The GONG Particle Filter: A Novel Framework for Data Assimilation in Stochastic PDEs

Colin Cotter, Dan Crisan, Maneesh Kumar Singh*
Department of Mathematics
Imperial College London
United Kingdom.
maneesh-kumar.singh@imperial.ac.uk

Abstract

This presentation will explore a data assimilation method using an advanced particle filtering framework that combines tempering, jittering, and global nudging. The entire filtering process is executed with the power of ensemble parallelism, ensuring efficient and scalable computations. We conduct extensive numerical experiments to examine the impact of different filtering procedures on the performance of the data assimilation process. Our analysis focuses on how observational data and the data assimilation step influence the stability of the obtained results.

1 Introduction

Data Assimilation (DA) is a process of estimating the state of a dynamical system by combining observation data with prior information on the state (often from a numerical model forecast). This technique is used to combine observational data with numerical models to increase forecast and simulation accuracy in a variety of fields, including meteorology, oceanography, and environmental research Asch et al. [2016], Van Leeuwen et al. [2019]. In the context of the filtering problem, data assimilation for stochastic systems can be rigorously stated as stochastic filtering. The computation of the signal's law given a sequence of collected observations is the crux of the nonlinear filtering problem. See Bain and Crisan [2009] and its references for further information on stochastic filtering.

In high dimensions, particle filtering usually needs to be adapted to overcome the dimensionality curse, which otherwise results in a loss of particle variety. In this work, we carry on the recent trend of combining approaches to prevent this loss of diversity by jittering, tempering, and nudging. This work explores the possibilities of global nudging the particle filter. Identical twin experiments with a stochastic incompressible Euler flow are used to illustrate the effect on the entire solution. Our experiments also show the emerging capabilities of our parallel library, based on the Firedrake automated code generation system, for particle filtering with SPDEs.

2 Model and numerics

Consider the stochastic incompressible Euler flow in $\Omega = (0,1)^2$:

$$\begin{aligned} &dq + (\vec{u}dt)\cdot\nabla q + dU = (Q - rq)dt \\ &\Delta\psi = q, \quad \psi = 0 \text{ on } \partial\Omega \\ &\vec{u} = \nabla^{\perp}\psi \end{aligned} \tag{1}$$

with zero boundary conditions and source term $Q = 0.5\sin(8\pi x)$. The additive noise dU satisfies

$$(1-\nabla^2)^k dU = \eta dW,$$

where $dW(x,t)$ is a space-time white noise.

*Corresponding author: Maneesh Kumar Singh

1st Conference on Applied AI and Scientific Machine Learning (CASML 2024).

2.1 Numerical Scheme

Mixed FEM ($W_h = Q_h \times V_h$) for spatial discretization

$$Q_h = \{p_h \in W^1(\Omega) | p_h \in C(\Omega), p_h|_K \in \mathbb{P}^1(K)\},$$
$$V_h = \{\phi_h \in H^1(\Omega) | \phi_h = 0 \text{ on } \partial\Omega\}.$$

The DG method for the hyperbolic vorticity equation

$$(q_t, p_h) - (\vec{u}q, \nabla p_h) + (\vec{u} \cdot \hat{n} q, p_h)_{\partial K} = (Q - rq, p_h).$$

and for the elliptic equation of the stream function

$$(\nabla \psi, \nabla \phi_h) = (q, \phi_h).$$

We have used the implicit midpoint scheme for the time discretization, *i.e.*, $q_t \approx q^{n+1/2} \equiv (q^n + q^{n+1})/2$. Therefore, the fully discrete scheme is expressed as follows:

$$(q^{n+1} - q^n, p_h) + \Delta t(rq^{n+1} - Q, p_h) - \Delta t(q^{n+1/2} \nabla^\perp \psi_h, \nabla p_h)$$
$$+\Delta t \Big((\vec{u} \cdot \hat{n})^+ q^{n+1/2,+} - (\vec{u} \cdot \hat{n})^- q^{n+1/2,-}, [[p_h]] \Big) + (\nabla \psi^{n+1}, \nabla \phi_h) + (q^{n+1}, \phi_h)$$
$$+\Delta t \sqrt{\Delta t}(dU, p_h) = 0$$

3 Data assimilation methods

In this section, we will offer the basic foundation of stochastic filtering. Let X and Y be two processes defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The process X is usually called the signal process or the "truth", with range in a specified function space V in the SPDE case (approximated by a finite element space in this work) and Y is the observation process, with range $\mathbb{R}^M$. The nonlinear filtering problem: find the best approximation of the posterior distribution of the signal X_t, denoted by π_t given the observations $Y_1, Y_2, \ldots, Y_t$. In our context, the observations consist of noisy measurements of the true state recorded at discrete times and they are taken at discrete locations.

In this work, we discuss the approximation of the posterior distribution of the signal by *particle filters*. These sequential Monte Carlo methods generate approximations of the posterior distribution using sets of *particles*, which represent samples from the conditional distribution of X. Particle filters are employed to make inferences about the signal process. This involves utilizing Bayes' theorem, considering the time-evolution induced by the signal X_t, and taking into account the observation process Y_t. The observation data Y_t consists of noisy measurements of the the stream function ψ taken at a point belonging to the data grid $\mathcal{G}_d$,

$$Y_t := \mathcal{P}_d^s(X_t) + V_t, \tag{2}$$

where $V_t \sim \mathcal{N}(\mathbf{0}, I_\sigma)$. While we assumed standard normal distributions for V_t, the methodology presented is valid for any observation likelihood with a computable pdf. The ensemble of particles evolved between assimilation times according to the law of the signal.

3.1 Particle Filters: Basic terminology

In this subsection, we briefly discuss the basics of particle filters, so that we can present our results in context. We mostly describe the methodology; more details on why it works can be found in the references Pons Llopis et al. [2018], Cotter et al. [2020b,a]. We briefly state the various particle filters used in this work:

- **Bootstrap Particle Filter (Sampling importance resampling):**
  - Given an initial distribution of particles, each particle is propagated forward according to SPDEs.
  - Based on partial observations, weights of new particles and ESS are computed
  - if the ESS drops below the critical value M^*, the particles are resampled.
- **Tempering:**

- To artificially flatten the weights by rescaling the likelihood function by a factor $\theta \in (0, 1]$. This gives a much more diverse ensemble as the ESS will have more reasonable values

- **Jittering (Metropolis-Hastings MCMC):**
  - To improve the diversity of the ensemble by computing new particles that have been duplicated during resampling.

3.2 Gloabl nudging

In the nudging particle filter framework, a time-dependent control variable directs particles toward observation-informed regions, while maintaining consistency with the prior distribution through adjusted weights.

Mathematical description: Consider an SDE of the form

$$\mathrm{d}x = f(x)\mathrm{d}t + G(x)\mathrm{d}W,\, x(0) = x_0.$$

The perturbed SDE:

$$d\hat{x} = f(\hat{x})dt + G(\hat{x})(\lambda(t)dt + dW),\, \hat{x}(0) = x_0.$$

In the discrete-time setting, to account for the perturbation, the particles will have new weights according to Girsanov's theorem, given by

$$w_i = \exp(-\Phi_i), \quad \Phi_i = \|y - h(x_i^N)\|_\Sigma^2 + \sum_{n=1}^{N} \left(\frac{\Delta t}{2} (\lambda_i^n + \Delta\lambda_i)^2 - (\lambda_i^n + \Delta\lambda_i)\Delta W_i^n \right),$$

(simplified to the case where all data is measured at time T). Here h is the observation operator and $\| \cdot \|_\Sigma$ is the appropriately weighted norm using the observation error covariance Σ.

Globally nudging algorithm:

1. Set $\Delta W_i^n = 0, \quad \lambda_i^n = 0$ for $n = 1, \ldots, N$, $i = 1, \ldots, M$.
2. Set $n = 1$
3. Set $\Delta W_i^n \sim \mathcal{N}(0, \Delta t)$ random variable, independently for each $i = 1, \ldots, M$.
4. Find $(\Delta\lambda_1, \Delta\lambda_2, \ldots, \Delta\lambda_M)$ that jointly minimise

$$\sigma \sum_{i=1}^{M} \Phi_i - \log \left(\frac{\left(\sum_{i=1}^{M} w_i\right)^2}{\sum_{i=1}^{M} w_i^2} \right), \quad w_i = \exp(-\Phi_i),$$

5. For $n \leq m \leq N$ and $1 \leq i \leq M$, replace $\lambda_i^m \mapsto \lambda_i^m + \Delta\lambda_i$
6. if $n < N$, replace $n \mapsto n + 1$ and return to step 3.

The first term of the functional drives the ensemble towards the data and the second term stops the nearest particles from getting too close. In general, these optimisation problems are nonlinear since the observations at τ_n depend on the entire history of $\mathrm{d}\,W$ from $t = \tau_{n-1}$ to τ_n through the nonlinear SDE. This algorithm is a significant update from Cotter et al. [2024], where each Φ_i was optimized separately for each time step in every assimilation window. The optimization problem is solved by utilizing PETSc's TAO optimization algorithm (Unconstrained Minimization) and particularly the Limited-Memory Variable-Metric Method (LMVM) is implemented. In our implementation, this is automated using Firedrake et al. [2023]. After the nudging step, tempering and jittering may still be needed, requiring the θ_k-adjusted weight formulas to replace W with $\tilde{W}$, as in Cotter et al. [2020a].

3.2.1 Ensemble parallelism

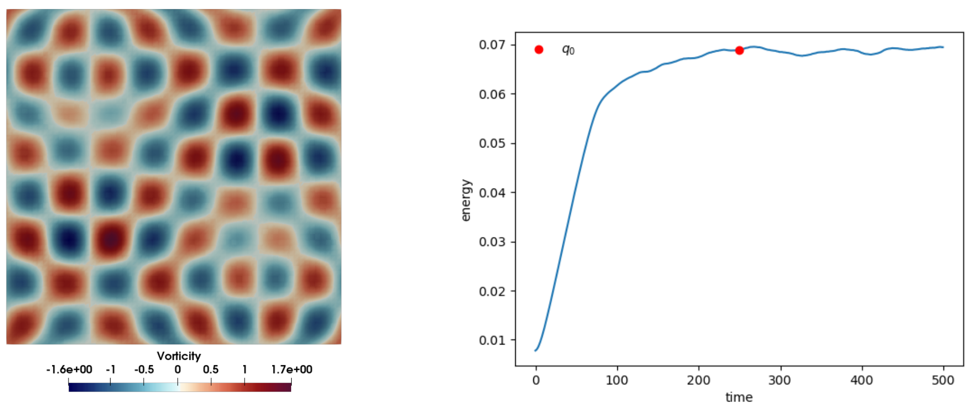
Our Firedrake implementation merges spatial domain decomposition with ensemble parallelism. The algorithms allow independent calculations for each particle, except when states and noise increments are copied during resampling. Weight normalization requires minimal inter-particle communication. Ensemble parallelism divides particles into batches for independent calculations, followed by updates from copies across batches.

4 Numerical experiment

Numerical setup: The domain $\Omega = (0,1)^2$ is discretized with cells 32×32, *i.e.*, 33×33 points. The stream function is observed on observation points $N_x = 81$. The assimilation steps $\{\tau_k\}_{k=1}^{50}$ with $\Delta\tau_k = 5\Delta t$ and time steps $\Delta t = 0.025$. All observation processes are perturbed with iid $\mathcal{N}(0, 0.005)$ measurement errors. The total number of particles is 60. The initial condition for the vorticity is

$$\begin{aligned} q_{spin} = \quad & \sin(8\pi x) sin(8\pi y) + 0.4\cos(6\pi x)\cos(6\pi y) + 0.3\cos(10\pi x)\cos((4\pi y) \\ & +0.02\sin(2\pi x) + 0.02\sin(2\pi y), \end{aligned}$$

from which we spin up the system until an energy equilibrium state seems to have been reached. This equilibrium state, denoted by $q_{initial}$, is then chosen as the initial condition for our numerical experiments. The system reaches an approximate energy equilibrium state after 250-time units, also confirmed by Fig 1. The system reaches an approximate energy equilibrium state after 250-time units,



(a) Initial configuration q_{spin}

(b) Plot of Energy time series

Figure 1: Plot of the chosen initial configuration and energy equilibrium at $t_{initial} = 250$.

and the solution at this equilibrium point is set to be the initial condition $q_{initial}$ from which we start our numerical experiments. The vorticity and the stream function at the initial time $t_{initial} = 250$ is plotted in Fig. 2. We use the initial condition to obtain an ensemble which contains particles that are

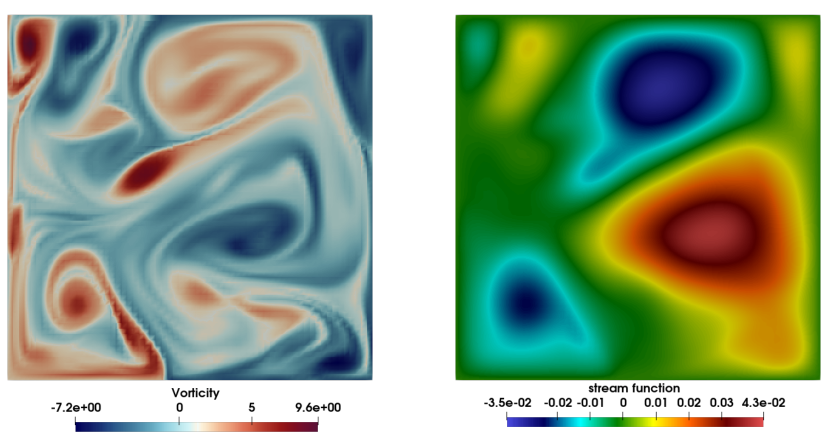
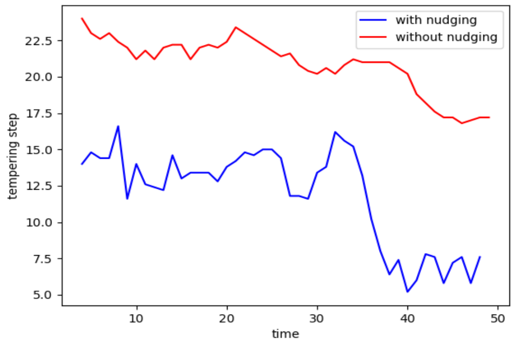


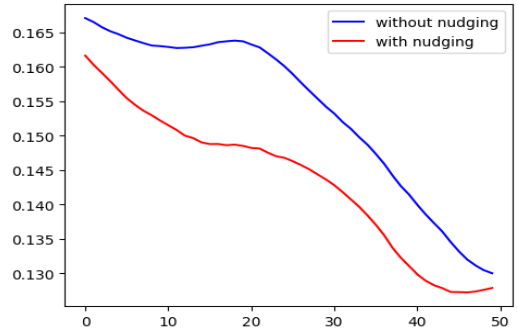
Figure 2: Plot of the numerical PDE solution at the initial time $t_{initial} = 250$.

reasonably *close* to the truth. For the validation of various filtering procedures, we use the ensemble mean l_2-norm relative error (EMRE), which is defined as follows,

$$EMRE(u^a, u^p) := \frac{1}{N_p}\sum_{n=1}^{N_p} \frac{\|u^a - u_n^p\|_2}{\|u^a\|_2},$$



(a) Tempering steps of nudging vs without nudging filters



(b) Evolution of the EMRE of nudging vs without nudging filter

Figure 3: Comparison of nudging vs without nudging particle filter

From Fig. 3, we observe that the particle filter is dealing well with this filtering problem. Also, we can confirm that the proposed global nudging filter is performing better in every perspective than the traditional filters.

References

M. Asch, M. Bocquet, and M. Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

A. Bain and D. Crisan. *Fundamentals of stochastic filtering*, volume 3. Springer, 2009.

C. J. Cotter, D. Crisan, D. D. Holm, W. Pan, and I. Shevchenko. Data assimilation for a quasi-geostrophic model with circulation-preserving stochastic transport noise. *Journal of Statistical Physics*, 179(5-6):1186–1221, 2020a.

C. J. Cotter, D. Crisan, D. D. Holm, W. Pan, and I. Shevchenko. A particle filter for stochastic advection by Lie transport: A case study for the damped and forced incompressible two-dimensional Euler equation. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1446–1492, 2020b.

Colin John Cotter, Dan Crisan, and Maneesh Kumar Singh. Data assimilation for the stochastic camassa-holm equation using particle filtering: a numerical investigation. *arXiv preprint arXiv:2402.06927*, 2024.

Ham D. A. et al. *Firedrake User Manual*. Imperial College London and University of Oxford and Baylor University and University of Washington, first edition, 5 2023.

F. Pons Llopis, N. Kantas, A. Beskos, and A. Jasra. Particle filtering for stochastic Navier–Stokes signal observed with linear additive noise. *SIAM Journal on Scientific Computing*, 40(3):A1544–A1565, 2018.

P. J. Van Leeuwen, H. R. Künsch, L. Nerger, R. Potthast, and S. Reich. Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723): 2335–2365, 2019.