Sequential Monte Carlo Methods for High-Dimensional Inverse Problems

Nikolas Kantas

Imperial College

joint work with A. Beskos (UCL), A. Jasra, A. Thiery (NUS) and D. Crisan (Imperial)
Inverse problems ubiquitous in many applications:
  meteorology, atmospheric & oceanic sciences, seismology, petroleum engineering, finance, medical imaging.....
  very often the state space is very high-dimensional

This talk:
  can we develop generic efficient and accurate and particle methods?

Case study for 2D Navier Stokes equations:
  what is the initial condition given some noisy measurements of the velocity field?
  to some degree relevant to numerical weather forecasting
Outline

- Motivation & Problem formulation:
  - Bayesian inverse problems on Hilbert spaces
  - the case study with Navier-Stokes
- Monte Carlo methods for problems in high dimensions
  - Markov Chain Monte Carlo
  - Sequential Monte Carlo
- Numerical examples
- Discussion and current work (on high dimensional filtering)
Motivation: inference for a 2D velocity vector field

Let now the state (or phase) space be a torus

\[ \mathbb{T} = [0, 2\pi) \times [0, 2\pi) \]

We are interested in the space and time evolution of a velocity vector field \( v : \mathbb{T} \times [0, \infty) \rightarrow \mathbb{R}^2 \) given by:

\[ v(x, t) = G_t \left( u \right)(x), \quad v(x, 0) = u(x), \quad x \in \mathbb{T}. \]

What is the initial condition \( u \) given some noisy measurements of the velocity field?

We will specify \( G_t \) with the deterministic 2D Navier-Stokes equations.
Dynamics: the Navier-Stokes equations

- Evolution of the velocity \( v = (v_1(x, t), v_2(x, t))' \) of an incompressible fluid defined on a 2D torus, \( x \in \mathbb{T} \):

\[
\begin{align*}
\partial_t v - \nu \Delta v + (v \cdot \nabla) v + \nabla p &= f \\
\nabla \cdot v &= 0, \quad \int_{\mathbb{T}} v_i(x, \cdot) dx = 0, \; i = 1, 2 \\
v(x, 0) &= u(x)
\end{align*}
\]

under periodic boundary conditions.

- \( \nu \) viscosity parameter, \( p \) pressure, \( f \) exogenous forcing.

- Advection \( (v \cdot \nabla) v = (v_1 \partial_{x_1} v_1 + v_2 \partial_{x_2} v_1, v_1 \partial_{x_1} v_2 + v_2 \partial_{x_2} v_2)' \)

- Compute numerically a weak solution using discrete Fourier transforms (FFT) on a 64\(^2\) mesh.
Likelihood

- **Eulerian data assimilation**: take measurements on a fixed grid of points, \( x_{1:M} \), at regular time intervals as before

\[
Y_{l,m} = \nu(x_m, l\delta) + \gamma V_{l,m}
\]

where \( V_{l,m} \) i.i.d zero mean standard normal.

- Likelihood

\[
I(y, u) = \prod_{l=1}^{T} \prod_{m=1}^{M} \exp \left( -\frac{1}{2\gamma^2} (y_{l,m} - G_{l\delta}(u)(x_m))^2 \right)
\]

\[
= \prod_{l=1}^{T} I_l(y_l, u)
\]
Challenges

- High dimensional state space
  - but fortunately low dimensional observations.
- Expensive to compute likelihood (requires PDE solver)
- Challenging scenario to apply particle methods without stochastic dynamics.
- Computational approach should be robust to mesh refinement.
  - there is a benchmark of Markov Chain Monte Carlo as in Law & Stuart 13, Iglesias, Law & Stuart 13,...
Introduction to inverse problems

- Let \((U, \| \cdot \|_U)\) and \((Y, \| \cdot \|_Y)\) be given normed vector spaces.
- Inverse problem: find some unknown \(u \in \mathcal{U}\) that generates data \(y \in \mathcal{Y}\):
  \[ y = G(u) \]
  where \(G\) is a complicated observation operator.
- There are many possible choices for \(u, y\) and \(G\):
  - simplest case: real vector and matrices,
  - in case study \(\mathcal{U}\) will be a Hilbert space (Stuart 10, Cotter et. al. 12,....)
  - setup can also include stochastic processes & noisy observations
Introduction to inverse problems

- Direct inversion most of the times not possible.
- Least squares: find instead
  \[ u^* = \arg \min_{u \in U} \left\| \Gamma^{1/2} (y - G(u)) \right\|_Y^2 \]
- Inverse problem now like:
  \[ y = G(u) + \mathcal{N}(0, \Gamma) \]
- \( \Gamma \) appropriate operators on \( Y \) respectively
Introduction to inverse problems

- Often problem underdetermined and least squares are *ill-posed*
- Remedy: include a *Tikhonov-Phillips regularization* term

\[ u^* = \arg \min_{u \in \mathcal{U}} \left( \left\| \Gamma^{1/2} (y - G(u)) \right\|^2_Y + \left\| C^{1/2} (u - u_0) \right\|^2_U \right) \]

- \( \Gamma, C \) appropriate operators on \( \mathcal{Y}, \mathcal{U} \) respectively
- Would like to quantify uncertainty around \( u^* \)
Bayesian interpretation using Gaussian processes

- Construct a posterior probability measure $\mu$ on $U$
  \[
  \frac{d\mu}{d\mu_0}(u) \propto l(y, u)
  \]

- Prior: Gaussian reference measure as prior
  \[
  \mu_0 = \mathcal{N}(u_0, C)
  \]

implied by regularisation

- Likelihood
  \[
  l(y, u) = \exp\left(-\frac{1}{2} \left\| \Gamma^{1/2} (y - G(u)) \right\|_{Y}^{2}\right)
  \]

References:
Diaconis 88, Rue & Held 05, Kaipio & Somersalo 05, Stuart 10, Cotter et.al. 12
Bayesian inference for inverse problems on Hilbert spaces

- $u$ is a 2D vector field
- Assume $u$ belongs to a Hilbert space $H$, $u \in L^2(\mathbb{T})$:
  - can extend Bayesian methodology for these spaces
  - preserves well-posedness, regularity, etc.
- Need to specify a basis for $H$.
  - Since we have periodic boundary conditions can use spectral method
  - use the eigenfunctions of the laplacian as basis of $H$
  - for Gaussian priors can use Karhunen-Loeve expansions
Specifying the basis of the space

- Look at incompressible functions $\nabla \cdot u = 0$ on Hilbert space spanned by orthonormal basis:

$$\psi_k(x) = \frac{k^\perp}{2\pi|k|} \exp(ik \cdot x)$$

where $k \in \mathbb{Z}^2 \setminus \{0\}$, $k^\perp = (-k_2, k_1)'$.

- Fourier decomposition of $u$:

$$u(x) = \sum_{k \in \mathbb{Z}^2 \setminus \{0\}} u_k \psi_k(x)$$

with Fourier coefficients $u_k = \langle u, \psi_k \rangle = \int_{\mathbb{T}} u \cdot \overline{\psi_k}(x) dx$. 
The prior

- Prior:
  \[ u \sim \mu_0 = \mathcal{N}(0, \beta^2 (-P \Delta)^{-\alpha}) \]

- Karhunen-Loeve: in Fourier domain the prior coefficients
  \[ \text{Re}(u_k), \text{Im}(u_k) \sim \mathcal{N}(0, \frac{1}{2} \beta^2 |k|^{-2\alpha}) \]

- \( \Delta = \partial^2_{x_1} + \partial^2_{x_2} \) is the Laplacian operator.
- \( \alpha, \beta \) affect the roughness and magnitude.

References
Stuart 10, Law & Stuart 12, Cotter et.al. 12, Iglesias, Law & Stuart 13
Some samples from the prior

Figure: $\beta^2 = 5, \alpha = 2.2$. Top: velocity diagram; bottom: vorticity

\[ w = \nabla \times u = \partial_{x_1} u_2 - \partial_{x_2} u_1. \]
Computational challenges

- Recall: algorithm should be robust to mesh refinement.
  - state space is infinite-dimensional, but in practice a high-dimensional discretisation is used (i.e. $64^2$ mesh).

- Computational approaches:
  - Variational methods & Gaussian approximations, 3DVAR, 4DVAR (Sasaki (58), Le Dimet & Talagrand (86), Courtier & Talagrand (87))
  - Ensemble Kalman filter (Evensen (94))
  - Markov Chain Monte Carlo (MCMC) (Law & Stuart (12), Cotter et. al. (12))

- MCMC is considered the most accurate but is much more expensive.
Markov Chain Monte Carlo (MCMC)...

- ...so that it is robust to the mesh size:
- (Vanilla version) Run a $\mu$-invariant Markov chain $(u_n; n \geq 0)$ as follows:
  - Initialise $u_0 \sim \mu_0$. For $n \geq 1$
    - Propose:
      $$u' = \rho u_{n-1} + \sqrt{1 - \rho^2} Z, \quad Z \sim \mu_0 = \mathcal{N}(0, \beta^2 (-\Delta)^{-\alpha})$$
    - Accepted $u_n = u'$ with probability:
      $$1 \wedge \frac{l(y, u')}{l(y, u_n)}$$
      otherwise $u_n = u_{n-1}$.

References
Neal 98, Tierney 98,...,Beskos et. al. 08, Stuart 10, Cotter et.al. 12, Law 13
MCMC in practice for Navier-Stokes

(Inference for Dataset A) $\rho = 0.9997$, acceptance ratio $\approx 0.30$, Computational effort: $10^5$ iterations / day

![Graphs showing real and imaginary parts of $k=(1,2)$ with corresponding plots and axes labels.]
Sequential Monte Carlo samplers (SMC)

- We have a natural target sequence \((\mu_n)_{n=0}^T\) with \(\mu_T = \mu\), \(\mu_0\) is prior
  \[
  \mu_n \propto \mu_0 \prod_{p=1}^{n} l_p
  \]
- SMC is an algorithm sampling sequentially particles from
  \[
  \mu_n \propto \mu_0 \prod_{p=1}^{n} l_p \cdot K_p \text{ where } \mu_n = \mu_n K_n
  \]
- Goal is combine robustness of MCMC to dimension with efficiency natural in SMC.
- Note unlike particle filtering we are dealing with a target space \(\mathcal{U}\) of fixed dimension
- Framework as in (Chopin 02, Del Moral, Doucet, Jasra 06)
Sequential Monte Carlo samplers (SMC)

- (Chopin 02) Each intermediate target $\mu_n$ can be approximated sequentially using:
  - **Selection:**
    - Importance Sampling using samples/particles from $\mu_{n-1}$
    - Resampling to discard samples with low weights
  - **Mutation:**
    - MCMC steps that are $\mu_n$-invariant to jitter the particle population
    - otherwise diversity of particles might die due to successive selection.
SMC Algorithm

- At $n=0$, for $j = 1, \ldots, N$ sample $u_{0}^{j} \sim \mu_{0}$
- For $n = 1, \ldots, T$, (for each $j$)
  - Weight $W_{n}^{j} \propto \frac{d\mu_{n}}{d\mu_{n-1}}(u_{n-1}^{j})$, $j = 1, \ldots, N$, $\sum_{j=1}^{N} W_{n}^{j} = 1$
  - Resample: if $\text{ESS}_{n} = (\sum_{j=1}^{N} (W_{n}^{j})^{2})^{-1} < N_{\text{thresh}}$
    - sample offspring $j' \sim \text{Mult}(W_{n}^{1:N})$
    - set $\overline{u}_{n}^{j} = u_{n-1}^{j'}$.
    - set $W_{n}^{j} = \frac{1}{N}$
  - Mutation: $\mu_{n}$-invariant MCMC step: $u_{n}^{j} \sim \mathcal{K}_{n}(\overline{u}_{n}^{j}, \cdot)$ where $\mu_{n} \mathcal{K}_{n} = \mu_{n}$

Particle approximations $\mu_{n}^{N} = \sum_{j=1}^{N} W_{n}^{j} \delta_{u_{n}^{j}}$
SMC samplers in high dimensions

- Scheme needs some adjustments to work in high dimensions.
- Some intuition from the results in Beskos, Crisan & Jasra 11

- Stability of Importance Sampling:
  - weights should have always relatively low variance
  - $\mu_n \propto \mu_{n-1}/l_n$ needs to be close to $\mu_{n-1}$

- MCMC Mutation kernels $K_n$ need to be uniformly efficient over bridging sequence
Tempering between intermediate $\mu_n$-s

- bridge two successive targets via tempering
  
  $\mu_{n,r} = \mu_{n-1} l_n^{\phi_r}$

  with temperature sequence $0 < \phi_1 < \ldots < \phi_q \leq 1$

- Let $\{ u^j_{n,r} \}$ be the particles for $r$-th target $\mu_{n,r}$. The incremental weights at $r + 1$ are
  
  $W_{n,r+1}^j = \frac{l_n(y_n, u^j_{n,r})^{\phi_{r+1} - \phi_r}}{\sum_{p=1}^N l_n(y_n, u^p_{n,r})^{\phi_{r+1} - \phi_r}}$

- adaptive tempering:
  
  - after SMC has completed for step $r$, use the particles to compute $\phi_r^N$ so that it satisfies:

  $ESS_n(\phi_r^N) = (\sum_{j=1}^N (W_{n,r+1}^j)^2)^{-1} \approx N_{thresh}$

  - easily implemented using a bisection on $(\phi_r, 1)$, (Jasra et. al. 11, Schafer & Chopin 12, Zhou, Johansen & Aston 13)
Improving MCMC efficiency

- MCMC Mutation kernels $\mathcal{K}_n$ need to be uniformly efficient over bridging sequence
  - to re-introduce diversity lost from resampling
  - use RW moves in MCMC that look like $\mu_n$ (use the particles)

- Would like to improve on previous MCMC with RW proposal:
  \[
  \tilde{u}^j_{k,n,r} = \rho u^j_{k,n,r} + \sqrt{1 - \rho^2} \mathcal{N}(0, \frac{1}{2} \beta^2 |k|^{-2\alpha})
  \]

- Main idea: use adaptive scaling of on proposal so that it resembles $\mu_n$
Improving MCMC efficiency

- **adaptive scaling:**
  - Estimate mean and covariance using particles

\[
\hat{u}_{n,r}^N = \sum_{j=1}^{N} \mathcal{W}_{n,r}^j u_{n,r}^j, \quad \hat{\Sigma}_{n,r}^N = \sum_{j=1}^{N} \mathcal{W}_{n,r}^j (u_{n,r}^j - \hat{u}_{n,r}^N)(u_{n,r}^j - \hat{u}_{n,r}^N)'
\]

- use at iteration \( n \) a RW that uses the a Gaussian approximation to posterior:

\[
\tilde{u}_{n,r}^j = \hat{u}_{n,r}^N + \rho(u_{n,r}^j - \hat{u}_{n,r}^N) + \sqrt{1 - \rho^2} \mathcal{N}(0, \hat{\Sigma}_{n,r}^N)
\]

- acceptance ratio is same as for an independent sampler with proposal \( \mathcal{N}(\hat{u}_{n,r}^N, \hat{\Sigma}_{n,r}^N) \)
For some inverse problems with PDEs including the Navier-Stokes it also makes sense to distinguish between:

- low frequencies
  - marginal posterior shows we learn from the data
  - $|k| \leq K$ or $k_1 \lor k_2 \leq K$
  - use particle-adapted MCMC tuned to posterior

- high frequencies:
  - do not learn so much (posterior similar to prior)
  - $|k| > K$ or $k_1 \lor k_2 > K$
  - use standard proposal tuned to prior

This ensures the MCMC kernel mixing is robust to mesh refinement and suitable for high dimensions
Adaptive scaling with SMC

- For a minute ignore tempering (and high dimensions)
- Adaptive scaling means at each time we are actually targeting a ideal sequence

$$\mu_n \propto \mu_0 \prod_{p=1}^{n} l_p \cdot K_{p, \mu_n(\zeta)} \text{ where } \mu_n = \mu_n K_{p, \mu_n(\zeta)}$$

- $\mu_n(\zeta)$ is covariance used in the proposal of MCMC
- We are implementing instead

$$\mu_n \propto \mu_0 \prod_{p=1}^{n} l_p \cdot K_{p, \mu^N_n(\zeta)}$$

- $\mu^N_n(\zeta)$ is particle approximation of covariance
Adaptive tempering with SMC

- Now for a minute ignore adaptive scaling

- At each $n$, adaptive tempering means we are targeting

$$\mu_{n,r} \propto \mu_{n-1} \prod_{p=1}^{\tau} l_n^{\phi_{n,p} - \phi_{n,p-1}} \cdot \mathcal{K}_{p,r} \text{ where } \mu_{n,r} = \mu_{n,r} \mathcal{K}_{p,r}$$

assuming there is a sequence $(\phi_{n,r})_{r=1}^{\tau}$ with $\phi_{\tau} = 1$ and

$$\phi_{n,r+1} = \inf \{ \phi : \phi > \phi_r, : \text{ESS}(\mu_{n,r}, \phi) = N_{thresh} \}$$

- Adaptive SMC targets

$$\mu_{n,r} \propto \mu_{n-1} \prod_{p=1}^{\tau^{N}} l_n^{\phi_{n,p}^{N} - \phi_{n,p-1}^{N}} \cdot \mathcal{K}_{p,r}$$

- $\tau^{N}, (\phi_{n,r}^{N})_{r=1}^{\tau^{N}}$ are particle approximations
Some recent theoretical results on adaptive SMC

- (Giraud and Del Moral 12) Non-asymptotic (in \( N \)) results for adaptive tempering
  - \( L^p \) bounds, concentration inequalities, annealing properties
  - assume perfect scaling for mutations

- (Beskos, Jasra, K. & Thiery 14) Asymptotic results in \( N \)
  - treat separately adaptive scaling and adaptive tempering
  - weak Law of Large numbers, CLT

- Adaptive tempering algorithm behaves as \( N \) increases closer to “ideal” non-adaptive SMC
  - asymptotic variance is the same
  - MCMC kernels \( K_{p,r} \) preserve target always
SMC in practice

<table>
<thead>
<tr>
<th>Data-set A</th>
<th>( \delta = 0.02 ), ( \Upsilon = 16 ), ( T = 5 )</th>
<th>( u^\dagger \sim \mathcal{N}(0, 5A^{-2.2}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-set B</td>
<td>( \delta = 0.2 ), ( \Upsilon = 4 ), ( T = 20 )</td>
<td>( u^\dagger \sim \mathcal{N}(0, A^{-2}) )</td>
</tr>
</tbody>
</table>

Table: The true initial condition \( u^\dagger \) was sampled from the prior and \( \nu = \frac{1}{50} \), \( f(x) = \nabla^\perp \cos ((5, 5)' \cdot x) \). Data-set A is a scenario of a short-time data-set with a dense observation grid; data-set B is a scenario of a long data-set with a sparse observation grid.
### SMC in practice

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Algorithmic</th>
<th>number of PDE solver calls (over $T$)</th>
<th>execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MCMC A</strong></td>
<td>$\rho = 0.9999$</td>
<td>$0.9 \times 10^6$</td>
<td>9 days</td>
</tr>
<tr>
<td><strong>SMC A</strong></td>
<td>$N = 500$</td>
<td>$7.266 \times 10^5$</td>
<td>3 days</td>
</tr>
<tr>
<td></td>
<td>$M = 20$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SMC A with parallel</strong></td>
<td>$N = 1020$</td>
<td>$1.403 \times 10^6$</td>
<td>7.4 hours</td>
</tr>
<tr>
<td></td>
<td>$M = 20$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SMC B with parallel</strong></td>
<td>$N = 1020$</td>
<td>$1.447 \times 10^6$</td>
<td>3.5 days</td>
</tr>
<tr>
<td></td>
<td>$M = 20$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table:* For SMC we use $\rho_L = 0.99$, $\rho_H = 0.991$, $K = 7$. For simulations we use Matlab and for the parallel implementation of SMC we used 60 workers on the cluster of CSML-UCL.
Posterior means for data-set (A)

Figure: left SMC, middle true, right MCMC
A: PDFs $k = (0, 1), (1, 1), (2, 1), (4, 4), (9, 9)$
Scatter plots (A)

Figure: (re-scaled) vorticity Fourier coefficients at frequencies $k = (1,1), (2,1), (4,4)$.
A: Posterior mean using smoother prior $\mu_0 = \mathcal{N}(0, 5A^{-2.5})$
Posterior mean for data-set B
B: PDFs $k = (0, 1), (1, 1), (2, 1), (4, 4), (9, 9)$

Figure:
Data-set B: Blue lines are estimated posterior PDFs for $v_{\text{ort}}(u_k)$ | $k$ | $a$
(solid SMC, dotted MCMC.) Black line is prior.
B: Scatter Plot

Figure: (re-scaled) Fourier coefficients at frequencies $k = (1,1), (2,1), (4,4)$. 

\begin{align*}
\text{Re}(\xi_{(1,1),T}) & \quad \text{Im}(\xi_{(1,1),T}) \\
\text{Re}(\xi_{(2,1),T}) & \quad \text{Im}(\xi_{(2,1),T}) \\
\text{Re}(\xi_{(4,4),T}) & \quad \text{Im}(\xi_{(4,4),T})
\end{align*}
B: Checking low/high frequencies

Figure:

Data-set B
Discussion

Sources of efficiency

► distinguishing behaviour between high/low frequencies provides a useful dimension reduction

► methods compares very favourably to MCMC in cases when earlier parts of the data are more informative
  ► after assimilating few data-points the posterior does not change much/quickly
  ► less tempering needed and hence less PDE solver calls

► massive parallelisation over particles useful (Lee et. al. 10, Murray, Lee, Jacob, Del Moral 13)
Discussion

Some drawbacks:

- Described algorithm has computational cost $O(NT^2)$
  - as it is it is not useful for on-line applications with very long data-sets.
  - this is due to using a state with deterministic dynamics.
- Could make computational cost $O(NTL)$ if receding horizon truncations are used
  - use this methods in segments of data of size $L$
  - each time plug in a smoothed version of the current particle approximation as new prior.
  - this is biased! but possibly this bias can be controlled.
Extensions:

- Parameter inference: could also be used to infer \( \nu \), forcing \( f \), jointly with \( u \) in the spirit of (Cotter, Dashti & Stuart 12)
- Can use for different PDE, boundary conditions and basis functions
- Method suitable for other priors (e.g. Sieve or Besov priors) or likelihoods (Langrangian onbservations)
- Ideas relevant to non-linear filtering
Conclusions

► SMC methodology is generic for some class of inverse problems.

► Still challenging task to tackle problems with higher effective dimension

  ▶ here $u$ is a high-dimensional target but data provides information for a relatively small part of the components.

► Current work filtering for particular SPDEs

  ▶ interesting questions on how these ideas compare or could be used together with ideas from Rebeschini & Van Handel 12, Van Leeuwen 09,10, Chorin et. al 10, Reich 12,14, and recent space time particle filters
References

This talk:


Relevant material:

References: background material

SMC samplers

- Chopin (02), *A sequential particle filter method for static models*, Biometrika.

MCMC for Hilbert spaces

- Cotter, Roberts, Stuart & White (12), *MCMC methods for functions: modifying old algorithms to make them faster*, submitted.