Stochastic parareal: an application of probabilistic methods to time-parallelisation

Kamran Pentland[1], Massimiliano Tamborrino[2], Debasmita Samaddar[3], Lynton Appel[3]

[1]Mathematics Institute, University of Warwick
[2]Department of Statistics, University of Warwick
[3]Culham Centre for Fusion Energy, UK Atomic Energy Authority

Motivation and Aims

Complex models in science often require the, computationally expensive, **numerical integration** of large-scale systems of **ordinary or partial differential equations (ODEs or PDEs)**. For spatially dependent problems, **domain decomposition methods** can be exploited to achieve significant **parallel speed-up** on **high performance computers** (HPCs). For **initial value problems (IVPs)**, integration wallclock speeds do, however, bottleneck in the time dimension, forcing one to consider using **time-parallel methods**.

Parareal[1,2] is a well established time-parallel numerical method for solving a variety of IVPs - including **fusion plasma dynamics**[3]. It locates a solution **deterministically** in $k_d \in \{1, \dots, N\}$ iterations, yielding a **fixed parallel speed up** (roughly N/k_d) compared to a serial numerical integrator.

The **aims of this project** were to:

- develop a **stochastic parareal algorithm** that locates a solution to an IVP in **fewer than** k_d **iterations**, thus **increasing parallel speed-up**.
- **illustrate the numerical performance** of stochastic parareal on small IVPs.

The parareal algorithm

The **problem** is to solve the following (nonlinear) system of $d \in \mathbb{N}$ ODEs **in parallel:**

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}(t), t) \quad \text{on} \quad t \in [T_0, T_N], \quad \text{with} \quad \mathbf{u}(T_0) = \mathbf{u}^0. \tag{1}$$

Setup

- **Discretise** problem (1) into N sub-problems on N sub-intervals - **assiging one processor to each** ($N = 6$ in Fig. 1).
- Choose **two numerical integrators** to carry out integration from T_n to T_{n+1} :
  - ➔ $\mathcal{F}$ – **fine integrator** with slow execution but high accuracy.
  - ➔ $\mathcal{G}$ – **coarse integrator** with fast execution but low accuracy.

Goal

- Integrating N sub-problems in parallel using $\mathcal{F}$ requires the true **initial values** $\mathbf{U_n}$ at each T_n ($n \geq 1$) → parareal iteratively locates these $\mathbf{U_n}$ using runs of $\mathcal{F}$ and $\mathcal{G}$.

Pseudocode

Step 1: Set counter $k = 0$, defining $\mathbf{U}_n^k$ as the numerical solution to (1) at time T_n and iteration k. Note $\mathbf{U}_0^k = \mathbf{u}^0 \ \forall k$.

Step 2: Calculate initial guesses $\mathbf{U}_n^0$ using $\mathcal{G}$ serially: $\mathbf{U}_n^0 = \mathcal{G}(\mathbf{U}_{n-1}^0)$.

Step 3: For $k = 1$ to N

(i) Propagate solutions on each sub-interval using $\mathcal{F}$ in parallel, calculating $\mathcal{F}(\mathbf{U}_{n-1}^{k-1})$.

(ii) Sequentially calculate $\mathcal{G}(\mathbf{U}_{n-1}^k)$, then use the predictor-corrector (PC):

$$\mathbf{U}_n^k = \underbrace{\mathcal{G}(\mathbf{U}_{n-1}^k)}_{\text{Predict}} + \underbrace{\mathcal{F}(\mathbf{U}_{n-1}^{k-1}) - \mathcal{G}(\mathbf{U}_{n-1}^{k-1})}_{\text{Correct}}. \tag{2}$$

(iii) If the tolerance $\|\mathbf{U}_n^k - \mathbf{U}_n^{k-1}\|_\infty < \varepsilon$ is met for all n, break the loop and return $\mathbf{U}_n^k$. Else continue iterations for the unconverged T_n.

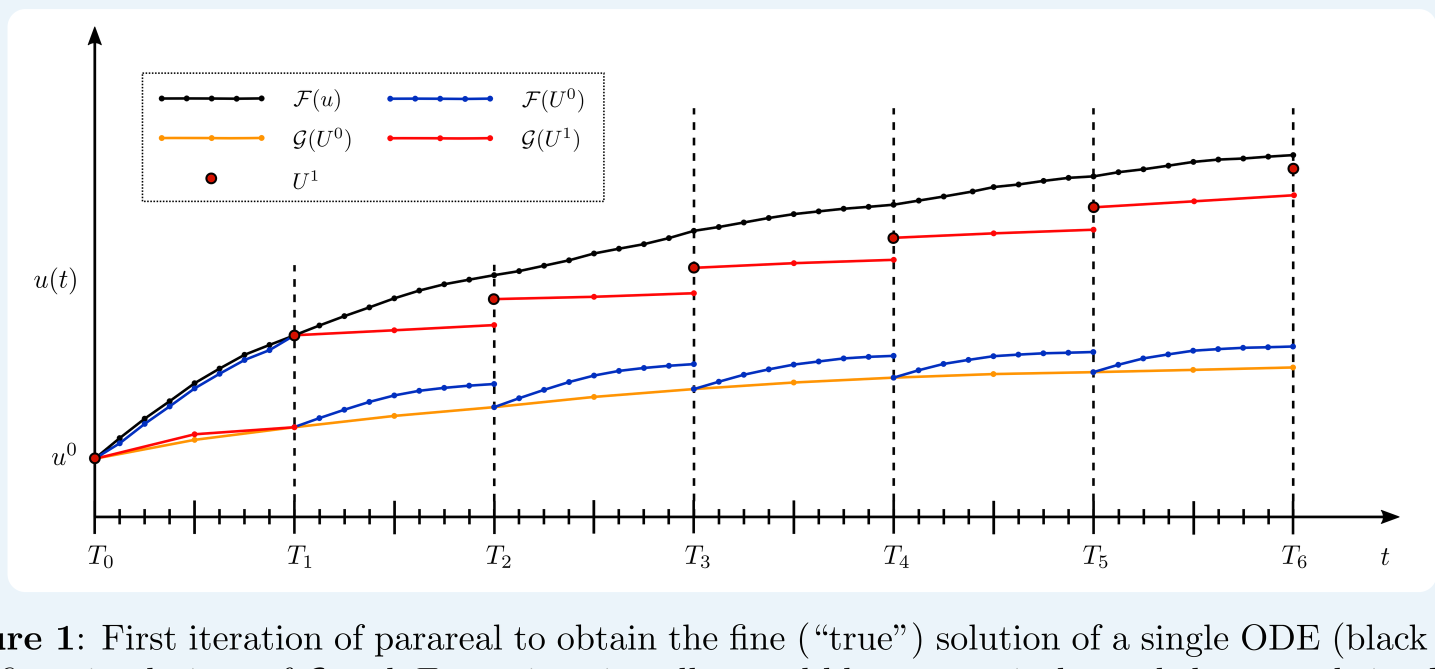

Figure 1: First iteration of parareal to obtain the fine ("true") solution of a single ODE (black line). The first simulations of $\mathcal{G}$ and $\mathcal{F}$ are given in yellow and blue respectively; and the second simulation of $\mathcal{G}$ in red. The red dots represent the PC solutions after applying rule (2).

The stochastic parareal algorithm

Aim: Parareal is **deterministic**, providing fixed parallel speed-up (N/k_d) for a given IVP. We want to incorporate **randomness** to converge in $k_s < k_d$ iterations ⇒ increased speed-up ($N/k_s > N/k_d$).

Deterministic to stochastic

- At each T_n, a **single deterministic** initial value, $\mathbf{U}_{n-1}^{k-1}$, is used in the correction term of eq. (2).
- We want to **improve the correction at each** T_{n-1} by choosing more accurate initial values.
- To do this, we **sample** M initial values

$$\boldsymbol{\alpha}_{n-1,m}^{k-1} \sim \Phi \quad \text{for} \quad m = 1, \dots, M,$$

randomly from a d-dimensional **probability distribution** Φ.

- **All samples** are propagated in parallel using $\mathcal{F}$, after which the **most accurate sample** (see next section) is chosen to replace $\mathbf{U}_{n-1}^{k-1}$ in eq. (2) and thus obtain faster convergence.

Stochastic sampling rules

- Information about initial values at the different temporal resolutions is used to **construct** Φ. It uses:
  - ➔ **marginal means**, $\boldsymbol{\mu}_{n-1}^{k-1}$.
  - ➔ **marginal standard deviations**, $\boldsymbol{\sigma}_{n-1}^{k-1}$.
  - ➔ **correlation matrix**, $\mathbf{R}_{n-1}^{k-1}$.
- We test **four sampling rules** to determine whether the **distribution family** or its **parameters** has the greater impact on performance.
- Sampling rules 1 & 2 are **multivariate Gaussians** and rules 3 & 4 are ***t*-copulas**:
  - ➔ $\boldsymbol{\mu}_{n-1}^{k-1} = \mathcal{F}(\mathbf{U}_{n-2}^{k-2})$ (rules 1 & 3).
  - ➔ $\boldsymbol{\mu}_{n-1}^{k-1} = \mathbf{U}_{n-1}^{k-1}$ (rules 2 & 4).
  - ➔ $\boldsymbol{\sigma}_{n-1}^{k-1} = |\mathcal{G}(\mathbf{U}_{n-2}^{k-1}) - \mathcal{G}(\mathbf{U}_{n-2}^{k-2})|$ (all rules).

The stochastic parareal algorithm cont.

Pseudocode

Step 1: Run parareal up to the end of iteration $k = 1$.

Step 2: For $k = 2$ to N

(i) If $d > 1$, calculate correlations matrices at each T_n using fine propagations from previous iteration[4].

(ii) At each unconverged T_n, sample $M - 1$ initial values $\boldsymbol{\alpha}_{n,1}^{k-1}, \dots, \boldsymbol{\alpha}_{n,M-1}^{k-1}$ from Φ, fixing the final sample $\boldsymbol{\alpha}_{n,M}^{k-1} = \mathbf{U}_n^{k-1}$. Propagate them *all* in parallel using $\mathcal{F}$.

(iii) Select most accurate $\hat{\boldsymbol{\alpha}}_n^{k-1}$ at each T_n by locating the most continuous trajectory, using all $\mathcal{F}(\boldsymbol{\alpha}_{n,m}^{k-1})$ trajectories over $[T_0, T_N]$. Propagate the optimal samples using $\mathcal{G}$.

(iv) Predict and correct at each T_n using the more accurate initial values:

$$\mathbf{U}_n^k = \underbrace{\mathcal{G}(\mathbf{U}_{n-1}^k)}_{\text{predict}} + \underbrace{\mathcal{F}(\hat{\boldsymbol{\alpha}}_{n-1}^{k-1}) - \mathcal{G}(\hat{\boldsymbol{\alpha}}_{n-1}^{k-1})}_{\text{new correction}}. \tag{4}$$

(v) Carry out the convergence check - Step 3(iii) of parareal.

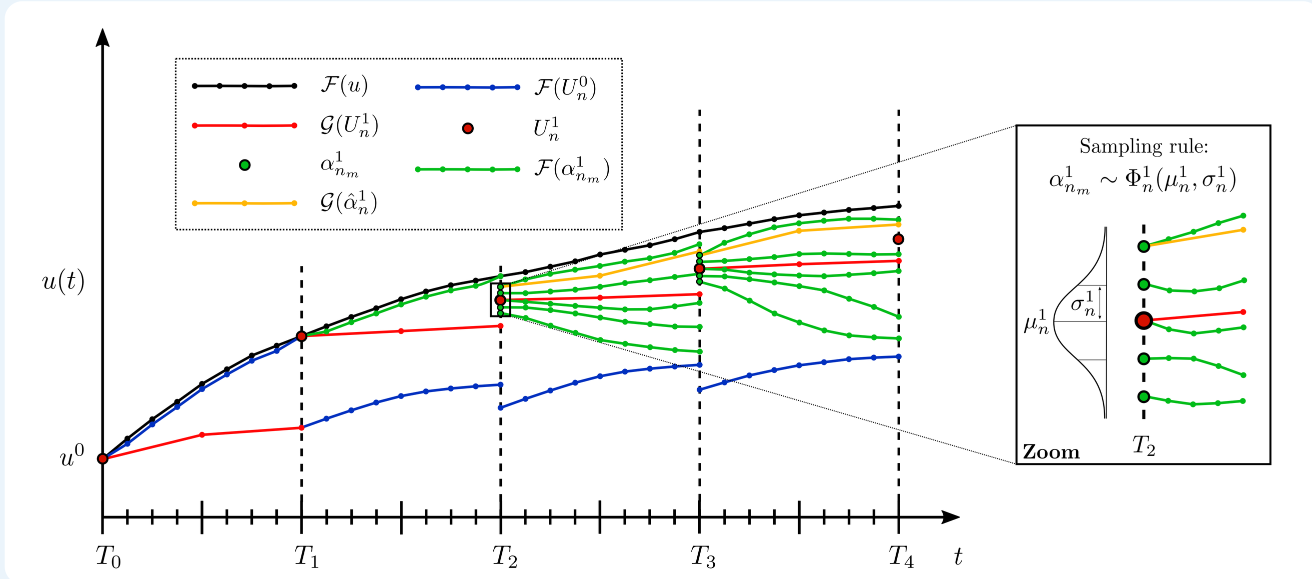

Figure 2: Sampling and propagation process within stochastic parareal following iteration $k = 1$. The "true" solution is given in black, the $k = 0$ fine solutions in blue, the $k = 1$ coarse solutions in red, and the $k = 1$ PC solutions as red dots. With $M = 5$, four samples $\alpha_{n,m}^1$ (green dots) are taken at T_2 and T_3 from some Φ. These values, along with U_2^1 and U_3^1 themselves, are propagated (in parallel) forward in time using $\mathcal{F}$ (green lines). The optimally chosen $\hat{\alpha}_n^1$ are also propagated using $\mathcal{G}$.

Test problem: the Lorenz system

We consider the **chaotic regime** of the Lorenz system

$$\frac{du_1}{dt} = 10(u_2 - u_1), \quad \frac{du_2}{dt} = 28u_1 - u_1u_3 - u_2, \quad \frac{du_3}{dt} = u_1u_2 - \frac{8}{3}u_3, \tag{4}$$

that generates **exponentially diverging trajectories** upon small perturbations of the initial values. Equation (4) is solved for $t \in [0, 18]$ with $\mathbf{u}(0) = (-15, -15, 20)^{\mathrm{T}}$. Parareal solves (4) in $k_d = 20$ **(out of 50) iterations**, stopping at tolerance $\varepsilon = 10^{-8}$. The **numerical results for stochastic parareal** (Fig. 3) show that

- given a sufficient value of M, the estimated **probability that** $k_s < k_d$ **approaches one**, regardless of the sampling rule chosen.
- the estimated **expected value** of k_s, $\mathbb{E}(k_s)$, **decreases for increasing** M.
- generating **correlated samples**, rather than uncorrelated ones, improves performance.

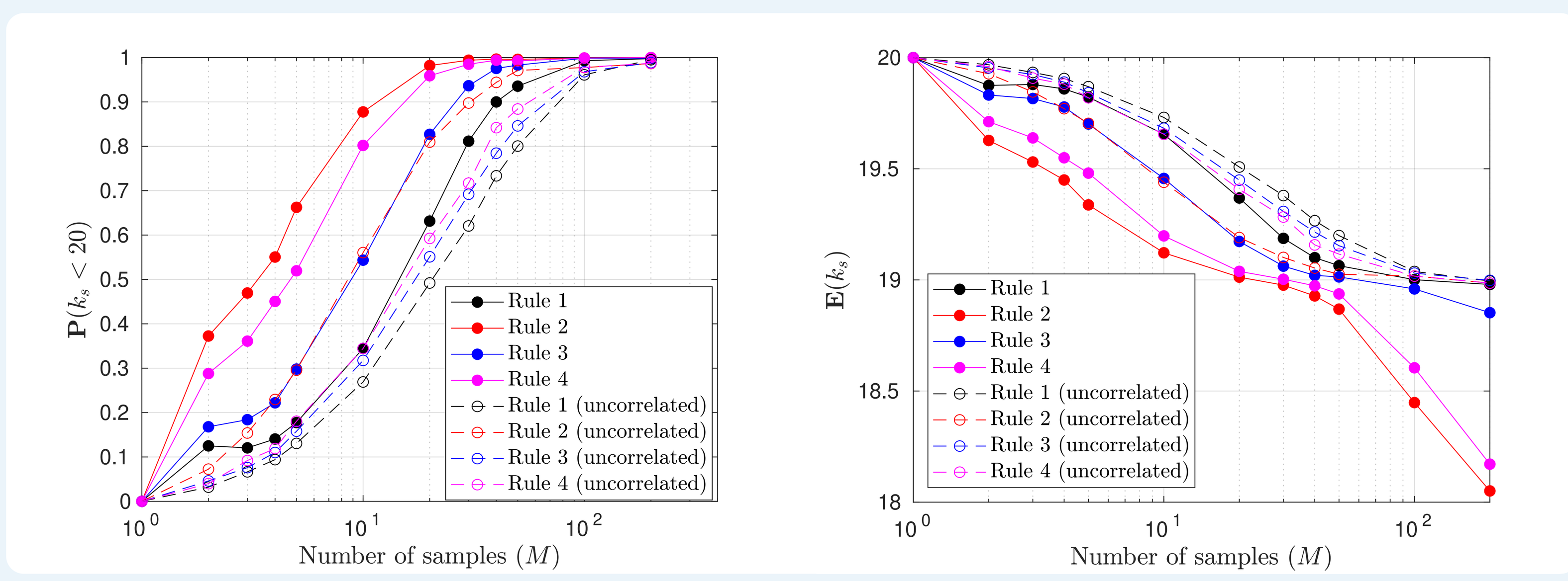

Figure 3: (Left panel) Estimated probabilities that $k_s < k_d$ against sample number M for the four correlated (solid lines) and uncorrelated (dashed lines) sampling rules. (Right panel) Estimated expectation of k_s against M for each sampling rule. Distributions in both panels were calculated by simulating 2000 independent realisations of stochastic parareal for each M.

Conclusions and future work

- Given sufficiently many samples M, **stochastic parareal converges in fewer iterations** ($k_s < k_d$) **than parareal with probability one** ⇒ increased parallel efficiency.
- Stochastic solutions (on average) **maintain accuracy** compared to the solution given by parareal (results not shown[4]).

Future work involves developing methods that **scale for much larger systems**. The processors required scale with M - problematic if high sampling needed. We plan to develop methods from a more **Bayesian perspective**, utilising existing work from the field of **probabilistic numerics**.

Acknowledgements and References

The authors gratefully acknowledge funding from CCFE, EPSRC (EP S022244/1), the EUROfusion consortium, and the Euratom programme (no. 633053).

[1] Lions *et al.*, 2001, *Résolution d'EDP par un schéma en temps pararéel*. Comptes Rendus de l'Academie des Sciences - Series I: Mathematics, 332(7), pp. 661–668.

[2] Gander and Vandewalle, 2007, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29(2), pp. 556–578.

[3] Samaddar *et al.*, 2010, *Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm*, J. of Comp. Phys. 229(18), pp. 6558–6573.

[4] Pentland *et al.*, 2021, *Stochastic parareal: an application of probabilistic methods to time-parallelisation*. In preparation.