Performance of parallel-in-time integration for Rayleigh Bénard Convection

Andrew Clarke · Chris Davies · Daniel Ruprecht · Steven Tobias · Jeffrey S. Oishi

Received: date / Accepted: date

Abstract Rayleigh-Bénard convection (RBC) is a fundamental problem of fluid dynamics, with many applications to geophysical, astrophysical, and industrial flows. Understanding RBC at parameter regimes of interest requires complex physical or numerical experiments. Numerical simulations require large amounts of computational resources; in order to more efficiently use the large numbers of processors now available in large high performance computing clusters, novel parallelisation strategies are required. To this end, we investigate the performance of the parallel-in-time algorithm Parareal when used in numerical simulations of RBC. We present the first parallel-in-time speedups for RBC simulations at finite Prandtl number. We also investigate the problem of convergence of Parareal with respect to statistical numerical quantities, such as the Nusselt number, and discuss the importance of reliable online stopping criteria in these cases.

Keywords Parareal · Rayleigh-Bénard · parallel-in-time

1 Introduction

Rayleigh-Bénard convection (RBC) is an archetypal problem in fluid dynamics, describing the buoyancy driven flow of a fluid heated from below and cooled from above [1]. It allows for studying fundamental properties of fluid dynamics and is used as a simplified analogue for astrophysical and geophysical systems such as planetary interiors, stars, and the atmosphere [8], [17].

RBC is the convection of a fluid driven by a vertical temperature gradient $\Delta T$ between two horizontal plates separated by a distance $L$. The problem can be characterised by three non-dimensional parameters. The Rayleigh number is given by

$$Ra = \frac{\alpha g \Delta T L^3}{\nu \kappa},$$

where $\alpha$ is the coefficient of thermal expansion, $\nu$ is the kinematic viscosity of the fluid, $g$ is gravity, and $\kappa$ is the thermal diffusivity. The Prandtl number is

$$Pr = \frac{\nu}{\kappa}.$$
The third controlling feature of the flow is the aspect ratio of the domain, \( L_x/L_z \) where \( L_x, L_z \) are the horizontal and vertical size of the domain. The Rayleigh number is a measure of how much the flow is driven by the temperature, while the Prandtl number is an inherent property of the fluid.

Very high or infinite Prandtl number is used as a model for convection in the Earth’s mantle \([33]\), while a Prandtl number \( \sim 1 \) is commonly used in simulations of the Earth’s core, see for example \([31, 28]\). In this work we investigate cases where \( Pr = 1, L_x/L_z = 2 \) and focus on the effects of changes in the Rayleigh number.

Rayleigh-Bénard convection has been studied intensively throughout the last few decades and before, see for example the papers by Siggia \([36]\) or Verzicco and Camussi \([40]\). Some notable studies utilising Rayleigh-Bénard convection include Cattaneo et al. \([9]\) who studied solar magnetic field interactions, Glatzmaier and Roberts \([13]\), who produced the first simulation of a geomagnetic field reversal, and McKenzie et al. \([29]\) who studied the effect of mantle flow in the earth. For more in depth reviews of the subject, see for example Bodenschatz et al. \([10]\) or Ahlers et al. \([7]\).

Much interest has developed in the behaviour of a fluid convecting at high Rayleigh numbers. This is an important area of study, as high Rayleigh numbers are thought to be present in many geophysical and astrophysical bodies. Different scaling regimes are believed to exist at different orders of Rayleigh number, and much work has been done to find the exact scaling behaviour of the Nusselt number \((Nu, \text{defined below})\), with \( Ra \), see for example Grossmann and Lohse \([19]\), Cioni et al. \([11]\), Kerr \([22]\), and Siggia \([36]\).

To test the theories describing this behaviour, experiments at higher and higher \( Ra \) are required, a difficult task to achieve, either numerically or experimentally. Much of this work is now done through direct numerical simulations, see for example Zhu et al. \([11]\) and Schumacher \([34]\). These studies require an enormous amount of computational power \([26]\) and, due to constraints on parallel performance, there is a need to investigate further options for increasing the degree of parallelism in simulation codes.

One such option which has gained much interest in recent years is parallel-in-time integration. This allows the time domain to be parallelised in a similar way to how the spatial domain is commonly parallelised. The recent interest in parallel in time methods was sparked by the introduction of the Parareal algorithm by Lions et al. \([27]\). Subsequently, much research has been carried out in this area; new parallel in time algorithms such as Parallel Full Approximation Scheme in Space and Time (PFASST, Minion \([30]\), Parallel implicit time-integrator (PITA, Farhat and Chandesris \([13]\)), and Multigrid Reduction in Time (MGRIT, Friedhoff et al. \([14]\)) have been proposed. For a comprehensive review see for example Gander \([15]\).

In this work, we present the first reported speedup from parallel-in-time integration for the problem of RBC at finite Prandtl number. We extend the work of Samuel \([35]\), who studied the performance of Parareal for infinite Prandtl, into a regime with more varied geo- and astro-physical applications. For infinite Prandtl number, the time derivative in the momentum equation vanishes and temperature is the only prognostic variable. In contrast, for finite Prandtl number, both velocity and temperature have to be integrated in time. Samuel reported speedups of up to 10 were found when using up to 40 CPUs for infinite Prandtl number, when combining Parareal with spatial parallelisation. These results were largely in line with the theoretical performance model they developed. Recently, Kooij \([25]\) discussed parallel-in-time methods as an attractive option for simulations of Rayleigh Bénard convection, but did not supply any results in this direction.

Our results show that Parareal can faithfully reproduce the relationship between Rayleigh- and Nusselt number found in the literature. Given that the number of studies of Parareal for problems with non-linear complex dynamics is limited, this is a useful result in itself. We further investigate the convergence properties of Parareal with respect to the \( L^2 \) error between individual trajectories as well as averaged quantities. While the former is typically used as a termination criterion for Parareal, the latter is often more relevant for applications. Our results show that, particularly for flows at high Rayleigh number, Parareal can fail to converge to the fine trajectory while still converging to the correct averaged dynamics. Only at Rayleigh numbers beyond \( 10^7 \) does Parareal’s convergence start to deteriorate. This suggests that research into alternative termination criteria for Parareal, aimed at reproducing correct statistics instead of individual trajectories, would be a useful direction for future research.

2 Rayleigh Bénard Convection

2.1 Equations and Domain

We use the Boussinesq approximation to the Navier-Stokes equations for fluid flow in a 2D Cartesian domain. The non-dimensional Oberbeck-Boussinesq equations modelling Rayleigh - Bénard convection can be written as

\[
\frac{1}{Pr} \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = - \nabla p + Ra \theta \cdot \widehat{z} + \nabla^2 u, \tag{3}
\]
\[ \nabla \cdot \mathbf{u} = 0, \quad (4) \]
\[ \frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla^2 T, \quad (5) \]
with fixed temperature
\[ T|_{z=-0.5} = 1, \quad T|_{z=0.5} = 0, \quad \mathbf{u}|_{z=-0.5,0.5} = 0, \quad (6) \]
and fixed flux
\[ \frac{\partial T}{\partial z}|_{z=-0.5,0.5} = -1, \quad \mathbf{u}|_{z=-0.5,0.5} = 0 \quad (7) \]
boundary conditions.

Here, \( \mathbf{u} = (u, w) \) represents the horizontal and vertical velocity of the fluid, \( T \) represents the temperature, \( t \) represents time and \( p \) is pressure. The fundamental time scale is taken as a thermal diffusion time \( \tau_T \sim L^2/\kappa \), \( T \) is scaled by \( \Delta T \), and length is scaled by \( L \). We use a domain of size \( (x = 2, \ z = 1) \), where \( x \) is the horizontal direction, and \( z \) the vertical, giving an aspect ratio of 2. We begin with a linear temperature profile with small perturbations and \( \mathbf{u} = 0 \).

In the fixed flux case, we use the flux Rayleigh number \( Ra_f \), defined as
\[ Ra_f = \frac{\alpha g \beta L^4}{\nu \kappa}, \quad (8) \]
where \( \beta \) is the imposed vertical heat flux, instead of the standard Rayleigh number in the momentum equation. The flux Rayleigh number can be related to the standard Rayleigh number as \( Ra_f = Ra Nu \) \[21\].

2.2 Consistency Checks

The Reynolds number can be computed from the velocity of the fluid. A characteristic speed \( U \) is determined as \( \langle u^2 + w^2 \rangle^{1/2} \) where the overbar denotes the time average and \( \langle \cdot \rangle \) the volume average. Our parameters are chosen such that \( Re = U \)

The amount of heat transport due to convection is represented by the Nusselt number
\[ Nu_V = \frac{1}{V} \int_V \left( -\frac{\partial T}{\partial z} + uT \right) dV, \quad (9) \]
where the subscript \( V \) indicates that it has been calculated using a volume integral over the domain. A Nusselt number of 1 indicates that all heat transport is due to conduction, whilst Nusselt > 1 indicates advection is present. A larger Nusselt number indicates more heat transport by advection.

In order to confirm the accuracy of our simulations, we carry out three internal consistency checks. We calculate the Nusselt number in three ways. First, integrated over the domain volume via Equation \(9\) Second, on the bottom plate via
\[ Nu_b = \frac{-\langle \partial T / \partial z \rangle|_{H=0.5}}{\alpha g \beta L}, \quad (10) \]
where \( \langle u \rangle_H = L x^{-1} \int_{x=0}^{x=L} a \ dz \) is a horizontal plane average. Third, on the top plate via
\[ Nu_t = \frac{-\langle \partial T / \partial z \rangle|_{H=0.5}}{\alpha g \beta L}, \quad (11) \]
Conservation of energy requires
\[ Nu = \frac{Nu_b = Nu_t = Nu_V}, \quad (12) \]
\[23\]. The standard test in the literature is for the Nusselt numbers calculated at different heights of the domain to be within 1\% of each other \[20,31,39\].

Thus, we calculate the maximum relative difference between the bulk Nusselt number and the Nusselt numbers at the top, bottom as well as the difference between the top and bottom Nusselt number
\[ Nu_{int} = \frac{\max \left( |Nu_b - Nu_V|, |Nu_b - Nu_t|, |Nu_V - Nu_t| \right) \int \frac{Nu_V - Nu_t}{Nu_V}}{13} \]
As a second consistency check, we verify that buoyancy generation is balanced with viscous dissipation. If we average over a sufficiently long time, the \( D\tau \) term of the momentum equation goes to zero. We then take the dot product of the momentum equation with \( \mathbf{u} \) and integrate to find the energy balance
\[ \langle \mathbf{u} \cdot \nabla^2 \mathbf{u} \rangle = -\frac{1}{P} \langle \mathbf{u} \cdot \mathbf{R} \rangle, \quad (14) \]
where the first term represents the viscous dissipation \( \epsilon_U \), and the second term represents the buoyancy production \( \mathbf{P} \). The standard test in the literature is for simulations to find these quantities within 1\% of each other \[23,31\]. We check this by calculating
\[ \frac{|P - \epsilon_U|}{P} \quad (15) \]
As a third test, we make sure that the boundary layers are resolved with a minimum number of nodes. The thermal boundary layer can be defined using the peak value of \( T_{rms} \), calculated as
\[ T_{rms}(z) = \sqrt{\frac{\left( T - \langle T \rangle_H \right)^2}{\langle T \rangle_H}} \quad (16) \]
as in King et al. \cite{24}. Figure 1 shows the relationship between $T_{rms}$ and the thermal boundary layers, and the relationship between the viscous boundaries and the mean horizontal velocity magnitude. The thickness of the thermal boundary layer $\delta_T$ is defined by the height at which the peak value of $T_{rms}$ occurs. The boundary layer scales with the Nusselt number as

$$\delta_T = \frac{1}{2} LN u^{-1},$$  \hspace{1cm} (17)

see Grossman and Lohse \cite{19}. The thermal boundary layer plays a significant role in the properties of Rayleigh-Bénard convection, and it is essential that they are fully resolved in any numerical simulation \cite{33}. Amati et al. \cite{2} showed that at least 4 grid points are required in the thermal boundary layer, while Verzicco and Camussi \cite{40} stated that 6 cells are needed. Stevens et al. \cite{39} say that up to 7 points could be the minimum number of points required. In this work, we specify that at least 6 points are in the boundary layer. The number of points in the thermal boundary layer will be denoted as $N_{BL}$.

Figure 2 shows example temperature fields for the cases we study, at a snapshot in time after the flow has equilibrated. It also shows the different temperature profiles found in these cases (bottom), and compares them to the linear conductive state. We can see that as $Ra$ increases, the profile becomes more uniform in the bulk, with a steeper temperature gradient in the boundary layers.

3 Implementation

3.1 Parareal Algorithm

The Parareal algorithm, first introduced in Lions et al. in 2001 \cite{27}, is briefly outlined here. For a more in depth explanation, see for example \cite{16}.

Parareal is a method used to speedup numerical solutions of initial value problems (IVPs) of the form

$$\frac{\partial U}{\partial t} = f(U(t), t), \quad U(0) = U_0, \quad 0 \leq t \leq t_{end}. \hspace{1cm} (18)$$

Parareal makes use of a coarse solver $G$ and a fine solver $F$. The time domain is split into $N$ time slices, where $N$ is the number of processors available for parallelisation in the time domain. The fine solver is the numerical method with properties designed to give the solution to the system to a required degree of accuracy. The coarse method is a cheaper method designed to give an answer quicker than the fine method, and with reduced accuracy. The Parareal method iterates over the fine and coarse solvers to improve the accuracy of the initial solution given by the coarse solver, until it is as accurate as the fine solver. This is done using the correction step

$$U_{n+1}^{k+1} = G(t_{n+1}, t_n, U_n^{k+1})$$

$$+ F(t_{n+1}, t_n, U_n^k) - G(t_{n+1}, t_n, U_n^k), \hspace{1cm} (19)$$

where $n$ denotes the current time slice, and $k$ denotes the Parareal iteration number. The coarse solver operates in serial, hence the need for a cheaper solution method, whilst the fine solver is able to operate in parallel, the key to reducing solution times.

3.2 Spatial discretization

We use a collocation-based pseudo-spectral method for the spatial discretisation, using Fourier bases with periodic boundaries for the horizontal ($x$) direction, and Chebyshev polynomial bases for the vertical ($z$) direction. The spatial resolution of a simulation is described by the number of collocation points in $x$ ($N_x$), and in $z$ ($N_z$). Simulations use the open source code Dedalus \cite{7}, with the parareal_dedalus \cite{12} module used to implement the Parareal algorithm in the Dedalus solver. Time stepping is done using Implicit-Explicit Runge-Kutta timestepping methods (Ascher et al. \cite{3}). Linear terms (diffusion, pressure and buoyancy forcing) are
3.3 Validation

The code was validated against the data in Johnston and Doering [21], see Figure 3. Both fixed flux and fixed temperature boundary conditions were simulated. We calculated a Rayleigh Nusselt scaling of $0.135 Ra^{0.286}$ from our fixed flux data, very close to the $0.138 Ra^{0.285}$ reported in [21]. The slightly higher Nusselt numbers that they found for fixed flux cases at low Rayleigh number were also replicated. Finally, we calculated the critical Rayleigh number by running multiple simulations near $Ra_c$ and checking the growth rate of the kinetic energy. We found that it was in agreement with Chandresakhar [10] to within 0.1%.

3.4 Determining Accuracy of Fine Solution

We set a tolerance level of less than 1% for $Nu_{int}$ defined in Equation 13 and $|P - \epsilon_U|/P$ defined in Equation 15. We also require a minimum of 6 points in the thermal...
boundary layers, that is $N_{BL} \geq 6$. At each $Ra$ we start with a low resolution $(N_x, N_z) = (16, 8)$ for $Ra = 10^5$ and $10^6$ and $(32, 16)$ for $Ra = 10^7$ and then double the resolution in both spatial directions until all three conditions are met.

For comparison, we also carry out spatial convergence tests, comparing results obtained from the low resolution simulations with those obtained from a high resolution simulation for each $Ra$ in the $L^2$ norm. These are not used to determine the spatial resolution, though. We calculate the relative difference in the final state temperature field by taking the $L^2$ norm with the high resolution (double resolution of shown values for each $Ra$) final state. The second test is for $Nu$, for which we calculate

$$Nu_{rel} = \frac{|Nu - Nu_{HR}|}{Nu_{HR}},$$

where HR denotes the high resolution simulation.

Table 1 shows the resolution required to meet various convergence tests. $L^2$, $Nu_{int}$, $Nu_{rel}$, and $|P - \epsilon_U|/P$ all have tolerance values of 1%. $Ra$ is the Rayleigh number, $N_{BL}$ denotes the resolution required for 6 points to be in the thermal boundary layer, $L^2$ denotes the defect of the end state temperature field to the high res simulation, $Nu_{int}$ shows max$(|Nu_V - Nu_b|, |Nu_V - Nu_a|, |Nu_b - Nu_a|)/Nu_V$, $Nu_{rel}$ is the Nusselt number compared with the high resolution simulation, and $|P - \epsilon_U|/P$ is the buoyancy/ dissipation internal consistency check.

3.5 Duration of Simulation

We determined the duration of a simulation based on a fixed number of advective times. There are three main timescales for Rayleigh-Bénard flow which can be found from dimensional arguments; the thermal diffusive timescale, thermal advective timescale, and the viscous timescale. Here we ignore the viscous timescale, as we set $Pr$ to 1. In the non-dimensionalisation we have chosen, the diffusive and advective timescales are linked by $\tau_{\text{adve}} = Re \times \tau_{\text{diff}}$. Following Mound et al., we run our simulations for a set number (in this case 100) of advective times, after the initial transient has balanced out. However, in the $Ra = 10^6$ case, we restrict the simulation to 1 diffusive time unit, since the solution is effectively steady state.

### Table 1: Table showing resolution required to meet various convergence tests

| $Ra$  | $N_{BL}$ | $L^2$ | $Nu_{int}$ | $Nu_{rel}$ | $|P - \epsilon_U|/P$ |
|------|----------|-------|------------|------------|-----------------|
| $10^5$ | $(64,32)$ | $(32,16)$ | $(32,16)$ | $(32,16)$ | $(32,16)$ |
| $10^6$ | $(128,64)$ | $(64,32)$ | $(64,32)$ | $(64,32)$ | $(32,16)$ |
| $10^7$ | $(128,64)$ | $(-, -)$ | $(64,32)$ | $(64,32)$ | $(64,32)$ |

3.6 Choice of Coarse Solver

There are several options for choosing a coarse solver for Parareal. These include a lower order timestep, a larger timestep, reduced spatial resolution, reduced physics, or a different method of solving the equations. In this work, we reduce the spatial resolution and reduce the timestep. In tests, we used different levels of spatial coarsening to find the optimal amount for speedup. We tested coarsening factors (CF) of 2, 4, and 8, where $(N_x, N_z)$ of the coarse solver is equal to 1/CF $(N_x, N_z)$ of the fine method. A coarsening factor of 2 did not lead to a speedup. Convergence was quick, but the runtime of the coarse solver was too close to the that of the fine solver. A coarsening factor of 4 worked better, allowing for quick convergence along with a significant difference in the cost of the fine/coarse solvers. A factor of 8 reduction showed slow convergence, and was not pursued further.

When choosing a coarse time step, we found situations where a Parareal simulation could be unstable even when a stable coarse solver was combined with a stable fine solver. This is likely due to the stability of Parareal itself, which has its own stability criterion, separate to the individual solvers. This leads to lower speedups as we had to use smaller coarse time steps, making the coarse solver more costly. We also investigated using lower order timesteppers for the coarse solver, along with the reduced resolution. However, as the stability region of Runge-Kutta tends to increase
3.7 Determining Convergence in Parareal

The most simple and widely used check for convergence in Parareal is to monitor the defect between two consecutive iterate [32, 4, 5]. This has the benefit of being easy to implement, and can be done whilst running the simulation. However, as discussed in Section 3.4, using the $L^2$ can lead to substantial over-resolution of the problem if one is interested only in the averaged dynamics. Therefore, the typical online Parareal convergence test is not suitable in this case. Since, at the moment, no termination criteria for averaged dynamics has been published, we perform a fixed number of Parareal iterations and assess convergence in post processing. While useful for benchmarking, this is obviously not a reasonable approach for production runs. Research into alternative and more application-oriented termination criteria for Parareal therefore seems to be an area where further studies are urgently needed.

4 Results

4.1 Kinetic energy in the Parareal solution

Figures 5a, 5b show the kinetic energy against time, for Rayleigh numbers $10^5$, $10^7$, for different numbers of Parareal iterations $k$. The number of time slices was kept constant at 10. For $Ra = 10^5$, an initial Parareal coarse run shows significant differences from the subsequent Parareal iterations. The overall kinetic energy is higher in the low resolution coarse solver, and varies over time periodically. This increased kinetic energy in the coarse solver is due to dissipation of the system being under resolved at the coarse resolution. The periodicity is not present in the fine solution, and the effect can be seen to reduce in the subsequent iterations. The kinetic energy quickly reduces to the correct level after the first iteration for each time slice. Subsequent iterations still have a small bump in kinetic energy at the correction time, but the overall level is in accordance with the fine solver. The $Ra = 10^7$ case shows problems with the Parareal convergence. The correction steps increase the error, which can be seen in the large jumps at the time slice boundaries. This is the first indication that Parareal has reached the limit of usability in this parameter space.
Fig. 5: Kinetic energy against time for different numbers of Parareal iterations $K$ for $Ra = 10^5$, $10^7$. The coarse solver has $\frac{1}{4}$ the number of modes in $x$ and $z$ as the fine solver, the coarse timestep is $\approx 2\times$ the fine timestep, and the simulation used 10 time slices. The coarse solver for $Ra = 10^5$ shows higher kinetic energy levels, along with periodic behaviour not present in the serial solution ($K = 11$). For $10^7$, large jumps in the solution for $k > 0$ are due to the Parareal correction step. The error at the jumps is growing, rather than shrinking, as the iteration number increases, showing the inability of Parareal to converge in this parameter regime.

4.2 Parareal convergence

Figure 6 shows how the calculated Nusselt number changes with increasing Parareal iterations. The Nusselt number found from the initial coarse solve is outside the accuracy requirement with an error of around 10% rather than 1%. In the case of $Ra = 10^5$, the Nusselt number converges to within the accuracy envelope after 1 iteration, but then in iterations 2-4 it falls back outside this region before converging again from iteration 5. For $Ra = 10^7$, the Nusselt number converges after a single iteration.

Figure 7 shows the comparison of the $L^2$ error with the Nusselt number error in Nusselt number for $Ra = 10^6$, $10^7$. In the smaller $Ra$ case, there is smooth convergence in both the $L^2$ error and in the Nusselt error, although the Nusselt convergence is slightly more erratic. In the $Ra = 10^7$ case, we see that the Nusselt number error falls just underneath the tolerance threshold after the first iteration. This is followed by a shallow decline in the error until the final iteration. The $L^2$ error behaves very differently, with a constant error of around 10% right up until the 9th iteration. We see here the mismatch in the error with respect to time averaged quantities with errors with respect to snapshots of the solution ($L^2$).

Figure 8 shows the internal consistency errors ($Nu_{int}$, $|P - \epsilon_U|/P$) for all three $Ra$ tested. In all three cases, the $|P - \epsilon_U|/P$ and $Nu_{int}$ converge to within the 1% tolerance after one iteration. However, the results for $Ra = 10^7$ show that $|P - \epsilon_U|/P$ then returns above the tolerance level, and does not fall reliably until 8 iterations have been completed.

We have also carried out numerical experiments for different numbers of time slices. Here, we would expect to see a trend where the number of iterations required to converge slowly increases with the number of time slices. In our results, we found that the number of iterations required did not behave like this for $Ra = 10^7$. The number of iterations required increased and decreased with no clear pattern up to 20 time slices. Beyond this the iteration count was always higher than 1, and gradually increased with the number of time slices.

4.3 Scaling and Performance

Figure 9 shows the scaling performance for simulations with $Ra = 10^5$, $10^6$, $10^7$. We see standard scaling behaviour for both $10^5$, and $10^6$, where speedup increases with processor count until the scaling limit is reached, and no further performance gains are possible. This is due to an increase in the number of Parareal iterations required at higher time slice count. We also see that
performance is better at $10^6$ than at $10^5$, likely because the bigger problem size due to higher resolutions improves scaling. However, the performance of Parareal at $Ra = 10^7$ is much more mixed. This is in part due to the errors being very close to the tolerance level for all iterations after $k = 1$, see Figure 7b. The error does not fall with increasing iterations in the way it does for $Ra = 10^5, 10^6$, rather, it hovers very close to the tolerance value. Convergence behaviour with number of time slices is unpredictable in this case. For some numbers of time slices, such as in Figure 7, the Nusselt error falls below tolerance after one iteration and remains there. In other cases, such as five or 16 time slices, see Figure 7, the error falls below the tolerance and then rises back again.

5 Conclusions

5.1 Parareal for Rayleigh-Bénard convection

We have shown that the Parareal algorithm allows for reliable speedup of simulations in a limited range of Rayleigh numbers at finite Prandtl number. The algorithm converges quickly with respect to averaged quantities like the Nusselt number and internal energy balance. Although slower, Parareal also converges with respect to the $L^2$ defect between subsequent iterations. Speedups of up to 2.4 are possible, with around 20 processors, with parallel efficiencies of around 0.2 for Rayleigh numbers as high as $10^6$. However, in all cases, speedups were limited to at most 20 processors. Beyond that, increases in the number of required iteration balanced out any gains from using more processors.

At $Ra = 10^7$, we find that convergence of Parareal degrades substantially. The errors in $Nu$ do not fall monotonically with increasing iteration number. For some simulations, the error falls below the tolerance level at a low number of iterations, only to increase in successive iterations. This erratic behaviour leads to irregular scaling performance at $10^7$; sometimes the simulation converges in one iteration, sometimes it takes two or three. These findings are in contrast to what Samuel [33] found for $Ra = 10^7$ with infinite Prandtl number where he observed a small number of iterations independent of the number of time slices being required for convergence and increasing speedup up to 40 processors. Clearly, performance of Parareal is very different in the finite versus infinite Prandtl number case.

Parareal is not expected to be useful for simulations of Rayleigh-Bénard convection at Rayleigh numbers above $10^7$ as we expect the performance to degrade further as the flow becomes more turbulent. This difference in performance is caused in part by the well known general degradation of Parareal with increasing Reynolds numbers [33]. It is also caused by the choice of convergence criteria. The correction step of Parareal depends on pointwise amplitude corrections at the boundary between time slices. In Rayleigh-Bénard convection
studies, the particular state of a given field at an instant in time is not of primary concern, therefore we relaxed the accuracy conditions of the fine solution, so that we did not enforce that the $L^2$ error be below a threshold value. In the cases of $10^5$ and $10^6$, the $L^2$ error is of roughly the same magnitude as the time- and space-averaged quantities ($\text{Nu}_{\text{int}}$, $|P - \epsilon_U|/P$), used to determine accuracy of the solution. In the $10^7$ simulations, we can find a good level of accuracy in the $\text{Nu}_{\text{int}}$ and $|P - \epsilon_U|/P$, whilst the $L^2$ error is still high in spatial convergence tests, (see Figure 5). As the Parareal algorithm effectively operates on the $L^2$ error, Parareal convergence is slow. Exploring the performance of other parallel-in-time methods like PFASST or MGRIT, and potentially a comparison with Parareal, would be an interesting direction for future research.
5.2 Convergence of Statistical Quantities in Parareal

For larger Rayleigh numbers, our tests show a significant disparity between the instantaneous $L^2$ error in a variable field such as temperature and the error in statistically calculated quantities such as the Nusselt number. In one example, Parareal reached a 1% error with respect to the Nusselt number while the $L^2$ error stalled for 7 iterations and only fell below 1% after iteration 8. In a case like Rayleigh-Bénard convection, statistical quantities like the Nusselt number are typically the most informative for understanding the behaviour of the physical system and what domain scientists are interested in. Therefore, we argue that this should be the criteria for determination of convergence, similar to what is used in time serial studies. However, for a reliable estimate of this kind of quantity, a time average is required across multiple time slices, in addition to a spatial average. Obtaining this kind of data during a simulation to monitor and terminate Parareal’s convergence with respect to statistical quantities is a problem that presents an interesting challenge, and would be a useful avenue for further investigation.

References

1. Ahlers, G., Grossmann, S., Lohse, D.: Heat transfer and large scale dynamics in turbulent rayleigh-bénard convection. Reviews of modern physics 81(2), 503 (2009)
Fig. 9: speedup vs number of timeslices/processors for $Ra = 10^5$, $10^6$, and $10^7$. We can see that performance appears best for $Ra = 10^6$. Peak speedup is around 2 ~ 2.4 for all $Ra$. For $10^7$, performance is predictable, with speedup increasing with number of cores until a scaling limit is reached. For $10^7$, the scaling behavior is erratic, due to the errors being very close to the tolerance limit. This leads to more iterations being required for convergence at some processor counts, causing the smaller speeds (black triangles).

2. Amati, G., Koal, K., Massaioli, F., Sreenivasan, K., Verzicco, R.: Turbulent thermal convection at high Rayleigh numbers for a Boussinesq fluid of constant Prandtl number. Physics of Fluids 17(12), 121701 (2005)
3. Ascher, U.M., Ruuth, S.J., Spiteri, R.J.: Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Applied Numerical Mathematics 25(2-3), 151–167 (1997)
4. Aubanel, E.: Scheduling of tasks in the Parareal algorithm. Parallel Computing 37(3), 172–182 (2011)
5. Berry, L.A., Elwasif, W., Reynolds-Barredo, J.M., Samaddar, D., Sanchez, R., Newman, D.E.: Event-based parareal: A data-flow based implementation of parareal. Journal of Computational Physics 231(17), 5945–5954 (2012)
6. Bodenschatz, E., Pesch, W., Ahlers, G.: Recent developments in Rayleigh–Bénard convection. Annual review of fluid mechanics 32(1), 709–778 (2000)
7. Burns, K.J., Vasil, G.M., Oishi, J.S., Lecoanet, D., Brown, B.P.: Dedalus: A Flexible Framework for Numerical Simulations with Spectral Methods. arXiv e-prints arXiv:1905.10388 (2019)
8. Busse, F.: Non-linear properties of thermal convection. Reports on Progress in Physics 41(12), 1929 (1978)
9. Cattaneo, F., Emontet, T., Weiss, N.: On the interaction between convection and magnetic fields. The Astrophysical Journal 588(2), 1183 (2003)
10. Chandrasekhar, S.: Hydrodynamic and hydromagnetic stability. International Series of Monographs on Physics, Oxford: Clarendon, 1961 (1961)
11. Cioni, S., Ciliberto, S., Sommeria, J.: Strongly turbulent Rayleigh–Bénard convection in mercury: comparison with results at moderate Prandtl number. Journal of Fluid Mechanics 335, 111–140 (1997)
12. Clarke, A.: parareal, dedalus (2019). DOI 10.5281/zenodo.3562481
13. Farhat, C., Chandesris, M.: Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. International Journal for Numerical Methods in Engineering 58(9), 1397–1434 (2003)
14. Friedhoff, S., Falgout, R.D., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: A Multigrid-in-Time Algorithm for Solving Evolution Equations in Parallel. In: Presented at: Sixteenth Copper Mountain Conference on Multi-grid Methods, Copper Mountain, CO, United States. Mar 17 - Mar 22, 2013 (2013). URL http://www.osti.gov/scitech/servlets/purl/1079108
15. Gander, M.J.: 50 years of time parallel time integration. Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg, May 6-8, 2013 9, 69 (2015)
16. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM Journal on Scientific Computing 29(2), 556–578 (2007)
17. Getting, A.V.: Rayleigh–Bénard Convection: Structures and Dynamics, vol. 11. World Scientific (1998)
18. Glazmaier, G.A., Roberts, P.H.: A three-dimensional convective dynamo solution with rotating and finitely conducting inner core and mantle. Physics of the Earth and Planetary Interiors 91(1-3), 63–75 (1995)
19. Grossmann, S., Lohse, D.: Scaling in thermal convection: a unifying theory. Journal of Fluid Mechanics 407, 27–56 (2000)
20. Hunter, J.D.: Matplotlib: A 2d graphics environment. Computing in science & engineering 9(3), 90–95 (2007)
21. Johnston, H., Doering, C.R.: Comparison of turbulent thermal convection between conditions of constant temperature and constant flux. Physical review letters 102(6), 064501 (2009)
22. Kerr, R.M.: Rayleigh number scaling in numerical convection. Journal of Fluid Mechanics 310, 139–179 (1996)
23. King, E., Stellmach, S., Aurnou, J.: Heat transfer by rapidly rotating Rayleigh–Bénard convection. Journal of Fluid Mechanics 691, 568–582 (2012)
24. King, E., Stellmach, S., Buffet, B.: Scaling behaviour in Rayleigh–Bénard convection with and without rotation. Journal of Fluid Mechanics 717, 449–471 (2013)
25. Kooij, G.L.: Towards parallel-in-time simulations of turbulent Rayleigh–Bénard convection. University of Twente (2017)
26. Kooij, G.L., Botchev, M.A., Frederix, E.M., Geurts, B.J., Horn, S., Lohse, D., van der Poel, E.P., Shishkina, O., Stevens, B.J., Verzicco, R.: Comparison of computational codes for direct numerical simulations of turbulent Rayleigh–Bénard convection. Computers & Fluids 166, 1–8 (2018)
27. Lions, J.L., Maday, Y., Turinici, G.: Résolution d’edp par un schéma en temps parallélé. Comptes Rendus de l’Académie des Sciences-Sciences I-Mathematics 332(7), 661–668 (2001)
28. Matsui, H., Heien, E., Aubert, J., Aurnou, J.M., Avery, M., Brown, B., Buffet, B.A., Busse, F., Christensen, U.R., Davies, C.J., et al.: Performance benchmarks for a next generation numerical dynamo model. Geochemistry, Geophysics, Geosystems 17(5), 1586–1607 (2016)
29. McKenzie, D.P., Roberts, J.M., Weiss, N.O.: Convection in the earth’s mantle: towards a numerical simulation. Journal of Fluid Mechanics 62(3), 465–538 (1974)
30. Minion, M.L.: A Hybrid Parareal Spectral Deferred Corrections Method. Communications in Applied Mathematics and Computational Science 5(2), 265–301 (2010). URL http://dx.doi.org/10.2140/camcos.2010.5.265
31. Mound, J.E., Davies, C.J.: Heat transfer in rapidly rotating convection with heterogeneous thermal boundary conditions. Journal of Fluid Mechanics 828, 601629 (2017). DOI 10.1017/jfm.2017.539
32. Samaddar, D., Coster, D., Bonnin, X., Bergmeister, C., Havlíčková, E., Berry, L.A., Elwasif, W.R., Batchelor, D.B.: Temporal parallelization of edge plasma simulations using the parareal algorithm and the SOLPS code. Computer Physics Communications 221, 19–27 (2017)
33. Samuel, H.: Time domain parallelization for computational geodynamics. Geochemistry, Geophysics, Geosystems 13(1) (2012)
34. Schumacher, J., Bandaru, V., Pandey, A., Scheel, J.D.: Transitional boundary layers in low-Prandtl-number convection. Physical Review Fluids 1(8), 084402 (2016)
35. Shishkina, O., Stevens, R.J., Grossmann, S., Lohse, D.: Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution. New journal of Physics 12(7), 075022 (2010)
36. Siggia, E.D.: High Rayleigh number convection. Annual review of fluid mechanics 26(1), 137–168 (1994)
37. Staff, G.A., Rnquist, E.M.: Stability of the parareal algorithm. In: R. Kornhuber, et al. (eds.) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 40, pp. 449–456. Springer, Berlin (2005). DOI 10.1007/3-540-26825-1_46. URL http://dx.doi.org/10.1007/3-540-26825-1_46
38. Steiner, J., Ruprecht, D., Speck, R., Krause, R.: Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number. In: Numerical Mathematics and Advanced Applications-ENUMATH 2013, pp. 195–202. Springer (2015)
39. Stevens, R.J., Verzicco, R., Lohse, D.: Radial boundary layer structure and Nusselt number in Rayleigh–Bénard convection. Journal of fluid mechanics 643, 495–507 (2010)
40. Verzicco, R., Camussi, R.: Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell. Journal of Fluid Mechanics 477, 19–49 (2003)
41. Zhu, X., Mathai, V., Stevens, R.J., Verzicco, R., Lohse, D.: Transition to the ultimate regime in two-dimensional Rayleigh–Bénard convection. Physical review letters 120(14), 144502 (2018)