Laplace's approximation for subsurface applications

Kristian Fossum[1], *Mathias M. Nilsen*[1] *& Andreas S. Stordal*

[1] NORCE Research

Introduction

Ensemble-based methods are popular for subsurface data assimilation.

Introduction

Ensemble-based methods are popular for subsurface data assimilation.

- but they require localization to work with small ensembles,
- measurements are typically non-local, and localization is hard to determine a priori.

Introduction

Ensemble-based methods are popular for subsurface data assimilation.

- but they require localization to work with small ensembles,
- measurements are typically non-local, and localization is hard to determine a priori.

Laplace's approximation is an alternative:

- builds a Gaussian approximation of the posterior,
- no localization required,
- requires exact sensitivities from the simulator.

Laplace approximation

Posterior for parameters m given data d_{obs}:

$$\pi(m \mid d_{\mathrm{obs}}) \;\propto\; \exp\big(-\mathcal{J}(m)\big), \quad \mathcal{J}(m) = \tfrac{1}{2}\|g(m) - d_{\mathrm{obs}}\|^2_{C_d^{-1}} + \tfrac{1}{2}\|m - \mu\|^2_{C_m^{-1}}.$$

Laplace approximation

Posterior for parameters m given data d_{obs}:

$$\pi(m \mid d_{\mathrm{obs}}) \;\propto\; \exp\big(-\mathcal{J}(m)\big), \quad \mathcal{J}(m) = \tfrac{1}{2}\|g(m) - d_{\mathrm{obs}}\|^2_{C_d^{-1}} + \tfrac{1}{2}\|m - \mu\|^2_{C_m^{-1}}.$$

Expand $\mathcal{J}$ to second order about its minimiser m_{MAP}:

$$\mathcal{J}(m) \approx \mathcal{J}(m_{\mathrm{MAP}}) + \tfrac{1}{2}\,(m - m_{\mathrm{MAP}})^{\top} H\,(m - m_{\mathrm{MAP}}).$$

Laplace approximation

Posterior for parameters m given data d_{obs}:

$$\pi(m \mid d_{\mathrm{obs}}) \;\propto\; \exp\!\big(-\mathcal{J}(m)\big), \quad \mathcal{J}(m) = \tfrac{1}{2}\|g(m) - d_{\mathrm{obs}}\|^2_{C_d^{-1}} + \tfrac{1}{2}\|m - \mu\|^2_{C_m^{-1}}.$$

Expand $\mathcal{J}$ to second order about its minimiser m_{MAP}:

$$\mathcal{J}(m) \approx \mathcal{J}(m_{\mathrm{MAP}}) + \tfrac{1}{2}\,(m - m_{\mathrm{MAP}})^{\top} H\,(m - m_{\mathrm{MAP}}).$$

Gaussian approximation — curvature at the mode gives the covariance:

$$\pi(m \mid d_{\mathrm{obs}}) \approx \mathcal{N}\big(m_{\mathrm{MAP}},\, H^{-1}\big), \qquad H = \nabla^2 \mathcal{J}(m_{\mathrm{MAP}}).$$

Laplace approximation

Challenges

Two ingredients, both expensive:

- find the MAP point $m_{\mathrm{MAP}} = \arg\min_m \mathcal{J}(m)$,
- sample $\mathcal{N}(m_{\mathrm{MAP}}, H^{-1})$ — the inverse Hessian.

Laplace approximation

Challenges

Two ingredients, both expensive:

- find the MAP point $m_{\mathrm{MAP}} = \arg\min_m \mathcal{J}(m)$,
- sample $\mathcal{N}(m_{\mathrm{MAP}}, H^{-1})$ — the inverse Hessian.

Gauss–Newton Hessian:

$$H = \underbrace{C_m^{-1}}_{\text{prior precision}} + \underbrace{J^\top C_d^{-1} J}_{\text{data Hessian}}.$$

Laplace approximation

Challenges

Two ingredients, both expensive:

- find the MAP point $m_{\mathrm{MAP}} = \arg\min_m \mathcal{J}(m)$,
- sample $\mathcal{N}(m_{\mathrm{MAP}}, H^{-1})$ — the inverse Hessian.

Gauss–Newton Hessian:

$$H = \underbrace{C_m^{-1}}_{\text{prior precision}} + \underbrace{J^\top C_d^{-1} J}_{\text{data Hessian}}.$$

- prior covariance C_m is dense — cannot be inverted or stored for large models,
- the Jacobian J is available only at high cost.

SPDE approach for precission matrix

A computationally possible approach

Build a sparse prior precision C_m^{-1} directly

SPDE approach for precission matrix

A computationally possible approach

Build a sparse prior precision C_m^{-1} directly

Matérn field = solution of an SPDE (Lindgren, Rue & Lindström, 2011):

$$(\kappa^2 - \Delta)^{\alpha/2} u(s) = \mathcal{W}(s).$$

SPDE approach for precission matrix

A computationally possible approach

Build a sparse prior precision C_m^{-1} directly

Matérn field = solution of an SPDE (Lindgren, Rue & Lindström, 2011):

$$(\kappa^2 - \Delta)^{\alpha/2}\, u(s) = \mathcal{W}(s).$$

GMRF representation $\Rightarrow$ sparse precision C_m^{-1}:

- finite-difference discretisation $\rightarrow$ 9-point stencil (2D) on reservoir grid,
- only local neighbours couple $\Rightarrow$ C_m^{-1} sparse, C_m dense.

Sampling the prior precission

Factorize once, $C_m^{-1} = LL^\top$ (sparse Cholesky), then *solve*:

$$L^\top m = z, \quad z \sim \mathcal{N}(0, I) \implies \mathrm{Cov}(m) = C_m.$$

Sampling the prior precission

Factorize once, $C_m^{-1} = LL^\top$ (sparse Cholesky), then *solve*:

$$L^\top m = z, \quad z \sim \mathcal{N}(0, I) \implies \mathrm{Cov}(m) = C_m.$$

- one sparse triangular solve per sample,
- factorization reused across draws,
- additional samples are essentially free.

Sensitivities

Adjoint capabilities in JutulDarcy

JutulDarcy is a general-purpose open-source porous media simulator.

JutulDarcy provides discrete adjoints. One backward solve = one vector–Jacobian product:

$$w^\top J \in \mathbb{R}^{1\times n_m}, \qquad w = \frac{\partial \mathcal{Q}}{\partial g} \text{ (the seed).}$$

Sensitivities

Adjoint capabilities in JutulDarcy

JutulDarcy is a general-purpose open-source porous media simulator.

JutulDarcy provides discrete adjoints. One backward solve = one vector–Jacobian product:

$$w^\top J \in \mathbb{R}^{1\times n_m}, \qquad w = \frac{\partial \mathcal{Q}}{\partial g} \text{ (the seed).}$$

The seed selects which combination of Jacobian rows is returned:

- misfit gradient: $w = C_d^{-1} r$ — one solve per iteration,
- single row i: $w = e_i$ — n_d solves give the full J.

Sensitivities

Adjoint capabilities in JutulDarcy

JutulDarcy is a general-purpose open-source porous media simulator.

JutulDarcy provides discrete adjoints. One backward solve = one vector–Jacobian product:

$$w^\top J \in \mathbb{R}^{1\times n_m}, \qquad w = \frac{\partial \mathcal{Q}}{\partial g} \text{ (the seed).}$$

The seed selects which combination of Jacobian rows is returned:

- misfit gradient: $w = C_d^{-1} r$ — one solve per iteration,
- single row i: $w = e_i$ — n_d solves give the full J.

Cost is independent of n_m but proportional to n_d

Calculating the MAP

Minimise the objective

$$m_{\mathrm{MAP}} = \arg\min_{m} \ \tfrac{1}{2}\|g(m) - d_{\mathrm{obs}}\|^2_{C_d^{-1}} + \tfrac{1}{2}\|m - \mu\|^2_{C_m^{-1}}.$$

Calculating the MAP

Minimise the objective

$$m_{\mathrm{MAP}} = \arg\min_{m} \ \tfrac{1}{2}\|g(m) - d_{\mathrm{obs}}\|^2_{C_d^{-1}} + \tfrac{1}{2}\|m - \mu\|^2_{C_m^{-1}}.$$

Two schemes considered (others possible, e.g. CG):

- Gauss–Newton — exact Jacobian, fast convergence, costly steps,
- Quasi-Newton (L-BFGS) — gradient only, cheap steps.

Gauss-Newton optimization

Update step:

$$m_{k+1} = m_k - \left(C_m^{-1} + J_k^\top C_d^{-1} J_k\right)^{-1} \nabla \mathcal{J}(m_k).$$

Gauss-Newton optimization

Update step:

$$m_{k+1} = m_k - \left(C_m^{-1} + J_k^\top C_d^{-1} J_k\right)^{-1} \nabla \mathcal{J}(m_k).$$

Jacobian cost:

- the full J requires n_d adjoint solves (one per observation),
- yields the Gauss–Newton Hessian at the MAP,
- when converged – all elements of the Laplace covariance.

Quasi-Newton optimization

L-BFGS builds an implicit inverse Hessian from limited gradient history:

$$m_{k+1} = m_k - \widehat{H}_k^{-1} \nabla \mathcal{J}(m_k).$$

Quasi-Newton optimization

L-BFGS builds an implicit inverse Hessian from limited gradient history:

$$m_{k+1} = m_k - \widehat{H}_k^{-1} \nabla \mathcal{J}(m_k).$$

- one adjoint per iteration (misfit gradient only),
- limited memory $\Rightarrow$ no dense matrices,
- curvature around the MAP must be reconstructed for sampling.

Sampling the posterior

Need samples with covariance $H^{-1} = (C_m^{-1} + J^\top C_d^{-1} J)^{-1}$ — never form H^{-1}.

Sampling the posterior

Need samples with covariance $H^{-1} = (C_m^{-1} + J^\top C_d^{-1} J)^{-1}$ — never form H^{-1}.

Two square-root samplers, both starting from a prior draw:

- full-rank — Krylov / conjugate-gradient sampler (for GN Hessian),
- low-rank — compressed Jacobian + closed-form square root.

Sampling the posterior

Full-rank: Krylov / conjugate gradients

Draw $r \sim \mathcal{N}(0, H)$ from square roots of the pieces, then solve $Hs = r$:

$$r = C_m^{-1}\, m_p + J^\top C_d^{-1/2}\, \xi, \qquad s = H^{-1} r \text{ (preconditioned CG).}$$

Sampling the posterior

Full-rank: Krylov / conjugate gradients

Draw $r \sim \mathcal{N}(0, H)$ from square roots of the pieces, then solve $Hs = r$:

$$r = C_m^{-1} m_p + J^\top C_d^{-1/2} \xi, \qquad s = H^{-1} r \text{ (preconditioned CG)}.$$

- matrix-free: only the action $v \mapsto C_m^{-1} v + J^\top C_d^{-1} J v$,
- prior preconditioner $\Rightarrow$ mesh-independent, $\leq n_d + 1$ iterations,
- one CG solve per sample.

Sampling the posterior

Low-rank: building the ensemble basis

$\tilde{J} = C_d^{-1/2} J$ is low rank in output space — few independent well/phase/time response patterns. Find a basis U for $\mathrm{col}(\tilde{J})$.

Sampling the posterior

Low-rank: building the ensemble basis

$\tilde{J} = C_d^{-1/2} J$ is low rank in output space — few independent well/phase/time response patterns. Find a basis U for $\mathrm{col}(\tilde{J})$.

Draw a small ensemble around the MAP, $m^{(k)} = m_{\mathrm{MAP}} + \xi^{(k)}$, $\xi^{(k)} \sim \mathcal{N}(0, C_m)$:

- simulate each member, $d^{(k)} = g(m^{(k)})$,
- centre and whiten: $P_{:,k} = C_d^{-1/2}\big(d^{(k)} - \bar{d}\big)$,
- thin SVD $P = USV^{\top}$, keep the leading p modes (95 % energy).

Sampling the posterior

Low-rank: building the ensemble basis

$\tilde{J} = C_d^{-1/2} J$ is low rank in output space — few independent well/phase/time response patterns. Find a basis U for $\mathrm{col}(\tilde{J})$.

Draw a small ensemble around the MAP, $m^{(k)} = m_{\mathrm{MAP}} + \xi^{(k)}$, $\xi^{(k)} \sim \mathcal{N}(0, C_m)$:

- simulate each member, $d^{(k)} = g(m^{(k)})$,
- centre and whiten: $P_{:,k} = C_d^{-1/2}\big(d^{(k)} - \bar{d}\big)$,
- thin SVD $P = USV^\top$, keep the leading p modes (95 % energy).

First-order Taylor: $d^{(k)} - \bar{d} \approx J\,\xi^{(k)} \Rightarrow P_{:,k} \approx \tilde{J}\xi^{(k)}$.

- ensemble columns sample $\mathrm{col}(\tilde{J})$ — randomized range-finding,
- MAP-centring aligns U with $\tilde{J}$ at the linearisation point.

Sampling the posterior

Low-rank: compressed Jacobian

Probe J with $p \ll n_d$ informative seeds (MAP-centred ensemble basis U):

$$\tilde{G} = U^\top C_d^{-1/2} J \in \mathbb{R}^{p \times n_m}, \qquad H = C_m^{-1} + \tilde{G}^\top \tilde{G}.$$

Sampling the posterior

Low-rank: compressed Jacobian

Probe J with $p \ll n_d$ informative seeds (MAP-centred ensemble basis U):

$$\tilde{G} = U^\top C_d^{-1/2} J \in \mathbb{R}^{p \times n_m}, \qquad H = C_m^{-1} + \tilde{G}^\top \tilde{G}.$$

Closed-form square root of H^{-1} (eigenpairs λ_i, v_i):

$$m = m_{\mathrm{MAP}} + z + V \,\mathrm{diag}\Big(\tfrac{1}{\sqrt{1+\lambda_i}} - 1\Big)(C_m^{-1} V)^\top z.$$

Sampling the posterior

Low-rank: compressed Jacobian

Probe J with $p \ll n_d$ informative seeds (MAP-centred ensemble basis U):

$$\tilde{G} = U^\top C_d^{-1/2} J \in \mathbb{R}^{p \times n_m}, \qquad H = C_m^{-1} + \tilde{G}^\top \tilde{G}.$$

Closed-form square root of H^{-1} (eigenpairs λ_i, v_i):

$$m = m_{\mathrm{MAP}} + z + V \operatorname{diag}\Big(\tfrac{1}{\sqrt{1+\lambda_i}} - 1\Big)(C_m^{-1} V)^\top z.$$

- p adjoints instead of n_d,
- one $p \times p$ eigensolve, then no per-sample solve.

Numerical experiment

2D synthetic 5-spot example, with poorly known log-permeability; 60 well-rate data measurements (WOPR, WGPR, WWPR).

Numerical experiment

2D synthetic 5-spot example, with poorly known log-permeability; 60 well-rate data measurements (WOPR, WGPR, WWPR).

- Laplace approximation:
  1. Gauss-Newton optimizer with full-rank sampler – GN-Hessian,
  2. L-BFGS optimizer with low-rank ($p = 3$) sampler – QN-Hessian,
- Ensemble reference methods: RML, EnRML - with localization, ES-MDA - with localization,
- RML optimized with the L-BFGS scheme.

Numerical experiment

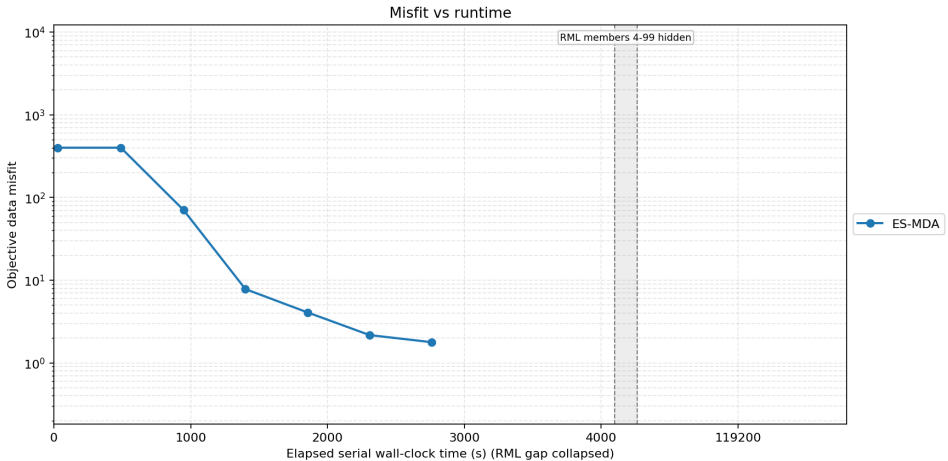
2D synthetic 5-spot example, with poorly known log-permeability; 60 well-rate data measurements (WOPR, WGPR, WWPR).

- Laplace approximation:
  1. Gauss-Newton optimizer with full-rank sampler – GN-Hessian,
  2. L-BFGS optimizer with low-rank ($p = 3$) sampler – QN-Hessian,
- Ensemble reference methods: RML, EnRML - with localization, ES-MDA - with localization,
- RML optimized with the L-BFGS scheme.

Compared on wall-clock vs. misfit, posterior mean, spread, and data match.

Results

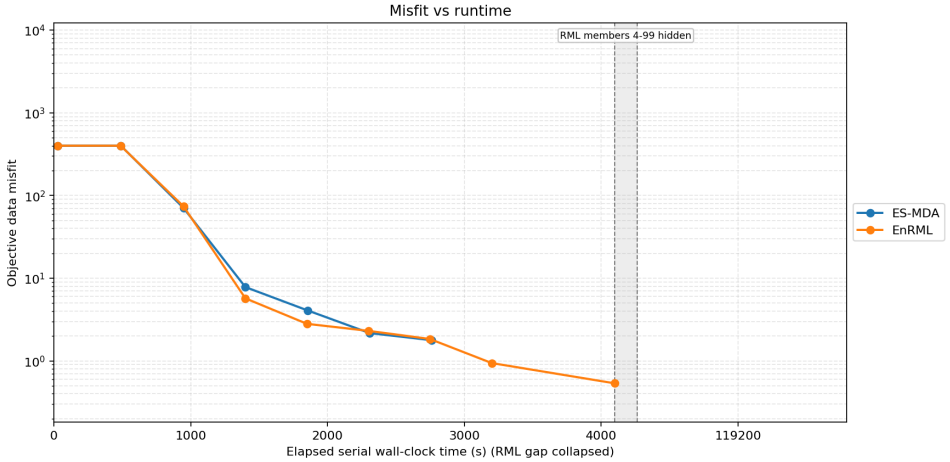
Wall-clock time vs. misfit



ES-MDA — ensemble baseline.

Results

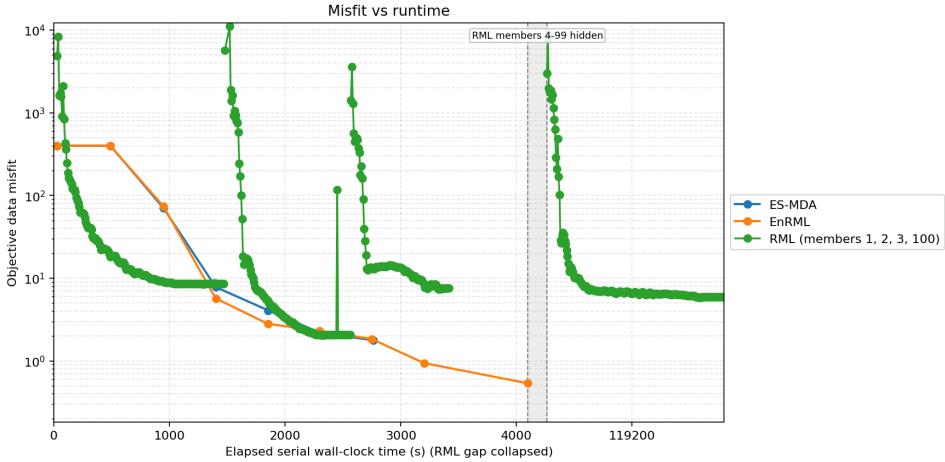
Wall-clock time vs. misfit



+ EnRML.

Results

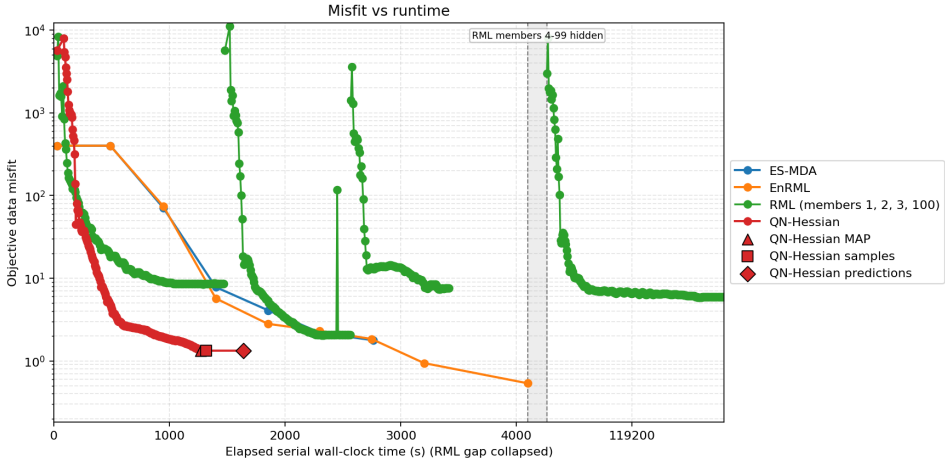
Wall-clock time vs. misfit



+ RML — per-member QN optimisation (members 4–99 hidden).

Results

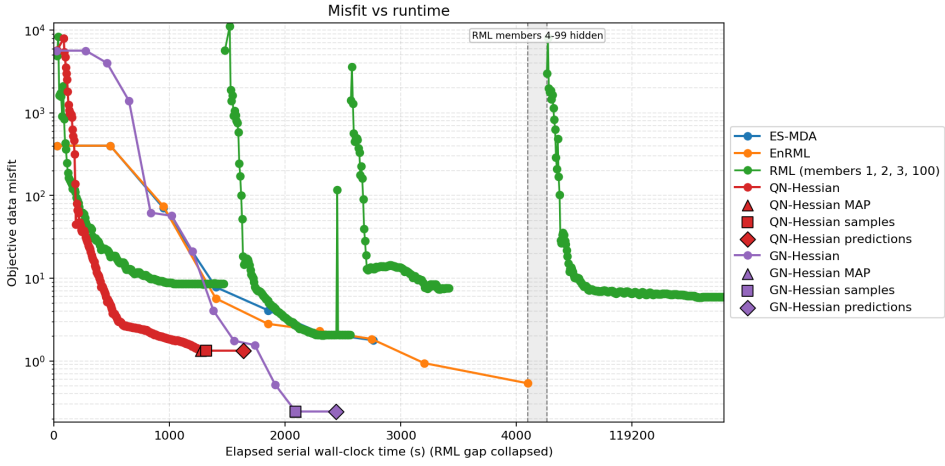
Wall-clock time vs. misfit



+ QN-Hessian Laplace (L-BFGS).

Results

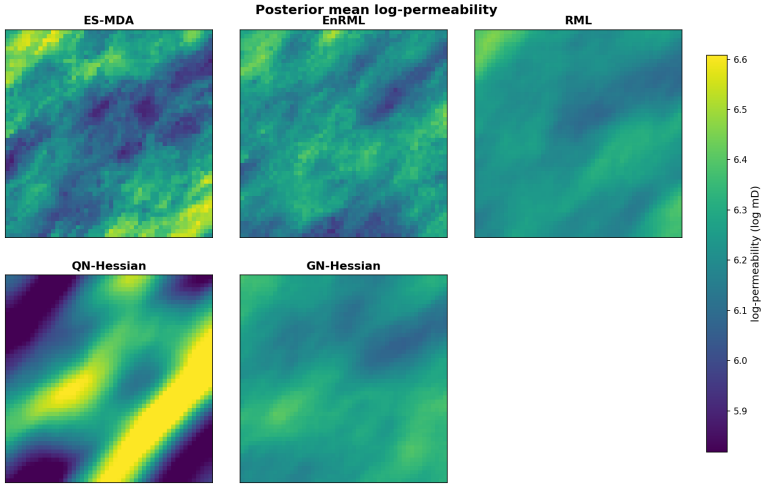
Wall-clock time vs. misfit



+ GN-Hessian Laplace (exact Gauss–Newton).

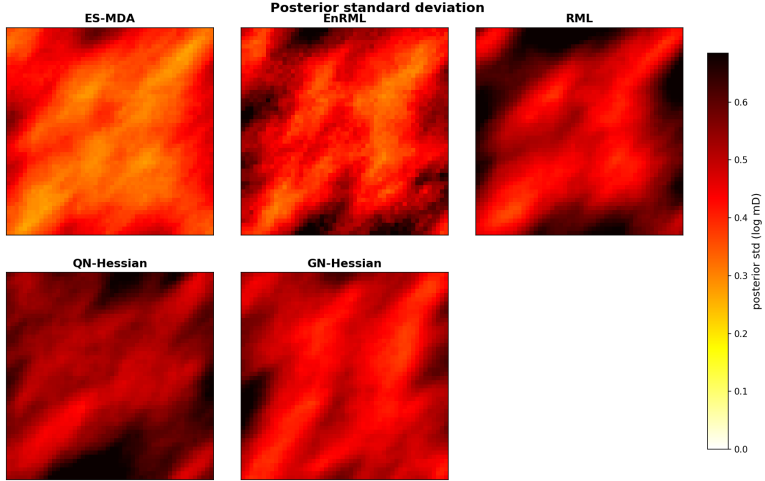
Results

Posterior mean



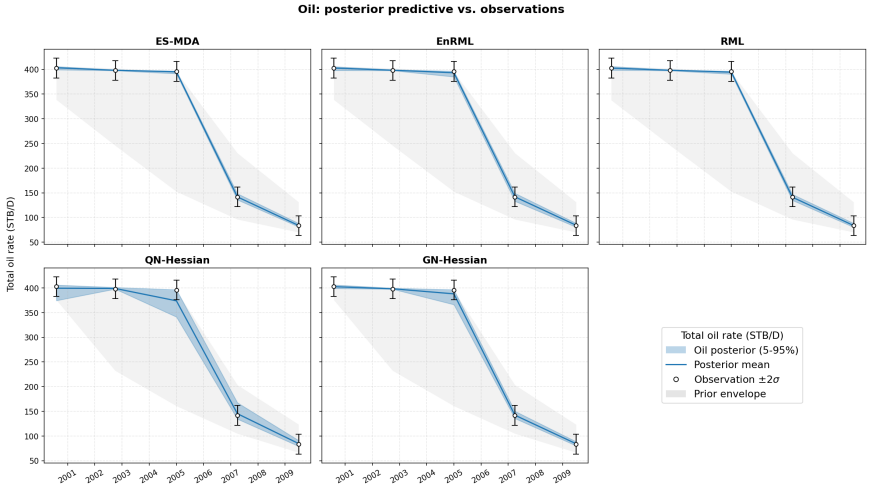
Results

Posterior spread



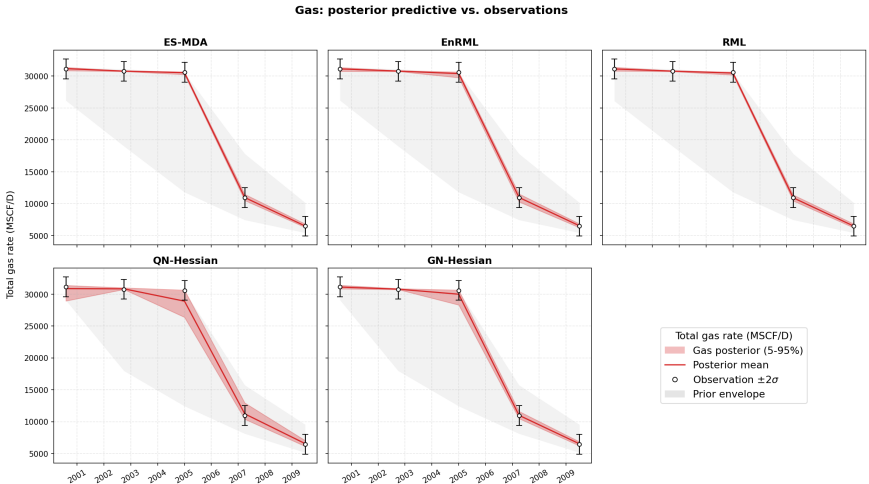
Results

Data match — oil



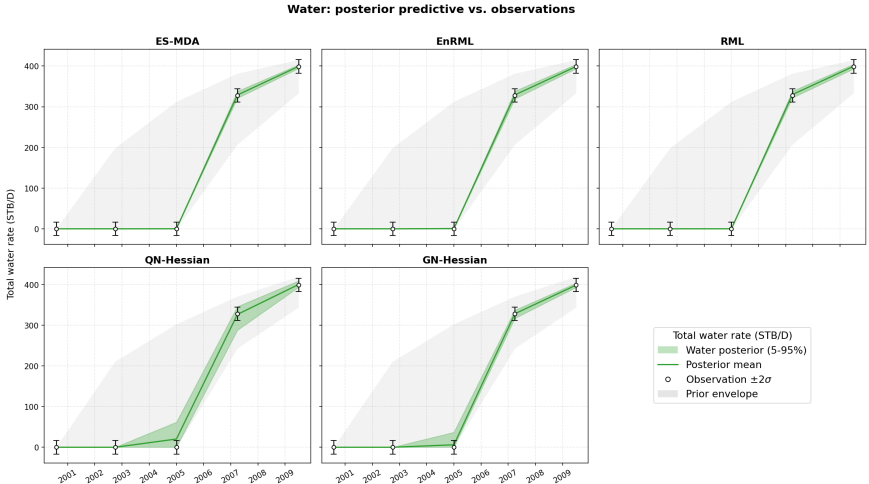
Results

Data match — gas



Results

Data match — water



Summary and conclusion

- Laplace's approximation gives a Gaussian posterior without localization.
- SPDE / Matérn $\Rightarrow$ sparse prior precision; prior samples easily available.
- JutulDarcy adjoints $\Rightarrow$ MAP and Hessian via Gauss-Newton or L-BFGS.
- Two efficient methods for sampling the Hessian.

Summary and conclusion

- Laplace's approximation gives a Gaussian posterior without localization.
- SPDE / Matérn $\Rightarrow$ sparse prior precision; prior samples easily available.
- JutulDarcy adjoints $\Rightarrow$ MAP and Hessian via Gauss-Newton or L-BFGS.
- Two efficient methods for sampling the Hessian.
- Comparable results to RML,
- ensemble methods are competitive but depend on localization.

Future work

- larger 3D field cases,
- low-rank Hessian for Gauss-Newton,
- implementation of adjoint solver in OPM-flow
- optimize choice of rank in Jacobian

The authors acknowledge financial support from the Knowledge-building Project "makeSENSE – Ensemble based data assimilation with exact SENSitivities" which is funded by industry partners Equinor Energy AS, and A/S Norske Shell, as well as the Research Council of Norway.