Parallel-in-time integrators for Hamiltonian systems

Claude Le Bris

ENPC and INRIA

Visiting Professor, The University of Chicago

joint work with X. Dai (Paris 6 and Chinese Academy of Sciences), F. Legoll (ENPC and INRIA) and Y. Maday (Paris 6 and Brown)

SSC Seminar, October 2012
Parallel in time algorithm for ODEs

\[ \dot{x} = f(x), \quad x \in \mathbb{R}^d \]

Most algorithms are sequential in nature: \( x_{n+1} = \Phi_{\Delta T}(x_n) \).

The parareal algorithm (Lions, Maday and Turinici, 2001) is the first efficient version of a time integrator that is parallel in time.

It is based upon two integrators to propagate the system over a time \( \Delta T \):

- a fine integrator \( F_{\Delta T} \), to advance of \( \Delta T \) by many tiny timesteps;
- a coarse integrator \( G_{\Delta T} \), to advance of \( \Delta T \) by a few large timesteps (hence much cheaper).

\[ F_{\Delta T} = (\Phi_{\delta t})^{\Delta T/\delta t} \quad \text{and} \quad G_{\Delta T} = (\Phi_{dT})^{\Delta T/dT} \quad \text{with} \quad \delta t \ll dT \]

\( \Phi_{\delta t} \equiv \) a single step of any one-step integrator.
The parareal paradigm - 1

The parareal algorithm combines the two integrators as follows:

- first, coarse propagation that yields \( \{ x_{n}^{k=0} \} \):
  \[
  \forall n, \quad x_{n+1}^{k=0} = G_{\Delta T}(x_{n}^{k=0})
  \]

- Then, iterate over \( k \geq 0 \):
  - compute jumps (in parallel):
    \[
    J_{n}^{k} = F_{\Delta T}(x_{n}^{k}) - G_{\Delta T}(x_{n}^{k})
    \]
  - sequential update to obtain \( \{ x_{n}^{k+1} \} \):
    \[
    \forall n, \quad x_{n+1}^{k+1} = G_{\Delta T}(x_{n}^{k+1}) + J_{n}^{k}
    \]

The fine solver is called only in the parallel part of the algorithm.
The parareal paradigm - 2

\[ x_{n+1}^{k+1} = G_{\Delta T}(x_{n+1}^{k+1}) + F_{\Delta T}(x_n^k) - G_{\Delta T}(x_n^k) \]

- If \( F \equiv G \), then \( x_{n+1}^{k+1} = F_{\Delta T}(x_n^{k+1}) \): similar to a fine sequential integration.

- It turns out that, at iteration \( k \),

\[ x_k^k = (F_{\Delta T})^k(x_0), \]

as if the fine scheme was used sequentially all the way from \( x_0 \) up to time \( k\Delta T \).

- In practice, convergence occurs much faster: only a few iterations are needed to obtain a good accuracy on a large range.

The approach has been successfully tested, in the past few years, on many equations (ODEs or PDEs, e.g. heat equation).
A simple test-case: $\dot{x} = -x$ (coarse and fine propagators: forward Euler)

At iteration $k = 10$, trajectory is exact up to time $10\Delta T = 5$, and is accurate over a much longer range.
Consider a Hamiltonian function $H(q, p)$, and the Hamiltonian dynamics

$$
\dot{q} = \frac{\partial H}{\partial p}(q, p), \quad \dot{p} = -\frac{\partial H}{\partial q}(q, p), \quad q \in \mathbb{R}^d, \quad p \in \mathbb{R}^d.
$$

Introducing $x = (q, p) \in \mathbb{R}^{2d}$, recast the dynamics as

$$
\dot{x} = f(x) = J \nabla H(x), \quad J = \begin{pmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{pmatrix}.
$$

**Separated case:** $H(q, p) = \frac{1}{2}p^Tp + V(q)$. Then

$$
\dot{q} = p, \quad \dot{p} = -\nabla V(q).
$$

**Verlet algorithm** (explicit, second order, symmetric, symplectic):

$$
\begin{align*}
p_{n+1/2} &= p_n - \frac{\delta t}{2} \nabla V(q_n), \\
q_{n+1} &= q_n + \delta t p_{n+1/2}, \\
p_{n+1} &= p_{n+1/2} - \frac{\delta t}{2} \nabla V(q_{n+1}).
\end{align*}
$$
Reversibility

\[
\dot{x} = f(x) = J \nabla H(x)
\]

Let \( \varphi_t(x) \) be the flow of the equation, namely the solution at time \( t \) with I.C. \( x \):

\[
\frac{d\varphi_t(x)}{dt} = J \nabla H(\varphi_t(x)), \quad \varphi_{t=0}(x) = x.
\]

We will focus on the case \( H(q,p) = \frac{1}{2} p^T p + V(q) \), for which the flow is reversible:

\[
\forall t, \quad \rho \circ \varphi_t = (\varphi_t)^{-1} \circ \rho
\]

where \( \rho(x) = \rho(q,p) = (q, -p) \).
Reversibility

\[ \dot{x} = f(x) = J \nabla H(x) \]

Let \( \varphi_t(x) \) be the flow of the equation, namely the solution at time \( t \) with I.C. \( x \):

\[
\frac{d\varphi_t(x)}{dt} = J \nabla H(\varphi_t(x)), \quad \varphi_{t=0}(x) = x.
\]

we will focus on the case \( H(q, p) = \frac{1}{2}p^T p + V(q) \), for which the flow is reversible:

\[
\forall t, \quad \rho \circ \varphi_t = (\varphi_t)^{-1} \circ \rho
\]

where \( \rho(x) = \rho(q, p) = (q, -p) \).

a subset of such systems is the case of integrable reversible systems: up to a reversible change of variables \( (q, p) \mapsto (a, \theta) \), the dynamics reads

\[
\dot{a} = 0, \quad \dot{\theta} = \omega(a), \quad a \in \mathbb{R}^d, \quad \theta \in \mathbb{T}^d.
\]

Thus \( a(q, p) \in \mathbb{R}^d \) is left invariant by the dynamics.
Backward error analysis for ODEs

\[
\dot{x} = f(x)
\]

numerical scheme

\[
\dot{x} = f_{\delta t}(x)
\]

exact flow

\[
\varphi_{\delta t}(x)
\]

Exact flow of the modified equation \textbf{at time} \(\delta t\)

\[
\Phi_{\delta t}(x)
\]

\text{Flow } \Phi_{\delta t}(x) \text{ given by the numerical scheme \textbf{after one-time step}.}
Symmetric integrators on reversible systems:

\[ \forall \delta t, \quad \Phi_{\delta t} \circ \Phi_{-\delta t} = \text{Id} \]

- Symmetric integrator \( \Rightarrow \) Reversible modified equation

- Reversible modified equation + integrable reversible system + non resonant condition \( \Rightarrow \) Conservation of invariants (hence of energy).

Another class of efficient schemes: Symplectic integrators
Parareal scheme for Hamiltonian dynamics

\[ \dot{x} = f(x) = J \nabla H(x), \quad x = (q, p) \in \mathbb{R}^{2d} \]

Standard parareal algorithm:

\[ x_{n+1}^{k+1} = G_{\Delta T}(x_{n}^{k+1}) + F_{\Delta T}(x_{n}^{k}) - G_{\Delta T}(x_{n}^{k}) \]

with \( F_{\Delta T} = (\Phi_{\delta t})^{\Delta T/\delta t} \) and \( G_{\Delta T} = (\Phi_{dT})^{\Delta T/dT} \) with \( \delta t \ll dT \), where \( \Phi_{\delta t} \equiv \) one step of (say) Verlet algorithm with \( \delta t \) time-step.

Symplectic or symmetric algorithms are known to be efficient for Hamiltonian dynamics.

PROBLEM: Even if \( F_{\Delta T} \) and \( G_{\Delta T} \) are symplectic (or symmetric), the resulting parareal algorithm is NOT symplectic (or symmetric) . . .

Long-term (and geometric) properties of the approach are to be clarified!
Harmonic oscillator case: \( \dot{q} = p, \quad \dot{p} = -q \)

Error on the energy \( \frac{1}{2}q^2 + \frac{1}{2}p^2 \)
Adapt the parareal algorithm to the Hamiltonian framework by using

- symmetrization

- projection on the constant energy manifold

\[ \mathcal{M} = \{(q, p) \in \mathbb{R}^{2d}; H(q, p) = H_0\} \]
Symmetric parareal algorithm

Symmetrizing by hand a given integrator is easy:

- consider a numerical flow $\Phi_{\Delta T}$, that reads $x_{n+1} = \Phi_{\Delta T}(x_n)$
- consider the adjoint of $\Phi_{\Delta T}$, defined by $\Phi^*_{\Delta T} = \Phi^{-1}_{-\Delta T}$
- then the scheme $x_{n+1} = \Phi_{\Delta T/2} \circ \Phi^*_{\Delta T/2}(x_n)$ is symmetric:

$$x_{n+1/2} = \Phi^*_{\Delta T/2}(x_n), \quad x_{n+1} = \Phi_{\Delta T/2}(x_{n+1/2})$$

  e.g. $\Phi_{-\Delta T/2}(x_{n+1/2}) = x_n, \quad x_{n+1} = \Phi_{\Delta T/2}(x_{n+1/2})$

Use this observation to make a symmetric version of parareal:

- $\Phi_{\Delta T} \equiv$ the standard parareal algorithm;
- we can write explicitly what is $\Phi^*_{\Delta T}$.

Remark: it is much more difficult to make symplectic a given integrator . . .
The standard form of the parareal algorithm writes

\[ x_{n+1}^{k+1} = G_{\Delta T}(x_{n+1}^{k+1}) + F_{\Delta T}(x_n^k) - G_{\Delta T}(x_n^k) \] (⋆)

It is NOT a one-step integrator on \( x_n \)!
The standard form of the parareal algorithm writes

\[ x_{n+1}^{k+1} = G_{\Delta T}(x_{n+1}^{k+1}) + F_{\Delta T}(x_n^k) - G_{\Delta T}(x_n^k) \]  

(\ast)

It is NOT a one-step integrator on \( x_n \)! Define

\[ U_n = (x_n^0, x_n^1, \ldots, x_n^K) \]

Knowing \( U_n \), the parareal algorithm defines \( U_{n+1} \): we write

\[ U_{n+1} = \Phi_{\Delta T}(U_n). \]

The symmetrized form of \( \Phi \), that is \( \Phi_{\Delta T/2} \circ \Phi_{\Delta T/2}^* \), reads

\[ U_n = \Phi_{-\Delta T/2}(U_{n+1/2}), \quad U_{n+1} = \Phi_{\Delta T/2}(U_{n+1/2}). \]
**Symmetric parareal**

The standard form of the parareal algorithm writes

\[
x_{n+1}^{k+1} = G_{\Delta T}(x_{n+1}^{k+1}) + F_{\Delta T}(x_{n+1}^k) - G_{\Delta T}(x_n^k)
\]  

\((\star)\)

It is NOT a one-step integrator on \(x_n\)! Define

\[
U_n = (x_n^0, x_n^1, \ldots, x_n^K)
\]

Knowing \(U_n\), the parareal algorithm defines \(U_{n+1}\): we write

\[
U_{n+1} = \Phi_{\Delta T}(U_n).
\]

The symmetrized form of \(\Phi\), that is \(\Phi_{\Delta T/2} \circ \Phi^{\star}_{\Delta T/2}\), reads

\[
U_n = \Phi^{-\Delta T/2}(U_{n+1/2}), \quad U_{n+1} = \Phi_{\Delta T/2}(U_{n+1/2}).
\]

In the specific context of the parareal algorithm \((\star)\), this two-step iteration yields the **symmetric parareal integrator**.
Symmetric parareal: some basic remarks - 1

At the initial iteration \((k = 0)\), we use the symmetric version of \(G\):

\[
x_{n+1/2}^{k=0} = G^{-1}_{-\Delta T/2}(x_n^{k=0}), \quad x_{n+1}^{k=0} = G_{\Delta T/2}(x_{n+1/2}^{k=0}).
\]

we next iterate over \(k\):

\[
\begin{align*}
x_{n+1/2}^{k+1} &= G^{-1}_{-\Delta T/2}[x_n^{k+1} - F_{-\Delta T/2}(x_{n+1/2}^k) + G_{-\Delta T/2}(x_{n+1/2}^k)], \\
x_{n+1}^{k+1} &= G_{\Delta T/2}(x_{n+1/2}^{k+1}) + F_{\Delta T/2}(x_{n+1/2}^k) - G_{\Delta T/2}(x_{n+1/2}^k)
\end{align*}
\]

The flow is **symmetric** in the following sense: if, for any \(k\) and any \(n\),

\[
(x_0^n, \ldots, x_n^k, x_n^{k+1}) \longrightarrow (x_{n+1}^0, \ldots, x_{n+1}^k, x_{n+1}^{k+1})
\]

by the previous integrator, then

\[
(x_0^{n+1}, \ldots, x_{n+1}^k, x_{n+1}^{k+1}) \longrightarrow (x_0^n, \ldots, x_n^k, x_n^{k+1})
\]

by the exact same algorithm reversing the time.
\[
x^{k+1}_{n+1/2} = G^{-1}_{-\Delta T/2} \left[ x^k_n - F_{-\Delta T/2}(x^{k+1}_{n+1/2}) + G_{-\Delta T/2}(x^k_{n+1/2}) \right],
\]
\[
x^{k+1}_n = G_{\Delta T/2}(x^{k+1}_{n+1/2}) + F_{\Delta T/2}(x^k_{n+1/2}) - G_{\Delta T/2}(x^k_{n+1/2})
\]

- If \( F \equiv G \), then the symmetrized form reads
\[
x^{k+1}_n = F_{\Delta T/2} \circ F^{-1}_{-\Delta T/2}(x^k_n)
\]
standard symmetrized version of the fine (= coarse) propagator.

- Formally taking the limit \( k \to +\infty \) yields
\[
x^\infty_{n+1} = F_{\Delta T/2} \circ F^{-1}_{-\Delta T/2}(x^\infty_n).
\]
Limit of the algorithm in terms of parareal iterations: standard symmetrized form of \( F_{\Delta T} \). Note also that we never call \( F^{-1} \).

- All the expensive computations (involving \( F \)) are performed in parallel.
Harmonic oscillator case: $\dot{q} = p$, $\dot{p} = -q$, symmetric algorithm

Error on the energy $\frac{1}{2}q^2 + \frac{1}{2}p^2$
A parareal integrator may be seen, at parareal iteration $k$, as an integrator of a system consisting of $k + 1$ identical replicas of the original system.

Consider the symmetric parareal integrator, used on an integrable reversible Hamiltonian system:

- The first replica ($k = 0$) is integrated by the symmetric algorithm $G_{\Delta T/2} \circ G_{-\Delta T/2}^{-1}$: energy preservation.

- Replicas are noninteracting: the system of replicas is an integrable reversible system, with energy $= \sum_{j=0}^{k} H(q_{j}, p_{j})$.

- At iteration $k + 1$, we have integrated an integrable reversible system with a symmetric algorithm: its energy is preserved.

Hence, for all $k$, $\sum_{j=0}^{k} H(q_{j}, p_{j})$ is preserved: the energy of each replica $H(q_{j}, p_{j})$ is preserved.

However, the non resonant condition is NOT satisfied: replicas . . . !
Resonances

Consider a set of $k$ identical replicas of the original system:

- before time discretization, they are resonant (identical!), but uncoupled: energy preservation, . . . OK
- the parareal time discretization introduces coupling.
- these resonances are present in the standard version as well as in the symmetric version of the algorithm.
- even if the symmetric algorithm is used, these resonances may impede preservation of the invariants (energy, . . . ) in the long-time limit.

One possibility to avoid resonances:

use a timestep $dT$ (for $G_{\Delta T}$) that depends on $k$.

\[
\begin{cases}
  x_{n+1}^{k=0} = G_{\Delta T}^{(k=0)}(x_{n}^{k=0}) \\
  x_{n+1}^{k=1} = G_{\Delta T}^{(k=1)}(x_{n}^{k=1}) + \mathcal{F}_{\Delta T}(x_{n}^{k=0}) - G_{\Delta T}^{(k=1)}(x_{n}^{k=0})
\end{cases}
\]

+ symmetrization.
Fix resonances by using $dT_k$

Kepler, symmetric parareal based on Verlet, $dT_k$

Error on the energy $H(q, p) = \frac{p^T p}{2} - \frac{1}{|q|}$
Going from $dT_k$ to a smaller $dT$ with no $k$ dependence

Kepler, symmetric parareal based on Verlet

Error on the energy

Left: $dT_k$

Right: $dT$ smaller

Both pictures with equal cost
Another idea

On top of symmetrizing, let us project on the constant energy manifold

\[ \mathcal{M} = \{ x = (q, p) \in \mathbb{R}^{2d}; \ H(q, p) = H_0 \} , \]

in a way that preserves symmetry.
Another idea

On top of symmetrizing, let us project on the constant energy manifold

\[ \mathcal{M} = \{ x = (q, p) \in \mathbb{R}^{2d}; \ H(q, p) = H_0 \} , \]

in a way that preserves symmetry.

Take any symmetric integrator \( \Psi_{\Delta T} \), and consider \( x_n \mapsto x_{n+1} \) defined by

\[
\begin{align*}
\tilde{x}_n &= x_n + \mu \nabla H(x_n), \\
\tilde{x}_{n+1} &= \Psi_{\Delta T}(\tilde{x}_n), \\
x_{n+1} &= \tilde{x}_{n+1} + \mu \nabla H(x_{n+1}),
\end{align*}
\]

with \( \mu \) chosen such that \( x_{n+1} \in \mathcal{M} \).

As the same Lagrange multiplier \( \mu \) is used in the first and third lines, and as \( \Psi_{\Delta T} \) is symmetric, the integrator \( x_n \mapsto x_{n+1} \) is symmetric.
Symmetric projection

\[ \begin{aligned}
\tilde{x}_n &= x_n + \mu \nabla H(x_n), \\
\tilde{x}_{n+1} &= \Psi_{\Delta T}(\tilde{x}_n), \\
x_{n+1} &= \tilde{x}_{n+1} + \mu \nabla H(x_{n+1}),
\end{aligned} \]

Need to solve a nonlinear problem to compute \( x_{n+1} \) and \( \mu \).

In practice: use a Newton-like algorithm, and stop after a few iterations, or when the residu is small enough.

In the following examples, stop after 2 iterations.
We need to start from a symmetric algorithm. We again introduce

\[ U_n := (x_n^0, x_n^1, \cdots, x_n^K), \]

and consider the symmetric parareal algorithm previously obtained:

\[ U_{n+1} = \Psi_{\Delta T}^{\text{sym}}(U_n), \]

where the map \( \Psi_{\Delta T}^{\text{sym}} \) is symmetric in the classical sense.

Introduce

\[ H_k(U_n) := H(x_n^k), \quad 1 \leq k \leq K \]

and consider a symmetric algorithm, based on the symmetric map \( \Psi_{\Delta T}^{\text{sym}} \) and on a symmetric projection on the manifold where all the energies \( H_k \) are preserved.
Application to the parareal setting - 2

\[ \tilde{U}_n = U_n + \sum_{k=1}^{K} \mu_k \nabla H_k(U_n) \]

\[ \tilde{U}_{n+1} = \Psi_{\Delta T}^{\text{sym}}(\tilde{U}_n) \]

\[ U_{n+1} = \tilde{U}_{n+1} + \sum_{k=1}^{K} \mu_k \nabla H_k(U_{n+1}), \]

with the Lagrange multipliers \( \mu_k \) such that \( H_k(U_{n+1}) = H_0 \) for any \( 1 \leq k \leq K \).

Resulting algorithm:

- **Initialization:** at \( k = 0 \), use the symmetrized version of \( G \):

\[ x_{n+1/2}^0 = G_{-\Delta T/2}^{-1}(x_n^0), \quad x_{n+1}^0 = G_{\Delta T/2}(x_{n+1/2}^0). \]

Set \( \tilde{x}_{n+1/2}^0 = x_{n+1/2}^0 \).
Symmetric parareal integrator with projection

Assume iteration $k$ is completed. We compute the solution at iteration $k+1$ by:

- Set $x_0^{k+1} = u_0$;
- For $0 \leq n \leq N - 1$, compute $x_{n+1}^{k+1}$ from $x_n^{k+1}$ as

$$
\begin{align*}
\tilde{x}_n^{k+1} &= x_n^{k+1} + \mu_{k+1} \nabla H(x_n^{k+1}) \\
\tilde{x}_{n+1/2}^{k+1} &= G_{-\Delta T/2}^{-1} \left[ \tilde{x}_n^{k+1} - F_{-\Delta T/2}(\tilde{x}_{n+1/2}^{k}) + G_{-\Delta T/2}(\tilde{x}_{n+1/2}^{k}) \right] \\
\tilde{x}_{n+1}^{k+1} &= G_{\Delta T/2}(\tilde{x}_{n+1/2}^{k+1}) + F_{\Delta T/2}(\tilde{x}_{n+1/2}^{k+1}) - G_{\Delta T/2}(\tilde{x}_{n+1/2}^{k+1}) \\
x_{n+1}^{k+1} &= \tilde{x}_{n+1}^{k+1} + \mu_{k+1} \nabla H(x_{n+1}^{k+1})
\end{align*}
$$

where $\mu_{k+1}$ is such that $H(x_{n+1}^{k+1}) = H_0$.

As before, $\mu_{k+1}$ and $x_{n+1}^{k+1}$ are determined iteratively.

Jumps are precomputed in parallel, before the nonlinear iterations at step $k + 1$.

All the expensive computations (fine propagator) are performed in parallel.
Kepler problem ($N_{\text{iter}}^{\text{max}} = 2$): energy preservation

Relative error on the energy $H(q, p) \quad (\delta t = 10^{-4}, dT = 0.01, \Delta T = 0.2)$.

Energy is well preserved at any parareal iteration $k \geq 1$, although we limit the number of iterations to solve the nonlinear projection step.
Kepler problem \((N_{\text{max}}^{\text{iter}} = 2)\): angular momentum preservation

For all parareal iterations \(k \geq 7\), the angular momentum is preserved with a relative accuracy of \(10^{-4}\) over the complete time range.
Kepler problem \( (N_{\text{max}}^{\text{iter}} = 2) \): trajectory accuracy

After only 6 iterations, errors on the trajectory are comparable to the errors on the trajectory of the fine scheme used sequentially on the complete time range.
Kepler problem: summary of the results

- energy is well preserved at any parareal iteration $k \geq 1$
- good preservation of the angular momentum (although we project on the manifold where only the energy is preserved)
- only 6 parareal iterations are needed to reach the best possible accuracy on the trajectory.

These results are better than those obtained with

- the standard parareal algorithm or its symmetrized version: energy was badly preserved
- the standard parareal algorithm coupled with a standard, not symmetrized, projection step: better preservation of energy (remember $N_{\text{iter}}^{\text{max}} < \infty$), of the angular momentum, and trajectory convergence is reached for fewer parareal iterations.

Both symmetrization and projection are useful.
The outer solar system

\[ H(q, p) = \frac{1}{2} p^T M^{-1} p + V(q), \quad q \in \mathbb{R}^{18}, \quad p \in \mathbb{R}^{18}, \]

with

\[ V(q) = -\sum_{1 \leq i < j \leq 6} \frac{G m_i m_j}{\|q_i - q_j\|}, \quad M = \text{diag}(m_i). \]

First particle \equiv\ Sun, with mass \( m_1 \geq 1000m_i \) for any planet \((i \geq 2)\).

We show here that

- the symmetric parareal algorithm with symmetric projection on the constant energy manifold again behaves well.

- that the coarse solver can be driven by a simplified dynamics without damaging the good properties of the algorithm.
A simplified coarse integrator

Consider the simplified dynamics

\[ H(q, p) = \frac{1}{2} p^T M^{-1} p + V_{\text{simp}}(q), \quad q \in \mathbb{R}^{18}, \quad p \in \mathbb{R}^{18}, \]

with

\[ V_{\text{simp}}(q) = - \sum_{2 \leq j \leq 6} \frac{G m_1 m_j}{\| q_1 - q_j \|}. \]

In \( V_{\text{simp}} \), we only take into account the gravitational interaction between the Sun (at position \( q_1 \)) and the other planets (at position \( q_j, 2 \leq j \leq 6 \)), and we ignore the interaction between pairs of planets.

- \( m_1 \gg m_i \rightarrow \) The exact dynamics is a perturbation of the simplified dynamics.

- The simplified system is less expensive to simulate than the exact one, since \( V_{\text{simp}} \) is a sum of 5 terms, whereas \( V \) is a sum of 15 terms.
**Numerical results** ($N_{\text{iter}}^{\text{max}} = 2$): energy preservation

Relative error on the energy ($\delta t = 10^{-2}$, $dT = 50$, $\Delta T = 200$).

For $k \geq 8$, energy is preserved up to the prescribed tolerance.
Numerical results: trajectory accuracy

![Graph showing trajectory accuracy](image)

**Error on the trajectory**

For $k \leq 5$, large trajectory error.
At $k = 15$, the trajectory error is comparable to the error made by the fine propagator used sequentially.
Numerical results: angular momentum preservation

Relative error on the angular momentum

Good preservation of the angular momentum (error < 1 %), although the trajectory may be wrong.
Numerical results: qualitative trajectory

$k = 1$ (top left), $k = 5$ (top right), $k = 10$ (bottom left), $k = 15$ (bottom right).

For $k = 5$, the trajectory is quantitatively wrong but qualitatively correct!
For $k \geq 10$, the trajectory is quantitatively correct.
With our parameters, the computation cost is mostly due to the fine-scale integrations. Hence, the speed-up is

\[ G = \frac{T}{K \Delta T} = \frac{\text{number of processors}}{\text{number of parareal iterations}} \]

For the outer-solar system test-case,

\[ G \approx 66 \]

provided we have 1000 processors.
Symplectic parareal algorithm

Unclear (no standard way to make a given algorithm symplectic).

- A first possible approach by G. Bal and Q. Wu (2008):
  - build a generating function $S$ by interpolating parareal results: in practice, for some chosen $\phi_i(q, P)$,
    \[
    S(q, P) = \sum_i a_i \phi_i(q, P), \quad a_i \text{ to determine.}
    \]
  - derive from this generating function a symplectic algorithm:
    \[
    p = P + \frac{\partial S}{\partial q}(q, P), \quad Q = q + \frac{\partial S}{\partial P}(q, P).
    \]

- How to go further?
Conclusions

- the standard parareal algorithm is neither symplectic nor symmetric, even if the building blocks are: this creates issues in the long-time behaviour.
- easy to symmetrize, not enough due to resonances . . .
- symmetrize + symmetric projection → good results! Both ingredients are needed.

Some open questions:

- Symplectic parareal algorithm?
- Understand better the behaviour of the integrators through their modified equations (work in progress).
