Multi Level MCMC

Andrea Boselli
Carlo Ghiglione
Eleonora Spizzi
Erica Manfrin
Randeep Singh

February 15th 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

Applicative interest of the project

Natural phenomena & Engineering applications	$\rightarrow$	Differential Models (**PDEs** and **ODEs**) depending on **unknown parameters**

Parameter Estimation: estimate the unknown parameters from a noisy and partial observation of the solution.

This problem can be tackled applying the **Bayesian framework**.

Differential Models and Bayesian framework

Consider a Differential Model defined in a **physical domain** Ω depending on some **unknown parameters** $\underline{\theta} \in \Theta$.

Let u_h be a **numerical solution** of such model:

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

Observed data: $\mathcal{D} := \{\underline{x}_i\}_{i=1}^N \subset \Omega, \quad \underline{y} := \{\mathrm{y}_i\}_{i=1}^N$

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

Bayesian Approach:

- Model a priori knowledge about $\underline{\theta}$ through a **prior** distribution $\pi(\underline{\theta})$
- Compute the **likelihood**: $y_i | \underline{\theta}, \tau \overset{\text{ind}}{\sim} \mathcal{N}(u_h(\underline{\theta}, \underline{x}_i), \tau^2)$ with $\tau^2 > 0$
- Estimate the **posterior** $\pi(\underline{\theta} | \underline{y})$

Multi Level MCMC method

Computing **likelihood** $L(\mathcal{D}, \underline{y} \mid \underline{\theta})$ requires a numerical solution $u_h(\underline{\theta}, \underline{x})$: this is **computationally intensive**.

In **MLMCMC** the model is solved at different **levels of accuracy**.

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

$$y_i|\underline{\theta}, \tau \overset{\text{ind}}{\sim} \mathcal{N}(u_{h_\ell}(\underline{\theta}, \underline{x}_i), \tau^2)$$

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

COARSE LEVEL

- fast
- less accurate

FINE LEVEL

- slow
- more accurate

Achieve more efficient samplings:

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

Case studies and applicative notes

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

Applicative notes:
MLMCMC is implemented in *PyMC3* Python library through **MLDA** (*Multi Level Delayed Acceptance*) algorithm.
It guarantees **Variance Reduction** properties to estimate **Quantities of Interest**, but we will focus on MLDA as a *sampling method*.

A. Application to a PDE: the Comet Equation

A. Comet Equation

It is a linear **Advection-Diffusion PDE**, featuring **2 parameters** (μ, θ), each with a clear physical interpretation:

Comet Equation

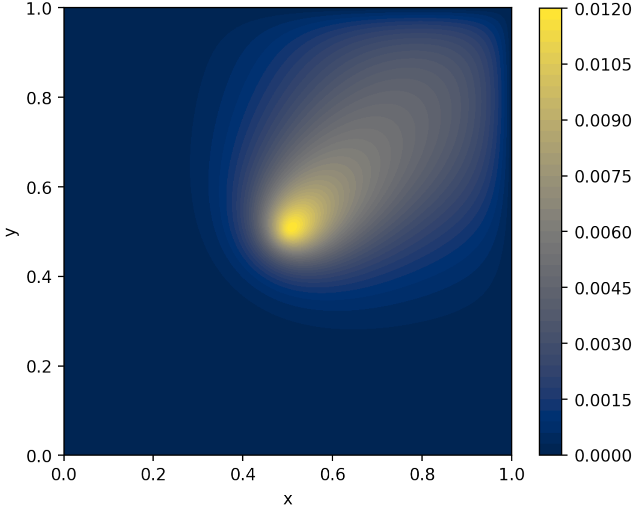
$$\begin{cases} -\mu \Delta u + 10(\cos\theta, \sin\theta) \cdot \nabla u = 10 e^{-50\|\underline{x} - \underline{x}_0\|_2} & \underline{x} \in \Omega = [0,1]^2 \\ u = 0 & \underline{x} \in \partial\Omega \end{cases}$$

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

True values: $\mu^* = 2$ $\quad\theta^* = \pi$ $\quad\tau^* = 10^{-4}$

Priors: $\mu \sim \mathcal{U}(0.1, 5)$ $\quad\theta \sim \mathcal{U}(0, 2\pi)$

A. Comet Equation



Plot of the solution for $\mu = 0.5$ and $\theta = \frac{\pi}{4}$

A.2. Surrogate Model

Fine level: FEM Solver (32x32 grid)

Coarse level:

A) FEM solver (16x16 grid)

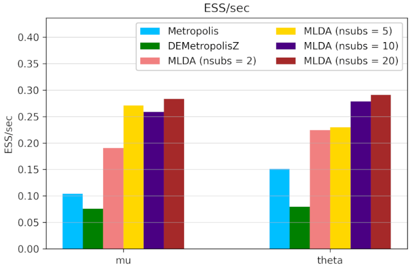
+ No training
+ Parameters in any range
– Slow execution

No improvement in efficiency.

B) Surrogate Model (Neural Network)

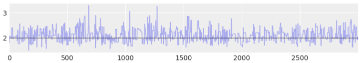
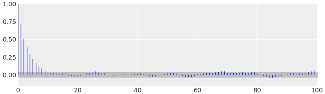
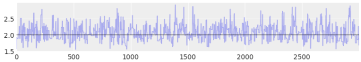
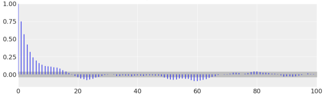
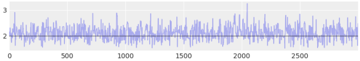
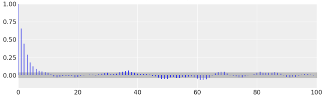
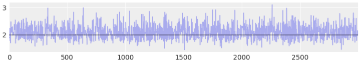
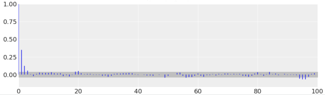
– Long training
– Parameters in specific range
+ Fast execution (x10)

Much more efficient!



The most efficient sampling for MLDA is with *subsampling rate* $n_{subs} = 20$.

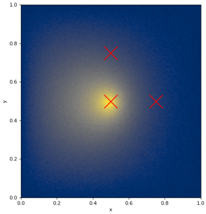
A.2. Surrogate Model

	Trace	Autocorrelation
(a)		
(b)		
(c)		
(d)		

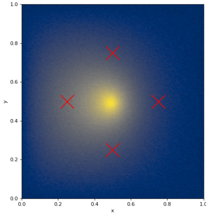
Traces (left) and Autocorrelation (right) for μ with MH (a), DEMZ (b), MLDA with $n_{subs} = 2$ (c) and MLDA with $n_{subs} = 20$ (d).

A.2. Data collection grid

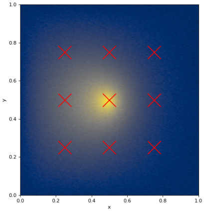
- Let $\mathcal{D}$ be the set where $u_h(\underline{\theta}, \cdot)$ is observed, up to an additive **Gaussian noise**.
- We inspect the **effect of the choice** of $\mathcal{D}$ on the **inference** of μ, θ: detect the **most informative points** w.r.t. each parameter.
- 3 sets are compared:



$\mathcal{D}_1$

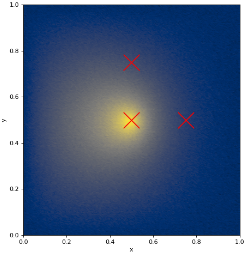


$\mathcal{D}_2$

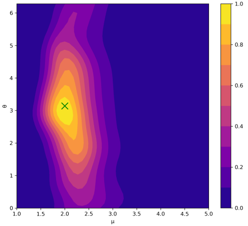


$\mathcal{D}_3$

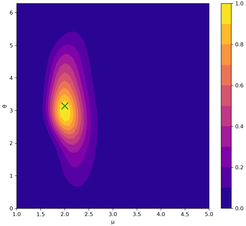
A.2. Set with 3 points



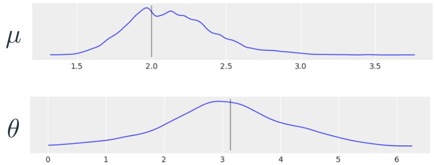
Set $\mathcal{D}_1$



Likelihood at $\ell = 0$



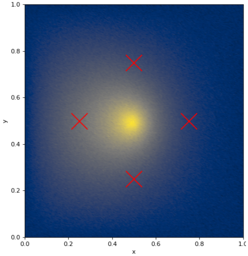
Likelihood at $\ell = 1$



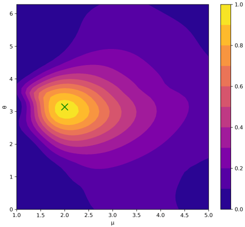
Posteriors estimated through MLDA.

p	$\mathbb{E}[p]$	$sd(p)$
μ	2.137	0.294
θ	3.087	1.139

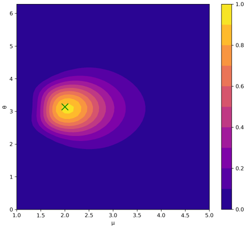
A.2. Set with 4 points



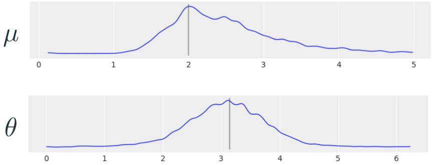
Set $\mathcal{D}_2$



Likelihood at $\ell = 0$



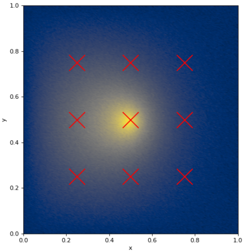
Likelihood at $\ell = 1$



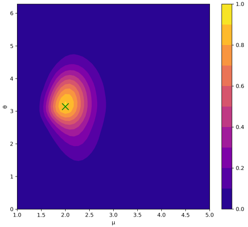
Posteriors estimated through MLDA.

p	$\mathbb{E}[p]$	$sd(p)$
μ	2.485	0.708
θ	3.094	0.753

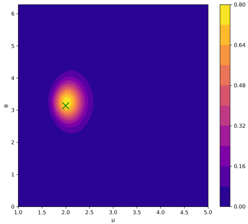
A.2. Set with 9 points



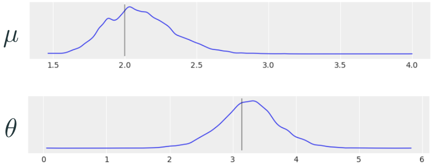
Set $\mathcal{D}_3$



Likelihood at $\ell = 0$



Likelihood at $\ell = 1$



Posteriors estimated through MLDA.

p	$\mathbb{E}[p]$	$sd(p)$
μ	2.108	0.234
θ	3.253	0.482

B. Application to an ODE:
the SEIR Epidemiological Model

B. SEIR Epidemiological Model

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

SEIR Differential Model

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

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

- $\beta \in (0, \infty)$: **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)
- $t \in [0, T]$ **time observation window**
- $\underline{u}_h(\underline{\theta}, t) := (E_h(\underline{\theta}, t), I_h(\underline{\theta}, t), R_h(\underline{\theta}, t))$ **numerical solution** of the differential system

Data are $\{(t_i, \underline{y}_i)\}_{i=1}^N$:

- $t_i \in [0, T]$
- $\underline{y}_i = \underline{u}_h(\underline{\theta}, t_i) + \underline{\epsilon}_i, \quad \underline{\epsilon}_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \tau^2 \mathbb{I}_3) \quad \Rightarrow \quad \underline{y}_i | \underline{\theta} \overset{\text{ind}}{\sim} \mathcal{N}(\underline{u}_h(\underline{\theta}, t_i), \tau^2 \mathbb{I}_3)$

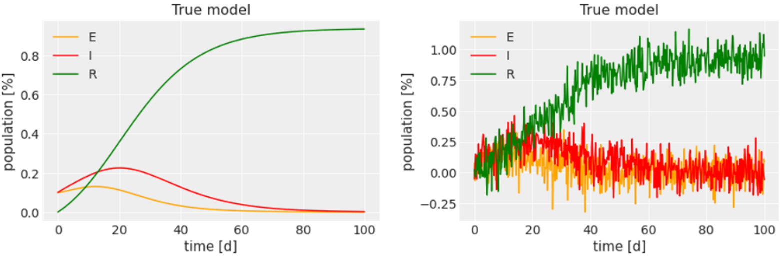
Data are **simulated** on uniform time grid on $[0, T]$ with step Δt.

True values: $\underline{\theta}^* = (0.27, 0.2, 0.1)$ from **Covid-19 infection in India**
$\tau^{*2} = 0.01$

B. SEIR data

Population is **normalized**.

Initial proportions of the four compartments are set as:
I(0)=0.1,   E(0)=0.1,   R(0)=0.



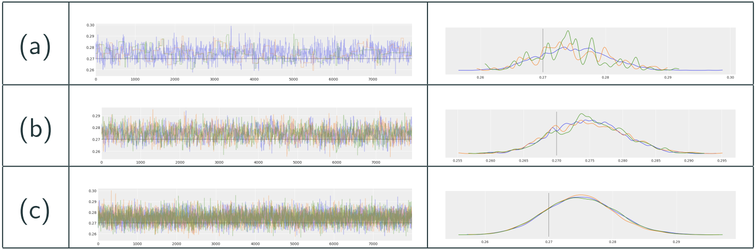
SEIR solution without (left) and with (right) noise.

B.1. SEIR-SEIR Settings

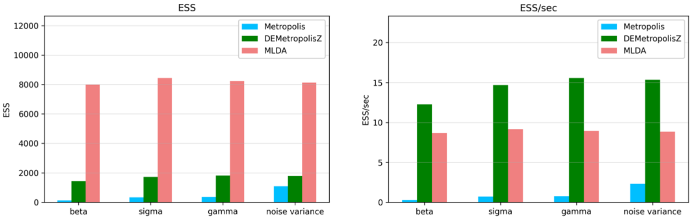
- **Subsampling rate**: $n_{subs} = 10$
- **Time grid**: $\Delta t_{fine} = 0.2$ and $\Delta t_{coarse} = 2\Delta t_{fine}$
- **Priors**: distributions from the **literature**; hyper-parameters such that means coincide with the true values:

$$\pi(\beta) \sim \mathit{LogNormal}(\mu = -1.31, \sigma = 0.037)$$
$$\pi(\sigma) \sim \mathit{LogNormal}(\mu = -1.61, \sigma = 0.04997)$$
$$\pi(\gamma) \sim \mathit{Weibull}(\alpha = 2, \beta = 0.12)$$
$$\pi(\tau^2) \sim \mathit{HalfCauchy}(0, 1)$$

B.1. SEIR-SEIR Results



Traces (left) and Posterior (right) for β with MH (a), DEMZ (b), MLDA (c).



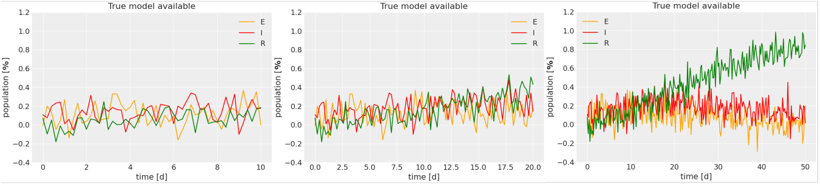
ESS and ESS/sec of the three different methods.

B.1. Prior informed on Historical Data

Framework:

- Data from **old** epidemic **fully available**;
- Data from **new** epidemic only **partially observed**.

Objective: estimate infectious **Peak Time** and **Peak Value**.



Data from new epidemic up to time $T = 10$ (left), 20 (center), 50 (right).

Remark: Old epidemic is simulated with infectious rate β^* 25% higher than new epidemic.

B.1. Prior informed on Historical Data

Procedure:

1. Parameter estimation on **historical data** (find posteriors);
2. Set **new priors** *s.t.* their means coincide with the ones of the estimated posteriors, variances analogously, up to a scaling factor.
3. Parameter estimation on **new partially observed data**.

Results: (Peak Time = 20.12, Peak Value = 0.225)

T	Peak Time	SE	Peak Value	SE
10	19.94	$7.47 \cdot 10^{-3}$	0.243	$2.33 \cdot 10^{-4}$
20	19.81	$5.45 \cdot 10^{-3}$	0.245	$1.34 \cdot 10^{-4}$
50	20.33	$3.25 \cdot 10^{-3}$	0.233	$8.56 \cdot 10^{-5}$

- If T increases $\Rightarrow$ SE decreases.
- Very **precise estimates** even from early stages of epidemic.

B.1. Prior informed on Historical Data

Comparison *w.r.t.* other methods ($T = 10$):

Method	Peak time SE	Peak value SE
MH	$1.86 \cdot 10^{-2}$	$4.63 \cdot 10^{-4}$
DEMZ	$1.165 \cdot 10^{-2}$	$5.05 \cdot 10^{-4}$
MLDA	$7.47 \cdot 10^{-3}$	$2.33 \cdot 10^{-4}$

For the same number of iterations, MLDA provides much **lower** SE
$\Rightarrow$ **more precise estimates**.

B.2. SIR model at coarse level

Low Fidelity Model as **Surrogate Model** in the coarse chain, trying to improve the **computational efficiency** of MLDA.

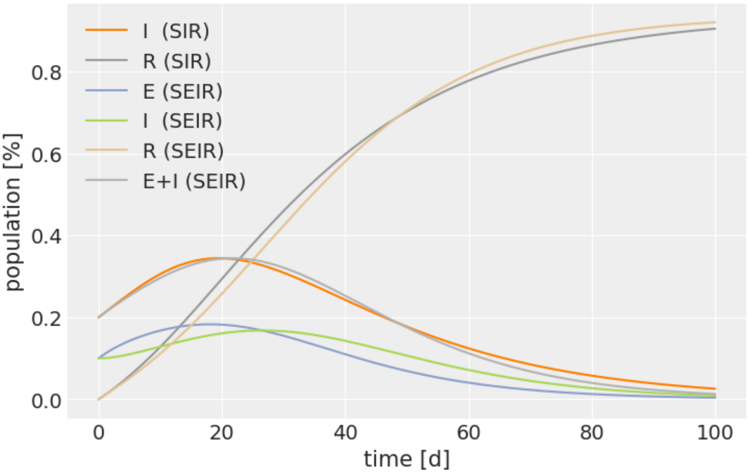
SIR Differential Model

$$\begin{cases} \tilde{S}'(t) = -\tilde{\beta}\tilde{S}\tilde{I} \\ \tilde{I}'(t) = \tilde{\beta}\tilde{S}\tilde{I} - \tilde{\gamma}\tilde{I} \\ \tilde{R}'(t) = \tilde{\gamma}\tilde{I} \\ \tilde{S}(0) = \tilde{S}_0, \tilde{I}(0) = \tilde{I}_0, \tilde{R}(0) = \tilde{R}_0 \end{cases}$$

Given a SEIR model, the SIR model that **best approximates** it is obtained by choosing:

$$\begin{cases} \tilde{\gamma} = \frac{\sigma}{\sigma+\gamma}\gamma \\ \tilde{\beta} = \frac{\sigma}{\sigma+\gamma}\beta \\ \tilde{S}_0 = S_0, \quad \tilde{I}_0 = I_0 + E_0, \quad \tilde{R}_0 = R_0 = 0 \end{cases}$$

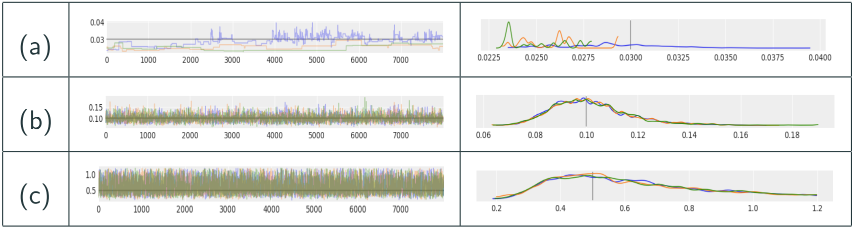
B.2. SIR model at coarse level



B.2. Hypotheses

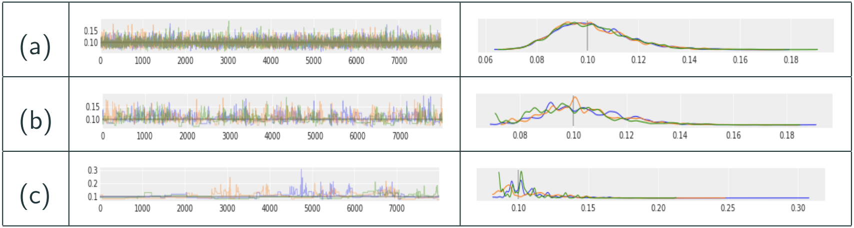
Two **qualitative assumptions** must be met in order to have a good approximation:

1. $\boldsymbol{\sigma}$ **must be in a suitable range**:
   - σ too small $\Rightarrow$ SIR heavily **differ** from SEIR;
   - σ too large $\Rightarrow$ **large posterior variance** of σ, difficult to infer the true parameter.



MLDA runs with $\sigma = 0.035$ (a), $\sigma = 0.1$ (b), $\sigma = 0.5$ (c)

2. $\boldsymbol{I_0}$, $\boldsymbol{\tilde{I}_0 \ll 1}$: technical hypothesis to mathematically **derive the expression** of SIR parameters. The bigger I_0, $\tilde{I}_0$, the less accurate the approximation.

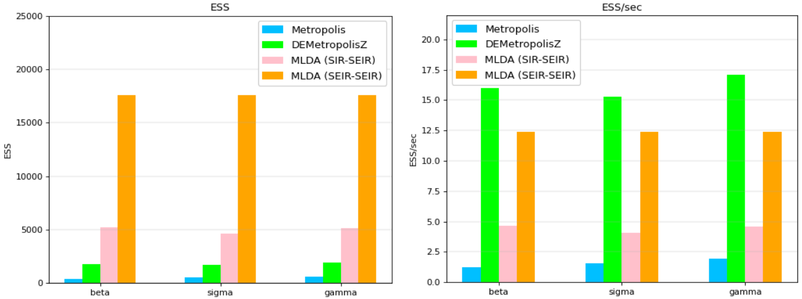


MLDA runs for σ with $(I_0, \tilde{I}_0) = (0.10, 0.20)$ (a), $(I_0, \tilde{I}_0) = (0.25, 0.35)$ (b), $(I_0, \tilde{I}_0) = (0.40, 0.50)$ (c)

B.2. SIR-SEIR vs SEIR-SEIR

SEIR-SEIR is the **clear winner** in terms of ESS and ESS/sec. $\Rightarrow$ The *Surrogate Model* brings **no advantage**.

Interpretation: The *Scipy* ODE solver is really efficient $\Rightarrow$ there is almost **no difference** in solving SIR or SEIR.



Conclusions

MLDA, compared to MH and DEMZ, features:

$\left.\begin{array}{l} + \text{ less correlated samples} \\ + \text{ higher ESS} \\ - \text{ higher computational time} \end{array}\right\}$ **Subsampling** at coarse level

The **less time consuming** is coarse level w.r.t. fine level, the higher the **efficiency** (ESS/sec) of MLDA $\Rightarrow$ Use **Surrogate Models** (true $\underline{\theta}$ must be defined in a given Applicability Range)

PDE case $\rightarrow$ Neural Network *V*

ODE case $\rightarrow$ Low Fidelity Model *X*

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

Cajo J. F. Ter Braak, *A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces*, Stat Comput 16, 239–249, 2006

Quarteroni, Alfio, and Silvia Quarteroni. Numerical models for differential problems. Vol. 2. Milan: Springer, 2009.

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

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

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))