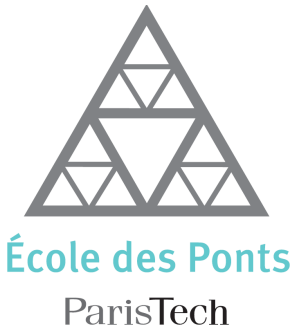


Parareal algorithms for molecular dynamics simulations

Frédéric Legoll

Ecole des Ponts & Inria Paris

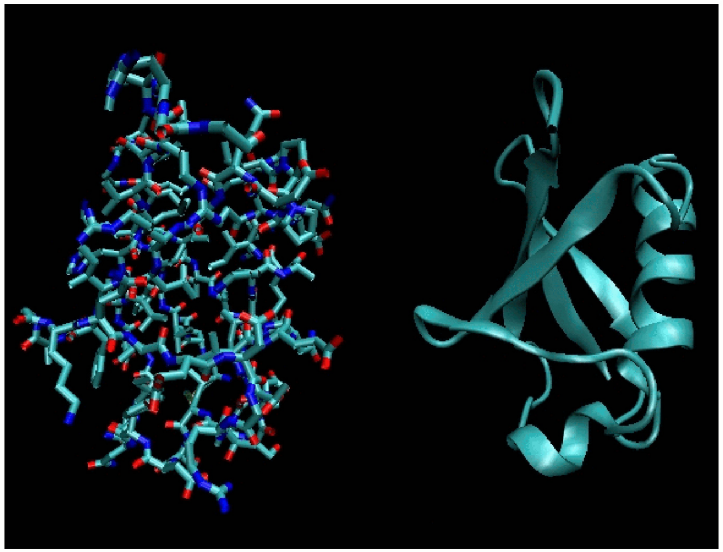
Joint with T. Lelièvre (ENPC), O. Gorynina (ENPC, now in Davos), D. Perez (LANL) and U. Sharma (Berlin, now in Sydney)

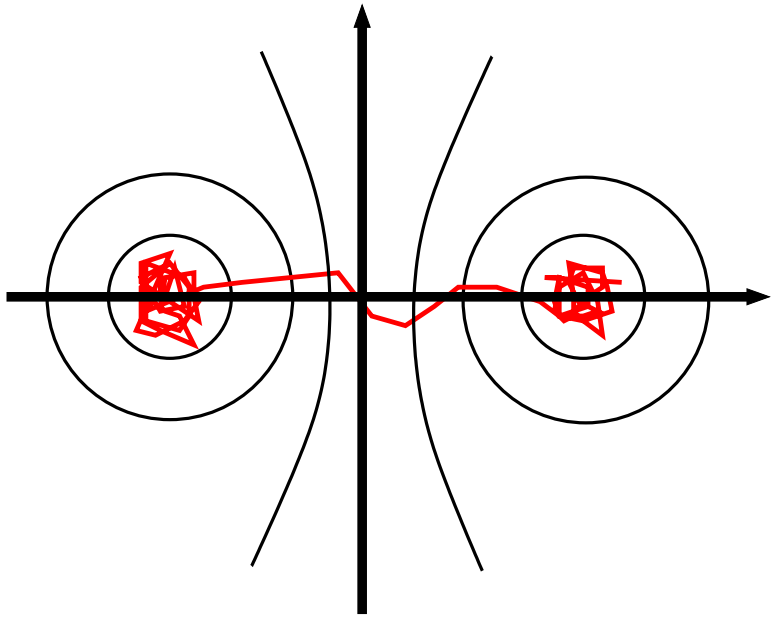
IPAM long program on *New Mathematics for the Exascale: Applications to Materials Science*

Workshop II: Scale-Bridging Materials Modeling at Extreme Computational Scales, April 2023

Motivation

This work is motivated by molecular simulation, where we often have to simulate long trajectories of complex, metastable systems.





Typical dynamics: the Langevin equation

$$dq_t = p_t\,dt, \qquad dp_t = -\nabla V(q_t)\,dt - \gamma\, p_t\, dt + \sqrt{2\gamma\beta^{-1}}\, dW_t$$

Outline

- Since we have to simulate long-time trajectories, it seems attractive to use the parareal algorithm, which solves initial value problems by parallel-in-time computations (domain-decomposition fashion)
- It turns out that this algorithm is not stable for MD problems when the time horizon is too large
- We are therefore going to introduce an adaptive parareal algorithm

F.L., T. Lelièvre and U. Sharma, SIAM Journal on Scientific Computing 2022

O. Gorynina, F.L., T. Lelièvre and D. Perez, arXiv preprint 2212.10508

Parallel in time algorithm for ODEs

Parallel in time algorithm for ODEs

$$\frac{dx}{dt} = f(x), \qquad x \in \mathbb{R}^d$$

The parareal algorithm (Lions, Maday and Turinici, 2001) is based upon two integrators to propagate the system over a time ΔT:

- a fine, accurate integrator $\mathcal{F}_{\Delta T}$
- a cheap coarse integrator $\mathcal{C}_{\Delta T}$

For instance,

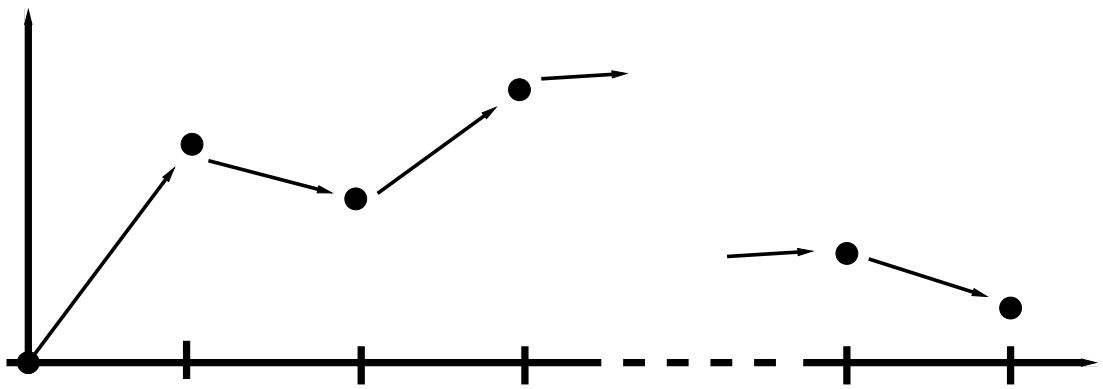
$$\mathcal{F}_{\Delta T} = (\Phi_{\delta t_F})^{\Delta T/\delta t_F} \quad \text{and} \quad \mathcal{C}_{\Delta T} = (\Phi_{\delta t_C})^{\Delta T/\delta t_C} \quad \text{with} \quad \delta t_F \ll \delta t_C$$

where $\Phi_{\delta t}$ is a one time step propagator

The parareal iterative procedure

- Initialization: coarse propagation that yields $\left\{x_n^{k=0}\right\}_n$:

$$\forall n, \quad x_{n+1}^{k=0} = \mathcal{C}_{\Delta T}(x_n^{k=0})$$

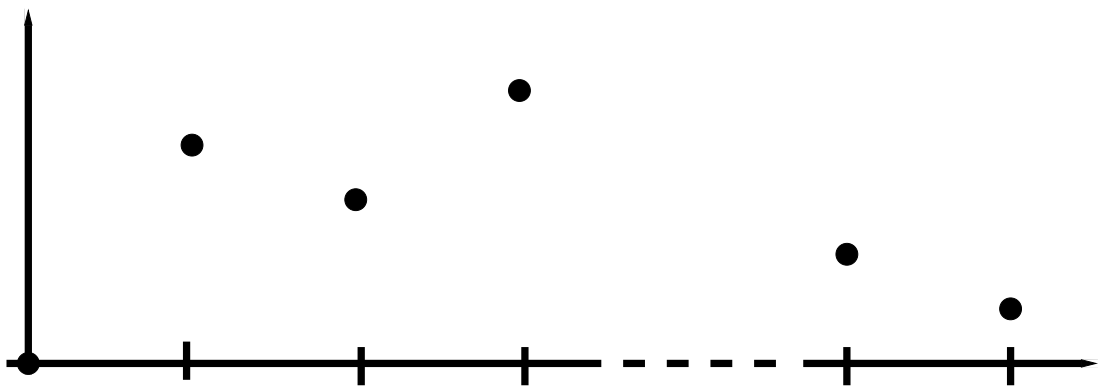


The parareal iterative procedure

- Initialization: coarse propagation that yields $\left\{x_n^{k=0}\right\}_n$:

$$\forall n, \quad x_{n+1}^{k=0} = \mathcal{C}_{\Delta T}(x_n^{k=0})$$

- Iterate over $k \geq 0$:



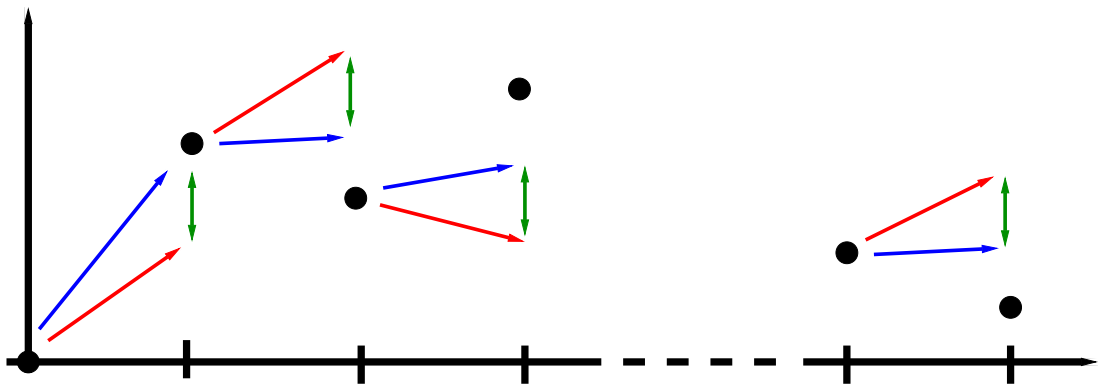
The parareal iterative procedure

- Initialization: coarse propagation that yields $\left\{x_n^{k=0}\right\}_n$:

$$\forall n, \quad x_{n+1}^{k=0} = \mathcal{C}_{\Delta T}(x_n^{k=0})$$

- Iterate over $k \geq 0$:
  - compute jumps (in parallel):

$$J_n^k = \mathcal{F}_{\Delta T}(x_n^k) - \mathcal{C}_{\Delta T}(x_n^k)$$



The parareal iterative procedure

- Initialization: coarse propagation that yields $\left\{x_n^{k=0}\right\}_n$:

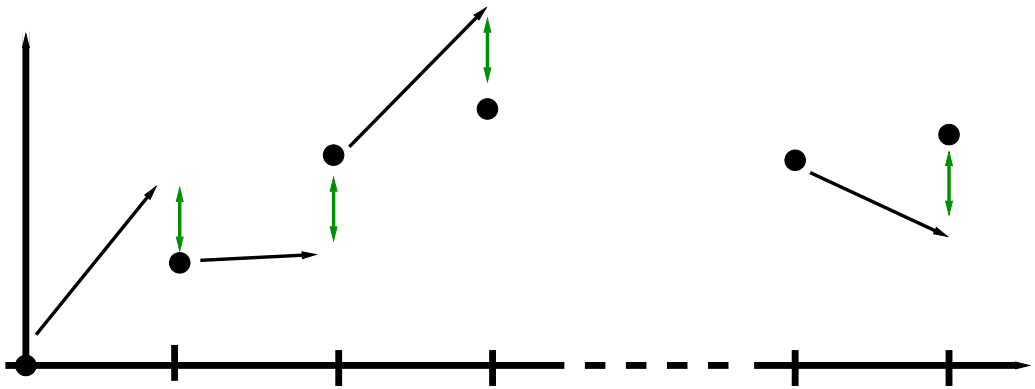
$$\forall n, \quad x_{n+1}^{k=0} = \mathcal{C}_{\Delta T}(x_n^{k=0})$$

- Iterate over $k \geq 0$:
  - compute jumps (in parallel):

$$J_n^k = \mathcal{F}_{\Delta T}(x_n^k) - \mathcal{C}_{\Delta T}(x_n^k)$$

  - sequential update to obtain $\left\{x_n^{k+1}\right\}_n$:

$$\forall n, \quad x_{n+1}^{k+1} = \mathcal{C}_{\Delta T}(x_n^{k+1}) + J_n^k$$



The parareal iterative procedure

- Initialization: coarse propagation that yields $\left\{x_n^{k=0}\right\}_n$:

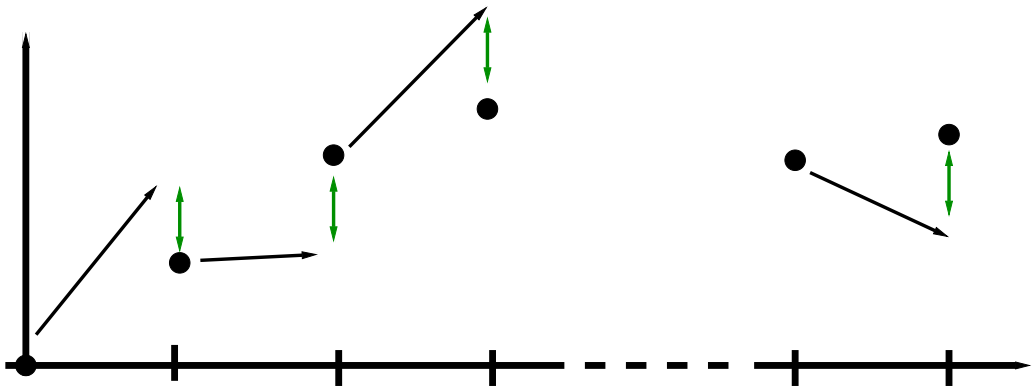
$$\forall n, \quad x_{n+1}^{k=0} = \mathcal{C}_{\Delta T}(x_n^{k=0})$$

- Iterate over $k \geq 0$:
  - compute jumps (in parallel):

$$J_n^k = \mathcal{F}_{\Delta T}(x_n^k) - \mathcal{C}_{\Delta T}(x_n^k)$$

  - sequential update to obtain $\left\{x_n^{k+1}\right\}_n$:

$$\forall n, \quad x_{n+1}^{k+1} = \mathcal{C}_{\Delta T}(x_n^{k+1}) + J_n^k$$



The fine solver is called only in the parallel part of the algorithm.

Parareal algorithm for MD simulations – 1

- The parareal iterations converge (when $k \to \infty$) to the solution of the reference dynamics

$$x_{n+1} = \mathcal{F}_{\Delta T}(x_n)$$

  This comes from the fact that $x_n^k = \mathcal{F}_{\Delta T}^n(x_0)$ whenever $k \geq n$.

- In practice, for many applications, convergence is observed in much fewer iterations

Parareal algorithm for MD simulations – 1

- The parareal iterations converge (when $k \to \infty$) to the solution of the reference dynamics

$$x_{n+1} = \mathcal{F}_{\Delta T}(x_n)$$

  This comes from the fact that $x_n^k = \mathcal{F}_{\Delta T}^n(x_0)$ whenever $k \geq n$.

- In practice, for many applications, convergence is observed in much fewer iterations

- In MD, we often run simulations with time steps just below the stability limit (this often provides sufficient accuracy on the quantities of interest): no room for choosing $\delta t_C \gg \delta t_F$

- We thus turn to a different paradigm where $\mathcal{C}_{\Delta T}$ integrates a simpler dynamics than $\mathcal{F}_{\Delta T}$ (say with the same time step)

Parareal algorithm for MD simulations – 2

- In our setting,
  - $\mathcal{F}_{\Delta T}$ integrates the original Langevin dynamics (with the reference potential $V_f \equiv V$)

$$dq_t = p_t\, dt, \qquad dp_t = -\nabla V_f(q_t)\, dt - \gamma\, p_t\, dt + \sqrt{2\gamma\beta^{-1}}\, dW_t$$

  - $\mathcal{C}_{\Delta T}$ integrates a Langevin dynamics run on a simplified (cheaper to compute) potential V_c:

$$dq_t = p_t\, dt, \qquad dp_t = -\nabla V_c(q_t)\, dt - \gamma\, p_t\, dt + \sqrt{2\gamma\beta^{-1}}\, dW_t$$

- Identical Gaussian increments for $\mathcal{F}_{\Delta T}$ and $\mathcal{C}_{\Delta T}$ and over all parareal iterations (to ensure as best as possible trajectorial convergence)
- Choice here: $\mathcal{F}_{\Delta T}$ and $\mathcal{C}_{\Delta T}$ use a single time-step to advance the system by the time interval ΔT

Similar paradigm (in terms of V_f vs V_c) in [Baffico et al, PRE 2002]

Simulations on toy model problems

Two model problems

- A quadratic model in 1D:

$$V_f(q) = \frac{q^2}{2}, \qquad V_c(q) = \omega \frac{q^2}{2} \quad \text{for some } \omega > 0$$

Simple enough to be amenable to theoretical analysis, and exhibits the same issues as those appearing with more complex models.

Two model problems

- A quadratic model in 1D:

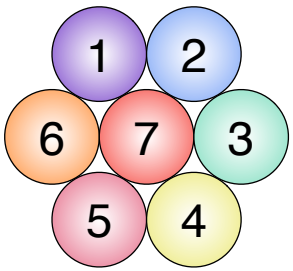
$$V_f(q) = \frac{q^2}{2}, \qquad V_c(q) = \omega \frac{q^2}{2} \quad \text{for some } \omega > 0$$

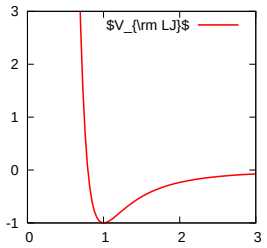
Simple enough to be amenable to theoretical analysis, and exhibits the same issues as those appearing with more complex models.

- A slightly less simple model: a 7-atom Lennard-Jones cluster in 2D:

$$V_f(q) = \frac{1}{2} \sum_{i,j \in \{1,\dots,7\},\ i \neq j} \phi_f(|q^i - q^j|), \qquad \phi_f(r) = r^{-12} - 2\, r^{-6}$$

$V_c \equiv$ harm. approx. of V_f at the global minimum (the initial condition is chosen in the corresponding well).





Convergence criteria

- Relative error between consecutive parareal trajectories:

$$E(k,N) = \frac{\sum_{n=1}^{N} |q_n^k - q_n^{k-1}|}{\sum_{n=1}^{N} |q_n^{k-1}|}.$$

- We stop the algorithm at the first parareal iteration $\overline{k}$ for which

$$E(\overline{k},N) < \delta_{\text{conv}} = 10^{-5}$$

Convergence criteria

- Relative error between consecutive parareal trajectories:

$$E(k,N) = \frac{\sum_{n=1}^{N} |q_n^k - q_n^{k-1}|}{\sum_{n=1}^{N} |q_n^{k-1}|}.$$

- We stop the algorithm at the first parareal iteration $\overline{k}$ for which

$$E(\overline{k},N) < \delta_{\text{conv}} = 10^{-5}$$

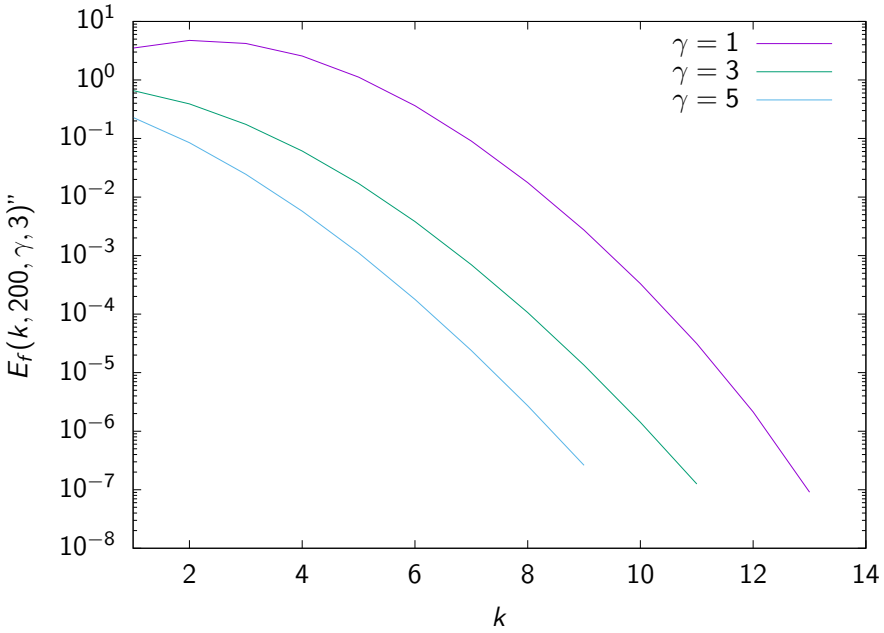
- The wall-clock gain is

$$\text{gain} = \frac{N}{\overline{k}} = \frac{\#\text{ fine propagations for a sequential algorithm}}{\#\text{ fine propagations for the parareal algorithm}}$$

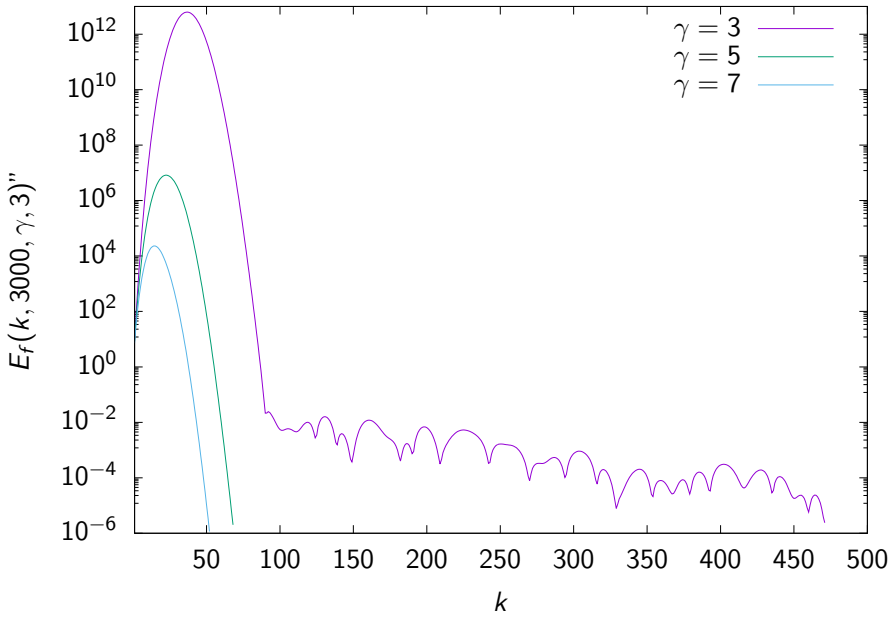
$E_f(k,N)$ = relative error with respect to the reference trajectory

Instability at large times

Plot of $E_f(k, N)$ as a function of k:



$N = 200$

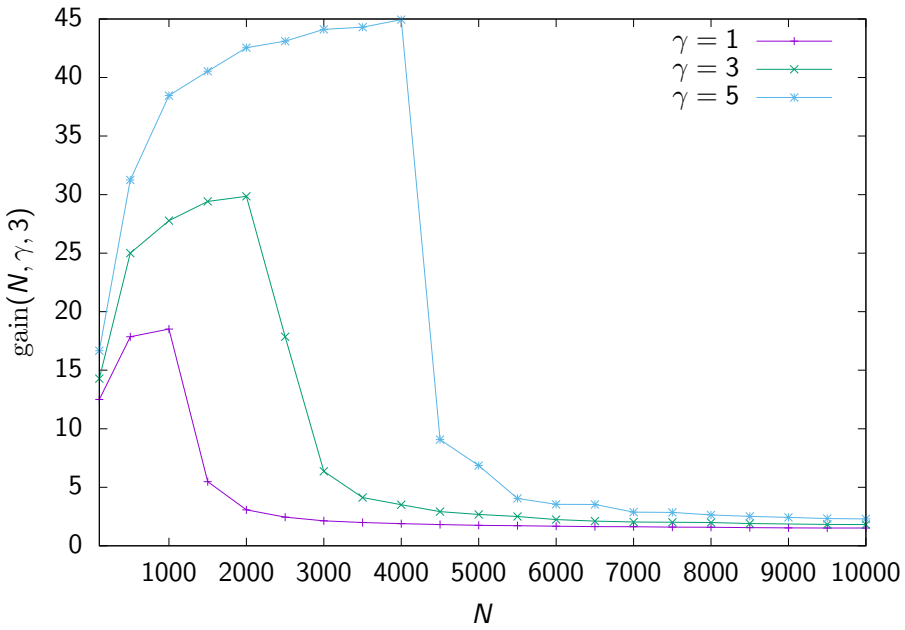


$N = 3000$

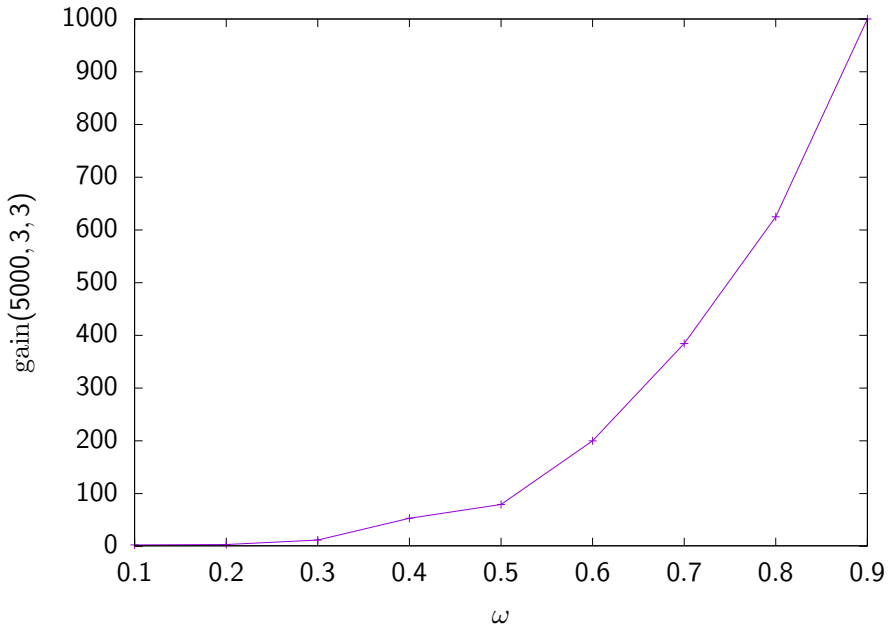
- For small N, fast convergence of the parareal iterations (gain ≈ 10)
- For large N, the error increases to large values when k increases (because the trajectory goes far away), before eventually converging

Harmonic model, $\omega = 0.1$, $\Delta T = 0.05$, $\beta = 3$

Gain as a function of N and ω



Gain as a function of N ($\omega = 0.1$)



Gain as a function of ω ($\gamma = 3$, $N = 5000$)

- Fixed coarse model: the gain decreases to 1 when N increases
- Fixed N: the gain decreases when the coarse model differs too much from the fine one

Harmonic model, $\Delta T = 0.05$, $\beta = 3$

Theoretical analysis on a toy problem (steepest descent)

- Reference problem: $V_f(x) = x^2/2$ and

$$\frac{dx}{dt} = -V_f'(x) = -x$$

Theoretical analysis on a toy problem (steepest descent)

- Reference problem: $V_f(x) = x^2/2$ and

$$\frac{dx}{dt} = -V_f'(x) = -x$$

Fine integrator (exact integ. over ΔT): $\mathcal{F}_{\Delta T}(x) = \exp(-\Delta T)\, x$

- Coarse model: $V_c(x) = \omega\, x^2/2$ and

$$\frac{dx}{dt} = -V_c'(x) = -\omega\, x$$

Coarse integrator (exact integ. over ΔT): $\mathcal{C}_{\Delta T}(x) = \exp(-\omega \Delta T)\, x$

Theoretical analysis on a toy problem (steepest descent)

- Reference problem: $V_f(x) = x^2/2$ and

$$\frac{dx}{dt} = -V_f'(x) = -x$$

Fine integrator (exact integ. over ΔT): $\mathcal{F}_{\Delta T}(x) = \exp(-\Delta T)\, x$

- Coarse model: $V_c(x) = \omega\, x^2/2$ and

$$\frac{dx}{dt} = -V_c'(x) = -\omega\, x$$

Coarse integrator (exact integ. over ΔT): $\mathcal{C}_{\Delta T}(x) = \exp(-\omega \Delta T)\, x$

- We have a complete understanding (see [SISC 2022]) of the relative error as a function of the trajectory length N and of $y = \dfrac{\mathcal{F}_{\Delta T}}{\mathcal{C}_{\Delta T}} - 1$, which quantifies how much the coarse and fine models differ.

This analysis on an oversimplified model is illustrative of the general situation.

A typical result

$R_{n,k}$ = relative error at time n and iteration # k, $\qquad y = \dfrac{\mathcal{F}_{\Delta T}}{\mathcal{C}_{\Delta T}} - 1$

- If y is such that $0 < y \leq c/n$ (the coarse model is very close to the fine; the longer the trajectory, the closer the models should be), then

$$k \mapsto R_{n,k} \quad \text{is decreasing and convex}$$

A typical result

$R_{n,k}$ = relative error at time n and iteration # k, $\qquad y = \dfrac{\mathcal{F}_{\Delta T}}{\mathcal{C}_{\Delta T}} - 1$

- If y is such that $0 < y \leq c/n$ (the coarse model is very close to the fine; the longer the trajectory, the closer the models should be), then

$$k \mapsto R_{n,k} \quad \text{is decreasing and convex}$$

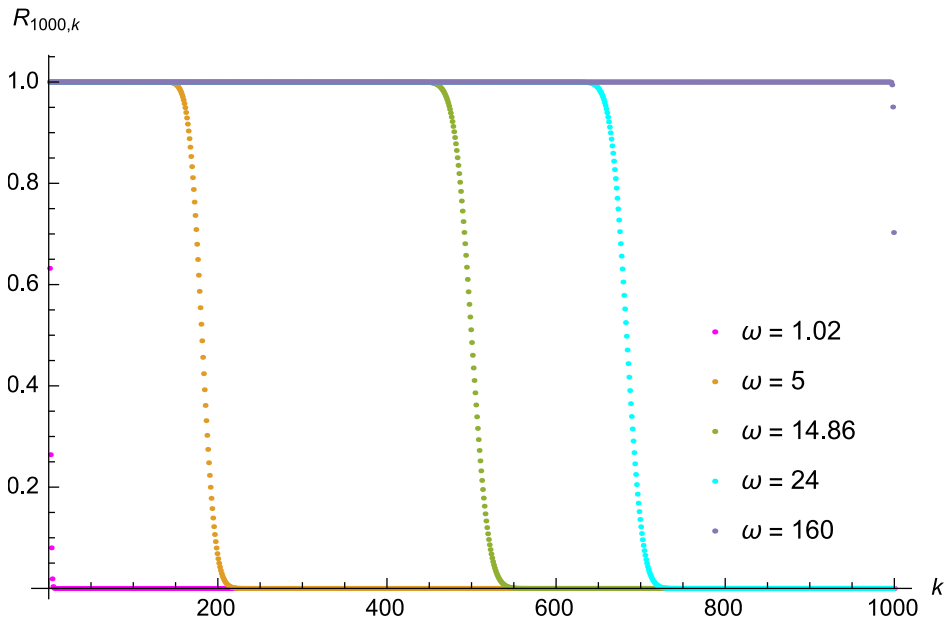
- If y is such that $y \geq c\, n$ (the coarse model is very different from the fine), then

$$k \mapsto R_{n,k} \quad \text{is decreasing and concave}$$

- If y is in-between, then there exists $\overline{p} \approx \dfrac{n|y|}{1+|y|} \in [1, n]$ such that

$$k \mapsto R_{n,k} \text{ is concave for } 1 \leq k \leq \overline{p} \text{ and convex for } \overline{p} \leq k \leq n$$

Numerical illustration



Plot of $k \mapsto R_{n,k}$ for $n = 1000$

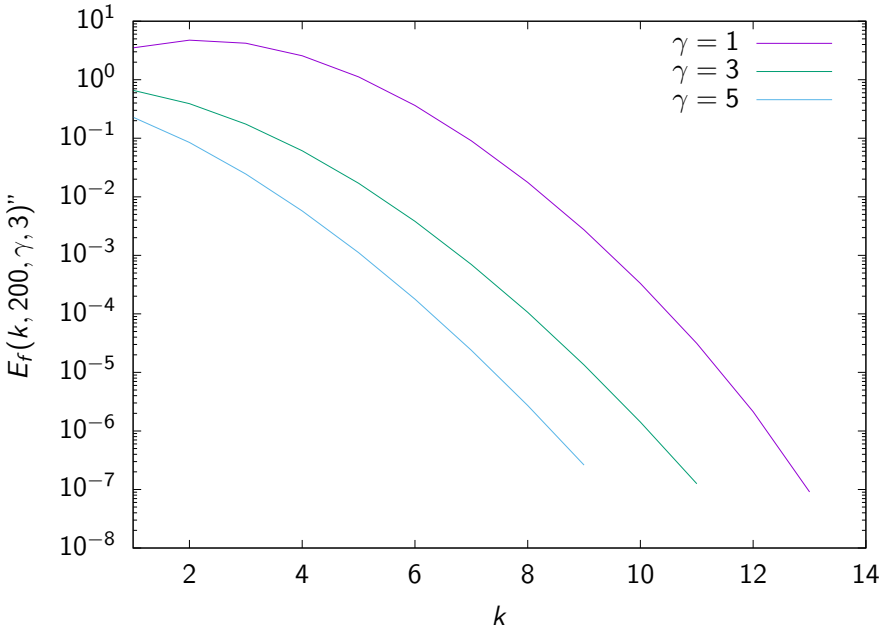
The error (as a function of k) is

- convex for $\omega = 1.02$ (excellent convergence)
- concave for $\omega = 160$ (error close to 100% for almost all k . . .)
- concave then convex for in-between ω (infl. point depends on y & n)

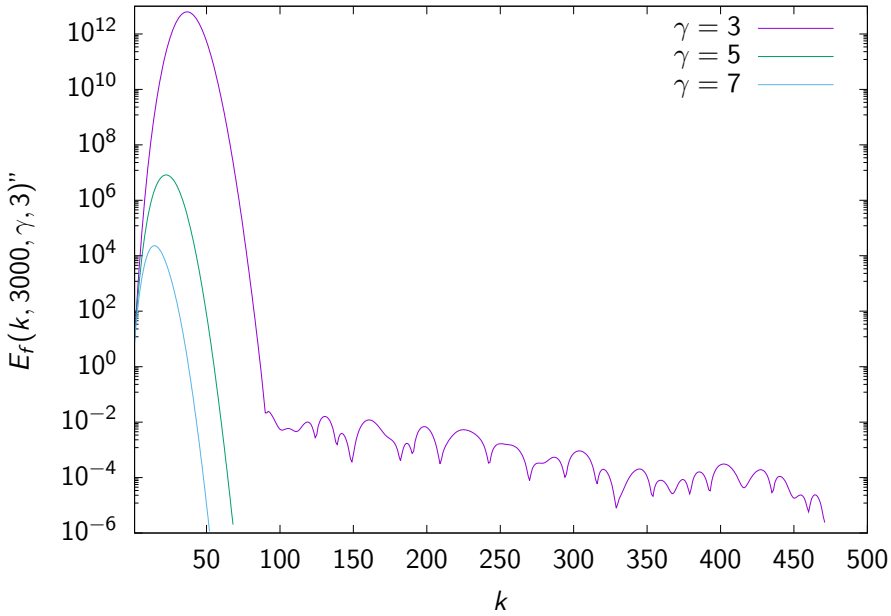
There are cases where $k \mapsto R_{n,k}$ is not a decreasing function (error grows when more iterations!)

Adaptive algorithm

Heuristics



$N = 200$



$N = 3000$

If N is not too large, the algorithm performs nicely ...

Adaptive algorithm

$E \equiv$ relative error between consecutive trajectories

- On the time-slab $[0, N\Delta T]$, we run the parareal algorithm until E is
  - either smaller than the convergence threshold δ_{conv}
  - or larger than an explosion threshold δ_{expl} (attained at parareal iteration $\#\ k_{\mathrm{cur}}$)

Adaptive algorithm

$E \equiv$ relative error between consecutive trajectories

- On the time-slab $[0, N\Delta T]$, we run the parareal algorithm until E is
  - either smaller than the convergence threshold δ_{conv}
  - or larger than an explosion threshold δ_{expl} (attained at parareal iteration $\#\ k_{\mathrm{cur}}$)

- In the blow-up case, for the parareal iteration k_{cur}, we find the first time iteration $1 + \widetilde{m}_1 \leq N$ for which E exceeds δ_{expl}, and we shorten the slab to $[0, \widetilde{m}_1 \Delta T]$.

Adaptive algorithm

$E \equiv$ relative error between consecutive trajectories

- On the time-slab $[0, N\Delta T]$, we run the parareal algorithm until E is
  - either smaller than the convergence threshold δ_{conv}
  - or larger than an explosion threshold δ_{expl} (attained at parareal iteration $\#\ k_{\mathrm{cur}}$)

- In the blow-up case, for the parareal iteration k_{cur}, we find the first time iteration $1 + \widetilde{m}_1 \leq N$ for which E exceeds δ_{expl}, and we shorten the slab to $[0, \widetilde{m}_1 \Delta T]$.

- We then proceed with the parareal iterations on the slab $[0, \widetilde{m}_1 \Delta T]$, that we possibly further shorten, until the relative error (on $[0, \widetilde{m}_1 \Delta T]$) is smaller than δ_{conv}.

Adaptive algorithm

$E \equiv$ relative error between consecutive trajectories

- On the time-slab $[0, N\Delta T]$, we run the parareal algorithm until E is
  - either smaller than the convergence threshold δ_{conv}
  - or larger than an explosion threshold δ_{expl} (attained at parareal iteration $\#\ k_{\mathrm{cur}}$)

- In the blow-up case, for the parareal iteration k_{cur}, we find the first time iteration $1 + \widetilde{m}_1 \leq N$ for which E exceeds δ_{expl}, and we shorten the slab to $[0, \widetilde{m}_1 \Delta T]$.

- We then proceed with the parareal iterations on the slab $[0, \widetilde{m}_1 \Delta T]$, that we possibly further shorten, until the relative error (on $[0, \widetilde{m}_1 \Delta T]$) is smaller than δ_{conv}.

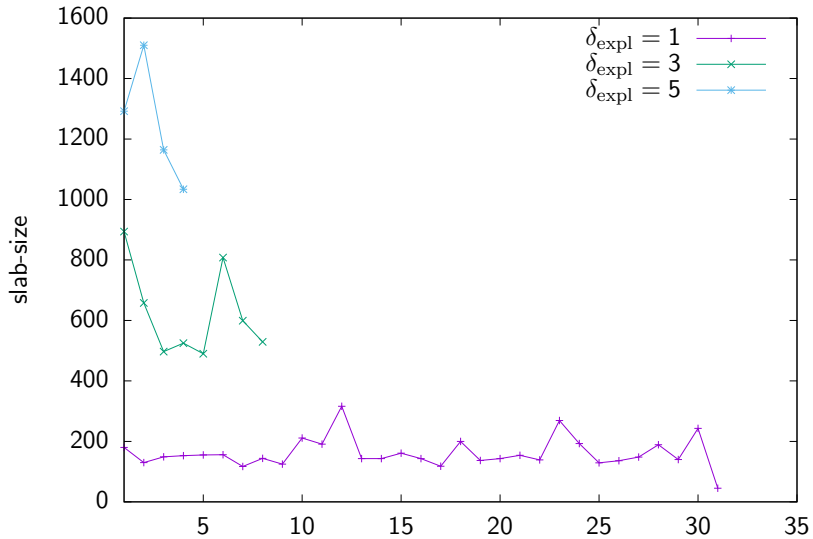
- Once we have converged on $[0, \widetilde{m}_1 \Delta T]$, we proceed and define the new (tentative) time-slab as $[\widetilde{m}_1 \Delta T, N \Delta T]$.

Explosion threshold

The slab sizes are such that $E \leq \delta_{\text{expl}}$:

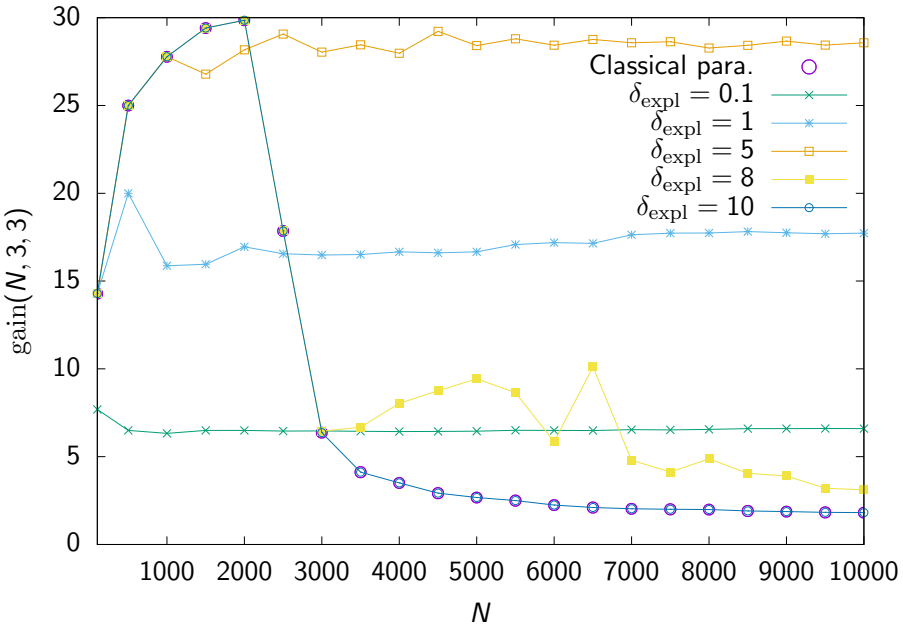
- if δ_{expl} is chosen large, the adaptive criterion is never triggered: vanilla parareal
- if δ_{expl} is chosen small, the slabs are short: no parallelism anymore
- the optimal choice of δ_{expl} is somewhere in-between

List of the sizes of the time-slabs found by the algorithm:

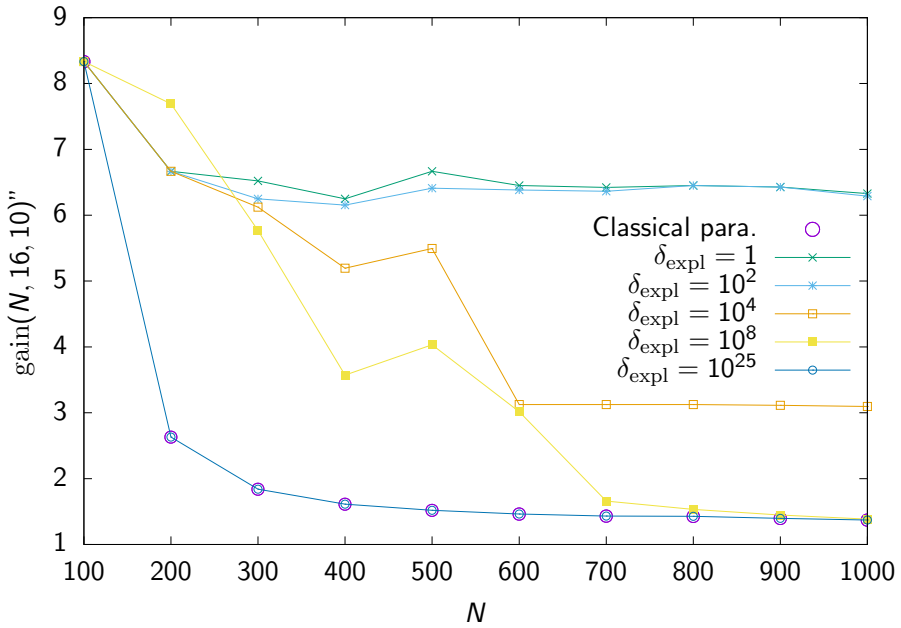


Harmonic model, $\omega = 0.1$, $\Delta T = 0.05$, $\beta = \gamma = 3$, $N = 5000$

Gain



Har-1d ($\omega = 0.1$)



LJ7-2d

- For moderate values of δ_{expl}, the gain seems independent of N
- For large N, the adaptive algorithm always outperforms the classical version (gain ≈ 30 for Har-1d, gain ≈ 7 for LJ7-2d)

Conclusions on this part

- In the long time limit, the trajectories provided by the classical parareal algorithm are far away from the reference trajectories: trajectorial accuracy is poor, and statistical accuracy is poor as well!
- The adaptive algorithm always outperforms the classical version

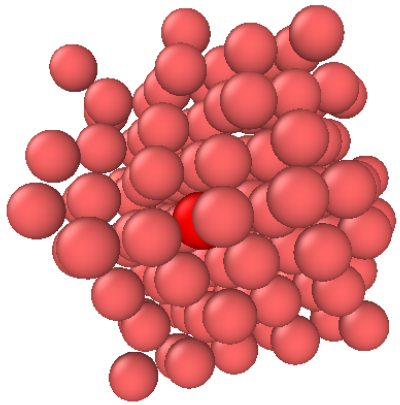
F.L., T. Lelièvre and U. Sharma, SIAM J. Scientific Computing 2022

Applications to the simulation of self-interstitial atoms in tungsten

Joint work with O. Gorynina, T. Lelièvre and D. Perez (arXiv preprint 2212.10508)

Our aim

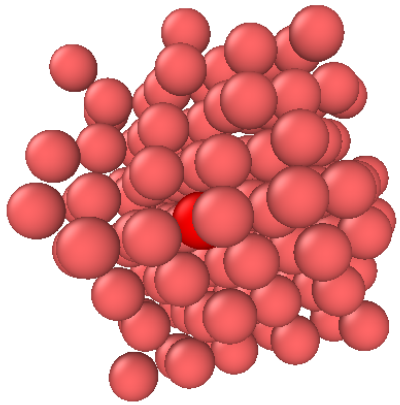
- We consider a periodic lattice of 128 tungsten atoms and add a defect in the lattice in the form of a self-interstitial atom (SIA)



- For the chosen physical parameters,
  - the system is metastable
  - within affordable trajectories, several jumps of the SIA are observed (the residence time within a basin is long, but not extremely long)

Our aim

- We consider a periodic lattice of 128 tungsten atoms and add a defect in the lattice in the form of a self-interstitial atom (SIA)



- For the chosen physical parameters,
  - the system is metastable
  - within affordable trajectories, several jumps of the SIA are observed (the residence time within a basin is long, but not extremely long)

- Quantity of interest: residence times $\equiv$ time spent by the SIA in a given well before jumping to another well

Potentials

- Coarse potential V_c: EAM $\equiv$ a cheap empirical potential based on physically informed parameterized expressions
- Fine (reference) potential V_f: SNAP $\equiv$ an expensive empirical potential.

  It is based on generic expressions, the parameters of which are optimized using machine-learning techniques to reproduce (on some small configurations) the energies, forces, etc, obtained by ab-initio computations

Cost ratio between V_c and V_f: 2600

Potentials

- Coarse potential V_c: EAM $\equiv$ a cheap empirical potential based on physically informed parameterized expressions
- Fine (reference) potential V_f: SNAP $\equiv$ an expensive empirical potential.

  It is based on generic expressions, the parameters of which are optimized using machine-learning techniques to reproduce (on some small configurations) the energies, forces, etc, obtained by ab-initio computations

Cost ratio between V_c and V_f: 2600

Non-intrusive implementation:

- the parareal algorithm is implemented in Python
- to perform the time-stepping, Python calls LAMMPS

The fact that we use LAMMPS in a particular setting (on short trajectories . . .) raises additional difficulties (not detailed here).

Trajectorial accuracy

Trajectorial accuracy

For a small enough value of the convergence threshold (here $\delta_{\text{conv}} = 10^{-10}$), it is possible to observe trajectorial accuracy:

	reference residence times
	[122, 23, 27, 476, 14, 32, 560, 245]

δ_{conv}	parareal residence times
10^{-5}	[63, 27, 16, 36, 19, 34, 332, 972]
10^{-10}	[122, 23, 27, 476, 14, 32, 575, 15, 28, 31, 156]

The first six residence times are identically reproduced on the parareal trajectory.

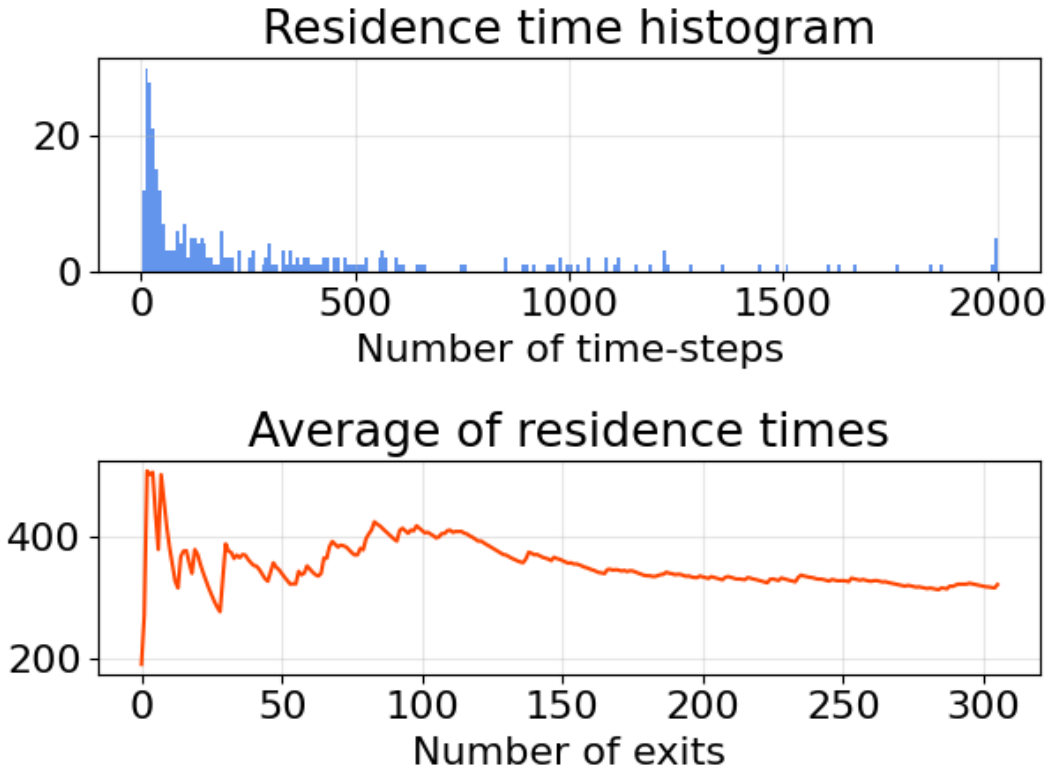
In this case, the reference trajectory and the parareal trajectory start from the same IC and are fed with the same Gaussian increments.

Distribution of residence times (reference results)

We now focus on statistical accuracy: trajectories run with different noises and ICs

Distribution of residence times (reference results)

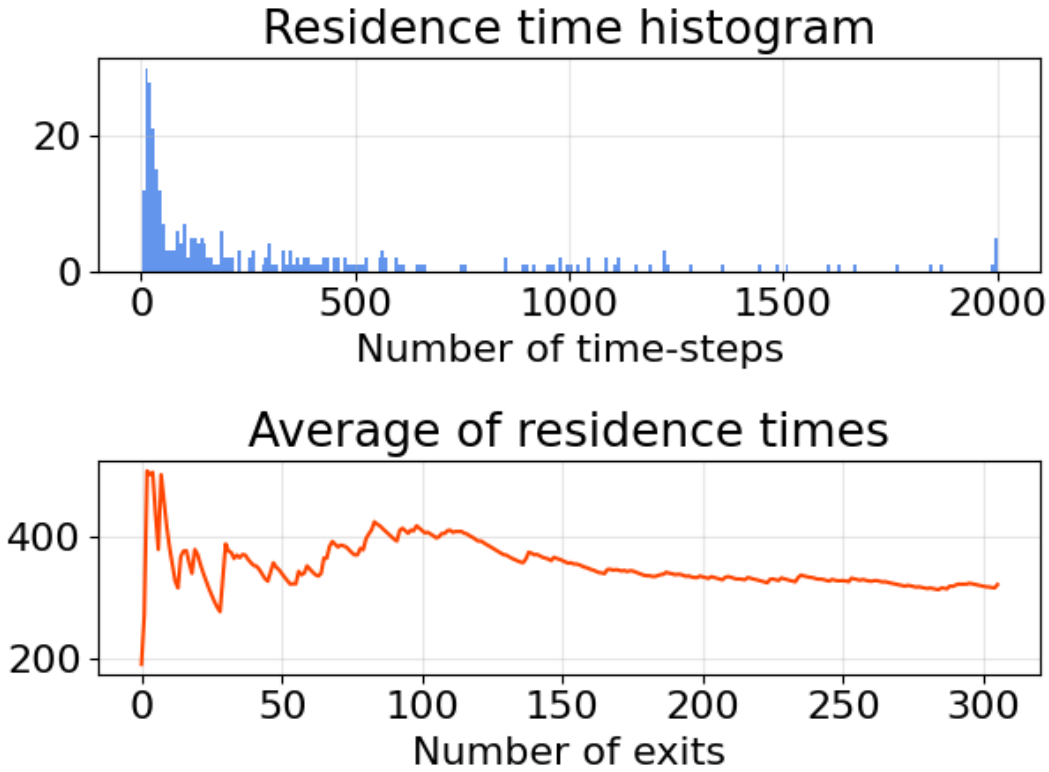
We now focus on statistical accuracy: trajectories run with different noises and ICs



Mean residence time: $T_{\text{mean}} = 320 \times \delta t$, confidence interval $[269; 372] \times \delta t$

Distribution of residence times (reference results)

We now focus on statistical accuracy: trajectories run with different noises and ICs

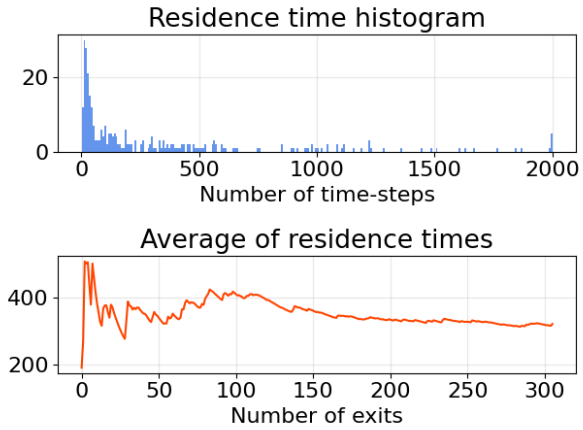


Mean residence time: $T_{\text{mean}} = 320 \times \delta t$, confidence interval $[269; 372] \times \delta t$

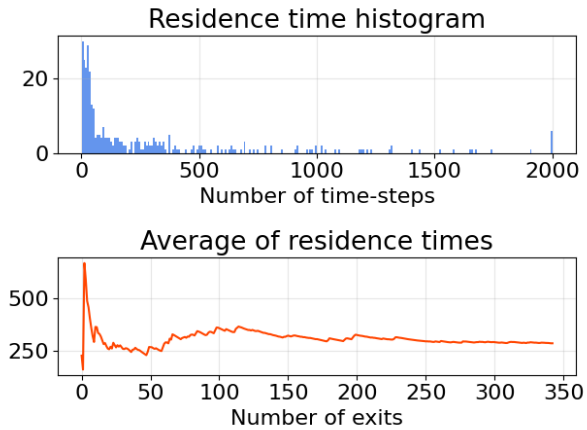
EAM trajectory is wrong: $T_{\text{mean}} = 100 \times \delta t$, confid. interval $[93; 107] \times \delta t$

Parareal results, statistical accuracy ($\delta_{\text{conv}} = 10^{-3}$)

Reference results: $T_{\text{mean}} = 320 \times \delta t$, confidence interval $[269; 372] \times \delta t$



Parareal results: $T_{\text{mean}} = 286 \times \delta t$, confidence interval $[240; 332] \times \delta t$



Very good statistical accuracy (overlapping confidence intervals), while no pathwise accuracy for this value of δ_{conv}. Gain $\approx$ 5 to 20

Conclusions

- Non-intrusive implementation within LAMMPS is possible
  - this allows to consider realistic systems
  - needs appropriate adjustement of time-scheme (not discussed here)
- We found a regime of intermediate δ_{conv} where
  - significant computational gains, of the order of 20 (resp. 5) for $\delta t =$ 0.5 fs (resp $\delta t = 2$ fs)
  - no pathwise accuracy but excellent statistical accuracy
- Similar gains for tungsten lattice and LJ7 cluster

Support of ANR (through project CINE-PARA) and of EuroHPC (through project TIME-X) are gratefully acknowledged