

Parareal computation of SDEs with time-scale separation

Frédéric Legoll¹, Tony Lelièvre¹, Keith Myerscough² and Giovanni Samaey²

¹Ecole des Ponts ParisTech, ²KU Leuven

In a nutshell

- Goal** Simulate slow-fast SDEs over long time, quickly
- Model** Slow-fast system of SDEs, and a macroscopic model taken from the “fast” limit
- Method** Parallel-in-time algorithm that iteratively improves the macroscopic result
- Result** Reduction in wall clock time
- Bonus** Lower variance than full microscopic model

Microscopic model

- Slow-fast system of coupled SDEs

$$dX = (-X_t^3 + X_t + Y_t^2) dt + \sqrt{\frac{2}{\beta}} dB_t^{(x)}$$

$$dY = \frac{1}{\epsilon}(X_t - Y_t) dt + \sqrt{\frac{2}{\beta\epsilon}} dB_t^{(y)}.$$

- Modeled as an ensemble \mathcal{X}_t^p of particles with positions (X_t^p, Y_t^p) and weight W^p
- Time integrator: a Lie-Trotter splitting, updating X_t first, then Y_t
- Validation: *deterministic* solution given by the Fokker-Planck equation, akin to the macroscopic model

Macroscopic model

- Only slow variable, *assume* the fast Y_t is equilibrated and use only the expected value of the term Y_t^2

$$dZ = (-Z_t^3 + Z_t + Z_t^2 + \frac{1}{\beta}) dt + \sqrt{\frac{2}{\beta}} dB^{(x)},$$

or in potential form

$$dZ = -\partial_z V_{\text{eff}}(Z_t) dt + \sqrt{\frac{2}{\beta}} dB^{(x)}.$$

- The associated Fokker-Planck equation reads

$$\partial_t \rho(z) = \partial_z (\rho(z) \partial_z V_{\text{eff}}) + \frac{1}{\beta} \partial_{zz} \rho(z).$$

- The macroscopic state is represented by integral quantities over a regular grid

Coupling

Restriction (\mathcal{R} , from micro to macro) sum the weights of all particles in each bin

Matching (\mathcal{M} , from macro to micro) reweight particles from a known microstate $(\bar{X}^p, \bar{Y}^p, \bar{W}^p)$

Resampling (\mathcal{M}^* , optional) retrieve an ensemble with all particles equal in weight

Parameters used below: $\beta = 5, \epsilon = 0.1, K = N = 20, \Delta t = 0.025, \delta t = 1.0 \times 10^{-4}$

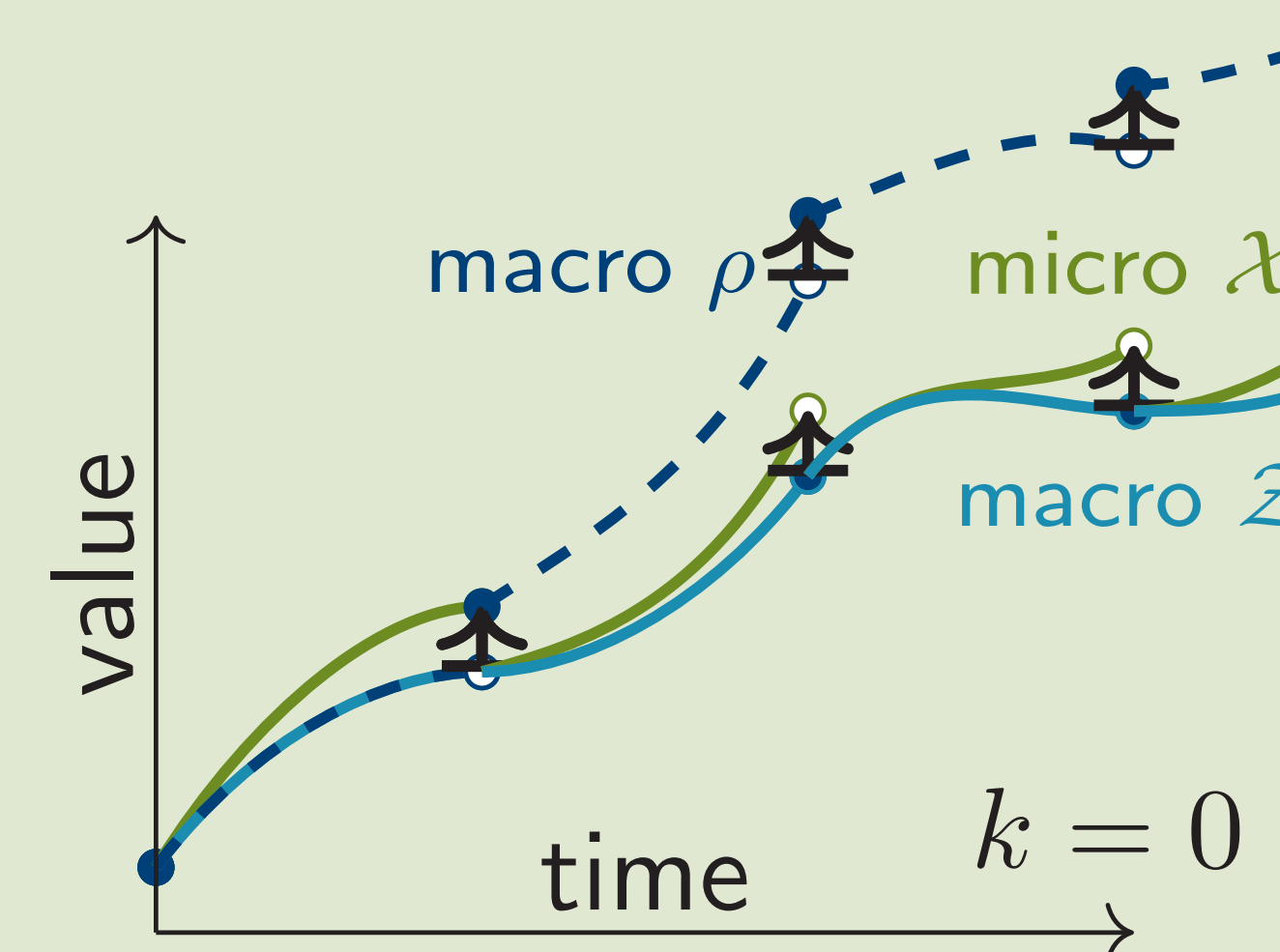
The parareal algorithm

Iteratively improves the macroscopic propagator by computing the discrepancies between the **macroscopic** and the **microscopic** models *in parallel*

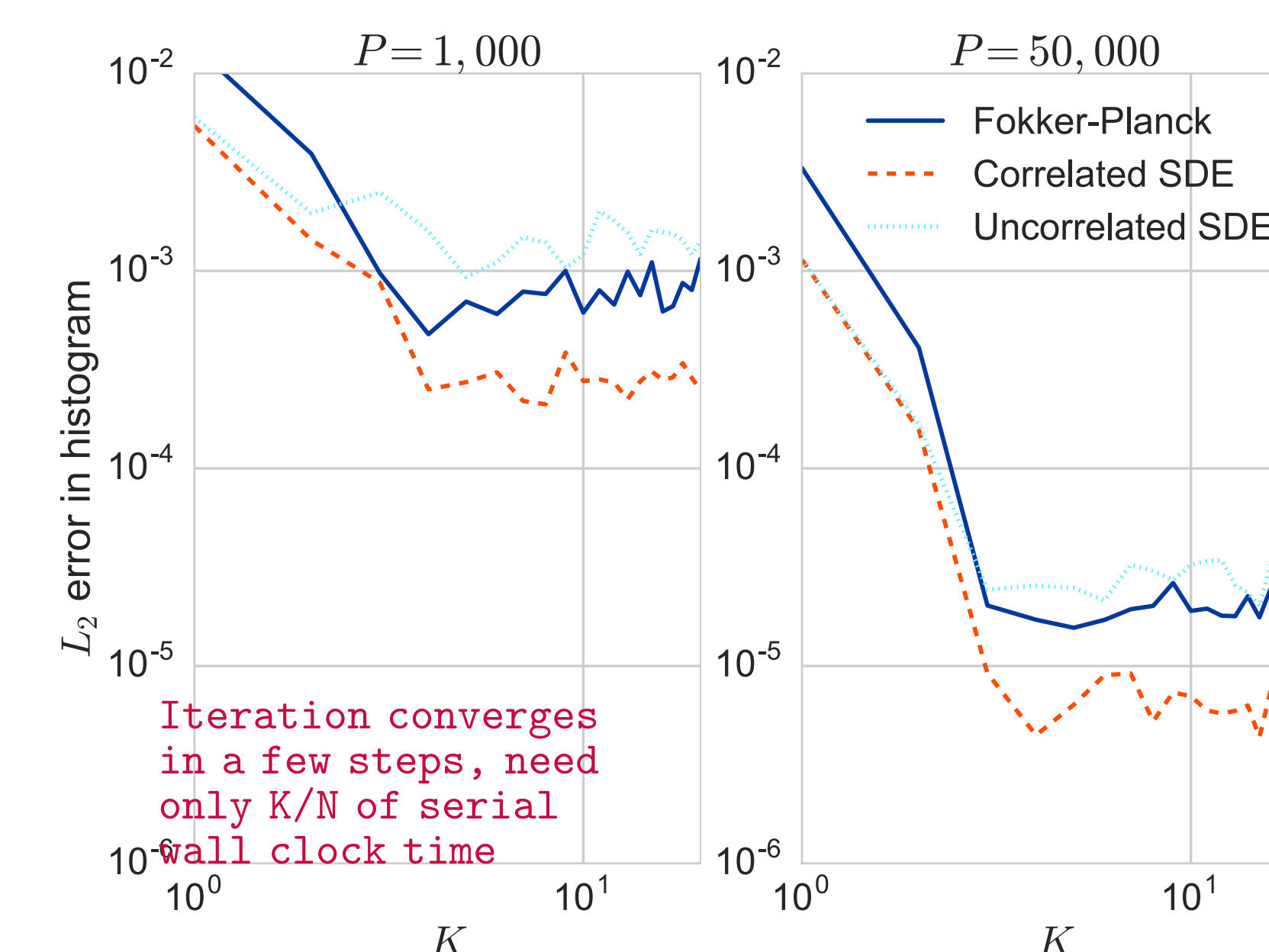
Parallel use of the microscopic propagator gives a reduction in *wall clock time* if there are fewer iterations needed than time steps

Variance of the stochastic microscopic propagator dominates the error (see figures →)

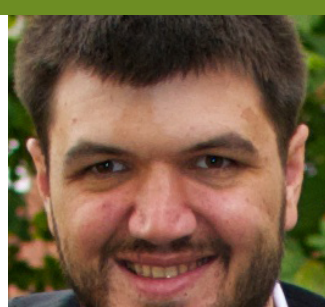
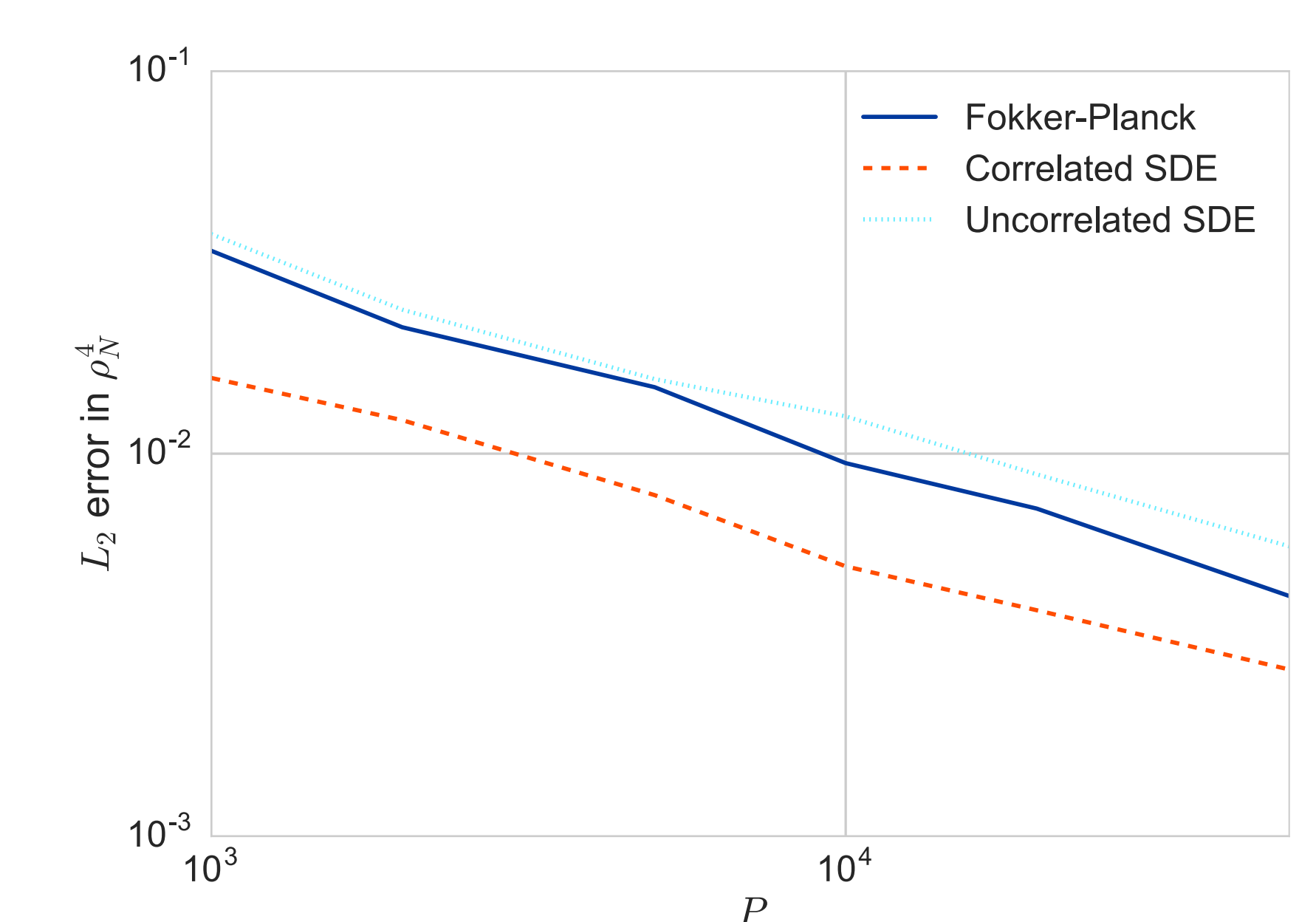
Reduction in variance by using a particle propagator with **correlated noise** for the macroscopic model in computing the discrepancies between the models



Convergence in iterations



Convergence in number of particles



Keith Myerscough
keith.myerscough@kuleuven.be
people.cs.kuleuven.be/~keith.myerscough/

In association with:



Department of Computer Science

