Towards a unified view of uncertainty quantification methods for full waveform inversion

Session: FWI - Imaging-2 - Theory and multi-parameter

P. Marchner[1], R. Brossier[1], L. Métivier[1,2]

Thursday 5th June, 2025

[1]ISTerre, Univ. Grenoble Alpes, France

[1,2]LJK, CNRS, Univ. Grenoble Alpes, France

Outline

Outline

An ill-posed inverse problem

Standard problem : L^2-minimization (Virieux et al. (2017))

$$\min_m C(m) = \frac{1}{2} \sum_{sources} \|d_{cal}[m] - d_{obs}\|_{L^2}^2$$

- d_{obs} : observed traces at receivers
- $d_{cal}[m] := \mathcal{F}(m)$: simulated traces, wave equation

Numerical resolution : gradient-based descent

$$m_{k+1} = m_k + \tau_k \delta m_k, \quad k \in \{1, \ldots, N_{iter}\}$$

- Descent direction δm_k, step size τ_k
- PDE-based method
  $\rightarrow$ large computational cost, $m(\mathbf{x}) \in \mathbb{R}^d$, $d \gg 1$

An ill-posed inverse problem

Standard problem : L^2-minimization (Virieux et al. (2017))

$$\min_m C(m) = \frac{1}{2} \sum_{sources} \| d_{\text{cal}}[m] - d_{\text{obs}} \|_{L^2}^2$$

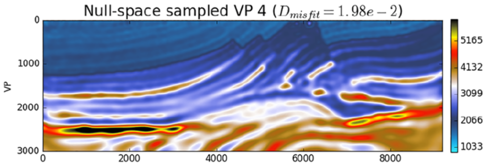
- d_{obs} : observed traces at receivers
- $d_{\text{cal}}[m] := \mathcal{F}(m)$: simulated traces, wave equation

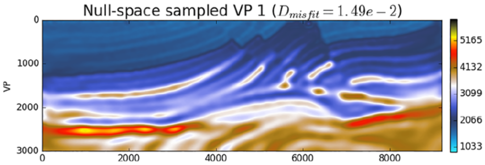
Numerical resolution : gradient-based descent

$$m_{k+1} = m_k + \tau_k \delta m_k, \quad k \in \{1, \ldots, N_{\text{iter}}\}$$

- Descent direction δm_k, step size τ_k
- PDE-based method
  $\rightarrow$ large computational cost, $m(x) \in \mathbb{R}^d$, $d \gg 1$

However, FWI has non-unique solutions !





Elastic Marmousi : inverted P-velocity maps within a 1% cost tolerance (Liu and Peter (2020))

An ill-posed inverse problem

Standard problem : L^2-minimization (Virieux et al. (2017))

$$\min_m C(m) = \frac{1}{2} \sum_{sources} \|d_{cal}[m] - d_{obs}\|_{L^2}^2$$

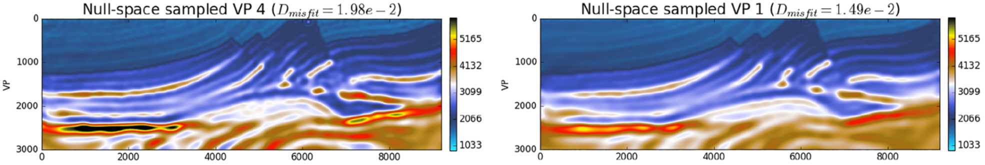
- d_{obs} : observed traces at receivers
- $d_{cal}[m] := \mathcal{F}(m)$: simulated traces, wave equation

Numerical resolution : gradient-based descent

$$m_{k+1} = m_k + \tau_k \delta m_k, \quad k \in \{1, \ldots, N_{iter}\}$$

- Descent direction δm_k, step size τ_k
- PDE-based method
  $\rightarrow$ large computational cost, $m(x) \in \mathbb{R}^d$, $d \gg 1$

However, FWI has non-unique solutions !



Elastic Marmousi : inverted P-velocity maps within a 1% cost tolerance (Liu and Peter (2020))

Uncertainty quantification aims to quantify our confidence in the reconstructed model

Outline

The Bayesian approach

Inverse problem formulation : we look for a model $m(\boldsymbol{x}) \in \mathbb{R}^d$ that satisfies, under Gaussian additive noise

$$\boxed{d_{\text{obs}} = d_{\text{cal}}[m] + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Sigma)}$$

- $d_{\text{cal}}[m] := \mathcal{F}(m)$ linked through the parameter-to-observations map $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^{n_{\text{obs}}}$

The Bayesian approach consists in seeking a posterior **probability distribution** (Kaipio and Somersalo (2006))

The Bayesian approach

Inverse problem formulation : we look for a model $m(\boldsymbol{x}) \in \mathbb{R}^d$ that satisfies, under Gaussian additive noise

$$d_{\text{obs}} = d_{\text{cal}}[m] + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Sigma)$$

- $d_{\text{cal}}[m] := \mathcal{F}(m)$ linked through the parameter-to-observations map $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^{n_{\text{obs}}}$

The Bayesian approach consists in seeking a posterior **probability distribution** (Kaipio and Somersalo (2006))

Bayes theorem for inverse problems

$$\pi_{\text{post}}(m) := \pi(m|d_{\text{obs}}) = \frac{\pi(d_{\text{obs}}|m)\pi_{\text{prior}}(m)}{\pi(d_{\text{obs}})}, \quad \pi(d_{\text{obs}}|m) \propto \exp(-C(m))$$

Given a prior $\pi_{\text{prior}}(m)$, how to infer on $\pi_{\text{post}}(m)$?

Inverse problem formulation : we look for a model $m(\mathbf{x}) \in \mathbb{R}^d$ that satisfies, under Gaussian additive noise

$$d_{obs} = d_{cal}[m] + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \Sigma)$$

- $d_{cal}[m] := \mathcal{F}(m)$ linked through the parameter-to-observations map $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^{n_{obs}}$

The Bayesian approach consists in seeking a posterior **probability distribution** (Kaipio and Somersalo (2006))

Bayes theorem for inverse problems

$$\pi_{post}(m) := \pi(m|d_{obs}) = \frac{\pi(d_{obs}|m)\pi_{prior}(m)}{\pi(d_{obs})}, \quad \pi(d_{obs}|m) \propto \exp(-C(m))$$

Given a prior $\pi_{prior}(m)$, how to infer on $\pi_{post}(m)$?

For a Gaussian prior $\pi_{prior}(m) \sim \mathcal{N}(m_0, \Sigma_0)$,

$$\pi_{post}(m) \propto \exp\left(-\frac{1}{2}\sum_{sources} \|d_{cal}[m] - d_{obs}\|^2_{\Sigma} - \frac{1}{2}\|m - m_0\|^2_{\Sigma_0}\right)$$

If $\mathcal{F}$ is linear, the posterior is also Gaussian with explicitly known mean and covariance

Problem : for FWI, $\mathcal{F}$ is nonlinear, and the parameter space $d \gg 1$ is very large

Outline

1D wave equation in $\Omega = (0, D) \times (0, T]$

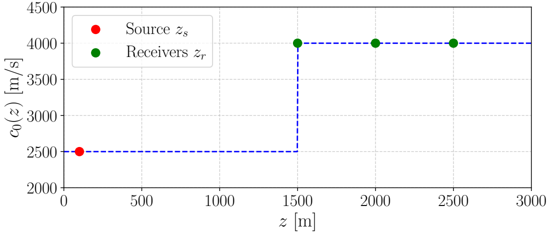
$$\frac{1}{c_0^2(z)} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial z^2} = s(z_s, t) \text{ in } \Omega$$

$$\frac{\partial u}{\partial t} \pm c_0(z) \frac{\partial u}{\partial z} = 0, \text{ at } z = \{0, D\} \quad \text{(absorbing BC)}$$

Setup

- Ricker wavelet source at $f_0 = 5$ Hz
- explicit, 2nd order finite differences
- zero initial conditions
- $m(\boldsymbol{x}) := c_0(z) \in \mathbb{R}^d$

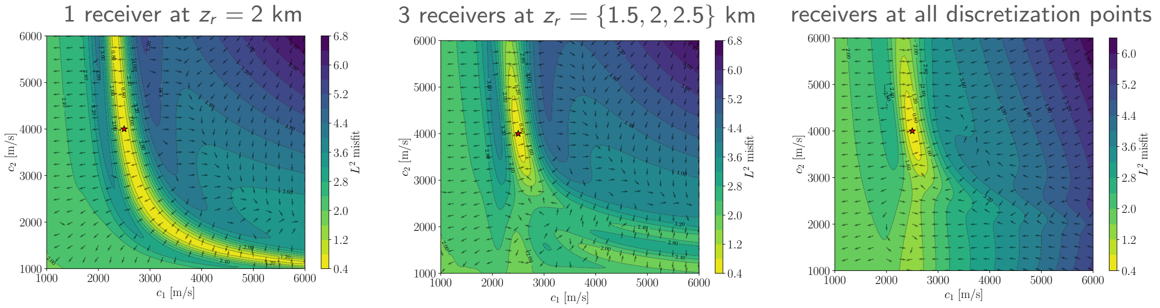
model $c_0(z) = (c_1, c_2)$



Bayesian inference with 2 parameters

- artificial observations d_{obs} with SNR ≈ 10
- play on the number of receivers z_r
- uniform prior $\pi_{\text{prior}}(m) = 1$
- we want to characterize the posterior $\pi_{\text{post}}(m) \propto \exp(-C(m))$

We fix the reference model at $c_0^*(z) = (2500, 4000)$ m/s



Strong nonlinearity

- the gradient rarely points to c_0^*
- receivers have a strong influence on the cost function

What do we want to infer ?

- the sensitivity around c_0^* ? (Hessian probing)
- identify local minima ?
- the entire shape of the nullspace ?

Outline

The deterministic gradient descent $m_{k+1} = m_k - \tau_k \nabla C(m_k)$ is an explicit Euler discretization

Gradient flow - Euclidean space $\mathbb{R}^d$

$$\frac{dm}{dt} = -\nabla C(m), \quad m(0) = m_0,$$

with time step τ_k.

The deterministic gradient descent $m_{k+1} = m_k - \tau_k \nabla C(m_k)$ is an explicit Euler discretization

Gradient flow - Euclidean space $\mathbb{R}^d$

$$\frac{dm}{dt} = -\nabla C(m), \quad m(0) = m_0,$$

with time step τ_k. Let $\{M_t\}_{t\geq 0}$ be a stochastic process with probability law μ_t

Gradient flow - Probability space $\mathcal{P}_2(\mathbb{R}^d)$

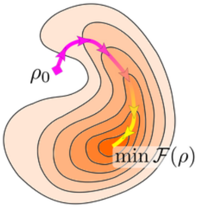
$$dM_t = -\nabla C(M_t)dt, \quad M_t \sim \mu_t$$

The density μ_t satisfies a *mass conservation* equation $\rightarrow$ continuous flow

$$\frac{\partial \mu_t}{\partial t} = \mathrm{div}(\mu_t \nabla C), \quad \boldsymbol{v}_t = -\nabla C$$

We get a gradient flow in probability space under the Wasserstein metric

$$\frac{\partial \mu_t}{\partial t} = \mathrm{div}(\mu_t \nabla_{W_2} \mathcal{E}(\mu_t)), \quad \mathcal{E}(\mu) = \int C(m)\, d\mu(m)$$



(Mokrov et al. (2021))

Gradient flows in the space of probabilities

The deterministic gradient descent $m_{k+1} = m_k - \tau_k \nabla C(m_k)$ is an explicit Euler discretization

Gradient flow - Euclidean space $\mathbb{R}^d$

$$\frac{dm}{dt} = -\nabla C(m), \quad m(0) = m_0,$$

with time step τ_k. Let $\{M_t\}_{t\geq 0}$ be a stochastic process with probability law μ_t

Gradient flow - Probability space $\mathcal{P}_2(\mathbb{R}^d)$

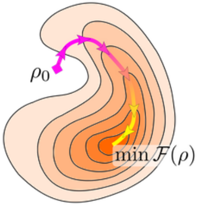
$$dM_t = -\nabla C(M_t)dt + \sqrt{2}dB_t, \quad M_t \sim \mu_t, \quad B_t : \text{Brownian motion}$$

The density μ_t satisfies a Fokker-Planck equation $\rightarrow$ continuous flow

$$\frac{\partial \mu_t}{\partial t} = \mathrm{div}(\mu_t \nabla C) + \Delta \mu_t \quad v_t = -\nabla C - \nabla \log \mu_t$$

We get a gradient flow in probability space under the Wasserstein metric

$$\frac{\partial \mu_t}{\partial t} = \mathrm{div}(\mu_t \nabla_{W_2} \mathcal{E}(\mu_t)), \quad \mathcal{E}(\mu) = \int C(m)\, d\mu(m) + \int \log(\mu(m))\, d\mu(m)$$



(Mokrov et al. (2021))

Sampling from different perspectives

The evolution of the density μ_t transports any prior π_{prior} to the posterior $\pi_{\text{post}} \propto \exp(-C)$

$$\frac{\partial \mu_t}{\partial t} = \text{div}(\mu_t \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}})), \ \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}}) = \nabla \log \left(\frac{\mu_t}{\pi_{\text{post}}} \right), \ \mathcal{E}(\mu) := \text{KL}(\mu || \pi_{\text{post}}) + \text{const},$$

and is the Wasserstein gradient flow of the KL divergence (Jordan et al. (1998))

Bayesian sampling → follow the steepest-descent of KL($\cdot || \pi_{\text{post}}$) in W_2

The evolution of the density μ_t transports any prior π_{prior} to the posterior $\pi_{\text{post}} \propto \exp(-C)$

$$\frac{\partial \mu_t}{\partial t} = \text{div}(\mu_t \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}})), \; \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}}) = \nabla \log \left(\frac{\mu_t}{\pi_{\text{post}}} \right), \; \mathcal{E}(\mu) := \text{KL}(\mu || \pi_{\text{post}}) + \text{const},$$

and is the Wasserstein gradient flow of the KL divergence (Jordan et al. (1998))

Bayesian sampling → follow the steepest-descent of $\mathbf{KL}(\cdot || \pi_{\text{post}})$ in W_2

1) Particle dynamics

evolve $\{M_t\}_{t \geq 0}$ realizations from μ_t

- Stochastic particle dynamics (SDE)
  $dM_t = -\nabla C(M_t)dt + \sqrt{2} dB_t$
  Langevin diffusion, MCMC (Ma et al. (2015))
- Deterministic particle dynamics (ODE)
  $dM_t = -\nabla \log(\mu_t / \pi_{\text{post}})dt,$
  with an approximation of μ_t

Sampling from different perspectives

The evolution of the density μ_t transports any prior π_{prior} to the posterior $\pi_{\text{post}} \propto \exp(-C)$

$$\frac{\partial \mu_t}{\partial t} = \text{div}(\mu_t \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}})), \; \nabla_{W_2} \text{KL}(\mu_t || \pi_{\text{post}}) = \nabla \log \left(\frac{\mu_t}{\pi_{\text{post}}} \right), \; \mathcal{E}(\mu) := \text{KL}(\mu || \pi_{\text{post}}) + \text{const},$$

and is the Wasserstein gradient flow of the KL divergence (Jordan et al. (1998))

Bayesian sampling $\rightarrow$ follow the steepest-descent of $\text{KL}(\cdot || \pi_{\text{post}})$ in W_2

1) Particle dynamics

evolve $\{M_t\}_{t \geq 0}$ realizations from μ_t

- Stochastic particle dynamics (SDE)
  $dM_t = -\nabla C(M_t)dt + \sqrt{2}dB_t$
  Langevin diffusion, MCMC (Ma et al. (2015))
- Deterministic particle dynamics (ODE)
  $dM_t = -\nabla \log(\mu_t / \pi_{\text{post}})dt$,
  with an approximation of μ_t

2) Variational inference (VI)

evolve a restricted distribution family $\hat{\mu}_t \approx \mu_t$

- Perform a "Wasserstein" gradient descent

$$\hat{\mu}_{k+1} = (\text{Id} - \tau \nabla_{W_2} \text{KL}(\hat{\mu}_k || \pi_{\text{post}}))_{\#} \hat{\mu}_k$$

  A common choice is the Gaussian family.
- VI is an optimization *over transport maps* (El Moselhy and Marzouk (2012))

Further details : Santambrogio (2015); Chewi et al. (2024)

Outline

Stochastic particle dynamics (SDE) - Langevin diffusion

The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

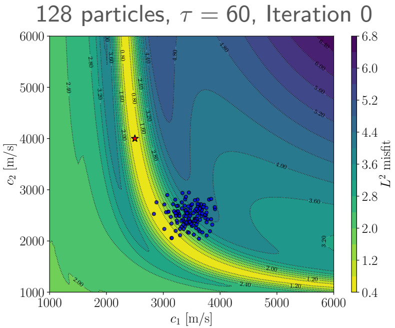
which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C

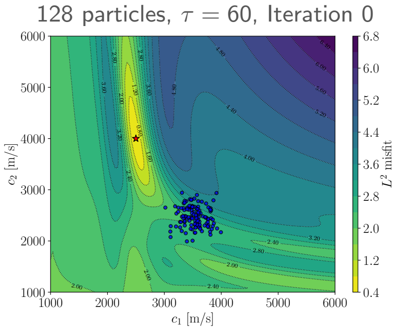
The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C



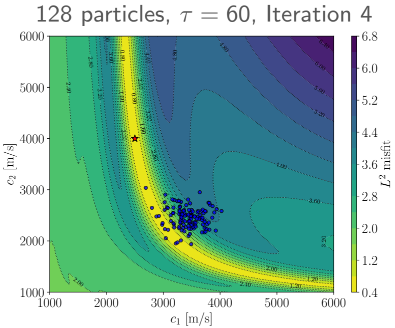


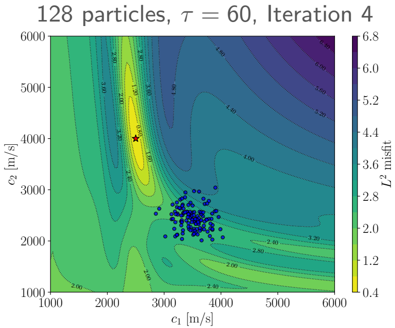
The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C





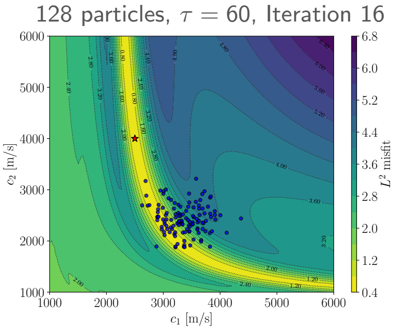
Stochastic particle dynamics (SDE) - Langevin diffusion

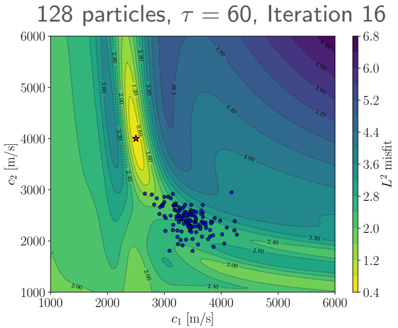
The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C





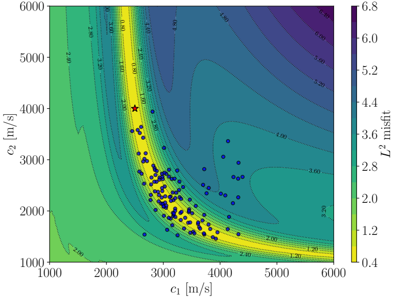
The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

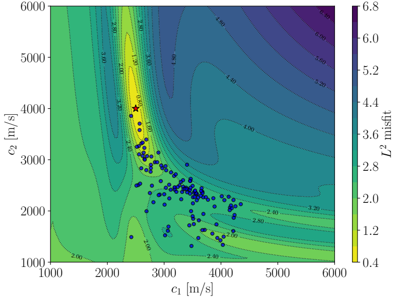
$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C

128 particles, $\tau = 60$, Iteration 64



128 particles, $\tau = 60$, Iteration 64

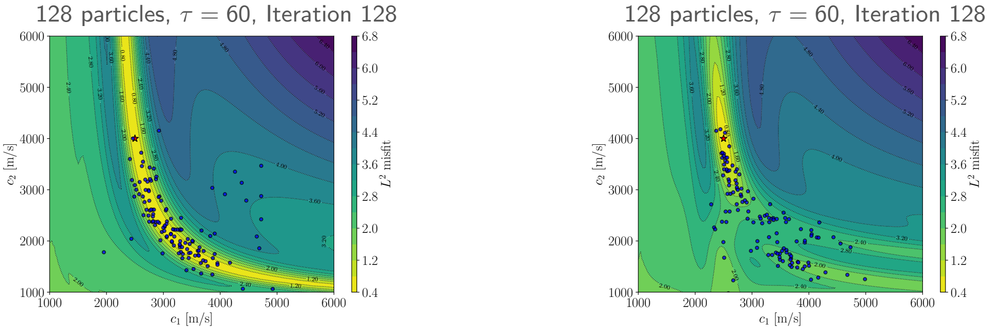


The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C



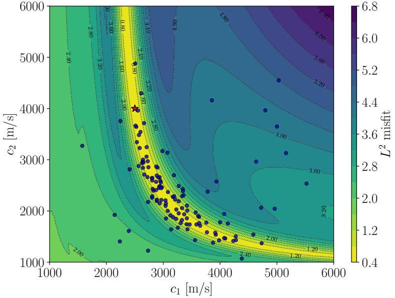
The Langevin diffusion can be discretized with an explicit Euler method (Euler-Maruyama)

Unadjusted Langevin algorithm

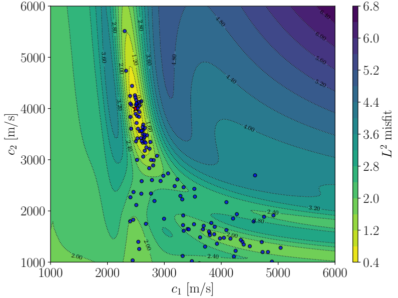
$$M_{k+1} = M_k - \tau \nabla C(M_k) + \sqrt{2\tau}\xi_k, \quad \xi_k \sim \mathcal{N}(0, Id),$$

which allows, in principle, to sample π_{post} from any prior, given the gradient of the cost ∇C

128 particles, $\tau = 60$, Iteration 256



128 particles, $\tau = 60$, Iteration 256



In practice : finite step size τ, finite time and finite number of particles $\Rightarrow$ sampling bias

Gradient descent in optimization $\Leftrightarrow$ Langevin algorithm in sampling

Outline

Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m, \cdot) \nabla_{W_2} \mathrm{KL}(\mu_k \| \pi_{\mathrm{post}}) \mu_k(m) dm$$
$$= \mathbb{E}_{\mu_k} \left(\nabla \kappa + \kappa \nabla \log \pi_{\mathrm{post}} \right)$$

and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k \| \pi_{\mathrm{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}\left(\nabla\kappa + \kappa\nabla \log \pi_{\mathrm{post}}\right)$$

and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

WGD simply uses a kernel density estimation for μ_t

$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k \| \pi_{\mathrm{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^{N}$ (Wang et al. (2022))

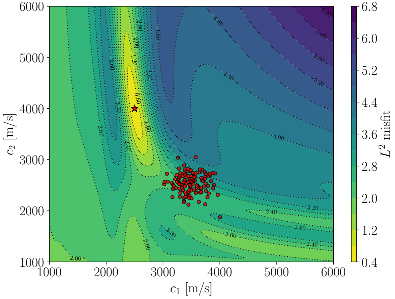
Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k \| \pi_{\mathrm{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}\left(\nabla\kappa + \kappa\nabla \log \pi_{\mathrm{post}}\right)$$

and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

128 particles, Iteration 0

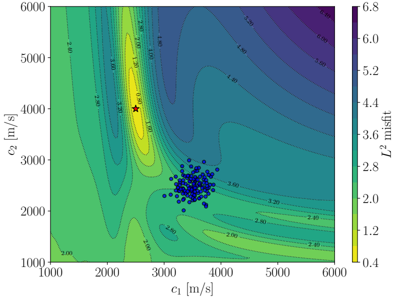


WGD simply uses a kernel density estimation for μ_t

$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k \| \pi_{\mathrm{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^N$ (Wang et al. (2022))

128 particles, Iteration 0



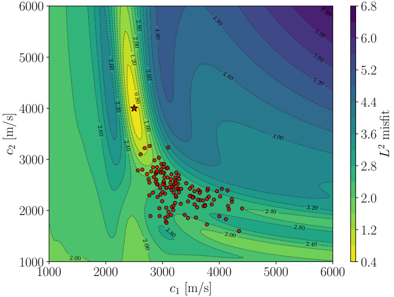
Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k\|\pi_{\text{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}\left(\nabla\kappa + \kappa\nabla\log\pi_{\text{post}}\right)$$

and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

128 particles, Iteration 4

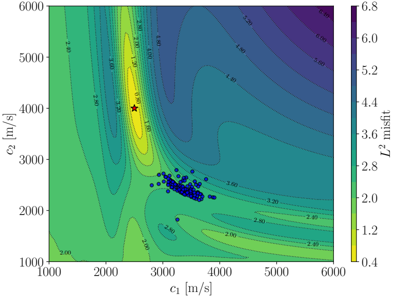


WGD simply uses a kernel density estimation for μ_t

$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k\|\pi_{\text{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^N$ (Wang et al. (2022))

128 particles, Iteration 4



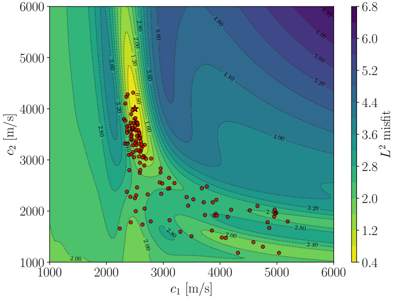
Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k \| \pi_{\text{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}(\nabla\kappa + \kappa\nabla \log \pi_{\text{post}})$$

and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

128 particles, Iteration 16

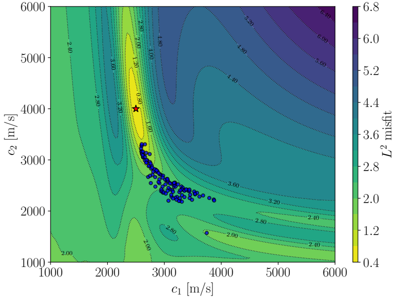


WGD simply uses a kernel density estimation for μ_t

$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k \| \pi_{\text{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^N$ (Wang et al. (2022))

128 particles, Iteration 16

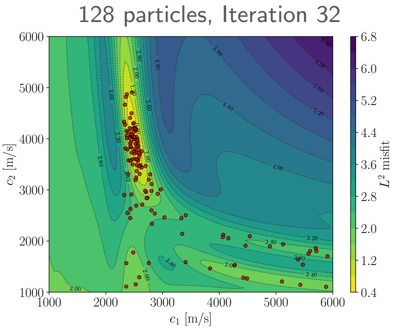


Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k \| \pi_{\mathrm{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}\left(\nabla\kappa + \kappa\nabla \log \pi_{\mathrm{post}}\right)$$

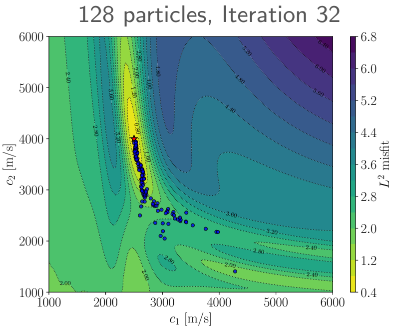
and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))



WGD simply uses a kernel density estimation for μ_t

$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k \| \pi_{\mathrm{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^N$ (Wang et al. (2022))



Deterministic (ODE) at the particle level with explicit time-stepping : $M_{k+1} = M_k - \tau_k \phi_k(M_k)$

SVGD first looks for a kernel-W^2 steepest descent

$$\phi_k = \int_{\mathbb{R}^d} \kappa(m,\cdot)\nabla_{W_2}\mathrm{KL}(\mu_k \| \pi_{\mathrm{post}})\mu_k(m)dm$$
$$= \mathbb{E}_{\mu_k}\left(\nabla\kappa + \kappa\nabla \log \pi_{\mathrm{post}}\right)$$

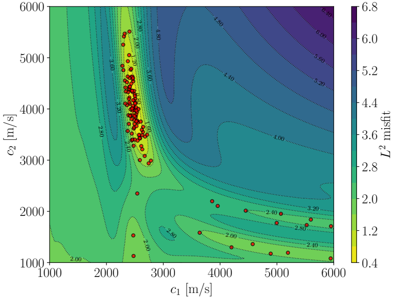
and second, uses particles for $\mathbb{E}_{\mu_k}(\cdot)$ (Liu and Wang (2016))

WGD simply uses a kernel density estimation for μ_t

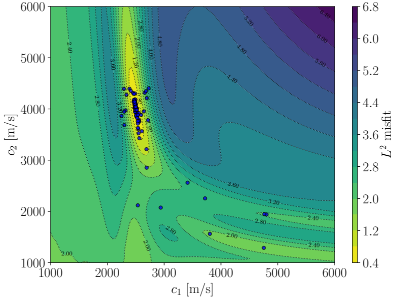
$$\phi_k = \nabla_{W_2}\mathrm{KL}(\hat{\mu}_k \| \pi_{\mathrm{post}}),$$
$$\hat{\mu}_k(M) = \sum_{i=1}^{N} \kappa(M, M_k^{(i)}),$$

for a set of particles $\{M_k^{(i)}\}_{i=1}^N$ (Wang et al. (2022))

128 particles, Iteration 64



128 particles, Iteration 64



The choice of the kernel $\kappa(\cdot,\cdot)$ is crucial $\rightarrow$ suffers from the curse of dimensionality

Outline

Gaussian variational inference

We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m - m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m - m^{(i)})\right]$$

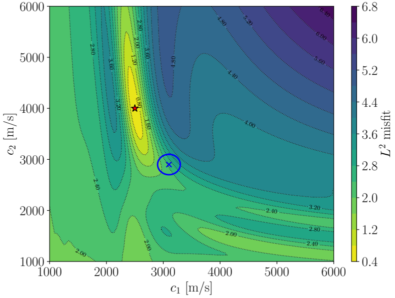
We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

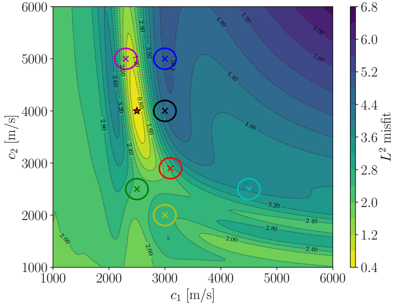
$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m - m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m - m^{(i)})\right]$$

Single Gaussian, Iteration 0



Gaussian mixture $N = 7$, Iteration 0



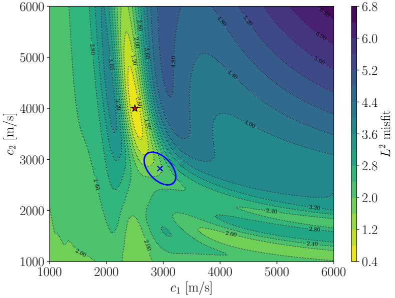
We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

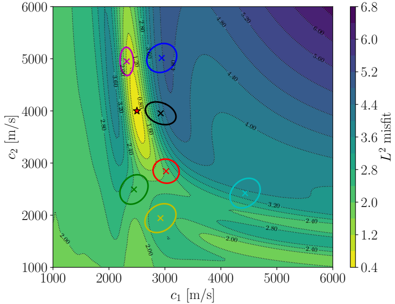
$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m - m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m - m^{(i)})\right]$$

Single Gaussian, Iteration 4



Gaussian mixture $N = 7$, Iteration 4



Gaussian variational inference

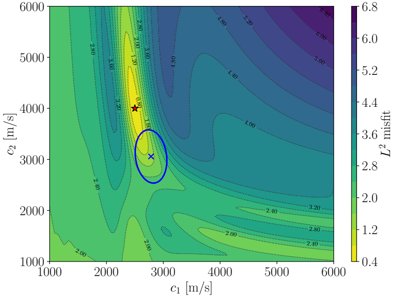
We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

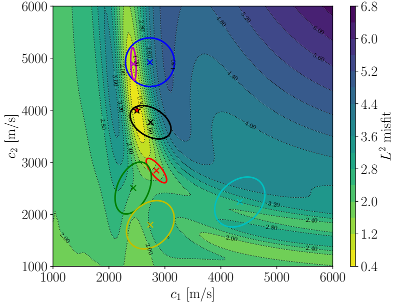
$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m - m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m - m^{(i)})\right]$$

Single Gaussian, Iteration 16



Gaussian mixture $N = 7$, Iteration 16



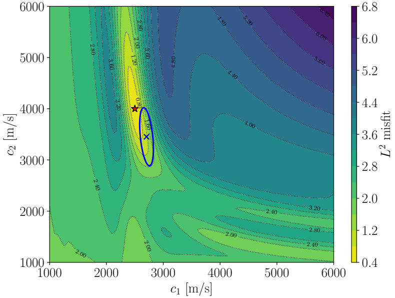
We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

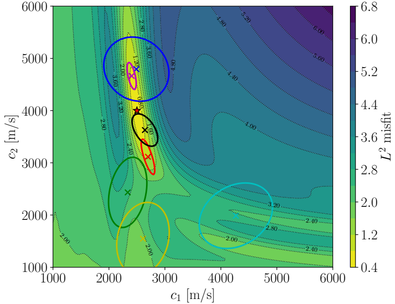
$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m-m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m-m^{(i)})\right]$$

Single Gaussian, Iteration 32



Gaussian mixture $N = 7$, Iteration 32



Gaussian variational inference

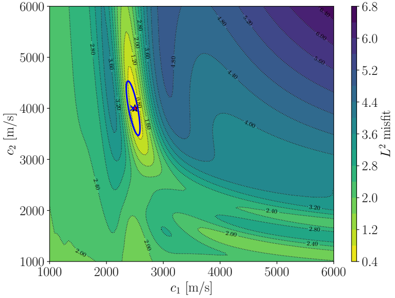
We choose $\hat{\mu}_t(m) = \frac{1}{N}\sum_{i=1}^{N} \hat{\mu}_i(m|m^{(i)}, \Sigma^{(i)})$ as a mixture of *Gaussians particles* with equal weights

Wasserstein gradient flow restricted to Gaussian mixture (Lambert et al. (2022))

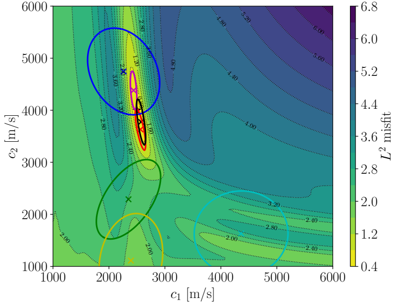
$$dm^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right]$$

$$d\Sigma^{(i)}/dt = -\mathbb{E}_{\hat{\mu}_i}\left[(m - m^{(i)}) \otimes \nabla \log(\hat{\mu}_t/\pi_{\text{post}})\right] - \mathbb{E}_{\hat{\mu}_i}\left[\nabla \log(\hat{\mu}_t/\pi_{\text{post}}) \otimes (m - m^{(i)})\right]$$

Single Gaussian, Iteration 64



Gaussian mixture $N = 7$, Iteration 64



The quality of the method depends on a low-variance estimator for $\mathbb{E}_{\hat{\mu}}$

Outline

Extended Kalman Filter : do a Bayes update for a **linearized** system $\mathcal{F}(m) = \mathcal{F}(\overline{m}) + \mathcal{J}_{\mathcal{F}}(\overline{m})(m - \overline{m})$

Given a Gaussian initial state $\mathcal{N}(m_0, \Sigma_0)$, the update is also Gaussian $\mathcal{N}(m_1, \Sigma_1)$

Extended Kalman Filter (mean update)

$$m_1 = m_0 + K(\mathcal{F}(m_0) - d_{\text{obs}}), \quad K = \Sigma_0 \mathcal{J}_{\mathcal{F}}^T (\mathcal{J}_{\mathcal{F}} \Sigma_0 \mathcal{J}_{\mathcal{F}}^T + \Sigma)^{-1}$$

Extended Kalman Filter : do a Bayes update for a **linearized** system $\mathcal{F}(m) = \mathcal{F}(\overline{m}) + \mathcal{J}_{\mathcal{F}}(\overline{m})(m - \overline{m})$

Given a Gaussian initial state $\mathcal{N}(m_0, \Sigma_0)$, the update is also Gaussian $\mathcal{N}(m_1, \Sigma_1)$

Extended Kalman Filter (mean update)

$$m_1 = m_0 + K(\mathcal{F}(m_0) - d_{\text{obs}}), \quad K = \Sigma_0 \mathcal{J}_{\mathcal{F}}^T (\mathcal{J}_{\mathcal{F}} \Sigma_0 \mathcal{J}_{\mathcal{F}}^T + \Sigma)^{-1}$$

We can use KF as an iterative method on $(m_0, m_1, \cdots, m^*)$ $\rightarrow$ Gauss-Newton method (Bell and Cathey (1993))

The EnKF relies on a **particle, Monte-Carlo approximation** for $\Sigma_0 \mathcal{J}_{\mathcal{F}}^T$ and $\mathcal{J}_{\mathcal{F}} \Sigma_0 \mathcal{J}_{\mathcal{F}}^T$ $\rightarrow$ low-rank update

Extended Kalman Filter : do a Bayes update for a **linearized** system $\mathcal{F}(m) = \mathcal{F}(\overline{m}) + \mathcal{J}_\mathcal{F}(\overline{m})(m - \overline{m})$
Given a Gaussian initial state $\mathcal{N}(m_0, \Sigma_0)$, the update is also Gaussian $\mathcal{N}(m_1, \Sigma_1)$

Extended Kalman Filter (mean update)

$$m_1 = m_0 + K(\mathcal{F}(m_0) - d_{\text{obs}}), \quad K = \Sigma_0 \mathcal{J}_\mathcal{F}^T (\mathcal{J}_\mathcal{F} \Sigma_0 \mathcal{J}_\mathcal{F}^T + \Sigma)^{-1}$$

We can use KF as an iterative method on $(m_0, m_1, \cdots, m^*)$ $\rightarrow$ Gauss-Newton method (Bell and Cathey (1993))
The EnKF relies on a **particle, Monte-Carlo approximation** for $\Sigma_0 \mathcal{J}_\mathcal{F}^T$ and $\mathcal{J}_\mathcal{F} \Sigma_0 \mathcal{J}_\mathcal{F}^T$ $\rightarrow$ low-rank update

EnKF as optimization : continuous time-limit

$$dM_t = -P(M_t)\nabla C(M_t)dt$$

where $P(M_t)\nabla C(M_t) \approx K(\mathcal{F}(M_t) - d_{\text{obs}})$ is obtained from the ensemble (Schillings and Stuart (2017))

Towards sampling in high-dimension : Ensemble Kalman filters (EnKF)

Extended Kalman Filter : do a Bayes update for a **linearized** system $\mathcal{F}(m) = \mathcal{F}(\overline{m}) + \mathcal{J}_{\mathcal{F}}(\overline{m})(m - \overline{m})$
Given a Gaussian initial state $\mathcal{N}(m_0, \Sigma_0)$, the update is also Gaussian $\mathcal{N}(m_1, \Sigma_1)$

Extended Kalman Filter (mean update)

$$m_1 = m_0 + K(\mathcal{F}(m_0) - d_{\text{obs}}), \quad K = \Sigma_0 \mathcal{J}_{\mathcal{F}}^T (\mathcal{J}_{\mathcal{F}} \Sigma_0 \mathcal{J}_{\mathcal{F}}^T + \Sigma)^{-1}$$

We can use KF as an iterative method on $(m_0, m_1, \cdots, m^*)$ $\rightarrow$ Gauss-Newton method (Bell and Cathey (1993))
The EnKF relies on a **particle, Monte-Carlo approximation** for $\Sigma_0 \mathcal{J}_{\mathcal{F}}^T$ and $\mathcal{J}_{\mathcal{F}} \Sigma_0 \mathcal{J}_{\mathcal{F}}^T$ $\rightarrow$ low-rank update

EnKF as optimization : continuous time-limit

$$dM_t = -P(M_t)\nabla C(M_t)dt$$

where $P(M_t)\nabla C(M_t) \approx K(\mathcal{F}(M_t) - d_{\text{obs}})$ is obtained from the ensemble (Schillings and Stuart (2017))

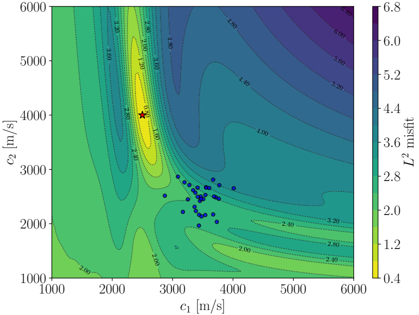
EnKF as sampling (SDE) : continuous time-limit

$$dM_t = -P(M_t)\nabla C(M_t)dt + \sqrt{2}P^{1/2}(M_t)dB_t$$

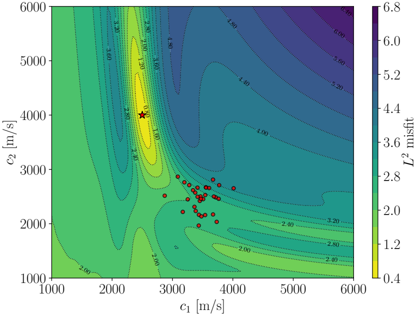
P-preconditioned Langevin diffusion (Chada et al. (2021)), $P^{1/2}$ is not computed explicitly

At each iteration, we may bias the prior $\hat{m}_k = BFGS(m_k)$ towards m^* (Thurin et al. (2019); Hoffmann et al. (2024))

EnKF as optimization : 32 particles, Iteration 0

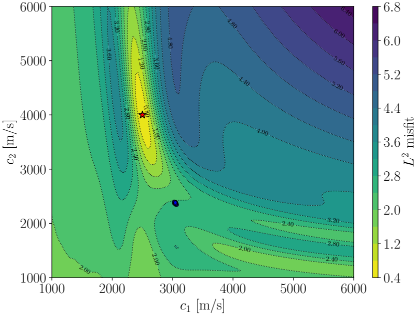


EnKF as sampling : 32 particles, Iteration 0

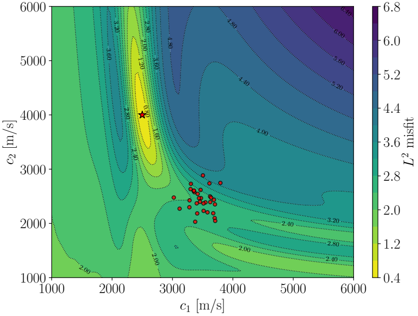


Ensemble Kalman filters : optimization vs sampling

EnKF as optimization : 32 particles, Iteration 1

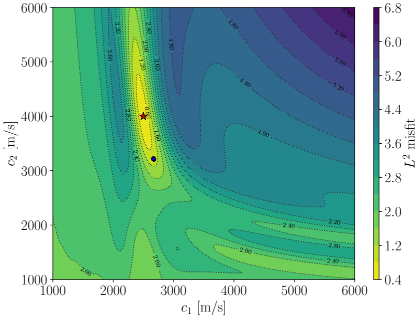


EnKF as sampling : 32 particles, Iteration 1

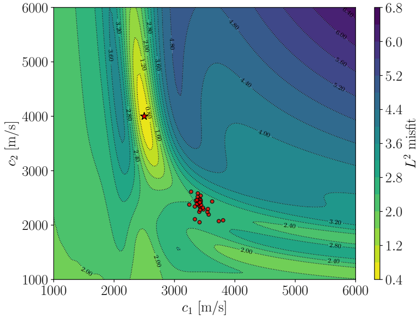


Ensemble Kalman filters : optimization vs sampling

EnKF as optimization : 32 particles, Iteration 4

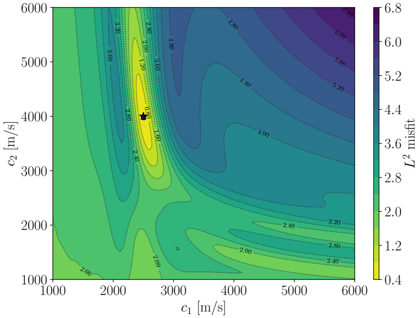


EnKF as sampling : 32 particles, Iteration 4

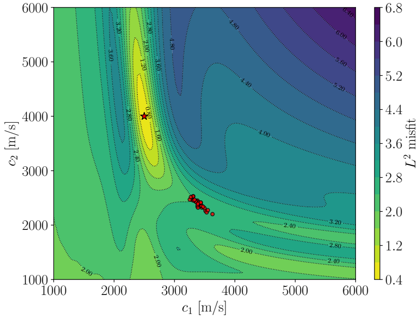


Ensemble Kalman filters : optimization vs sampling

EnKF as optimization : 32 particles, Iteration 16

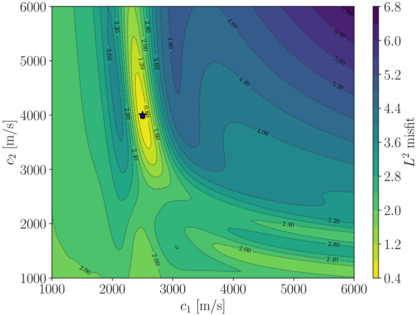


EnKF as sampling : 32 particles, Iteration 16

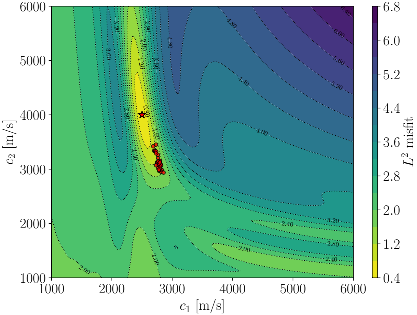


Ensemble Kalman filters : optimization vs sampling

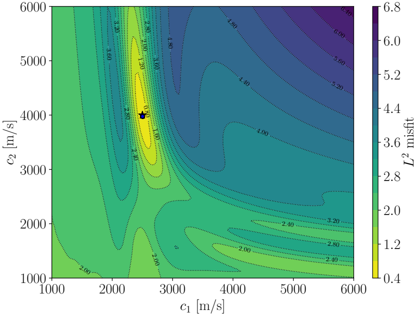
EnKF as optimization : 32 particles, Iteration 48



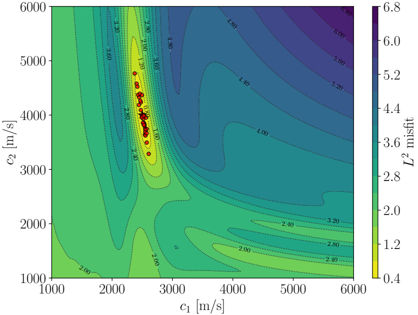
EnKF as sampling : 32 particles, Iteration 48



EnKF as optimization : 32 particles, Iteration 64

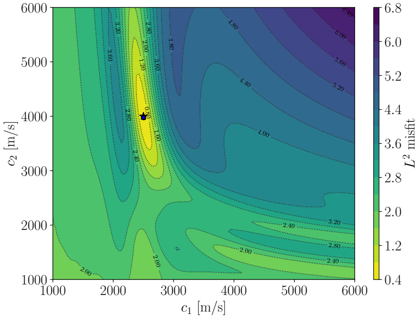


EnKF as sampling : 32 particles, Iteration 64

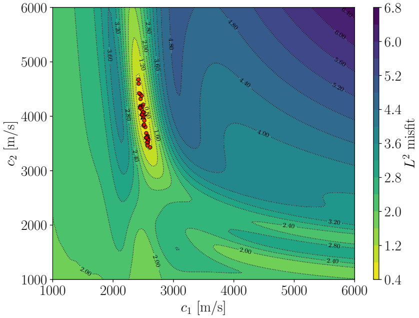


Ensemble Kalman filters : optimization vs sampling

EnKF as optimization : 32 particles, Iteration 128



EnKF as sampling : 32 particles, Iteration 128



- EnKF is a (derivative-free) 2nd order method : less calls to the forward map $\mathcal{F}$
- EnKF scales to high-dimension parameter space thanks to low-rank ensemble
- EnKF as sampling :
  - needs more iterations but avoids ensemble collapse
  - is limited to a Gaussian approximation of the posterior
  - does not need parameter tuning

We scratched the surface of modern Bayesian methods

- Uncertainty quantification involves computing a representation of the posterior distribution
- Gradient flow in probability space unifies sampling methods and suggest well-grounded strategies

Conclusion

We scratched the surface of modern Bayesian methods

- Uncertainty quantification involves computing a representation of the posterior distribution
- Gradient flow in probability space unifies sampling methods and suggest well-grounded strategies
- Ingredients towards efficient UQ in FWI
  1. few function evaluations $\rightarrow$ 2nd order method + efficient time-integration
  2. high-dimensional scaling $\rightarrow$ low-rank reduction
  3. sampling accuracy

There is hope for UQ in FWI - we need to develop solvers

Thank you for your attention and thanks to

- All SEISCOPE sponsors : AKERBP, DUG, EXXONMOBIL, GEOLINKS, JGI, PETROBRAS, SHEARWATER, SHELL, SINOPEC, TOTALENERGIES and VIRIDIEN
- IDRIS, TGCC and CINES, French national computing centers
- GRICAD, Grenoble computing center
- SWAN, Hewlett Packard Enterprise (HPE) Cray XC System
- All SEISCOPE project members

Questions ?

philippe.marchner@univ-grenoble-alpes.fr

Bell, B. M. and Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Bui-Thanh, T., Ghattas, O., Martin, J., and Stadler, G. (2013). A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35(6):A2494–A2523.

Chada, N. K., Chen, Y., and Sanz-Alonso, D. (2021). Iterative ensemble Kalman methods: A unified perspective with some new variants. *Foundations of Data Science*, 3(3):331–369.

Chewi, S., Niles-Weed, J., and Rigollet, P. (2024). Statistical optimal transport. *arXiv:2407.18163*.

El Moselhy, T. A. and Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850.

Fichtner, A. and Trampert, J. (2011). Resolution analysis in full waveform inversion. *Geophysical Journal International*, 187(3):1604–1624.

Gebraad, L., Boehm, C., and Fichtner, A. (2020). Bayesian elastic full-waveform inversion using Hamiltonian Monte Carlo. *Journal of Geophysical Research: Solid Earth*, 125(3).

Hoffmann, A., Brossier, R., Métivier, L., and Tarayoun, A. (2024). Local uncertainty quantification for 3-D time-domain full-waveform inversion with ensemble Kalman filters: application to a North Sea OBC data set. *Geophysical Journal International*, 237(3):1353–1383.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998). The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17.

Kaipio, J. and Somersalo, E. (2006). *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media.

Lambert, M., Chewi, S., Bach, F., Bonnabel, S., and Rigollet, P. (2022). Variational inference via Wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 35:14434–14447.

Liu, Q. and Peter, D. (2020). Square-root variable metric-based nullspace shuttle: A characterization of the nonuniqueness in elastic full-waveform inversion. *Journal of Geophysical Research: Solid Earth*, 125(2).

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29.

Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient MCMC. *Advances in neural information processing systems*, 28.

Martin, J., Wilcox, L. C., Burstedde, C., and Ghattas, O. (2012). A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487.

Mokrov, P., Korotin, A., Li, L., Genevay, A., Solomon, J. M., and Burnaev, E. (2021). Large-scale Wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34:15243–15256.

Mosegaard, K. and Tarantola, A. (2002). Probabilistic approach to inverse problems. In *International Geophysics*, volume 81, pages 237–265. Elsevier.

Santambrogio, F. (2015). *Optimal transport for applied mathematicians*. Birkäuser, Basel.

Schillings, C. and Stuart, A. M. (2017). Analysis of the ensemble Kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55(3):1264–1290.

Sen, M. K. and Biswas, R. (2017). Transdimensional seismic inversion using the reversible jump Hamiltonian Monte Carlo algorithm. *Geophysics*, 82(3):R119–R134.

Sen, M. K. and Stoffa, P. L. (1996). Bayesian inference, Gibbs' sampler and uncertainty estimation in geophysical inversion. *Geophysical Prospecting*, 44(2):313–350.

Tarantola, A. and Valette, B. (1982). Inverse problems= quest for information. *Journal of geophysics*, 50(1):159–170.

Thurin, J., Brossier, R., and Métivier, L. (2019). Ensemble-based uncertainty estimation in full waveform inversion. *Geophysical Journal International*, 219(3):1613–1635.

Virieux, J., Asnaashari, A., Brossier, R., Métivier, L., Ribodetti, A., and Zhou, W. (2017). *6. An introduction to full waveform inversion*, chapter 6, pages R1–40. Society of Exploration Geophysicists.

Wang, Y., Chen, P., and Li, W. (2022). Projected Wasserstein gradient descent for high-dimensional bayesian inference. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4):1513–1532.

Zhang, X., Lomas, A., Zhou, M., Zheng, Y., and Curtis, A. (2023). 3-D Bayesian variational full waveform inversion. *Geophysical Journal International*, 234(1):546–561.