Three Different Multigrid Interpretations of the Parareal Algorithm and an Adaptive Variant

Martin J. Gander

Time parallel time integration methods have a long history [1], and the parareal algorithm [2] sparked renewed interest in such methods. To define the parareal algorithm for the ordinary differential equation

$$\frac{du}{dt} = f(u), \quad \text{in } (0,T) \text{ with } u(0) = u_0, \tag{1}$$

one needs a coarse solver $G(t_2, t_1, u_1)$ which solves the differential equation in (1) starting with initial value u_1 at t_1 and gives an approximate solution at time t_2, and a fine solver $F(t_2, t_1, u_1)$ which does the same with much more accuracy. The parareal algorithm is then defined for a partition of the time interval $(0, T)$ into subintervals $0 = T_0 < T_1 < T_2 < \ldots < T_N = T$ by the iteration

$$U_{n+1}^{k+1} = F(T_{n+1}, T_n, U_n^k) + G(T_{n+1}, T_n, U_n^{k+1}) - G(T_{n+1}, T_n, U_n^k), \tag{2}$$

where k is the iteration index, $U_0^{k+1} = u_0$, and the initial approximation can be obtained for example using the coarse solver,

$$U_{n+1}^0 = G(T_{n+1}, T_n, U_n^0), \quad U_0^0 = u_0. \tag{3}$$

The values U_n^k approximate the solution $u(T_n)$ of (1). The most natural interpretation of the parareal algorithm is that it is a multiple shooting method with approximate Jacobian on a coarse grid, and its convergence properties are well understood, see [3] for linear partial differential equations, and [4] for the non-linear case.

Because of the two grids that are often used, a fine one for F and a coarse one for G, the parareal algorithm is also a two-grid method, and it is interesting to consider multigrid variants. There are three ways to obtain these, see [5]:

(1) In the linear case, one can write the parareal iteration (2) as a preconditioned Richardson iteration, where the preconditioner is given by the coarse solve, see also [6]. One can then easily apply again (2) to approximately invert the coarse solver and get a multilevel parareal method.

(2) The parareal iteration (2) can also be interpreted in the geometric multigrid setting as a two level method using one presmoothing step with block Jacobi, but not updating the coarse nodes (sometimes called an F-smoother), and using injection for the restriction R, with prolongation $P := R^T$, and for the coarse matrix a coarse time stepper. This holds also in the non-linear setting using the full approximation scheme, see for example [3]. Again then applying the algorithm recursively for the coarse time stepper leads to a multilevel version.

(3) Finally, one can consider the parareal algorithm (2) in the framework of algebraic multigrid, where the nodes are first partitioned into fine, so called F-nodes, and coarse, so called C-nodes. It can then be shown that the parareal algorithm in fact uses optimal restriction and prolongation

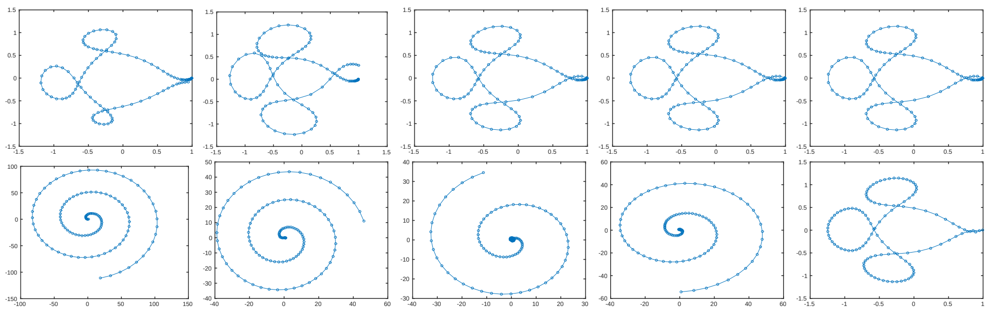

FIGURE 1. Initial approximation and first four iterations of the parareal algorithm applied to the Arenstorf problem. Top row: adaptive variant. Bottom row: fixed step size variant.

operators, and only approximates the optimal coarse correction by a simple coarse solve. This interpretation, together with a modification from the F-smoother to a so called FCF-smoother led to the MGRIT algorithm in [7], which corresponds to a parareal algorithm with overlap, see [5].

A new idea is to use the parareal algorithm adaptively as follows: one first determines the time intervals T_n using an adaptive coarse solver G in the initialization step (3). On this time grid, one then runs the correction iteration (2), also using adaptive fine solvers F and coarse solvers G, without changing the time partition T_n any more. To illustrate this, we now solve an Arenstorf orbit problem. Arenstorf orbits are non-trivial closed orbits of a light object moving in the gravity field of two heavy objects, following the equations of motion

$$\ddot{x} = x + 2\dot{y} - b\frac{x+a}{D_1} - a\frac{x-b}{D_2}, \quad \ddot{y} = y - 2\dot{x} - b\frac{y}{D_1} - a\frac{y}{D_2},$$

where D_j, $j = 1, 2$ are functions of x and y,

$$D_1 = ((x+a)^2 + y^2)^{\frac{3}{2}}, \; D_2 = ((x-b)^2 + y^2)^{\frac{3}{2}}.$$

If the parameters are $a = 0.012277471$ and $b = 1 - a$, with initial conditions $x(0) = 0.994$, $\dot{x}(0) = 0$, $y(0) = 0$, $\dot{y}(0) = -2.00158510637908$, then the solution is a nice closed orbit with period $T = 17.06521656015796$, see [8], which can be interpreted as a space craft that tries to return from moon to earth and unfortunately lands again back at the moon, as illustrated by the converged trajectory on top right in Figure 1. The top row in this figure represents the initial approximation and the first four iterations of the parareal algorithm when using `ode45` in Matlab with tolerance $1e-2$ for the coarse integration[1], determining in the first iteration also the adaptive time partition, and tolerance $1e-10$ for the fine integration. Convergence to an error tolerance of $1e-6$ is achieved

[1]A small modification was needed to reduce the minimum number of time steps Matlab takes from 10 to 1 to avoid over-resolution

in four iterations, as one can see in Figure 2. The accuracy in this converged solution is also $1e-6$, and the adaptive parareal algorithm needed a total of $36'458$ function evaluations, using 121 coarse time steps. We next use 121 equidistant coarse time steps and one time step of the classical 4th order Runge-Kutta method for the coarse integrator, and found that 4133 equidistant 4th order Runge-Kutta steps are needed in each coarse time interval for the fine integrator to reach the same error of size $1e-6$. Running the parareal algorithm with these fixed step sizes leads to the result shown in Figure 1 in the bottom row. We see that the initial guess and the first three iterations are very far away from the solution, and only the fourth iteration brings the trajectory closer to the recognizable shape of the Arenstorf orbit. It takes then almost twice the number of iterations to converge, see Figure 2, using a total of $20'011'948$ function evaluations! This is about 550 times more than the adaptive parareal algorithm, for the same accuracy. The adaptive parareal algorithm has the same potential for parallelism and multilevel extension.

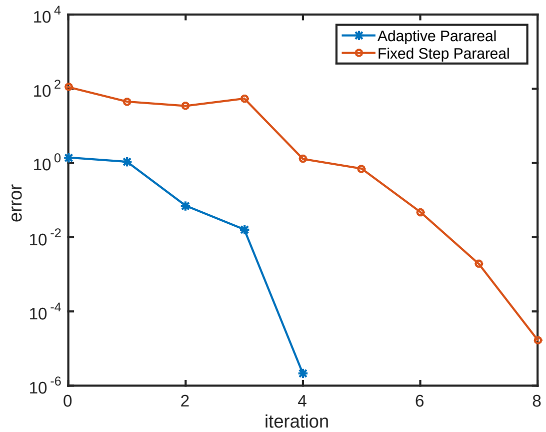

FIGURE 2. Decay of the error as a function of the iteration.

REFERENCES

[1] Martin J. Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods*, pages 69–113. Springer, 2015.

[2] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. A parareal in time discretization of PDEs. *C.R. Acad. Sci. Paris, Serie I*, 332:661–668, 2001.

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

[4] Martin J. Gander and Ernst Hairer. Nonlinear convergence analysis for the parareal algorithm. In Olof B. Widlund and David E. Keyes, editors, *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 45–56. Springer, 2008.

[5] Martin J. Gander, Felix Kwok and Hui Zhang. Multigrid Interpretations of the Parareal Algorithm Leading to an Overlapping Variant. In preparation, 2017.

[6] Daniel Ruprecht, Robert Speck, and Rolf Krause. Parareal for diffusion problems with space- and time-dependent coefficients. In *Domain Decomposition Methods in Science and Engineering XXII*, pages 371–378. Springer, 2016.

[7] Robert D. Falgout, Stephanie Friedhoff, T.V. Kolev, Scott P. MacLachlan, and Jacob B. Schroder. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing*, 36(6):C635–C661, 2014.

[8] Ernst Hairer, Syvert P. Nørsett and Gerhard Wanner. Solving Ordinary Differential Equations I: Nonstiff Problems. Springer-Verlag, Second Revised Edition, 1993.