Symplectic Parareal

Guillaume Bal and Qi Wu

Department of Applied Physics and Applied Mathematics, Columbia University, New York NY, 10027. {qw2107,gb2030}@columbia.edu

1 Introduction

The parareal algorithm, recalled in section 2 below, allows one to solve evolution equations on (possibly massively) parallel architectures. The two building blocks of the algorithms are a coarse-discretization predictor (solved sequentially) and a fine-discretization corrector (solved in parallel). First developed in [9] and slightly modified in [3], which is the algorithm presented below, the parareal algorithm has received quite a bit of attention lately; see e.g. [1, 2, 4, 5, 6, 7, 10] and their references. Section 3 recalls some results on the parareal algorithm when it is used to solve ordinary and partial differential equations. One of the main shortcomings of the parareal algorithm is that, as a predictor corrector scheme, it may generate high-frequency instabilities.

An area of great potential for the parareal algorithm may thus be the long time evolution of not-too-large systems of ordinary differential equations as they may arise e.g. in molecular dynamics and in the Keplerian problem. The parareal algorithm, however, does not preserve geometric properties such as the symplecticity of the continuous flow of a Hamiltonian system.

We propose in this note a framework to construct a symplectic parareal-type algorithm. The framework is based on the introduction of an interpolating step between the predicting step and the correcting step. The resulting Interpolated Predictor Corrector (IPC) scheme is presented in section 4. We first derive an IPC scheme for arbitrary systems of ordinary differential equations. We then show how the IPC can be rendered symplectic by using the interpolation of appropriate generating functions. Section 5 provides proof of concept by showing numerical simulations for a simple one-dimensional Hamiltonian system.

2 Parareal Algorithm

Let us consider a system of ordinary differential equations of the form

$$\frac{dX}{dt}(t) = b(t, X(t)), \qquad t \in [0, T], \qquad X(0) = X_0. \tag{1}$$

Here $X(t) \in \mathbb{R}^d$ for some finite d. We assume that the above system admits a unique solution. We set a time step $\Delta T > 0$ and a discretization $T^n = n\Delta T$ and introduce the solution operator $g(t,x)$ over the small interval ΔT given by $g(t,X) = X(t+\Delta T)$ when $X(t) = X$. Let now $g_\Delta(t,X)$ be a discretization of g and define the coarse solution

$$X_1^{n+1} = g_\Delta(T^n, X_1^n) \quad \text{for } 0 \le n \le N-1; \qquad X_1^0 = X_0. \tag{2}$$

We introduce the correction operator $\delta g(T^n, X) = g(T^n, X) - g_\Delta(T^n, X)$.

Then we define iteratively the **parareal** approximations

$$X_{k+1}^{n+1} = g_\Delta(T^n, X_{k+1}^n) + \delta g(T^n, X_k^n), \qquad k \ge 1. \tag{3}$$

Note that all the terms $\delta g(T^n, X_k^n)$ for $0 \le n \le N-1$ may be performed in parallel. Let us define the error $\varepsilon_k^n = X_k^n - X(T^n)$. Provided that g_Δ provides a scheme of order m and is such that g_Δ and δg are Lipschitz continuous (see e.g. [2] for the details), we obtain the following estimate:

$$|\varepsilon_k^n| = |X_k^n - X(T^n)| \le C(\Delta T)^{k(m+1)} \binom{n}{k} (1 + |X_0|). \tag{4}$$

For $n = N$ and $k = O(1)$ (the case of interest in practice), we thus obtain:

$$|X_k^N - X(T)| \le CT(\Delta T)^{km}(1 + |X_0|). \tag{5}$$

The iterative scheme (3) replaces a discretization of order m by a discretization of order km after $k-1$ iterations, involving k coarse solutions and $k-1$ fine solutions that can be calculated in parallel.

3 Two Remarks on the Parareal Algorithm

Provided that we seek a final solution at time T with an accuracy of order δt, we have four parameters at our disposal: (i) the coarse time step ΔT; (ii) the number of parareal iterations k; (iii) the number N of successive uses of the parareal scheme over intervals of size $\tau = \frac{T}{N}$ and (iv) the number of available processors P. An analysis of the choices for these parameters that maximize speedup and system efficiency is presented in [2]. The main conclusions are as follows. When the number of available processors is unlimited, i.e., at least of order $(\delta t)^{-1/2}$, then an optimal speedup is attained when ΔT, k, and N are chosen as $\Delta T \approx (\delta T)^{1/2}$, $k = 2$, and $N = 1$. Assuming that the number of processors is smaller and that it takes the form $P = (\delta t)^{-\alpha}$ for some $0 < \alpha < 1/2$, then optimality in the system efficiency (i.e., in the use of all available processors) is achieved provided that the parameters are chosen as $\Delta T \approx (\delta T)^{(1+\alpha)/3}$, $k = 2$, and $N \approx (\delta t)^{-2(1-2\alpha)/2}$.

The parareal algorithm is therefore quite efficient when the number of parareal iterations is $k = 2$, which means that the coarse solver is used twice in a sequential fashion and that the fine solver is used once in parallel. Larger values of k may be beneficial to obtain a better accuracy or to allow for more conservative choices of the (a priori unknown) parameters ΔT and N. Subsequent modifications of the parareal algorithm in this paper implicitly recognize that $k = 2$ is a reasonable choice.

The second remark pertains to the use of the parareal algorithm to solve partial differential equations. Several studies have shown that the parareal algorithm performed well for parabolic equations but showed some instabilities for hyperbolic equations; see e.g. [4, 6]. Analytical calculations performed for simple examples of partial differential equations in [3, 1] provide some explanations for this behavior. In the framework of equations with constant coefficients, we obtain in the Fourier domain the following evolution equation

$$\frac{\partial \hat{u}}{\partial t}(t,\xi) + P(\xi)\hat{u}(t,\xi) = 0,\ \xi \in \mathbb{R},\ t > 0, \qquad \hat{u}(0,\xi) = \hat{u}_0(\xi),\ \xi \in \mathbb{R}. \tag{6}$$

We define $\delta(\xi) = P(\xi)\Delta T$ and the propagator $g(\delta(\xi)) = e^{-\delta(\xi)}$.

Assume that the symbol $P(\xi)$ is approximated by $P_H(\xi)$ to model spatial discretization and that the time propagator $g(\delta)$ is approximated by $g_\Delta(\delta_H)$, where $\delta_H(\xi) = P_H(\xi)\Delta T$. We then define the parareal scheme as:

$$\hat{u}^{n+1}_{k+1}(\xi) = g_\Delta(\delta_H(\xi))\hat{u}^n_{k+1}(\xi) + \delta g(\xi)\hat{u}^n_k(\xi),\ \ \delta g(\xi) = g(\delta(\xi)) - g_\Delta(\delta_H(\xi)), \tag{7}$$

with $\hat{u}^0_{k+1}(\xi) = \hat{u}_0(\xi)$ and $\hat{u}^n_0(\xi) \equiv 0$. We verify that we have:

$$\hat{u}^n_{k+1}(\xi) = \sum_{m=0}^{k} \binom{n}{m} (\delta g(\xi))^m g_\Delta^{n-m}(\delta_H(\xi))\hat{u}_0(\xi). \tag{8}$$

The error term $\varepsilon^n_k(\xi) = \hat{u}^n(\xi) - \hat{u}^n_k(\xi)$ satisfies the following equation: $\varepsilon^{n+1}_{k+1}(\xi) = g_\Delta(\delta_H(\xi))\varepsilon^n_{k+1}(\xi) + (g(\delta(\xi)) - g_\Delta(\delta_H(\xi)))\varepsilon^n_k(\xi)$, with boundary conditions $\varepsilon^0_{k+1}(\xi) = 0$ and $\varepsilon^n_0(\xi) = \hat{u}^n(\xi)$. We may prove by induction that:

$$\varepsilon^n_k(\xi) = (\delta g(\xi))^k \sum_{p_1=1}^{n-1} \cdots \sum_{p_{k-1}=1}^{p_{k-2}-1} \sum_{p_k=0}^{p_{k-1}-1} g^{p_k}(\delta) g_\Delta^{n-p_k-k}(\delta_H)\hat{u}_0(\xi). \tag{9}$$

This provides the following bound for the error estimate

$$|\varepsilon^n_k(\xi)| \lesssim |\delta g(\xi)|^k \binom{n}{k} \sup_p |g_\Delta|^{n-p-k}(\delta_H)|g|^p(\delta)|\hat{u}_0(\xi)|. \tag{10}$$

The above equation shows a different behavior of the error estimate for low and for high frequencies. For low frequencies, $|\delta g(\xi)|^k$ is small by consistency and the error term $|\varepsilon^n_k(\xi)|$ is of the same order as in (4)-(5). For high frequencies however, all we can expect from $|\delta g(\xi)|^k$ is that it is bounded. The term $\binom{n}{k} \approx n^k$ for $k \ll n$ thus creates instabilities.

The lack of stability of the parareal scheme may be seen in (8). We observe that for $k + 1 \geq 2$, the large term $\binom{n}{k} \approx n^k$ can be compensated in three ways: when $|\delta g(\xi)|^k$ is small, which happens for sufficiently small frequencies; when $|g_\Delta|(\delta_H(\xi))$ is small because the scheme is sufficiently damping at high frequencies; or when $\hat{u}_0(\xi)$ is small because $u_0(x)$ is sufficiently smooth. There are however many schemes $g_\Delta(\delta_H)$, which are stable, in the sense that u^n_1 remains bounded uniformly in n, and yet which generate unstable parareal schemes; we refer to e.g. [1] for additional details.

4 Interpolated Predictor Corrector Scheme

A reasonable conclusion that can be drawn from what we have seen so far is that the parareal algorithm is adapted to solving small systems of equations over long times. Such systems do not possess instabilities caused by high frequencies and could greatly benefit from the high accuracy obtained by the parareal algorithm. In several practical applications of long term evolutions however, accuracy is not the only constraint. Users may also want their numerical solutions to satisfy some of the geometric constraints that the exact solutions verify. One such geometric constraint is symplecticity in the solution of Hamiltonian evolution equations:

$$\dot{\mathbf{q}} = \nabla_{\mathbf{p}} H(\mathbf{p}, \mathbf{q}), \qquad \dot{\mathbf{p}} = -\nabla_{\mathbf{q}} H(\mathbf{p}, \mathbf{q}), \tag{11}$$

where the symplectic two-form $d\mathbf{p} \wedge d\mathbf{q}$ is preserved by the flow.

It turns out that the parareal algorithm is not symplectic, even when g and g_Δ are symplectic. The reason is that the sum of symplectic operators appearing in (3) is in general not symplectic. In order to make a parallel algorithm such as parareal symplectic, we need to replace the addition of jumps in (3) by compositions of symplectic maps (since composition of symplectic maps is indeed clearly symplectic).

One way to do this gives rise to the following **Interpolated Predictor Corrector** (IPC) scheme. Let us forget about symplectic structures for the moment and consider an arbitrary system of ordinary differential equations such as (1). We still define the coarse predictor X_1^n as the solution of (2). Now instead of viewing the exact propagator as $g = g_\Delta + (g - g_\Delta)$, which is the main ingredient used in the parareal algorithm (3), we consider the following decomposition;

$$g = \psi_\Delta \circ g_\Delta, \qquad \psi_\Delta \equiv g \circ g_\Delta^{-1}. \tag{12}$$

This definition assumes that the approximation of identity g_Δ is indeed invertible on $\mathbb{R}^d$. We suppress explicit time dependency to simplify.

Once X_1^n is calculated sequentially for $n \geq 0$, we can calculate $\psi_\Delta(X_1^{n+1})$ for all $n \geq 0$ with the requested accuracy and in parallel for the sequence $0 \leq n \leq N-1$ provided that N processors are available. In the second step of the predictor-corrector algorithm, we need to be able to evaluate $\psi_\Delta \circ g_\Delta$ at the points X_2^n sequentially. Since $\psi_\Delta \circ g_\Delta$ has only been evaluated at the points X_1^n, an interpolation step is necessary.

Let us assume that the dynamical system has sufficiently smooth trajectories. Then ψ_Δ is a smooth function on $\mathbb{R}^d$. In fact, it is an approximation of Identity of order $(\Delta T)^{m+1}$ if the coarse scheme g_Δ is of order m. The function $\psi_\Delta : \mathbb{R}^d \to \mathbb{R}^d$ can then be approximated by an interpolated function, which we will denote by $\mathcal{I}(\psi_\Delta)$. Such an interpolation is chosen so that $\mathcal{I}(\psi_\Delta)(X_1^n) = \psi_\Delta(X_1^n)$ for all $0 \leq n \leq N-1$.

Once an interpolation $\mathcal{I}(\psi_\Delta)$ is chosen, we *define* the IPC scheme as:

$$X_2^{n+1} = \mathcal{I}(\psi_\Delta) \circ g_\Delta(X_2^n), \quad n \geq 0, \qquad X_2^0 = X_0. \tag{13}$$

See Fig. 1. We obtain the following result.

Theorem 1. *Let us assume that $\mathcal{I}(\psi_\Delta) - \psi_\Delta$ is a Lipschitz function on $\mathbb{R}^d$ with Lipschitz constant of order $(\Delta T)^{M+1}$. Then the IPC scheme is an accurate scheme of order M, so that e.g. $|X(N\Delta T) - X_2^N| \leq CT(\Delta T)^M(1 + |X_0|)$.*

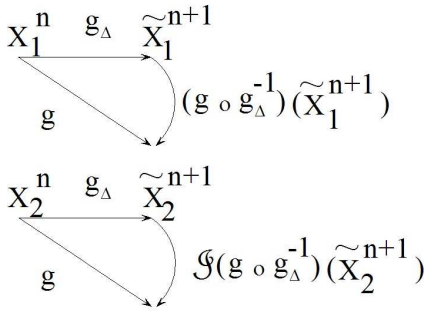

Fig. 1. Construction of the IPC scheme

The proof is classical: $\mathcal{I}(\psi_\Delta) \circ g_\Delta$ is consistent with an accuracy of order $(\Delta T)^{M+1}$ while $\mathcal{I}(\psi_\Delta) \circ g_\Delta$ generates a stable, thus convergent, scheme.

The main ingredient in the construction remains to find an appropriate choice for the interpolating operator $\mathcal{I}$. Note however that ψ_Δ is a smooth map of size $O(\Delta T^{m+1})$, which is known at N nearby points along a trajectory. Under sufficient geometric constraints, we may thus hope that polynomial interpolations may converge to the true map ψ_Δ with spectral accuracy in the vicinity of the discrete trajectory X_1^n. What would be the most accurate and least expensive way to obtain this interpolation remains to be investigated. Note that M above is arbitrary and not necessarily of the form $2m$ as for the parareal algorithm with $k = 2$. The two-step IPC scheme can be arbitrarily accurate provided that the flow is sufficiently smooth and the interpolation sufficiently accurate.

Symplectic scheme. We now come back to the original problem of devising a parallel scheme that would preserve the symplecticity of the continuous equations. The operator ψ_Δ constructed above is clearly symplectic as a composition of symplectic maps. The interpolation $\mathcal{I}$ however may not preserve symplecticity if e.g. polynomial approximation is used. In order to construct a symplectic interpolation, we use the concept of *generating function*; see [8].

We now assume that $X = (\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{2n}$ solves an equation of the form (11). Because ψ_Δ is an approximation of identity on $\mathbb{R}^{2d}$, there exists, at least locally [8], a generating function $S(\mathbf{q}^*, \mathbf{p}) = \mathbf{q}^* \cdot \mathbf{p} + \delta(\mathbf{q}^*, \mathbf{p})$, where δ maps a subset in $\mathbb{R}^{2d}$ to $\mathbb{R}$ and where $(\mathbf{q}^*, \mathbf{p}^*) = \psi_\Delta(\mathbf{q}, \mathbf{p})$. We assume here that S and δ are defined globally; an assumption that can be alleviated by appropriate partition of unity of $\mathbb{R}^{2d}$. The maps ψ_Δ and S are then related by the following equations

$$\mathbf{q}^* = \mathbf{q} - \frac{\partial \delta}{\partial \mathbf{p}}(\mathbf{q}^*, \mathbf{p}), \qquad \mathbf{p}^* = \mathbf{p} + \frac{\partial \delta}{\partial \mathbf{q}^*}(\mathbf{q}^*, \mathbf{p}). \tag{14}$$

The coarse scheme provides the set of N points $(g_\Delta(X_1^n), \psi_\Delta(g_\Delta(X_1^n)))$ of the form $((\mathbf{q}, \mathbf{p}), (\mathbf{q}^*, \mathbf{p}^*))$. We find an interpolation $\mathcal{I}(\delta)(\mathbf{q}^*, \mathbf{p})$ of $\delta(\mathbf{q}^*, \mathbf{p})$ so that (14) is exactly satisfied at such a set of points. Owing to (14), the interpolated generating function $\mathcal{I}(\delta)$ now implicitly generates a map on $\mathbb{R}^{2d}$, which we will call $\mathcal{I}(\psi_\Delta)$. This map is by construction symplectic, and provided that the interpolation $\mathcal{I}(\delta)$ of δ is accurate (say of order ΔT^{M+1}), then so is the interpolation $\mathcal{I}(\psi_\Delta)$. We may then apply Theorem 1 and obtain a **symplectic IPC** scheme of order M.

Note that the interpolation of the generating function may be performed locally by appropriate choice of a partition of unity. The interpolated map $\mathcal{I}(\delta)$ would then take the form $\sum_{i\in I} \mathcal{I}_i(\delta_i)\phi_i$ with obvious notation. The astute reader may also have noticed that the symplectic map $\mathcal{I}(\psi_\Delta)$ so constructed depends on the coarse trajectory X_1^n and thus on its seed X_0. When several trajectories are considered, then the interpolations cannot be performed independently if one wants a truly symplectic scheme. One may either perform one interpolation based on all coarse trajectories, or make sure that the interpolation performed on a new trajectory is compatible with the interpolations obtained from previous trajectories. Such complications also arise when the symplectic IPC is restarted in the sense considered in section 3. When the number N of successive uses of the symplectic IPC is greater than 2, then we need to ensure that the interpolations generated at each restart of the algorithm are compatible with each-other.

As we have noted earlier, the optimal way to perform the interpolation step is still research to be done, whether in the framework of symplectic maps or that of more general maps. In the next section, we show proof of concept by considering a one-dimensional Hamiltonian system and a symplectic IPC schemes based on a global interpolation. Such an interpolation is not optimal and may be computationally prohibitively expensive in higher dimensions.

5 Numerical Simulations

We consider the one-dimensional Hamiltonian (pendulum) system (11) with

$$H(q,p) = \frac{1}{2}p^2 + \sin q. \tag{15}$$

We choose a discretization g_Δ which is second-order and symplectic. The $N = 50$ locations of the parareal solution X_2^n presented in section 2 for $1 \le n \le N$ are shown for several choices of the coarse time step $\Delta T = 0.5$, $\Delta T = 0.65$ and $\Delta T = 0.7$, respectively, in Fig. 2 (they correspond to different final times). The fine time step is

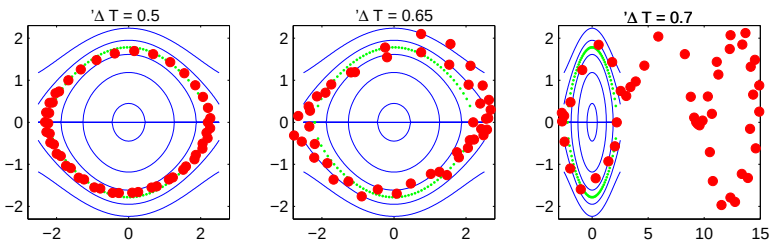

Fig. 2. Parareal solution X_2^n for $1 \le n \le 50$ and $\Delta T = 0.5$, 0.65, and 0.7.

chosen sufficiently small so that the operator g is estimated almost exactly, also by the second-order symplectic scheme. The parareal solution significantly departs from the surface of constant Hamiltonian for large values of ΔT (as it would for larger times and smaller values of ΔT). This is an indication that the parareal scheme

looses the symplectic structure of the flow, and this even though both g and g_Δ are symplectic.

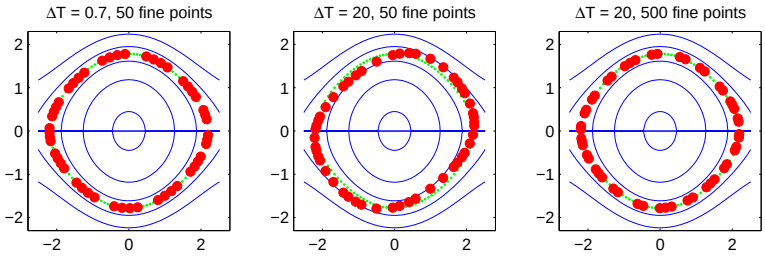

Fig. 3. Symplectic IPC parareal X_2^n for $1 \leq n \leq 50$ and $\Delta T = 0.7$, 20, and 40.

Let now M be the number of discretization points per ΔT for the fine solution operator g. The solution of the IPC scheme X_2^n presented in section 4 is shown in Fig. 3 for values of $(\Delta T, M)$ equal to $(0.7, 50)$, $(20, 50)$, and $(20, 500)$, respectively. The generating function $S(q^*, p)$ is constructed globally on the square $(-2.8, 2.8) \times (-2.3, 2.3)$. Its interpolation is a polynomial of sufficiently high degree so that the $2N$ constraints in (14) generate an under-determined system of linear equations, which is solved by standard least squares. The pseudo-inversion ensures that the resulting interpolation satisfies the constraints exactly and is smooth. The IPC scheme preserves symplecticity independent of ΔT and M. When the fine calculation is not sufficiently accurate (M is too small), ψ_Δ is not estimated accurately and the resulting trajectory may deviate from the true trajectory. With $M = 500$, the estimate of ψ_Δ becomes more accurate and so is its (global) interpolation.

Acknowledgement. This work was supported in part by NSF grant DMS-0239097. GB acknowledges support from an Alfred P. Sloan fellowship.

References

[1] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 425–432. Springer, Berlin, 2005.

[2] G. Bal. Parallelization in time of (stochastic) ordinary differential equations, 2006. Submitted; `www.columbia.edu/~gb2030/PAPERS/Paralleltime.pdf`.

[3] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In *Recent Developments in Domain Decomposition Methods (Zürich, 2001)*, volume 23 of *Lect. Notes in Comput. Sci. Eng.*, pages 189–202. Springer, Berlin, 2002.

[4] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *Internat. J. Numer. Methods Engrg.*, 58(9):1397–1434, 2003.

[5] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. *Internat. J. Numer. Methods Engrg.*, 67(5):697–724, 2006.

[6] P. F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Engrg.*, pages 433–440. Springer, Berlin, 2005.

[7] M. J. Gander and S. Vandewalle. On the superlinear and linear convergence of the parareal algorithm. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lect Notes Comput. Sci. Engrg.*, pages 291–298. Springer, Berlin, 2007.

[8] E. Hairer, L. Christian, and G. Wanner. *Geometric Numerical Integration*. Springer-Verlag, Berlin, 2002.

[9] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *C.R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2000.

[10] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C.R. Acad. Sci. Paris Sér. I Math.*, 335:387–391, 2002.