Multi Level MCMC methods

Andrea Boselli
Carlo Ghiglione
Eleonora Spizzi
Erica Manfrin
Randeep Singh

January 10^{th} 2022

Politecnico di Milano,
joint project Bayesian Statistics - Computational Statistics
Tutors:
Prof. Alessandra Guglielmi - Department of Mathematics, Politecnico di Milano
Prof. Andrea Manzoni - Department of Mathematics, Politecnico di Milano

The framework: differential problem

Consider an **equation** (PDE/ODE) depending on some **unknown parameters** $\underline{\theta} \in \Theta$, defined in a domain Ω.

Let u be the **solution** of such equation:

$$u : \Theta x \Omega \to \mathbb{R}$$
$$(\underline{\theta}, \underline{x}) \mapsto u(\underline{\theta}, \underline{x})$$

Suppose we observe $\{(\underline{x}_i, y_i)\}_{i=1}^{N}$ with:

- $\underline{x}_i \in \Omega$
- $y_i = u(\underline{\theta}, \underline{x}_i) + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \tau^2)$

$\Rightarrow$ **likelihood:** $y_i|\underline{\theta}, \tau \overset{\text{ind}}{\sim} \mathcal{N}(u(\underline{\theta}, \underline{x}_i), \tau^2)$ with $\tau^2 > 0$

Model a priori knowledge about $\underline{\theta}$ through a **prior** distribution $\pi(\underline{\theta})$, and estimate the **posterior** $\pi(\underline{\theta}|\underline{y})$.

The method: Multi Level MCMC

Computing **likelihood** $L(\underline{\theta}^*, \underline{x})$ requires solution $u(\underline{\theta}^*, \underline{x})$ (or an approximation $u^{(\ell)}(\underline{\theta}^*, \underline{x})$): this is **computationally intensive**.

MLMCMC solves the equation at different **levels of accuracy**.

At each level ℓ, being $u^{(\ell)}(\underline{\theta}, \underline{x})$ the numerical solution computed at that level, we set the following likelihood:

$$y_i | \underline{\theta}, \tau \overset{\text{ind}}{\sim} \mathcal{N}(u^{(\ell)}(\underline{\theta}, \underline{x}_i), \tau)$$

$$\text{COARSE LEVEL} \xrightarrow{\text{proposes samples to}} \text{FINE LEVEL}$$

COARSE LEVEL	FINE LEVEL
• fast	• slow
• less accurate	• more accurate

Achieve more efficient samplings:

- **higher** ESS and **ESS/sec**
- **less correlated** samples

It is implemented in PyMC3 Python library through **MLDA** algorithm.

Case studies

A. PDE: **Comet Equation**

   A.1 Coarse and fine model differ in the mesh refinement

   A.2 Coarse level features a NN as a surrogate model

B. ODE: **SEIR Epidemiological Model**

   B.1 Coarse and fine model differ in the solver time-step

   B.2 Coarse level merges two compartments of the fine model (E,I) $\Rightarrow$ SIR model

A. Comet Equation

It is a linear **advection-diffusion PDE**, featuring **2 parameters**, each with a clear physical interpretation:

Comet Equation

$$\begin{cases} -\mu \Delta u + 10(\cos\theta, \sin\theta) \cdot \nabla u = 10 e^{-50\|\underline{x}-\underline{x}_0\|_2} & \underline{x} \in \Omega = [0,1]^2 \\ u = 0 & \underline{x} \in \partial\Omega \end{cases}$$

- $\mu \in (0, \infty)$: **diffusion** parameter
- $\theta \in (0, 2\pi)$: angle of **advection** term
- $\underline{x}_0 = (0.5, 0.5)$: centre of the **forcing bump**

Data is produced through **simulation** with fixed and known values of parameters and error scale:

$$\mu^* = 2 \qquad \theta^* = \pi \qquad \tau^* = 10^{-4}$$

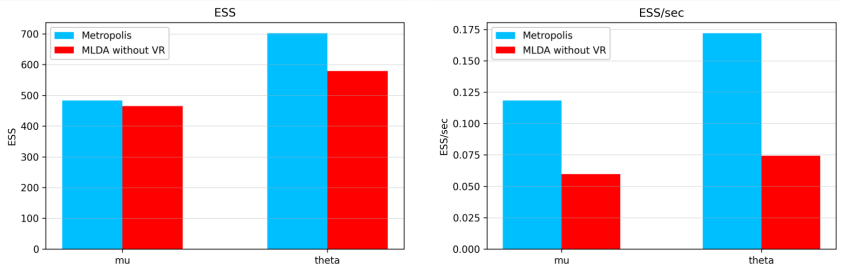
A1. Comet with Different Grids

$u(\underline{\theta}, \underline{x})$ is approximated through *Finite Element Method* (FEM). The **mesh refinement** affects the **accuracy** of the numerical solution.

In MLDA, coarse and fine model **differ** for the mesh **refinement**.

Comparison of:

- **Metropolis**: 32x32 grid
- **MLDA**:
  - Coarse model: 16x16 grid
  - Fine model: 32x32 grid



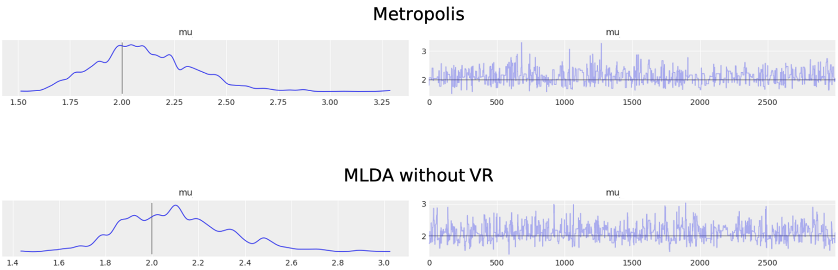
A1. Comet with Different Grids

$u(\underline{\theta}, \underline{x})$ is approximated through *Finite Element Method* (FEM). The **mesh refinement** affects the **accuracy** of the numerical solution.

In MLDA, coarse and fine model **differ** for the mesh **refinement**.

Comparison of:

- **Metropolis**: 32x32 grid
- **MLDA**:
  - Coarse model: 16x16 grid
  - Fine model: 32x32 grid



A2. Comet with Surrogate Model

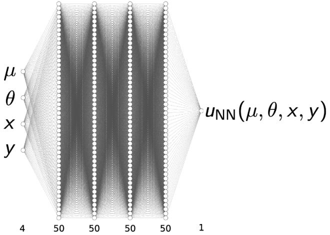
Metropolis overcomes **MLDA** evidently. Indeed, computing the coarse solution in MLDA is too **time consuming**.

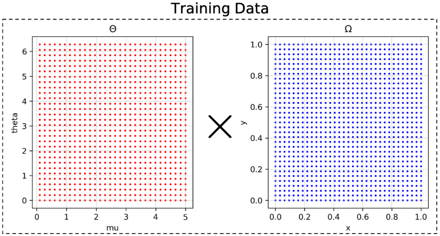
Hence, we place at coarse level a **surrogate model**, implemented through a **Neural Network**:

$$u_{\mathrm{NN}} : (\mu, \theta, x, y) \mapsto u_{\mathrm{NN}}(\mu, \theta, x, y)$$

- Long training time

+ Small execution time





A2. Comet with Surrogate Model

u_{NN} is **trained** on a dataset of **900 PDE solutions**, each corresponding to a different (μ_i, θ_i), evaluated on a grid of Ω.

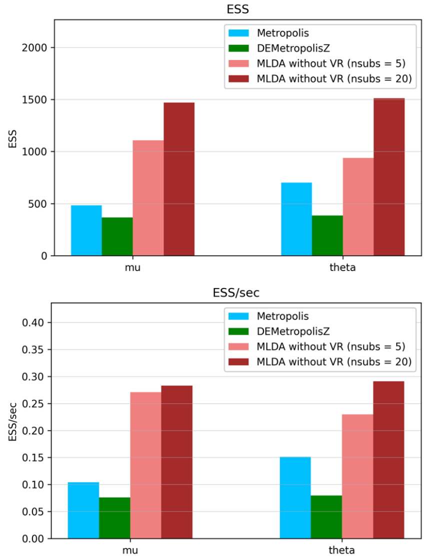
A first training with *Adam* optimizer is followed by a second training with *Stochastic Gradient Descent* (SGD).

Within this framework for MLDA, we investigate the performance of **Metropolis**, **DEMetropolisZ** and **MLDA** for different:

1. **frequencies of subsampling** (nsubs) of proposed samples from coarse level to fine level in **MLDA**
2. **grids** of physical points where data is available
3. choices of the **priors** for (μ, θ)

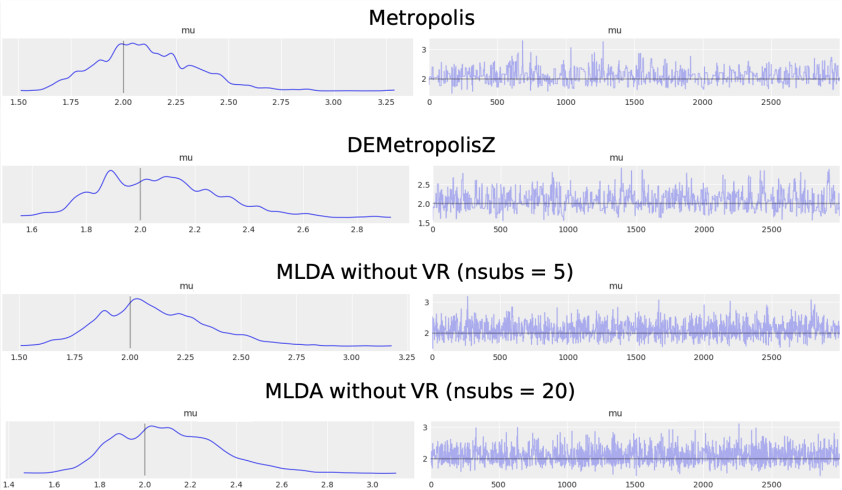
A2. Comet with Surrogate Model

Comparison of Metropolis, DEMetropolisZ and MLDA (nsubs = 5, 20):



A2. Comet with Surrogate Model

Comparison of Metropolis, DEMetropolisZ and MLDA (nsubs = 5, 20):



B. SEIR Epidemiological Model

It is a compartmental model, often used as a backbone in the analysis of infectious diseases. We have four compartments: **Susceptible, Exposed, Infectious, Recovered**.

SEIR Differential Model

$$\begin{cases} S'(t) = -\beta SI \\ E'(t) = \beta SI - \sigma E \\ I'(t) = \sigma E - \gamma I \\ R'(t) = \gamma I \\ S(0) = S_0, E(0) = E_0, I(0) = I_0, R(0) = R_0 \end{cases}$$

$$\mathrm{S(t) + E(t) + I(t) + R(t) = 1}$$

- $\beta \in (0, \infty)$: **Infection** rate
- $\sigma \in (0, \infty)$: **Incubation** rate
- $\gamma \in (0, \infty)$: **Recovery** rate

B. SEIR Framework

Let us consider:

- $\underline{\theta} := (\beta, \sigma, \gamma)$ model parameters (unknown)
- $t \in [0, T] \subset [0, \infty)$ time observation window
- $\underline{u}(\underline{\theta}, t) := (E(\underline{\theta}, t), I(\underline{\theta}, t), R(\underline{\theta}, t))$ solution of the differential system

Data are $(t_i, \underline{y}_i)$ where:

- $t_i \in [0, T]$
- $\underline{y}_i = \underline{u}(\underline{\theta}, t) + \underline{\epsilon}_i,\ \underline{\epsilon}_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \tau^2 \mathbb{I}_3) \Rightarrow \underline{y}_i | \underline{\theta} \overset{\text{ind}}{\sim} \mathcal{N}(\underline{u}(\underline{\theta}, t_i), \tau^2 \mathbb{I}_3)$ where τ^2 is the random noise variance

Data are generated through **simulation** with known values for the parameters and the error scale, and are observed on a specific time grid $\{t_i\}_{i=1}^n$ uniformly distributed on $[0, T]$ with time step Δt. We set:

$$\underline{\theta}^* = (0.27, 0.2, 0.1),\ T^* = 100, \Delta t^* = 0.2, \tau^* = 0.01$$

B1. SEIR MLMCMC setting

The goal is to perform bayesian inference on the **model parameters** $\underline{\theta}$ and on the **error scale** τ.
We fix a **Uniform prior** for each model parameter and a **Half-Cauchy prior** for τ.

In **MLDA**, coarse and fine models differ in the **time step Δt**. Our standard setting is:

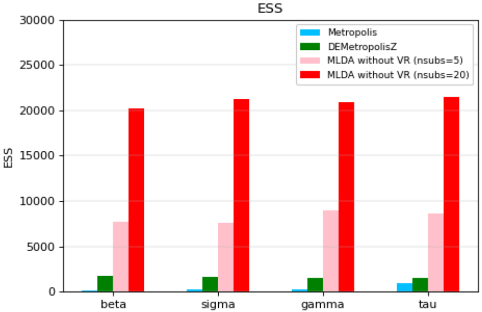
- $\Delta t_{\text{fine}} = 0.2$
- $\Delta t_{\text{coarse}} = 2\Delta t_{\text{fine}}$

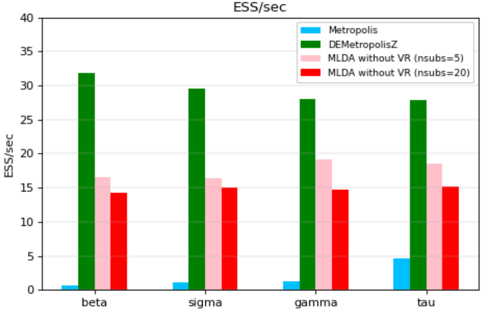
B1. SEIR MLMCMC Setting

Within this framework, we investigate the performance of **Metropolis**, **DEMetropolisZ** and **MLDA** for different:

1. **frequencies of subsampling** (nsubs) of proposed samples from coarse level to fine level in MLDA
2. **ratio** between coarse and fine **time steps** Δt in MLDA
3. choices of the **priors** for $\underline{\theta}, \tau$

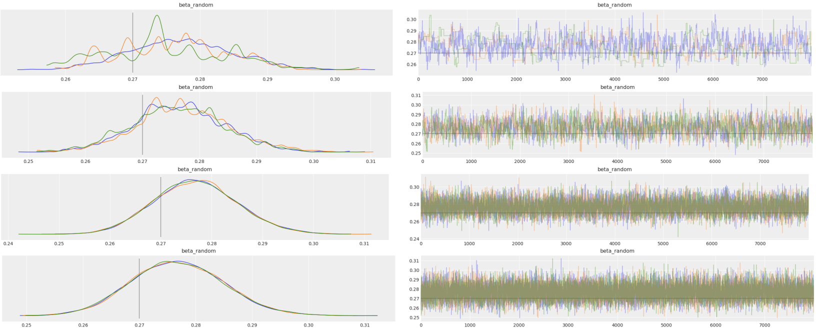
B1. Comparison nsubs=5,20





B1. Posteriors and Trace plots

Posteriors and *trace plots* for β using respectively **Metropolis, DEMetropolisZ, MLDA(nsubs=5), MLDA(nsubs=20)**.



Conclusive steps

The conclusive steps we are investigating are:

1. use of **historically informed priors** for the epidemiological model for **data scarcity frameworks**;
2. use of **different mathematical models** for the levels of the **MLDA** algorithm.

Bibliography

- T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup, *Multilevel Markov Chain Monte Carlo* SIAM Review 61(3), pp. 509–545, 2019.

- T. J. Dodwell, C. Ketelsen, R. Scheichl, andA. L. Teckentrup, *A Hierarchical Multilevel Markov Chain Monte Carlo Algorithm with Applications to Uncertainty Quantification in Subsurface Flow* SIAM/ASA J. Uncertainty Quantification 3, pp. 1075–1108, 2015.

- Parth Vipul Shah, *Prediction of the Peak, Effect of Intervention, and Total Infected by COVID-19 in India.* Cambridge University Press: 09 September 2020

- Heng, K., Althaus, C.L. *The approximately universal shapes of epidemic curves in the Susceptible-Exposed-Infectious-Recovered (SEIR) model*. Sci Rep 10, 19365 (2020).

Bibliography

- Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G. E. (2021). *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*. Nature Machine Intelligence, 3(3), 218-229.

- Python PyMC3 package (providing an implementation for MLMCMC and a few examples)

- FEniCS (providing an easy implementation for the discretization and solution of PDEs through Finite Element Methods (FEM))

Appendix: B1. Artificial Data