:

$$f = -\left[\left(\frac{\partial \log p(x,\lambda)}{\partial x}\right)^T\left(\frac{\partial \log p(x,\lambda)}{\partial x}\right)\right]^{\#}\left(\frac{\partial \log p(x,\lambda)}{\partial x}\right)^T\left[\log h(x) + Tr\left(\frac{\partial f}{\partial x}\right)\right] \tag{39}$$

Using the properties of the generalized inverse it is easy to further simplify this formula as follows:

$$\boxed{f(x,\lambda) = -\left(\frac{\partial \log p(x,\lambda)}{\partial x}\right)^{\#}\left[\log h(x) + Tr\left(\frac{\partial f(x,\lambda)}{\partial x}\right)\right] \qquad (40)}$$

In particular, we can see that the particle flow is in the direction of (our old friend) the gradient of the log-homotopy (exactly as in [11]-[13]), with the speed of the particle flow determined by the log-likelihood (as before in [11]-[13]) but modified by the divergence of f(x,λ). We happen to know, from numerical experiments, that changing the speed of particle flow by orders of magnitude has very little effect on overall filter performance. That is, the filter performance is very insensitive to the divergence of f, at least for the specific examples that we have tested by numerical experiments so far. This suggests that the incompressible flow derived in [11]-[12] is just as good as the more general compressible flow derived in this paper. But why? This deserves further investigation, and perhaps a bound or formula or theorem to explain it. Moreover, there might be other examples for which this insensitivity to the divergence of f(x,λ) breaks down. This is analogous to the situation in fluid dynamics, where incompressible flow is an excellent approximation for subsonic flight in air, but it breaks down for supersonic flight.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

The particle flow derived in [11]-[12] is the same as (40) but with zero divergence of f(x,λ). Hence the speed of particle flow in [11]-[12] is proportional to the log-likelihood. This makes sense intuitively, because the speed should be faster for very accurate measurements, and the speed should be exactly zero for uninformative measurements (i.e., very low signal-to-noise ratio data). That is, there is exactly no change in the conditional density of x as a result of measurements that are worthless. Thus, the theory in [11]-[12] is in perfect agreement with our intuition. However, this is not so obvious for (40), where there might be a non-zero flow of particles even for worthless measurements, owing to a non-zero divergence of f. This is counterintuitive and it demands an explanation. One possible explanation is that the divergence of f should be zero, but that is not necessary. All that we need is that logh(x) + div(f) = 0 when a given measurement is worthless, and this apparently happens automatically, because f is computed from logh(x); for example, see equations (31) & (32).

Notice that the perspicuous formula (40) derived by a rather circuitous route above could also have been deduced from (11) by inspection. That is, the particle flow computed by (40) is the unique minimum norm solution of (11), if we consider the divergence of f(x,λ) as given (but not otherwise). We emphasize that div(f) should be chosen such that logh(x) + div(f) = 0 when a given measurement is useless, but this is easy to arrange. Actually, even that condition on div(f) is not necessary; it is possible for particles to move in closed orbits and return to the starting position, thereby resulting in no change to the conditional density. Such orbits suggest non-zero vorticity in the particle flow, which is ruled out by choosing f(x,λ) as an irrotational vector field, which we did in section 3.0.

It is interesting to note that if f(x,λ) is linear in x (i.e. f(x,λ) = Ax), then the divergence of f(x,λ) is a constant, which could be absorbed into logh(x) in (40). But such constants have exactly no effect on the unnormalized conditional density of x, which is what we are trying to compute. Hence, to first order in x, the divergence of f(x,λ) should have no effect on our problem. Maybe that is the intuitive explanation for why the divergence of f(x,λ) should be zero. In any case, this simple theory is in perfect agreement with our numerical experiments. But of course, second and higher order terms in x will spoil this argument.

## 5.0 Exact flow for Gaussian and exponential families

As noted in section 2.0, we can solve equation (11) exactly for f(x, λ) when log(h) and log(g) are polynomials in the components of the d-dimensional vector x. For example, if the prior density and the likelihood are Gaussian, then we can solve (11) exactly, and we will do this shortly. More generally, when the prior and likelihood are from an exponential family we can also solve (11) exactly by equating like coefficients on the RHS & LHS. Moreover, one could approximate log(g) and log(h) in a Taylor series expansion (of arbitrarily high order) about the current estimate of x, analogous to the first order approximation used in the EKF. The only issue with displaying such a solution in a paper like this is notational, but the ODE can be coded in MATLAB without any difficulty.

The notational problems vanish for Gaussian densities. Consider equation (11) and assume that the prior and the likelihood are unnormalized Gaussian probability densities. Substituting these densities into (11) and equating like coefficients of x and terms that are quadratic in x results in the following exact solution for the particle flow corresponding to Bayes' rule:

$$\boxed{\frac{dx}{d\lambda} = A(\lambda)x + b(\lambda) \qquad (41)}$$

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

in which

$$A = -\frac{1}{2}PH^T(\lambda HPH^T + R)^{-1}H \tag{42}$$

$$b = (I + 2\lambda A)[(I + \lambda A)PH^T R^{-1}z + A\overline{x}] \tag{43}$$

where P is the covariance matrix of prediction error in x for the Gaussian prior density, and H is the measurement matrix (i.e., z = Hx+v), and R is the covariance matrix of the measurement noise (v). The symbol $\overline{x}$ denotes the predicted value of x, corresponding to the mean value of the prior Gaussian density. Equations (42) & (43) are the result of algebraic simplifications using Woodbury's matrix inversion lemma, similar to the standard equations for the Kalman filter (e.g., see problem 2 on page 357 of Bryson & Ho [32]). Without such algebraic simplifications the equations for A and b would be much uglier, and also they would require inversion of several dxd matrices. We have spared the reader these ugly equations.

It is easy to see that the eigenvalues of A(λ) are all negative or zero, owing to the assumption of the positive definiteness of the covariance matrices P and R, as well as the fact that A is the product of two symmetric matrices. This implies that the particle flow is Schur stable. Moreover, in all of our numerical experiments, the particle flow (as well as the filter) corresponding to (41) was indeed stable. However, it is well known (by some) that eigenvalues of A in the left half plane is a necessary but not a sufficient condition for stability of particle flow, owing to the dependence of A on λ; that is, one can construct examples of continuous time linear systems all of whose eigenvalues are in the left half plane, but nevertheless the system is unstable. A more careful stability analysis is called for.

We emphasize that equation (41) has no singularities. Moreover there are no obstructions to particle flow that are evident in our numerical experiments. These two facts are intimately related, as shown in [34] for d =1. This contrasts with our approximate (incompressible) particle flow, which assumes that the divergence of f is zero, and which results in both singularitites at isolated points in d-dimensional space for Gaussian problems, as well as obstruction to particle flow (see [34] as well as problem #11 in [35] for more details). The reason for this is that at the start of our derivation of equation (11), we have asked for the exact particle flow corresponding to Bayes' rule; intuitively it seems to us that there always should exist such a flow for problems in d-dimensional Euclidean space. However, for topologically "more interesting" state spaces with disconnected components (corresponding to dynamical systems that are not completely controllable), the situation is not so clear. This deserves further analysis. But for most practical engineering problems, including all of the five classes of examples that we have studied for our numerical experiments, there are no obstructions to particle flow using our new exact equations.

The method used above to solve (11) for f is called "separation of variables", which is the most popular method of solving PDEs exactly. It is essentially the same as the exact solution of the Fokker-Planck equation for the Kalman filter problem (with continuous time plant), as well as the exact solution of the Zakai equation for nonlinear filters with unnormalized densities in the exponential family (see [10] for more details on the exponential family). It turns out that the constant terms on the RHS & LHS of (11) are irrelevant, because we are only interested in the unnormalized probability density. Moreover, for the Gaussian problem solved above, it is obvious that the divergence of f is a constant; namely div(f) = Tr(A). Hence, for such Gaussian problems, the divergence of f only shifts the irrelevant constant that scales log(h); this comment is related to our previous discussion of the significance (or lack of significance) of div(f) in section 4.0.

## 6.0 Elementary solution of fundamental PDE

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

As noted in section 2.0, the most elementary and obvious approach to solving (11) for f is to set all but one component of f to zero, and integrate (11) with respect to the corresponding component of x. One could repeat this d times for the d components of f, and any linear combination of these d solutions is also a solution to (11), owing to the superposition principle for linear PDEs. Using the divergence form of our fundamental PDE (19) we obtain:

$$\boxed{\frac{dx_j}{d\lambda} = -\int^{x_j} p(x,\lambda)\log h(x)dx_j \,/\, p(x,\lambda) \qquad (44)}$$

in which j = 1, 2,…,d. We emphasize that the computational complexity to approximate this integral is relatively small, because it is an integral with repect to only one component of x, rather than the d-dimensional vector x. Also, there are many ways to approximate this integral. For example: (1) use the exact function h(x) but approximate g(x) as Gaussian (or a suitable exponential family), and compute the integral exactly using calculus; (2) mean value theorem for integrals; (3) Taylor series approximation of p(x,λ) but exact h(x), and compute the integral exactly with calculus; (4) Taylor series approximation of both p(x,λ) and log(h), and compute integral exactly with calculus; (5) exploit the similarity of (44) with Shannon's entropy or Kullback-Leibler measure or Chernoff distance or Bhattacharyya distance and borrow various approximations from information and decision theory; (6) use Battin's trick (sequential scalar valued updates of the filter), which reduces the complexity of h(x); (7) various numerical integration methods; and (8) if all else fails, try integration by parts.

## 7.0 Numerical experiments

We have made hundreds of plots that summarize the numerical experiments for the exact theory, a subset of which are published in a companion paper [33], owing to page limitations here. We compare the estimation accuracy of four filters: (1) EKF; (2) approximate (incompressible) flow without explicit stabilization; (3) approximate (incompressible) flow with stabilization; and (4) exact particle flow. Apparently, our new exact particle flow is many orders of magnitude more accurate than the EKF for difficult nonlinear problems, and sometimes it is an order of magnitude more accurate than the incompressible flow, but it is (almost) never worse than the incompressible flow. Evidently, our new exact flow does not require explicit stabilization, even for plants that are unstable. Moreover, as one would expect, the exact flow cannot be better than the Kalman filter for linear problems with stable or unstable plants, but it is nevertheless slightly better than our incompressible flow for such linear stable problems. In summary, our numerical results are in excellent agreement with both theory and intuition. Moreover, the exact flow (and the filter itself) is stable, and there are no singularities in the flow, and there are no obstructions to particle flow. Furthermore, the new exact flow produces a distribution of particles that is much closer to the exact probability density as compared with the approximate (incompressible) flow. In addition, the new filter has much faster computer run time per particle (see Figure 1). In particular, the new exact flow filter is many orders of magnitude faster per particle than standard particle filters, and it is roughly one order of magnitude faster per particle than our approximate (incompressible) flow filter.

## Acknowledgements

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

We thank Lennart Svensson and Mark Morelande for asking several excellent probing questions [20] that inspired the new theory in this paper. We are extremely grateful to Lingji Chen for sending his paper [34] to us many months before publication, as well as for a spirited exchange of emails, which motivated the new exact theory in this paper; Lingji's paper clarified the connection between singularities in particle flow and obstructions to particle flow for d = 1 problems. We thank Professor Guido Sandri for asking us (last year at a beautiful concert) whether we exploit symmetry or structure or smoothness or sparseness of our problem, and if not, then "where is the magic?".

## References

[1] Neil Gordon, David Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," IEE Proceedings-F 140(2), pp. 107-113 (1993).

[2] "Sequential Monte Carlo Methods in Practice," edited by Arnaud Doucet, Nando de Freitas and Neil Gordon, Springer-Verlag, 2001.

[3] Branko Ristic, S. Arulampalam and Neil Gordon, "Beyond the Kalman filter," Artech House, 2004.

[4] Mike Pitt and Neil Shephard, "Filtering via simulation: auxiliary particle filters," Journal of American Statistical Association, pages 590-599, June 1999.

[5] Christian Musso, Nadia Oudjane and Francois Le Gland, "Improving Regularised Particle Filters," Chapter 12 in "Sequential Monte Carlo Methods in Practice," edited by A. Doucet, N. de Freitas and N. Gordon, Springer-Verlag, 2001.

[6] Simon Godsill and Tim Clapp, "Improvement Strategies for Monte Carlo Particle Filters," Chapter 7 in "Sequential Monte Carlo Methods in Practice," edited by A. Doucet, N. de Freitas and N. Gordon, Springer-Verlag, 2001.

[7] Avishy Carmi and Yaakov Oshman, "Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations," Journal of Guidance, Control and Dynamics, vol. 32, No. 1, January-February 2009.

[8] Arnaud Doucet and Adam Johansen, "A Tutorial on Particle Filters and Smoothing: Fifteen years later," available on-line, version 1.1, December 2008; also, to be published in [14].

[9] Francois Le Gland and Nadia Oudjane, "A Robustification Approach to Stability and to Uniform Particle Approximation of Nonlinear Filters," INRIA Technical Report, March 2002.

[10] Fred Daum, "Nonlinear filters: Beyond the Kalman filter," Special tutorial issue of the IEEE Aerospace and Electronic Systems Magazine, August 2005.

[11] Fred Daum and Jim Huang, "Nonlinear filters with log-homotopy," Proceedings of SPIE Conference on signal & data processing," Orlando Florida, March 2007.

[12] Fred Daum and Jim Huang, "Particle flow for nonlinear filters with log-homotopy," Proceedings of SPIE Conference on signal & data processing," Orlando Florida, March 2008.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

[13] Fred Daum and Jim Huang, “Nonlinear filters with particle flow induced by log-homotopy,” Proceedings of SPIE Conference on Signal Processing, Orlando Florida, April 2009.

[14] Dan Crisan and Boris Rozovsky (editors), “The Oxford University handbook of nonlinear filtering,” Oxford University Press, 2010.

[15] Andrew Jazwinski, “Stochastic Processes and Filtering Theory,” Dover Books, 1970.

[16] J. E. Handschin and David Q. Mayne, “Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering” International Journal of Control, first series, Vol. 9, pp. 547-559, 1969.

[17] Christophe Andrieu, Arnaud Doucet & Roman Holenstein, “Particle Markov chain Monte Carlo methods,” read before a meeting of the Royal Statistical Society on 14 October 2009, available on-line, to be published in Journal of Royal Statistical Society series B, volume 72, Part 2, pages 1-33, 2010.

[18] Thomas Bengtsson, Peter Bickel and Bo Li, “Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems”, IMS Collections Probability and Statistics: Essays in Honor of David A. Freedman, Vol. 2 (2008) 316–334, Institute of Mathematical Statistics, 2008.

[19] Peter Bickel, Bo Li & Thomas Bengtsson, “Sharp failure rates for the bootstrap particle filter in high dimensions,” IMS Collections Volume 3 pages 318-329, 2008.

[20] Lennart Svensson and Mark Morelande, “Reflections on log-homotopy based particle filters,” unpublished notes, 19 November 2009.

[21] Adi Ben-Israel and Thomas Greville, “Generalized inverses,” 2nd edition, Canadian mathematical society, 2002.

[22] Richard Courant & David Hilbert, “Methods of mathematical physics, volume 2: partial differential equations,” Wiley-Interscience, 1962.

[23] Richard Courant & Karl Otto Friedrichs, “Supersonic flow and shock waves,” Wiley-Interscience, 1948.

[24] Cornelius Lanczos, “Linear differential operators,” (originally published in 1961) Dover books, 2006.

[25] Fred Daum and Jim Huang, “Generalized particle flow for nonlinear filters,” Proceedings of SPIE Conference, Orlando, April 2010.

[26] Flavia Lanzara, Vladimir Maz’ya and Gunther Schmidt, “On the fast computation of high dimensional volume potentials,” arXiv:0911.0443v1 [Math.NA], 2 November 2009.

[27] Boris Khoromskij and Ivan Oseledets, “Quantics-TT approximation of elliptic solution operators in high dimension,” Max Planck Institut für Mathematik in den Naturwissenschaften Leipzig, report no. 79, 2009.

[28] Etienne Pardoux and A. Yu. Veretennikov, “On the Poisson equation and diffusion approximation 3,” The Annals of Probability, 2005.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

[29] B. Z. Lu, Y. C. Zhou, M. J. Holst and J. A. McCammon, "Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications," http://www.global-sci-com/, 2008.

[30] John David Jackson, "Classical Electrodynamics," John Wiley & Sons, Inc., 1962.

[31] Fred Daum, Jim Huang, Misha Krichman and Talia Kohen, "Seventeen dubious methods to approximate the gradient for nonlinear filters with particle flow," SPIE Conference Proceedings, San Diego, August 2009.

[32] Arthur Bryson and Yu-Chi Ho, "Applied optimal control," Blaisdell Publishing Company, 1969.

[33] Fred Daum and Jim Huang, "Numerical experiments on nonlinear filters with exact particle flow induced by log-homotopy," Proceedings of SPIE Conference, Orlando Florida, April 2010.

[34] Lingji Chen, "A study of 'nonlinear filters with particle flow induced by log-homotopy'" *Proceedings of SPIE Conference 7697, Orlando Florida, April 2010.*

[35] Fred Daum and Jim Huang, "A fresh perspective on research for nonlinear filters with particle flow induced by log-homotopy," Proceedings of SPIE Conference, Orlando Florida, April 2010.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

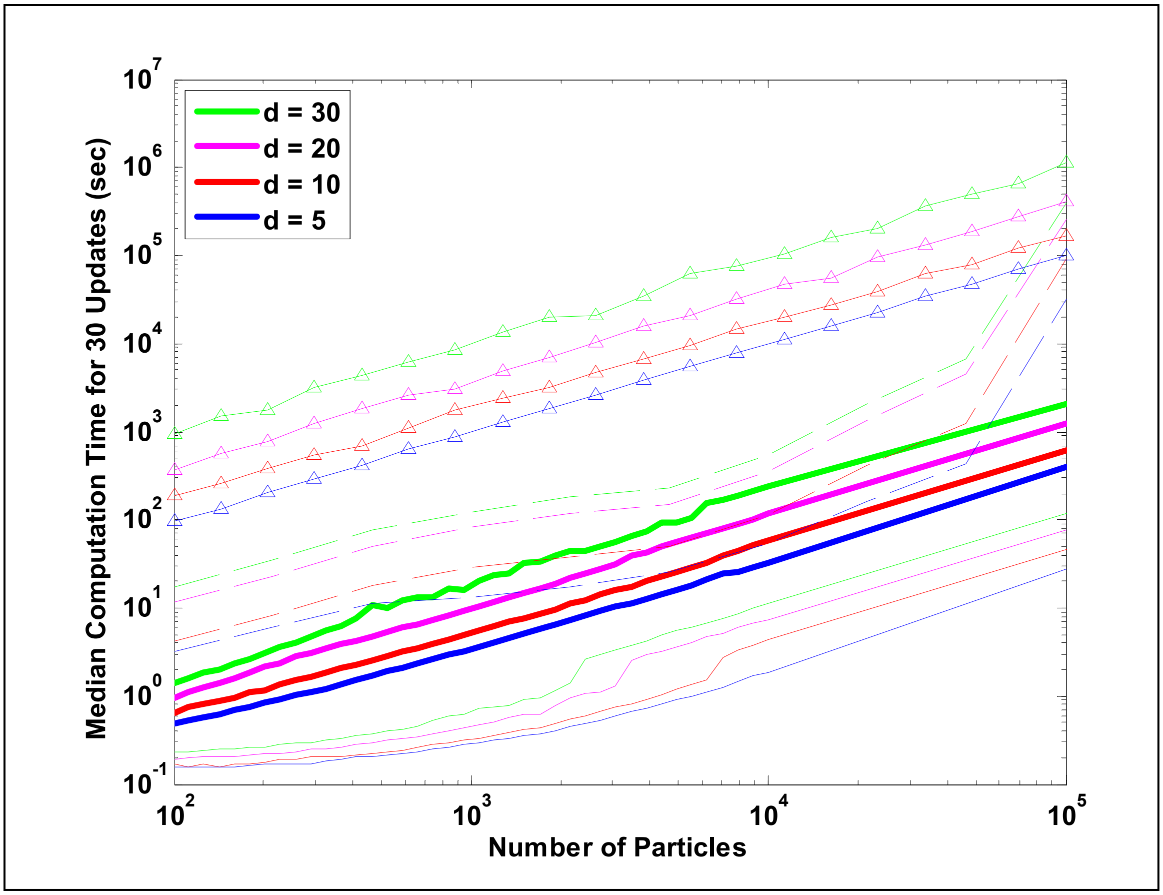

**Figure 1 Exact flow filter is many orders of magnitude faster than other particle filters (classic PF denoted by triangles, PF with proposal density from EKF denoted by dashed lines, approximate incompressible flow with thick solid lines and exact flow with thin solid lines); median computer run time is over 25 Monte Carlo runs on a 2 GHz PC using MATLAB coded by the same person (Jim Huang) for all four filters; the MATLAB was carefully designed to run fast in all cases (e.g., by avoiding loops and global variables and by using other coding tricks), but extreme care was given to the classic PF code, which would otherwise take forever to run; hence, this is an apples & apples comparison.**

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 07 Feb 2023
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use