Parareal as a Runge-Kutta Scheme

Martin J. Gander[0000−0001−8450−9223],
Aušra Pogoželskytė[0009−0004−5446−4721]

1 Introduction

Parareal is a Parallel-in-Time algorithm. Such algorithms have been designed to introduce parallelism in the time dimension, which seems unintuitive at first glance as time is a sequential dimension. Parareal is usually analyzed as an iterative method, but in early Parareal papers [10, 1], the algorithm is viewed as a one step method over the entire time interval. Using this approach, and with the right hypotheses, we show that Parareal is a Runge-Kutta scheme.

Consider a general time-dependent problem with $\boldsymbol{f} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ such that the problem is well-defined,

$$\frac{d\boldsymbol{u}(t)}{dt} = \boldsymbol{f}(t, \boldsymbol{u}(t)) \quad \text{for } t \in (0, T), \quad \boldsymbol{u}(0) = \boldsymbol{u}_0 \ . \tag{1}$$

Such a problem can be solved numerically using a Runge-Kutta scheme.

Definition 1 An s-stage *Runge-Kutta method* is given by

$$\begin{aligned} \boldsymbol{k}_i &= \boldsymbol{f}\left(t_n + c_i \Delta t, \boldsymbol{u}_n + \Delta t \textstyle\sum_{j=1}^{s} a_{ij} \boldsymbol{k}_j\right), \quad i = 1, \ldots, s \ , \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{k}_i \ , \end{aligned}$$

for coefficients $a_{ij}, b_i, c_i \in \mathbb{R}$.

Martin J. Gander
University of Geneva, rue du Conseil-Général 7-9, Geneva, Switzerland, e-mail: martin.gander@unige.ch

Aušra Pogoželskytė
University of Geneva, rue du Conseil-Général 7-9, Geneva, Switzerland, e-mail: ausra.pogozelskyte@unige.ch

A common practice is to summarize the coefficients of the method in a *Butcher table* as follows:

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} =: \begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^\top \end{array}.$$

Now, assume that we want to parallelize the numerical solution to (1) using N processors. For this, we partition the domain $(0, T)$ into N equal intervals, and assign each interval to a processor. We call such intervals *Parareal intervals* and denote by $\mathcal{F}$ the application of the numerical solution over a Parareal interval, e.g. several steps of a Runge-Kutta method. Since time is sequential, each processor has to wait for the preceding processor to finish its computations before starting their own.

To overcome this limitation, we define $\mathcal{G}$ to be an operator that approximates $\mathcal{F}$ but that is much cheaper to compute, such as a single Backward Euler step. The operator $\mathcal{G}$ is applied sequentially to give an approximation of the numerical solution at the beginning of each Parareal interval, from which each processor can compute a solution using $\mathcal{F}$ in parallel. This results in the Parareal algorithm [10]:

1. Set $\boldsymbol{u}_0^k = \boldsymbol{u}_0$ for $k = 0, \ldots, K$, with K being the number of iterations.
2. Compute $\boldsymbol{u}_n^0 = \mathcal{G}(\boldsymbol{u}_{n-1}^0)$ for $n = 1, \ldots, N$.
3. Correct $\boldsymbol{u}_n^k = \mathcal{F}(\boldsymbol{u}_{n-1}^{k-1}) + \mathcal{G}(\boldsymbol{u}_{n-1}^k) - \mathcal{G}(\boldsymbol{u}_{n-1}^{k-1})$ for $n = 1, \ldots, N$ and $k = 1, \ldots, K$.

The operator $\mathcal{F}$ is often defined as multiple steps of a Runge-Kutta method using a small time-step. A typical way to define $\mathcal{G}$ is to use a lower-order Runge-Kutta method and/or a method with a larger time-step. In that case, the operators $\mathcal{F}$ and $\mathcal{G}$ are Runge-Kutta methods because they are the composition of Runge-Kutta methods [3]. The following theorem explains how to compute their Butcher table.

Theorem 1 (Composition of RK methods, as presented in (12.4) in [8]) *Let A, b and c, and A', b' and c' be the coefficients of two Runge-Kutta methods. Let $\mathbf{1}$ be a vector of all ones. The Butcher table of the composition of the two methods is given by*

$$\begin{array}{c|cc} \boldsymbol{c} & A & 0 \\ (\boldsymbol{b}^\top \mathbf{1})\mathbf{1} + \boldsymbol{c}' & \mathbf{1}\boldsymbol{b}^\top & A' \\ \hline & \boldsymbol{b}^\top & \boldsymbol{b}'^\top \end{array} \tag{2}$$

Example 1 The Butcher table for Backward Euler (BE) is given by

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}. \tag{3}$$

Assume that Parareal is used with $\mathcal{F}$ being 4 steps of the Backward Euler scheme and $\mathcal{G}$ being a single step of it. So, the table in (3) is also the Butcher table of $\mathcal{G}$. To compute the Butcher table for $\mathcal{F}$, one needs to apply Theorem 1 recursively,

$$BE^2: \begin{array}{c|cc} 1 & 1 & 0 \\ 2 & 1 & 1 \\ \hline & 1 & 1 \end{array} \quad BE^3: \begin{array}{c|ccc} 1 & 1 & 0 & 0 \\ 2 & 1 & 1 & 0 \\ 3 & 1 & 1 & 1 \\ \hline & 1 & 1 & 1 \end{array} \quad BE^4: \begin{array}{c|cccc} 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 3 & 1 & 1 & 1 & 0 \\ 4 & 1 & 1 & 1 & 1 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

The Butcher table of BE^4 is given with respect to Δt, the fine time step, however the Butcher table of $\mathcal{F}$ needs to be given with respect to $\Delta T = 4\Delta t$ the size of the Parareal interval. For this, the table of BE^4 needs to be normalized by dividing every coefficient by 4. Thus, the Butcher table of $\mathcal{F}$ is given by

$$\begin{array}{c|cccc} 1/4 & 1/4 & 0 & 0 & 0 \\ 2/4 & 1/4 & 1/4 & 0 & 0 \\ 3/4 & 1/4 & 1/4 & 1/4 & 0 \\ 1 & 1/4 & 1/4 & 1/4 & 1/4 \\ \hline & 1/4 & 1/4 & 1/4 & 1/4 \end{array} \; .$$

2 Parareal as a Runge-Kutta Scheme

Assuming that the fine and coarse operators, $\mathcal{F}$ and $\mathcal{G}$, in Parareal are Runge-Kutta methods, we show what this implies for the algorithm.

Let (A_f, b_f, c_f) and (A_g, b_g, c_g) be the coefficients of the fine and coarse methods used in Parareal with $A_f \in \mathbb{R}^{s_f \times s_f}$, $A_g \in \mathbb{R}^{s_g \times s_g}$ and $s = s_f + s_g$.

Denote by I_n the $n \times n$ identity matrix, Γ_n the $n \times n$ lower shift matrix and L_n a strictly lower triangular $n \times n$ matrix filled with ones. Denote by $\mathbf{1}_n$ the vector of size n of all ones and $\boldsymbol{e}_n^i$ the i-th basis vector of size n.

Theorem 2 *If the fine and coarse solvers in Parareal are chosen to be consistent Runge-Kutta methods, then, for a fixed number of iterations $K \geq 1$, Parareal is also a Runge-Kutta method where the coefficients of its Butcher table are given by $A \in \mathbb{R}^{(K+1)Ns \times (K+1)Ns}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^{(K+1)Ns}$ with*

$$A = I_{K+1} \otimes P + \Gamma_{K+1} \otimes C, \quad \boldsymbol{b} = \boldsymbol{e}_{K+1}^{K+1} \otimes \boldsymbol{b}_P + \boldsymbol{e}_{K+1}^{K} \otimes \boldsymbol{b}_C, \quad \boldsymbol{c} = \mathbf{1}_{K+1} \otimes \boldsymbol{c}_{prog},$$

and

$$\begin{aligned} \boldsymbol{b}_P &= \mathbf{1}_N \otimes \boldsymbol{b}_{pred}, & \boldsymbol{b}_{pred} &= [\mathbf{0}; \boldsymbol{b}_g] \; . \\ \boldsymbol{b}_C &= \mathbf{1}_N \otimes \boldsymbol{b}_{corr}, & \boldsymbol{b}_{corr} &= [\boldsymbol{b}_f; -\boldsymbol{b}_g] \; . \\ P &= I_N \otimes A_s + L_N \otimes B_P, & B_P &= \mathbf{1}_s \boldsymbol{b}_{pred}^\top, & A_s &= \mathrm{diag}(A_f, A_g) \; . \\ C &= L_N \otimes B_C, & B_C &= \mathbf{1}_s \boldsymbol{b}_{corr}^\top. \\ \boldsymbol{c}_{prog} &= \boldsymbol{\iota}_N \otimes \mathbf{1}_s + \mathbf{1}_N \otimes \boldsymbol{c}_s, & \boldsymbol{c}_s &= [c_f; c_g], & \boldsymbol{\iota}_N &= [0, 1, \ldots, N-1]^\top. \end{aligned}$$

When $K = 0$, the coefficients $(A, \boldsymbol{b}, \boldsymbol{c})$ reduce to the N-fold composition of the coarse method $(A_g, \boldsymbol{b}_g, \boldsymbol{c}_g)$.

Proof. Notice that for $k \geq 0$, $\boldsymbol{u}_0^k = \boldsymbol{u}_0$ is a degenerate case of a Runge-Kutta method.

Let us proceed by induction on the iteration k. For $k = 0$,

$$\boldsymbol{u}_n^0 = \mathcal{G}(\boldsymbol{u}_{n-1}^0) = \cdots = \mathcal{G}^n(\boldsymbol{u}_0) \ .$$

From Theorem 1, as the composition of Runge-Kutta methods $\mathcal{G}^n$ is a Runge-Kutta method for $n \geq 1$. Its Butcher table can be computed by applying Theorem 1 recursively.

Now assume that for $k \geq 1$, $\boldsymbol{u}_n^k$ is a Runge-Kutta method for all $n \in \{0, \ldots, N\}$. We want to show that $\boldsymbol{u}_n^{k+1}$ is a Runge-Kutta method for all n. For this, we proceed by recurrence on n. For $n = 0$, $\boldsymbol{u}_0^{k+1}$ is a Runge-Kutta method. Assume that $\boldsymbol{u}_n^{k+1}$ is a Runge-Kutta method and let us show that $\boldsymbol{u}_{n+1}^{k+1}$ is one too. For this, recall the Parareal iteration formula,

$$\boldsymbol{u}_{n+1}^{k+1} = \mathcal{F}(\boldsymbol{u}_n^k) + \mathcal{G}(\boldsymbol{u}_n^{k+1}) - \mathcal{G}(\boldsymbol{u}_n^k) \ .$$

Since, $\mathcal{F}(\boldsymbol{u}_n^k)$, $\mathcal{G}(\boldsymbol{u}_n^k)$ and $\mathcal{G}(\boldsymbol{u}_n^{k+1})$ are Runge-Kutta methods there exist coefficients b_i^f, $\boldsymbol{k}_i^{f,k}$, b_i^g, $\boldsymbol{k}_i^{g,k}$, $\boldsymbol{k}_i^{g,k+1}$ such that

$$\begin{aligned} \mathcal{F}(\boldsymbol{u}_n^k) &= \boldsymbol{u}_n^k + \Delta T \sum_{i=1}^{s_f} b_i^f \boldsymbol{k}_i^{f,k} \ , \\ \mathcal{G}(\boldsymbol{u}_n^k) &= \boldsymbol{u}_n^k + \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k} \ , \\ \text{and} \quad \mathcal{G}(\boldsymbol{u}_n^{k+1}) &= \boldsymbol{u}_n^{k+1} + \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k+1} \ . \end{aligned}$$

Then,

$$\begin{aligned} \boldsymbol{u}_{n+1}^{k+1} &= \boldsymbol{u}_n^k + \Delta T \sum_{i=1}^{s_f} b_i^f \boldsymbol{k}_i^{f,k} + \boldsymbol{u}_n^{k+1} + \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k+1} - \boldsymbol{u}_n^k - \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k} \\ &= \boldsymbol{u}_n^{k+1} + \Delta T \sum_{i=1}^{s_f} b_i^f \boldsymbol{k}_i^{f,k} + \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k+1} - \Delta T \sum_{i=1}^{s_g} b_i^g \boldsymbol{k}_i^{g,k} \ . \end{aligned}$$

As $\boldsymbol{u}_n^{k+1}$ is a Runge-Kutta method by assumption, then $\boldsymbol{u}_{n+1}^{k+1}$ is also a Runge-Kutta method. When writing the following coefficients explicitly, we get the Butcher table given in the statement of the theorem. □

Remark 1 The assumption that the fine and coarse solvers in Parareal are consistent is not necessary for the proof. However, one needs to ensure that the Butcher tables of $\mathcal{F}$ and $\mathcal{G}$ are normalized to the same time step, that is, requiring that $\boldsymbol{b}_f^\top \mathbf{1}_{s_f} = \boldsymbol{b}_g^\top \mathbf{1}_{s_g}$.

$$
\begin{array}{c|ccc}
\boldsymbol{c}_g & A_g & & \\
1\mathbf{1}+\boldsymbol{c}_g & \mathbf{1}\boldsymbol{b}_g^\top & A_g & \\
2\mathbf{1}+\boldsymbol{c}_g & \mathbf{1}\boldsymbol{b}_g^\top & \mathbf{1}\boldsymbol{b}_g^\top & A_g \\
\hline
 & \boldsymbol{b}_g^\top & \boldsymbol{b}_g^\top & \boldsymbol{b}_g^\top
\end{array}
$$

Iteration 0

$$
\begin{array}{c|cccccc}
\boldsymbol{c}_f & A_f & & & & & \\
\boldsymbol{c}_g & & A_g & & & & \\
1\mathbf{1}+\boldsymbol{c}_f & & \mathbf{1}\boldsymbol{b}_g^\top & A_f & & & \\
1\mathbf{1}+\boldsymbol{c}_g & & \mathbf{1}\boldsymbol{b}_g^\top & & A_g & & \\
2\mathbf{1}+\boldsymbol{c}_f & & \mathbf{1}\boldsymbol{b}_g^\top & & \mathbf{1}\boldsymbol{b}_g^\top & A_f & \\
2\mathbf{1}+\boldsymbol{c}_g & & \mathbf{1}\boldsymbol{b}_g^\top & & \mathbf{1}\boldsymbol{b}_g^\top & & A_g \\
\hline
 & & \boldsymbol{b}_g^\top & & \boldsymbol{b}_g^\top & & \boldsymbol{b}_g^\top
\end{array}
$$

Iteration 1 — fine propagation

$$
\begin{array}{c|ccccccccc}
\boldsymbol{c}_f & A_f & & & & & & & & \\
\boldsymbol{c}_g & & A_g & & & & & & & \\
1\mathbf{1}+\boldsymbol{c}_f & & \mathbf{1}\boldsymbol{b}_g^\top & A_f & & & & & & \\
1\mathbf{1}+\boldsymbol{c}_g & & \mathbf{1}\boldsymbol{b}_g^\top & & A_g & & & & & \\
2\mathbf{1}+\boldsymbol{c}_f & & \mathbf{1}\boldsymbol{b}_g^\top & & \mathbf{1}\boldsymbol{b}_g^\top & A_f & & & & \\
2\mathbf{1}+\boldsymbol{c}_g & & \mathbf{1}\boldsymbol{b}_g^\top & & \mathbf{1}\boldsymbol{b}_g^\top & & A_g & & & \\
\boldsymbol{c}_g & & & & & & & A_g & & \\
1\mathbf{1}+\boldsymbol{c}_f & \mathbf{1}\boldsymbol{b}_f^\top & -\mathbf{1}\boldsymbol{b}_g^\top & & & & & \mathbf{1}\boldsymbol{b}_g^\top & A_f & \\
2\mathbf{1}+\boldsymbol{c}_g & \mathbf{1}\boldsymbol{b}_f^\top & -\mathbf{1}\boldsymbol{b}_g^\top & \mathbf{1}\boldsymbol{b}_f^\top & -\mathbf{1}\boldsymbol{b}_g^\top & & & \mathbf{1}\boldsymbol{b}_g^\top & \mathbf{1}\boldsymbol{b}_g^\top & A_g \\
\hline
 & \boldsymbol{b}_f^\top & -\boldsymbol{b}_g^\top & \boldsymbol{b}_f^\top & -\boldsymbol{b}_g^\top & \boldsymbol{b}_f^\top & -\boldsymbol{b}_g^\top & \boldsymbol{b}_g^\top & \boldsymbol{b}_g^\top & \boldsymbol{b}_g^\top
\end{array}
$$

Iteration 1

$$
\begin{array}{c|ccccccccc}
\boldsymbol{c}_s & A_s & & & & & & & & \\
1\mathbf{1}+\boldsymbol{c}_s & B_{pred} & A_s & & & & & & & \\
2\mathbf{1}+\boldsymbol{c}_s & B_{pred} & B_{pred} & A_s & & & & & & \\
\boldsymbol{c}_s & & & & A_s & & & & & \\
1\mathbf{1}+\boldsymbol{c}_s & B_{corr} & & & B_{pred} & A_s & & & & \\
2\mathbf{1}+\boldsymbol{c}_s & B_{corr} & B_{corr} & & B_{pred} & B_{pred} & A_s & & & \\
\boldsymbol{c}_s & & & & & & & A_s & & \\
1\mathbf{1}+\boldsymbol{c}_s & & & & B_{corr} & & & B_{pred} & A_s & \\
2\mathbf{1}+\boldsymbol{c}_s & & & & B_{corr} & B_{corr} & & B_{pred} & B_{pred} & A_s \\
\hline
 & & & & \boldsymbol{b}_{corr}^\top & \boldsymbol{b}_{corr}^\top & \boldsymbol{b}_{corr}^\top & \boldsymbol{b}_{pred}^\top & \boldsymbol{b}_{pred}^\top & \boldsymbol{b}_{pred}^\top
\end{array}
$$

Iteration 2

Table 1 Example of the construction of the Butcher table for Parareal with $N = 3$. Note that iteration 2 uses the notation from Theorem 2 for simplicity. Note that $2\mathbf{1}$ is a vector of all twos as it is the product of the vector $\mathbf{1}$ with the scalar 2.

Table 1 shows the iterations 0, 1 and 2 of the general Parareal Butcher table with $N = 3$ Parareal intervals. At iteration 2, we can see the block structure described in Theorem 2.

3 Reduction of Butcher Tables

Notice that Butcher tables are not unique, indeed two Butcher tables can lead to two methods that give the same approximate solution (up to numerical round-off error considerations). For analysis purposes, it is useful to know how to reduce a Butcher table.

Two criteria exist to reduce Butcher tables. For simplicity, we present them here as principles, for formal definitions see [4, Definition 2] and [11, Definition 3.2.5].

1. *DJ-reducibility*: if a stage does not influence the final solution, it can be removed.
2. *S-reducibility*: if we can compute a stage with less computations, we should do it.

Example 2 The following three methods reduce to the Crank-Nicolson method.

$$
\begin{array}{c|cc}
0 & 0 & 0 \\
1 & 1/2 & 1/2 \\
\hline
 & 1/2 & 1/2
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
1/2 & 1/4 & 1/4 & 0 \\
1 & 1/2 & 0 & 1/2 \\
\hline
 & 1/2 & 0 & 1/2
\end{array}
\qquad
\begin{array}{c|ccc}
0 & 0 & 0 & 0 \\
1 & 1/2 & 1/2 & 0 \\
1 & 1/2 & 0 & 1/2 \\
\hline
 & 1/2 & 1/4 & 1/4
\end{array}
$$

Crank-Nicolson method   DJ-reducible method   S-reducible method

The first is the Crank-Nicolson method as usually presented. The second is a method that DJ-reduces to the Crank-Nicolson scheme. Indeed, the second stage is not used by any of the other stages or the final approximation though the coefficients b_i. The third, S-reduces to the Crank-Nicolson scheme as its second and third stage do the same computation. We can thus sum the second and third columns and remove the third line.

An important property of Parareal that we will use to reduce its Butcher table is its *finite-time convergence property* which states that, at each iteration, we get the exact solution on one additional Parareal interval.

Theorem 3 (Parareal finite-time convergence, Theorem 2.4 in [6]) *The Parareal algorithm has the property that*

$$\boldsymbol{u}_n^k = \mathcal{F}^n(\boldsymbol{u}_0), \quad \textit{for } k \geq n \ .$$

We can now explain how to reduce the Butcher table of Parareal.

Proposition 1 *Consider Parareal with a fixed number of iterations K with the final approximation being $\boldsymbol{u}_N^K$, then the stages related to $\boldsymbol{u}_n^k$ where $k > n$ can be S-reduced to stages related to $\boldsymbol{u}_n^n$, and all stages $\boldsymbol{u}_n^k$ with $n - k > N - K$ can be DJ-reduced.*

Proof. By the finite time convergence property of Parareal, $\boldsymbol{u}_n^n = \boldsymbol{u}_n^k$ for $k > n$ and thus, as the stages related to $\boldsymbol{u}_n^k$ are the same as the stages related to $\boldsymbol{u}_n^n$, they can be S-reduced.

By the Parareal iteration formula, the computation of $\boldsymbol{u}_N^K$ depends on $\boldsymbol{u}_{N-1}^K$ and $\boldsymbol{u}_{N-1}^{K-1}$. Iterating this formula, we have that $\boldsymbol{u}_N^K$ depends on $\boldsymbol{u}_n^k$ for $k < K$ and for $n - k \leq N - K$. Thus, all the stages related $\boldsymbol{u}_n^k$ with $n - k > N - K$ can be DJ-reduced as they do not contribute to the computation of $\boldsymbol{u}_N^K$. □

Example 3 In Table 2, we present the reduced general Butcher table for Parareal with $N = 3$ Parareal intervals and $K = 0, 1, 2, 3$ iterations. Notice that for $K = 3$, we obtain the Butcher table of the N-fold composition of the fine method with itself as predicted by the finite-time convergence property of Parareal.

$$
\begin{array}{c|ccc}
c_g & A_g & & \\
1\mathbf{1} + c_g & \mathbf{1}b_g^\top & A_g & \\
2\mathbf{1} + c_g & \mathbf{1}b_g^\top & \mathbf{1}b_g^\top & A_g \\
\hline
 & b_g^\top & b_g^\top & b_g^\top
\end{array}
$$

Iteration 0

$$
\begin{array}{c|cccccccc}
c_f & A_f & & & & & & & \\
c_g & & A_g & & & & & & \\
1\mathbf{1} + c_f & & \mathbf{1}b_g^\top & A_f & & & & & \\
1\mathbf{1} + c_g & & \mathbf{1}b_g^\top & & A_g & & & & \\
2\mathbf{1} + c_f & & \mathbf{1}b_g^\top & & \mathbf{1}b_g^\top & A_f & & & \\
2\mathbf{1} + c_g & & \mathbf{1}b_g^\top & & \mathbf{1}b_g^\top & & A_g & & \\
1\mathbf{1} + c_g & \mathbf{1}b_f^\top & & & & & & A_g & \\
2\mathbf{1} + c_g & \mathbf{1}b_f^\top & & \mathbf{1}b_f^\top & -\mathbf{1}b_g^\top & & & \mathbf{1}b_g^\top & A_g \\
\hline
 & b_f^\top & & b_f^\top & -b_g^\top & b_f^\top & -b_g^\top & b_g^\top & b_g^\top
\end{array}
$$

Iteration 1

$$
\begin{array}{c|ccccccccc}
c_f & A_f & & & & & & & & \\
c_g & & A_g & & & & & & & \\
1\mathbf{1} + c_f & & \mathbf{1}b_g^\top & A_f & & & & & & \\
1\mathbf{1} + c_g & & \mathbf{1}b_g^\top & & A_g & & & & & \\
1\mathbf{1} + c_f & \mathbf{1}b_f^\top & & & & A_f & & & & \\
1\mathbf{1} + c_g & \mathbf{1}b_f^\top & & & & & A_g & & & \\
2\mathbf{1} + c_f & \mathbf{1}b_f^\top & & \mathbf{1}b_f^\top & -\mathbf{1}b_g^\top & & \mathbf{1}b_g^\top & A_f & & \\
2\mathbf{1} + c_g & \mathbf{1}b_f^\top & & \mathbf{1}b_f^\top & -\mathbf{1}b_g^\top & & \mathbf{1}b_g^\top & & A_g & \\
2\mathbf{1} + c_g & \mathbf{1}b_f^\top & & & & \mathbf{1}b_f^\top & & & & A_g \\
\hline
 & b_f^\top & & & & b_f^\top & & b_f^\top & -b_g^\top & b_g^\top
\end{array}
$$

Iteration 2

$$
\begin{array}{c|ccc}
c_f & A_f & & \\
1\mathbf{1} + c_f & \mathbf{1}b_f^\top & A_f & \\
2\mathbf{1} + c_f & \mathbf{1}b_f^\top & \mathbf{1}b_f^\top & A_f \\
\hline
 & b_f^\top & b_f^\top & b_f^\top
\end{array}
$$

Iteration 3

Table 2 Reduced Butcher tables of Parareal for with $N = 3$ Parareal intervals and for iterations $K = 0$ thought $K = 3$.

Example 4 Table 2 can be used to check the order conditions for Parareal using one step of the Implicit Midpoint rule (order 2) and one step of Backward Euler (order 1) as the fine and coarse operators in Parareal. The corresponding coefficients are given by $(A_f, b_f, c_f) = (1/2, 1, 1/2)$ and $(A_g, b_g, c_g) = (1, 1, 1)$.

The order conditions for the first three orders are given by: $b^\top \mathbf{1} = 1$ (order 1), $b^\top A\mathbf{1} = 1/2$ (order 2), $b^\top(A\mathbf{1} \cdot A\mathbf{1}) = 1/3$ and $b^\top AA\mathbf{1} = 1/6$ (order 3). At iteration $K = 0$, $b^\top \mathbf{1} = 1$ but $b^\top A\mathbf{1} = 3/4$. At iteration $K = 1$, $b^\top \mathbf{1} = 1$, $b^\top A\mathbf{1} = 1/2$ but $b^\top(A\mathbf{1} \cdot A\mathbf{1}) = 5/16$ and $b^\top AA\mathbf{1} = 3/16$. So, in that case, the order increases by one at each iteration as described in [10, Proposition 1].

4 Discussion

Current research on the development of Parareal focuses on choosing coarse operators to accelerate the convergence of the algorithm [5, 7, 9, 14]. Our new result allows us to take advantage of the numerous results that exist for Runge-Kutta methods and apply them to Parareal to better understand the algorithm. The knowledge acquired that way can inform the design of better Parareal-like algorithms. Some ideas in that direction can be found in [2, 12, 13].

The code to construct the Butcher table of Parareal is available at: `https://gitlab.unige.ch/Ausra.Pogozelskyte/parareal-as-runge-kutta`. Two examples in `parareal-as-runge-kutta/scripts/` verify numerically that classic Parareal and the Butcher table approach numerically yield the same solution and demonstrates how one can compute the order of Parareal as in Example 4.

References

1. Bal, G.: On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: Domain decomposition methods in science and engineering, *Lect. Notes Comput. Sci. Eng.*, vol. 40, pp. 425–432. Springer, Berlin (2005)
2. Bronasco, A.: Improving the Efficiency and Theoretical Understanding of Time-Parallel Multigrid Methods. Ph.D. thesis, Université de Genève (2025). URL https://archive-ouverte.unige.ch/unige:187048
3. Butcher, J.C.: An algebraic theory of integration methods. Math. Comp. **26**, 79–106 (1972)
4. Dahlquist, G., Jeltsch, R.: Generalized disks of contractivity for explicit and implicit Runge-Kutta methods. Tech. Rep. TRITA-NA-79-06, CM-P00069451 (1979)
5. De Sterck, H., Falgout, R.D., Krzysik, O.A., Schroder, J.B.: Parallel-in-time solution of scalar nonlinear conservation laws (2024). ArXiv:2401.04936
6. Gander, M.J., Lunet, T.: Time parallel time integration. SIAM, Philadelphia (2024)
7. Gander, M.J., Ohlberger, M., Rave, S.: A Parareal Algorithm with Spectral Coarse Solver (2025). ArXiv:2508.08873
8. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations. I, *Springer Series in Computational Mathematics*, vol. 8, second edn. Springer-Verlag, Berlin (1993)
9. Ibrahim, A.Q., Götschel, S., Ruprecht, D.: Space-Time Parallel Scaling of Parareal with a Physics-Informed Fourier Neural Operator Coarse Propagator Applied to the Black-Scholes Equation. In: Proceedings of the Platform for Advanced Scientific Computing Conference, PASC '25, pp. 1–11. Association for Computing Machinery, New York, NY, USA (2025)
10. Lions, J.L., Maday, Y., Turinici, G.: Résolution d'EDP par un schéma en temps "pararéel". C. R.Acad. Sci. Paris Sér. I. Math. **332**(7), 661–668 (2001)
11. Stetter, H.J.: Analysis of discretization methods for ordinary differential equations, *Springer Tracts in Natural Philosophy*, vol. Vol. 23. Springer-Verlag, New York-Heidelberg (1973)
12. Vargas, D.A.: Parallel Multigrid in Time for Chaotic Dynamical Systems. Ph.D. thesis, The University of New Mexico (2024)
13. Yoda, R., Bolten, M., Nakajima, K., Fujii, A.: Coarse-grid operator optimization in multigrid reduction in time for time-dependent Stokes and Oseen problems. Jpn. J. Ind. Appl. Math. **41**(3), 1315–1339 (2024)
14. Yue, X., Wang, Z., Wu, S.L.: Convergence Analysis of a Mixed Precision Parareal Algorithm. SIAM J. Sci. Comput. **45**(5), A2483–A2510 (2023)