# Cascadic Parareal Method for Explicit Time-Marching Schemes

Yen-Chen Chen
The University of Tokyo
Japan

Kengo Nakajima
The University of Tokyo
Japan

## 1 INTRODUCTION

Parallel computation with a very large number of cores reaches a saturation point in the spatial dimension due to the latency of supercomputers. With the increase of core numbers, the communication and synchronization costs will gradually dominants the computation cost. This problem limits the parallel efficiency of time-dependent problems. Parallel-in-time methods parallelize a problem in the time dimension to extract further parallel efficiency.. So far parallel-in-time (PinT) methods such as parareal[5] and multigrid reduction-in-time (MGRIT)[4] are shown to provide successful converged results for small problems.

However, PinT methods have yet to be applied to large-scale problems, and it still requires a huge amount of cores to achieve reasonable computation acceleration. Falgout et al. (2014)[4] shows that it requires at least 256 cores to achieve faster computation time than spatial parallelization of an implicit time-stepping for Poisson equation. Christopher et al. (2020)[2] shows that it requires at least 1024 cores to achieve faster computation time of an explicit time-stepping for a Couette flow problem. Moreover, very few PinT works have been conducted for explicit schemes since explicit schemes are very fast and highly scalable for spatial parallelization.

## 2 CASCADIC PARAREAL

We propose a cascadic parareal method, which is a PinT method optimized for explicit schemes. The cascadic parareal method optimized the parareal method to work with explicit schemes and by the number of parallel processes in the time domain. The whole time interval is divided into $P_t$ time subintervals, within where fine solvers of the parareal method is solved in parallel. Coarse solvers with larger time steps solve on coarse space grids to prevent violating the CFL-condition[3]. We further construct multiple levels with different time steps and apply solving process similar to the cascadic multigrid method[1], as described in Algorithm 1.

**Algorithm 1:** Cascadic Parareal

```
Sequential solve on the coarsest level L. u_c = Ψ^x(R u_j^{k+1})
for level l = L-1 to 1 do
    Take initial values from level l-1.
    for iterate until residual tolerance do
        On current core:
        Solve on the current level in parallel u_f = Φ^{2x}(u_j^k).
        for Core p = 1 to P do
            Solve on the coarsest level
            u_{j+1}^{k+1} = PΨ^x(R u_j^{k+1}) + Φ^{2x}(u_j^k) − PΨ^x(R u_j^k)
            Update values to level l with prolongation
        end
    end
end
```

## 3 PINT FOR ADVECTION EQUATION

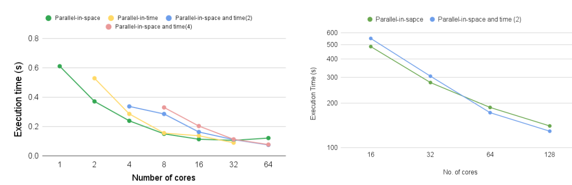
$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}$$

PinT methods converge slower on hyperbolic equations, and therefore advection equation has been a common numerical experiment for PinT methods. We show in Figure 1a that with more than 64 cores, cascadic parareal could achieve faster execution time than spatial parallelization.

## 4 PINT FOR COMPRESSIBLE SUBSONIC FLOW

$$\frac{\partial U}{\partial t} + \nabla \cdot \vec{F} = \nabla \cdot \vec{R}$$

For a larger example, we simulate the compressible flow around a cylinder. We solve from an initial velocity of 0.3 Mach (subsonic flow) to a steady state without perturbation. Figure 1b shows that cascadic parareal achieve faster execution time than spatial parallelization with more than 64 cores.




**(a) Parallel-in-space and time ($P_t$) compare to spatial parallelization for adevction equation. The problem size $N_x \times N_t$ = 4097 × 8192**

**(b) Parallel-in-space and time ($P_t$) compare to spatial parallelization for compressible flow simulation. The problem size $N_x \times N_t$ = 32000 × 100000.**

**Figure 1: Parallel performance for cascadic parareal.**

## 5 SUPERSONIC FLOW

Cascadic parareal coarsens the spatial grid for coarse solvers to satisfy the CFL-condition. Thus, it has trouble converging for discontinuous problems such as a shock wave simulation. We are working on applying adaptive mesh refinement with cascadic parareal for supersonic flow.

## REFERENCES

[1] Bornemann, F. A., and Deuflhard, P. The cascadic multigrid method for elliptic problems. *Numerische Mathematik 75*, 2 (1996), 135–152.
[2] Christopher, J., Falgout, R. D., Schroder, J. B., Guzik, S. M., and Gao, X. A space-time parallel algorithm with adaptive mesh refinement for computational fluid dynamics. *Computing and Visualization in Science 23*, 1 (2020), 1–20.
[3] Courant, R., Friedrichs, K., and Lewy, H. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen 100*, 1 (1928), 32–74.
[4] Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. Parallel time integration with multigrid. *SIAM Journal on Scientific Computing 36*, 6 (2014), C635–C661.
[5] Lions, J.-L., Maday, Y., and Turinici, G. Résolution d'edp par un schéma en temps «pararéel». *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics 332*, 7 (2001), 661–668.