Computing and Visualization in Science (2020) 23:10
https://doi.org/10.1007/s00791-020-00332-3

PINT 2019

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

Andrew Clarke[1] · Chris Davies[2] · Daniel Ruprecht[3] · Steven Tobias[4] · Jeffrey S. Oishi[5]

Received: 6 December 2019 / Accepted: 21 May 2020

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 ΔT between two horizontal plates separated by a distance L. The problem can be characterised by

Communicated by Robert Speck.

A.C. was supported by Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Fluid Dynamics (EP/L01615X/1). C.J.D is supported by a Natural Environment Research Council (NERC) Independent Research Fellowship (NE/L011328/1). D.R. thankfully acknowledges partial support by NERC grant NE/R008795/1 “Parallel Paradigms for Numerical Weather Prediction”. S.M.T. was supported by funding from the European Research Council (ERC) under the EU’s Horizon 2020 research and innovation programme (grant agreement D5S-DLV-786780). J.S.O. and A.C. were supported by NASA LWS grant no. NNX16AC92G. This work used the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk) as well as ARC2 and ARC3, part of the High Performance Computing facilities at the University of Leeds. Figures were produced using Matplotlib [20]. We thank the two anonymous reviewers for their useful comments. In particular, the second reviewer proposed some interesting ideas about the convergence of Parareal in terms of hyperbolic PDE’s with time averaged quantities.

✉ Andrew Clarke
scatc@leeds.ac.uk

Chris Davies
c.davies@leeds.ac.uk

Daniel Ruprecht
ruprecht@tuhh.de

Steven Tobias
s.m.tobias@leeds.ac.uk

Jeffrey S. Oishi
joishi@bates.edu

[1] Centre for Doctoral Training in Fluid Dynamics, University of Leeds, Leeds, UK

[2] School of Earth and Environment, University of Leeds, Leeds, UK

[3] Lehrstuhl Computational Mathematics, Institut für Mathematik, Technische Universität Hamburg, Hamburg, Germany

[4] Department of Applied Mathematics, University of Leeds, Leeds, UK

[5] Department of Physics and Astronomy, Bates College, Lewiston, ME, USA

Published online: 23 September 2020

three non-dimensional parameters. The Rayleigh number is given by

$$Ra = \frac{\alpha g \Delta T L^3}{\nu\kappa}, \tag{1}$$

where α is the coefficient of thermal expansion, ν is the kinematic viscosity of the fluid, g is gravity, and κ is the thermal diffusivity. The Prandtl number is

$$Pr = \frac{\nu}{\kappa}. \tag{2}$$

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 [35], while a Prandtl number ~ 1 is commonly used in simulations of the Earth's core [29,32]. 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 [38] or Verzicco and Camussi [42]. Some notable studies utilising Rayleigh–Bénard convection include Cattaneo et al. [9] who studied solar magnetic field interactions, Glatzmaier and Roberts [18], who produced the first simulation of a geomagnetic field reversal, and McKenzie et al. [30] who studied the effect of mantle flow in the earth. For more in depth reviews of the subject, see for example Bodenschatz et al. [6] or Ahlers et al. [1].

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, defined below), with Ra, see for example Grossmann and Lohse [19], Cioni et al.[11], Kerr [22], and Siggia [38].

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. [43] and Schumacher [36]. 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 [31]), 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 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

$$\frac{1}{Pr}\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right)=-\nabla p+RaT\cdot\hat{\boldsymbol{z}}+\nabla^2\boldsymbol{u}, \tag{3}$$

$$\nabla\cdot\boldsymbol{u}=0, \tag{4}$$

$$\frac{\partial T}{\partial t}+\boldsymbol{u}\cdot\nabla T=\nabla^2 T, \tag{5}$$

with fixed temperature

$$\begin{aligned}&T|_{z=-0.5}=1,\quad T|_{z=0.5}=0,\\&\boldsymbol{u}|_{z=-0.5}=\boldsymbol{u}|_{z=0.5}=0\end{aligned} \tag{6}$$

and fixed flux

$$\begin{aligned}&\left.\frac{\partial T}{\partial z}\right|_{z=-0.5}=\left.\frac{\partial T}{\partial z}\right|_{0.5}=-1,\\&\quad\boldsymbol{u}|_{z=-0.5}=\boldsymbol{u}|_{z=0.5}=0\end{aligned} \tag{7}$$

boundary conditions. For all work in this study, periodic horizontal boundary conditions are used. Here, $\boldsymbol{u}=(u,w)$ 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 $\tau_d \sim L^2/\kappa$, T is scaled by ΔT, 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 $\boldsymbol{u}=0$.

In the fixed flux case, we use the flux Rayleigh number Ra_f, defined as

$$Ra_f=\frac{\alpha g\beta L^4}{\nu\kappa}, \tag{8}$$

where β 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 $Ra_f=RaNu$ [21].

2.2 Consistency checks

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

The heat transported due to convection is represented by the Nusselt number

$$Nu_V=\frac{1}{V}\int_V\left(-\frac{\partial T}{\partial z}+wT\right)\mathrm{d}V, \tag{9}$$

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 Eq. 9. Second, on the bottom plate via

$$Nu_b=\left.\left\langle-\frac{\partial T}{\partial z}\right\rangle_H\right|_{z=-0.5}, \tag{10}$$

where $\langle a\rangle_H=L_x^{-1}\int_{x=0}^{x=L_x}a\,\mathrm{d}x$ is a horizontal plane average. Third, on the top plate via

$$Nu_t=\left.\left\langle-\frac{\partial T}{\partial z}\right\rangle_H\right|_{z=0.5}. \tag{11}$$

Conservation of energy requires

$$Nu=\overline{Nu_b}=\overline{Nu_t}=\overline{Nu_V}, \tag{12}$$

[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 [23,32,41]. In this work, the reported values have been calculated from Eq. 9.

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

$$Nu_{\text{int}}=\frac{\max\left(|\overline{Nu_b}-\overline{Nu_V}|,|\overline{Nu_b}-\overline{Nu_t}|,|\overline{Nu_V}-\overline{Nu_t}|\right)}{\overline{Nu_V}}. \tag{13}$$

As a second consistency check, we verify that buoyancy generation is balanced with viscous dissipation. If we average over a sufficiently long time, the $\frac{D\boldsymbol{u}}{Dt}$ term of the momentum equation goes to zero. We then take the dot product of the momentum equation with $\boldsymbol{u}$ and integrate to find the energy balance

$$|\overline{\boldsymbol{u}\cdot\nabla^2\boldsymbol{u}}|=|\overline{\boldsymbol{u}\cdot RaT\hat{\boldsymbol{z}}}|, \tag{14}$$

where the first term represents the viscous dissipation ϵ_U, and the second term represents the buoyancy production P, not to be confused with p for pressure. The standard test in the literature is for simulations to find these quantities within 1% of each other [23,32]. We check this by calculating

$$\frac{|P-\epsilon_U|}{P}. \tag{15}$$

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

$$T_{\mathrm{rms}}(z) = \overline{\left\langle \sqrt{\left(T - \overline{\langle T \rangle_H}\right)^2} \right\rangle_H} \tag{16}$$

as in King et al., [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 δ_T is defined by the height at which the peak value of T_{rms} occurs. The boundary layer scales with the Nusselt number as

$$\delta_t = \frac{1}{2} L Nu^{-1}, \tag{17}$$

see Grossman and Lohse [19]. The thermal boundary layers play a significant role in the behaviour of Rayleigh–Bénard convection, and it is essential that they are fully resolved in any numerical simulation [37]. Amati et al. [2] showed that at least 4 grid points are required in the thermal boundary layer, while Verzicco and Camussi [42] stated that 6 points are needed. Stevens et al. [41] 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 equilibriated. 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.

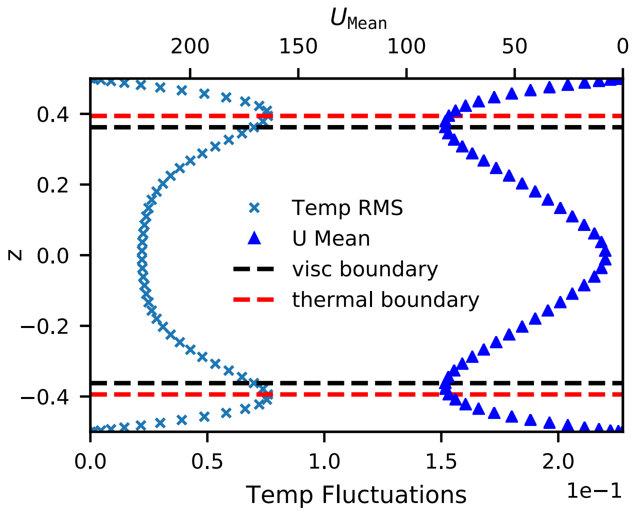


Fig. 1 Rayleigh Bénard flow at Rayleigh number = 10^5. Temperature fluctuations (left side of graph, bottom scale) denote the T_{rms} of the temperature field (defined in text), U_{Mean} (right side of graph, top scale) denotes the magnitude of the horizontal component of the velocity. The thermal boundary layer is defined by the height at which the peak T_{rms} is found, and the viscous boundary layer is defined by the height at which the peak U_{mean} is found [24]

3 Implementation

3.1 Parareal algorithm

The Parareal algorithm, first introduced in Lions et al. in 2001 [27], is briefly outlined here. A more in depth explanation is provided by Gander and Vandewalle [16].

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

$$\frac{\partial U(t)}{\partial t} = f\left(U(t), t\right), \quad U(0) = U_0, \quad 0 \le t \le t_{\mathrm{end}}. \tag{18}$$

Parareal makes use of a coarse solver $\mathcal{G}$ and a fine solver $\mathcal{F}$. 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

$$\begin{aligned} U_{n+1}^{k+1} &= \mathcal{G}(t_{n+1}, t_n, U_n^{k+1}) \\ &\quad + \mathcal{F}(t_{n+1}, t_n, U_n^k) - \mathcal{G}(t_{n+1}, t_n, U_n^k), \end{aligned} \tag{19}$$

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 [7], with the parareal_dedalus [12] module used to implement the Parareal algorithm in the

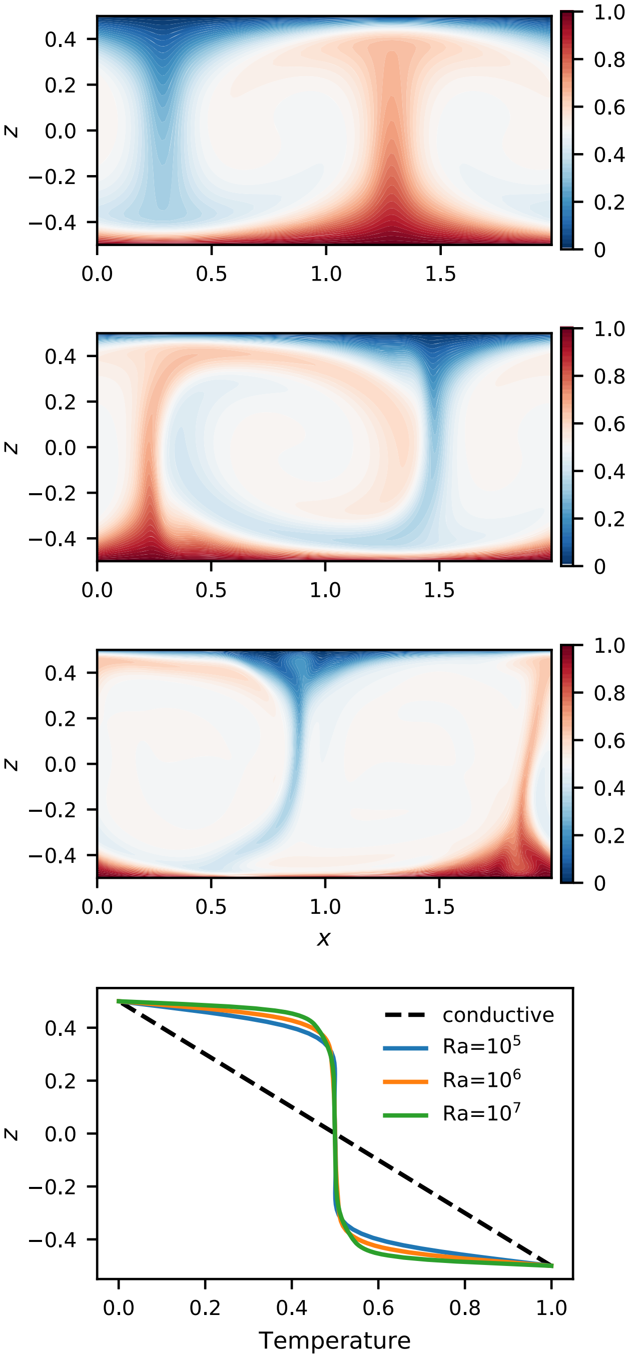


Fig. 2 Temperature field for flows with $Ra = 10^5$ (top), 10^6 (middle top), and 10^7 (middle lower) taken after a statistically steady state has been reached. The bottom plate is fixed at $T = 1$, whilst the top plate is fixed at $T = 0$, and both top and bottom plates are no-slip. There is steady flow for $Ra = 10^5$, with more unsteady and smaller plumes at 10^6, and even more so at 10^7. At $Ra = 10^7$, there is a small amount of entrainment of fluid into the base of the plumes. The bottom figure shows temperature profiles for all three cases, compared to the purely conductive case. Boundary layers get thinner as Ra increases

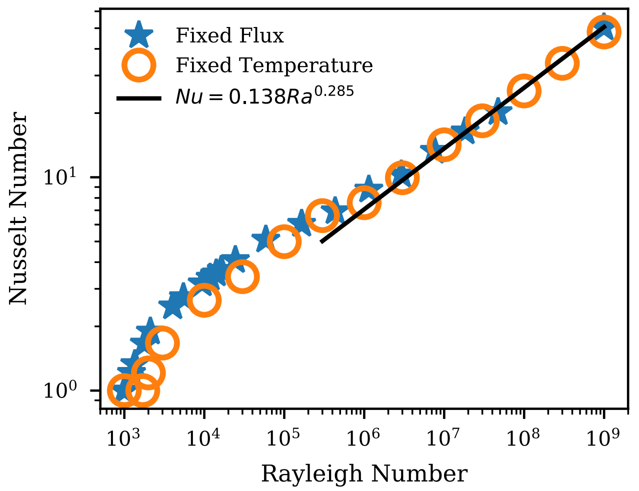


Fig. 3 Calculated Nusselt values ($\overline{Nu}_V$) compared with the scaling found in Johnston and Doering [21]. Scaling of $0.135Ra^{0.286}$ was calculated from our data, compared to $0.138Ra^{0.285}$ found in [21]. Fixed temperature and fixed flux boundary simulations collapse on to the same line at high Rayleigh number, in agreement with Johnston et al. [21] (black line)

Dedalus solver. Time stepping is done using Implicit-Explicit Runge–Kutta timestepping methods by Ascher et al. [3]). Linear terms (diffusion, pressure and buoyancy forcing) are treated implicitly, whilst non-linear terms are treated explicitly. This combination lends itself to the pseudo-spectral method, as transformations between spectral and grid space are carried out using the parallel FFTW package, allowing multiplications to take place in grid space.

3.3 Validation

The code was validated against the data in Johnston and Doering [21], see Fig. 3. Both fixed flux and fixed temperature boundary conditions were simulated. We calculated a Rayleigh Nusselt scaling of $Nu = 0.135Ra^{0.286}$ from our fixed flux data, very close to the $Nu = 0.138Ra^{0.285}$ reported in [21]. The slightly higher Nusselt numbers obtained in [21] 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 [10], 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 Eq. 13 and $|P - \epsilon_U|/P$ defined in Eq. 15. We also require a minimum of 6 points in the thermal boundary layers, that is $N_{\text{BL}} \geq 6$. 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 for the L^2 norm of the temperature field, comparing results obtained from the low resolution simulations with those obtained from a high resolution simulation for each Ra. These are not used to determine the spatial resolution. 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

$$Nu_{\text{rel}} = \frac{|Nu - Nu_{\text{HR}}|}{Nu_{\text{HR}}}, \tag{20}$$

where HR denotes the high resolution simulation.

Table 1 shows the resolution required to meet the consistency checks discussed above. We can see that the resolution required for 6 points in the boundary layer is higher than the resolution required for the other convergence tests, except for the L^2 error for $Ra = 10^7$. Figure 4 shows how the L^2 error compares with Nu_{int}. At $Ra = 10^5$, the resolution for a 1% L^2 error is the same as the resolution required for the 1% tolerance in the Nusselt numbers and buoyancy production and only half the resolution needed to have at least six nodes in the boundary layers.

At $Ra = 10^7$, the L^2 error is not yet below 1% even when all other tests are below tolerance, showing a significant difference in the L^2 error and the convergence tests we have set.

Given that the L^2 norm is not a very relevant quantity for understanding flow dynamics, if the internal checks and key quantities are converged before the L^2 error, then the lower resolution is deemed sufficient. The effect of timestep size on the accuracy of the solution was also investigated. However, it was found that for a given spatial resolution, the largest stable timestep was found to meet all of the accuracy criteria.

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 $\tau_{\text{advective}} = Re \times \tau_{\text{diffusive}}$. Following Mound et al. [32], 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 $Ra = 10^5$ case, we restrict the simulation to 1 diffusive time unit, since the solution is effectively steady state.

Table 1 Resolution required to meet various convergence tests. L^2 of the temperature field, 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 $\max(|Nu_V - Nu_b|, |Nu_V - Nu_t|, |Nu_b - Nu_t|)/Nu_V$, Nu_{rel} is the Nusselt number compared with the high resolution simulation, and $|P - \epsilon_U|/P$ is the buoyancy/ dissipation internal consistency check

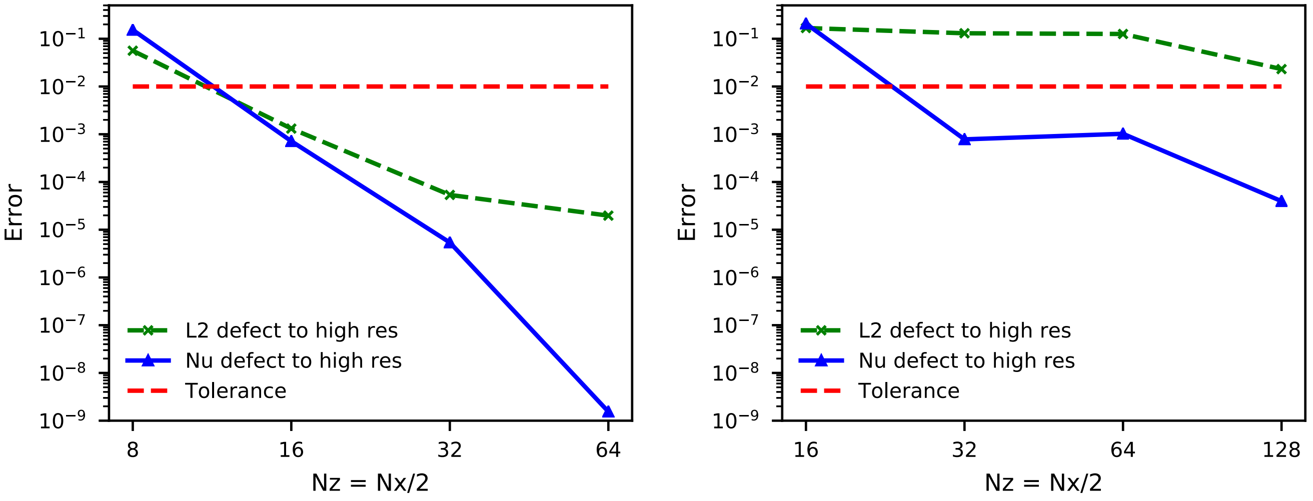
Ra		Resolution (N_x, N_z) for error $\leq$ 1%			
	$N_{\text{BL}} \geq 6$	L^2	Nu_{int}	Nu_{rel}	$\|P - \epsilon_U\|/P$
10^5	(64,32)	(32,16)	(32,16)	(32,16)	(32,16)
10^6	(128,64)	(64,32)	(64,32)	(64,32)	(32,16)
10^7	(128,64)	(-,-)	(64,32)	(64,32)	(64,32)

3.6 Choice of coarse solver

There are several options for choosing a coarse solver for Parareal. These include a lower order timestepper, 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. We tested 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.

Coarsening in space requires a method to transmit information from coarse grid to fine grid (interpolation), and back again (restriction). The order of operator for interpolation has been found to be important for the convergence of Parareal [28]; a high order method of interpolation helps the convergence of Parareal. In this work we use spectral interpolation, both because of its convergence properties, and because the use of spectral methods for spatial discretisation make it a natural choice.

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 [39]. This leads to lower speedups as we had to use



(a) $Ra = 10^5$

(b) $Ra = 10^7$

Fig. 4 Spatial convergence of Nusselt number $\overline{Nu}_V$ and L^2 errors relative to high resolution solution for $Ra = 10^5$ (left) and $Ra = 10^7$ (right). As expected, higher resolution is required for both quantities to meet the 10^{-2} tolerance for the higher Rayleigh number case. It can also be seen that the L^2 error requires much more resolution at higher Rayleigh number than the Nusselt number, where as at $Ra = 10^5$, the resolution required to give good answers for the Nusselt number and L^2 error are similar. The shown Nusselt number is calculated by averaging over time and space

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 with the order, we found that reduced timestep sizes were required for lower order coarse solvers. This cancelled out any speed increase from reduced computation, thus the higher order timestepper RK443 was used in both the fine and coarse solver. Table 2 shows the resolutions, timesteps and runtimes of the coarse and fine solvers used in this work.

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 [4,5,34]. This has the benefit of being easy to implement, and can be done whilst running the simulation. However, as discussed in Sect. 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 were further studies are urgently needed.

4 Results

4.1 Kinetic energy in the Parareal solution

Figure 5a, b 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 kinetic energy corrects quickly to the correct level (within tolerance of the fine method) at the start of each time slice, so that the time averaged value falls within tolerance values. The magnitude of the jump is also small, and does not grow significantly beyond the difference between the coarse and fine solvers. 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. These jumps

Table 2 Spatial resolution (N_x, N_z), timestep size (in diffusion times τ_d), time-serial runtimes (seconds), and simulation duration (in τ_d) for the coarse and fine solvers at different Rayleigh number (Ra)

Ra	Resolution		Timestep		Runtime		Duration
	Coarse	Fine	Coarse	Fine	Coarse	Fine	(τ_d)
10^5	(16,8)	(64,32)	10^{-4}	2×10^{-5}	95.4	300	1.0
10^6	(32,16)	(128,64)	5×10^{-5}	2×10^{-5}	1062	2996	0.6
10^7	(32,16)	(128,64)	5×10^{-6}	2×10^{-6}	5185	14,169	0.3

are of far larger magnitude than those found in the lower Ra case, which is a further reason to suspect that the method is failing for $Ra = 10^7$, whilst accepting that it is working for $Ra = 10^5$.

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. We believe this is due to the well known 'hump' that can be seen for problems with dominant imaginary eigenvalues where the error does not contract monotonically [16]. For $Ra = 10^7$, the Nusselt number converges after a single iteration in this case of 10 time slices. For different numbers of time slices, the Nusselt number sometimes takes more than one iteration to converge—see Figs. 7, 9.

Figure 7 shows the comparison of the L^2 error with the 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, from 5 to 32 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.

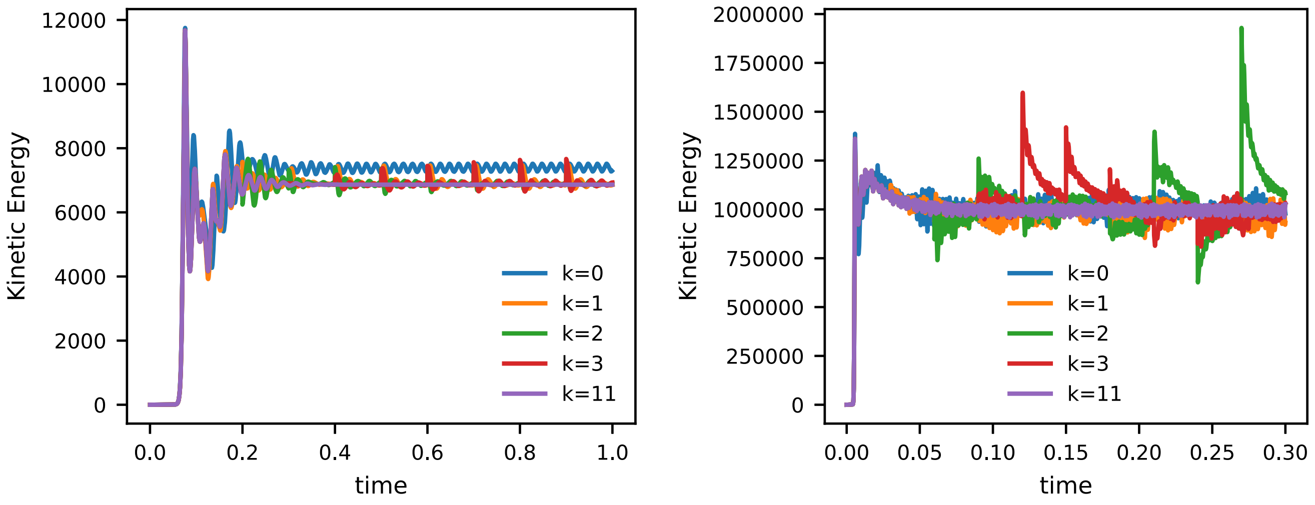
4.3 Scaling and performance

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 7b, the Nusselt error falls below tolerance after one iteration and remains there. In other cases, such as five or 16 time slices, see Figure 7c, the error falls below the tolerance and then rises back again.

5 Conclusions

5.1 Parareal for Rayleigh–Bénard convection

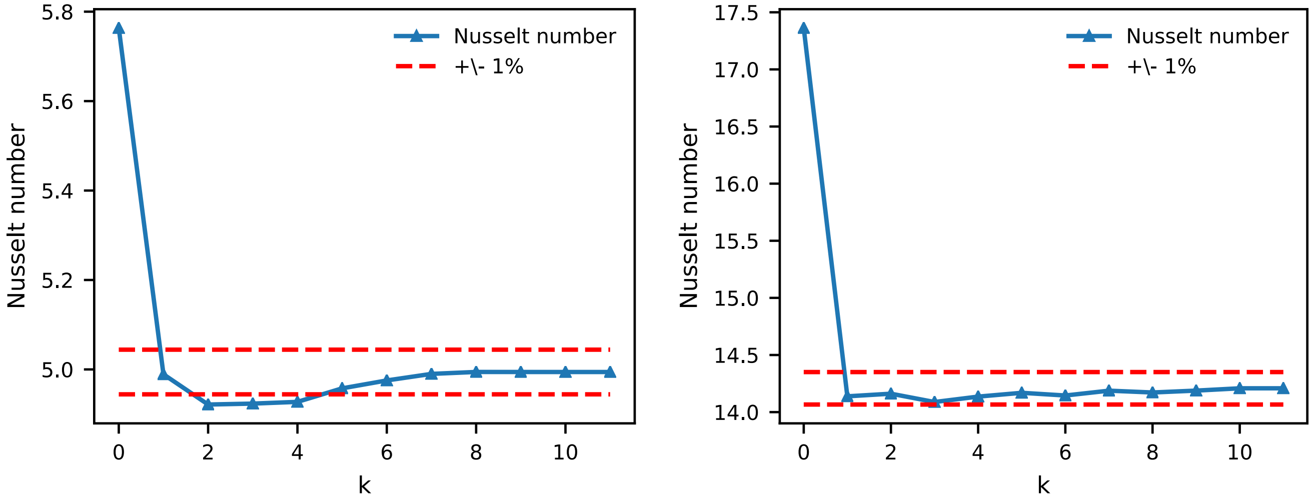
We have shown that the Parareal algorithm allows for reliable speedup of simulations in a limited range of Rayleigh numbers at finite Prandtl number. The algorithm converges quickly with respect to averaged quantities like the Nusselt number and internal energy balance. Although slower, Parareal also 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,



(a) $Ra = 10^5$

(b) $Ra = 10^7$

Fig. 5 Dimensionless kinetic energy against time for different numbers of Parareal iterations K for $Ra = 10^5$, 10^7. Time is measured in terms of the diffusion time τ_d, duration was determined as ≈ 100 advective time units after the transient settled. 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 (see table 2). The coarse solver for $Ra = 10^5$ shows higher kinetic energy levels, along with periodic behaviour not present in the fine solution, which is proven to be found when $k >$ number of processors ($k = 11$ in this case). 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



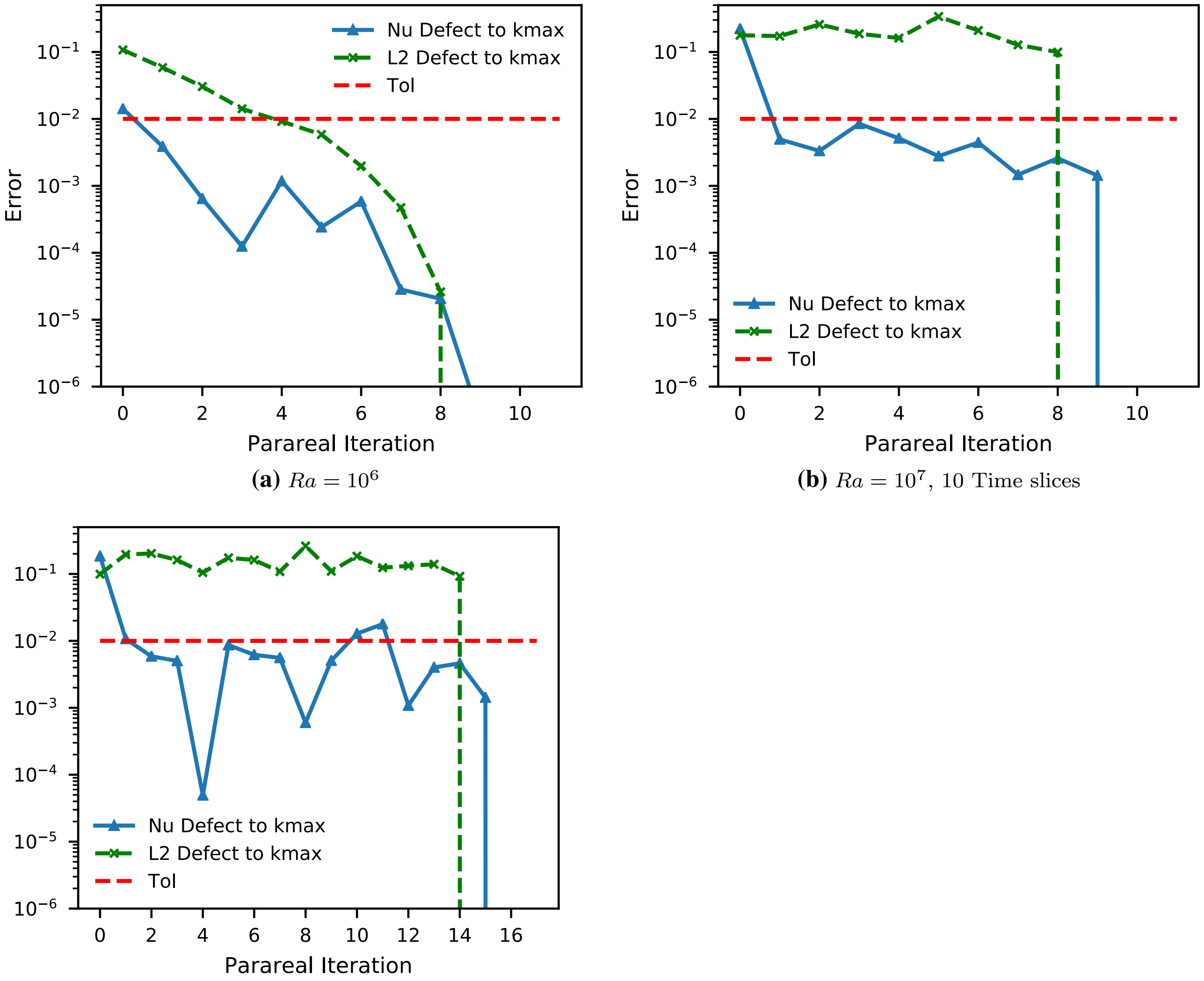
(a) $Ra = 10^5$

(b) $Ra = 10^7$

Fig. 6 Changing Nusselt number $\overline{Nu}_V$ with Parareal iteration k. There is a large error in the Nusselt number calculated from the coarse solver ($k = 0$), so that at least one iteration is required to calculate the correct Nusselt number (within 1%—dotted red lines). For the Nusselt number alone, convergence behaviour is encouraging, for $Ra = 10^5$ and $Ra = 10^7$. The simulation was carried out with 10 time slices

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. 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, in



(a) $Ra = 10^6$

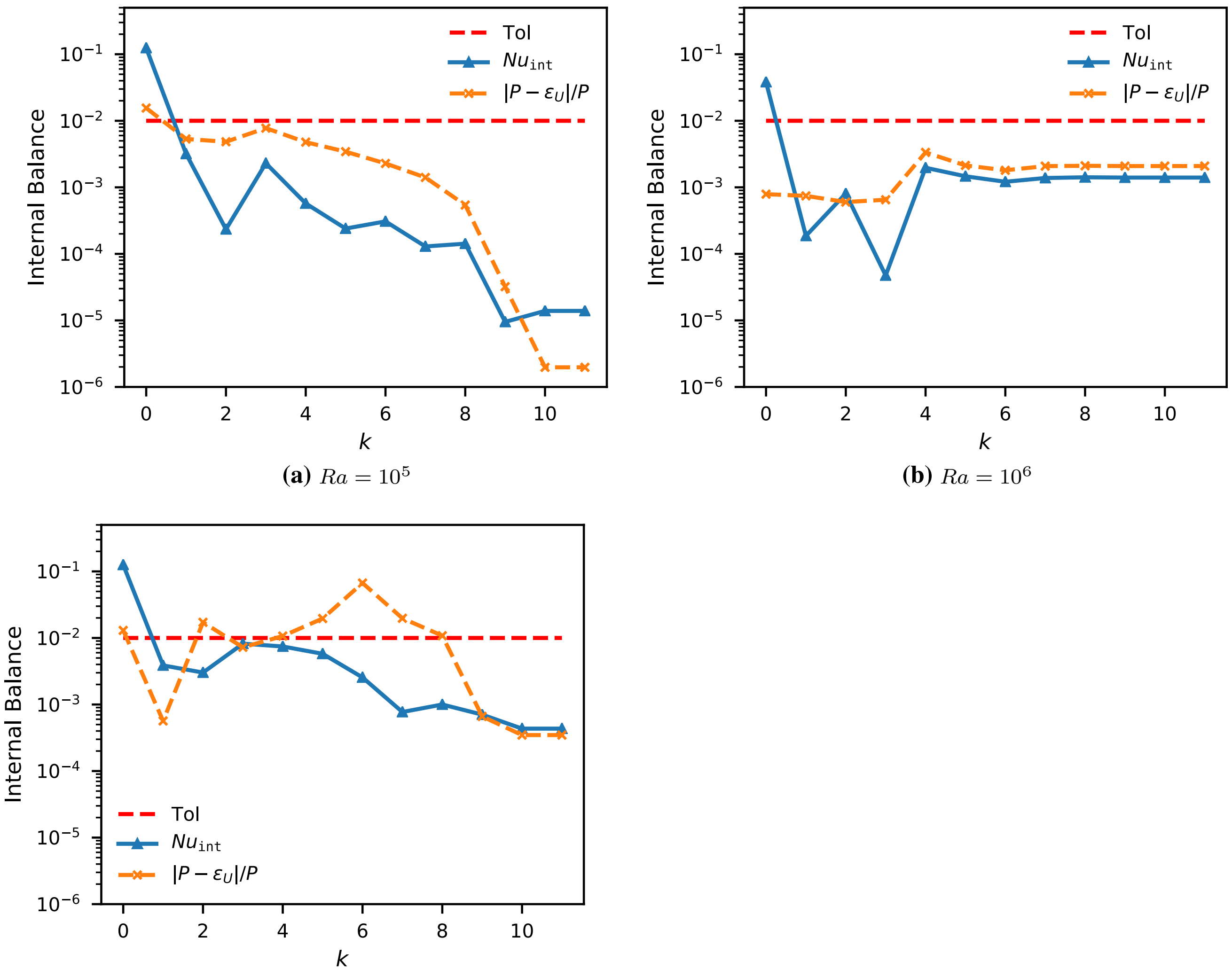
(b) $Ra = 10^7$, 10 Time slices

(c) $Ra = 10^7$, 16 Time slices

Fig. 7 Convergence of Nusselt number $\overline{Nu}_V$ and L^2 error with Parareal iteration for $Ra = 10^6$, 10^7, 10 time slices (a,b), 16 time slices(c). As kmax is greater than number of timeslices, the solution at kmax perfectly represents the serial fine solution. We can see that the L^2 error at $Ra = 10^6$ behaves as expected for good Parareal convergence, with a superlinear convergence behaviour. The Nusselt error at this Ra also shows convergence, but is more erratic. At $Ra = 10^7$, we see much worse convergence. The L^2 error does not converge until the last iteration, when k is equal to the number of time slices. The Nusselt number error behaves slightly better, but does not decrease monotonically. Figure(c) shows $Ra = 10^7$ but with 16 time slices. Here, it requires two iterations for the Nusselt number to reach the 1% tolerance

line with previous results [40]. These findings are in contrast to what Samuel [35] 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.

This difference in performance is caused in part by the well known general degradation of Parareal with increasing Reynolds numbers [40]. 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 (Nu_{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 Nu_{int} and $|P - \epsilon_U|/P$, whilst the L^2 error is still high in spatial convergence tests, (see Figure 4). As the Parareal algorithm effectively operates on the L^2 error, Parareal convergence



(a) $Ra = 10^5$

(b) $Ra = 10^6$

(c) $Ra = 10^7$

Fig. 8 Convergence of the internal checks carried out on the data of the Parareal simulations, for $Ra = 10^5$, 10^6, 10^7, 10 time slices. The internal energy balance ($|P - \epsilon_U|/P$) takes longer to converge than Nu_{int}. The Nusselt number is convergent for all three cases, but the internal energy balance is not convergent at the highest Rayleigh number

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 in 1 iteration 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.

The cause for the poor performance of Parareal for hyperbolic problems is that Parareal mostly performs an amplitude correction rather than a phase correction—if the wave speed

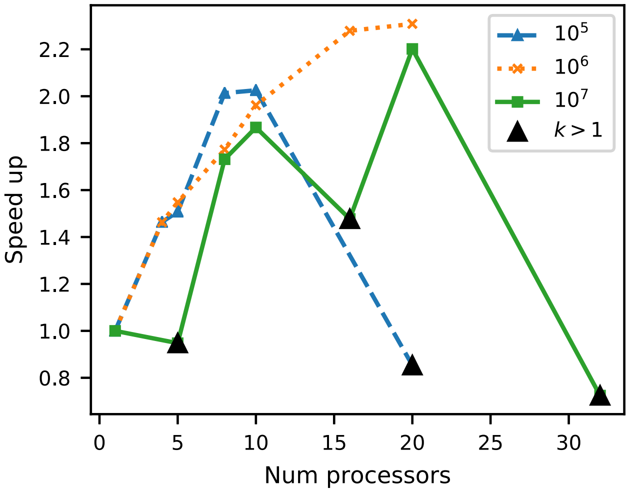


Fig. 9 speedup vs number of timeslices/processors for $Ra = 10^5, 10^6$, and 10^7. We can see that performance apeears best for $Ra = 10^6$. Peak speedup is around $2 \sim 2.4$ for all Ra. For 10^5, and 10^6, performance is predictable, with speedup increasing with number of cores until a scaling limit is reached. For 10^7, the scaling behaviour 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 speedups (black triangles)

is incorrect in the coarse propagator, Parareal will be unable to correct for the phase error this causes [33]. In a time average, the phase information might be much less important. Therefore, if a convergence criteria/correction for Parareal based on time averages could be constructed, this might also help to alleviate the problems Parareal faces for hyperbolic problems.

References

1. Ahlers, G., Grossmann, S., Lohse, D.: Heat transfer and large scale dynamics in turbulent Rayleigh–Bénard convection. Rev. Mod. Phys. **81**(2), 503 (2009)
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. Phys. 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. Appl. Numer. Math. **25**(2–3), 151–167 (1997)
4. Aubanel, E.: Scheduling of tasks in the Parareal algorithm. Parallel Comput. **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. J. Comput. Phys. **231**(17), 5945–5954 (2012)
6. Bodenschatz, E., Pesch, W., Ahlers, G.: Recent developments in Rayleigh–Bénard convection. Annu. Rev. Fluid Mech. **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. Phys. Rev. Res. **2**, 023068 (2020)
8. Busse, F.: Non-linear properties of thermal convection. Rep. Progress Phys. **41**(12), 1929 (1978)
9. Cattaneo, F., Emonet, T., Weiss, N.: On the interaction between convection and magnetic fields. Astrophys. J. **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. J. Fluid Mech. **335**, 111–140 (1997)
12. Clarke, A.: parareal_dedalus (2019). https://doi.org/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. Int. J. Numer. Methods Eng. **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 Multigrid Methods, Copper Mountain, CO, United States, Mar 17–Mar 22, 2013 (2013). http://www.osti.gov/scitech/servlets/purl/1073108
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 J. Sci. Comput. **29**(2), 556–578 (2007)
17. Getling, A.V.: Rayleigh–Bénard Convection: Structures and Dynamics, vol. 11. World Scientific, Singapore (1998)
18. Glatzmaier, G.A., Roberts, P.H.: A three-dimensional convective dynamo solution with rotating and finitely conducting inner core and mantle. Phys. Earth Planet. Interiors **91**(1–3), 63–75 (1995)
19. Grossmann, S., Lohse, D.: Scaling in thermal convection: a unifying theory. J. Fluid Mech. **407**, 27–56 (2000)
20. Hunter, J.D.: Matplotlib: a 2d graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007)
21. Johnston, H., Doering, C.R.: Comparison of turbulent thermal convection between conditions of constant temperature and constant flux. Phys. Rev. Lett. **102**(6), 064501 (2009)
22. Kerr, R.M.: Rayleigh number scaling in numerical convection. J. Fluid Mech. **310**, 139–179 (1996)
23. King, E., Stellmach, S., Aurnou, J.: Heat transfer by rapidly rotating Rayleigh–Bénard convection. J. Fluid Mech. **691**, 568–582 (2012)
24. King, E., Stellmach, S., Buffett, B.: Scaling behaviour in Rayleigh–Bénard convection with and without rotation. J. Fluid Mech. **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, R.J., Verzicco,

R.: Comparison of computational codes for direct numerical simulations of turbulent Rayleigh–Bénard convection. Comput. Fluids **166**, 1–8 (2018)

27. Lions, J.L., Maday, Y., Turinici, G.: Résolution d'edp par un schéma en temps pararéel. C. R. l'Acad. des Sci.-Ser. I-Math. **332**(7), 661–668 (2001)
28. Lunet, T., Bodart, J., Gratton, S., Vasseur, X.: Time-parallel simulation of the decay of homogeneous turbulence using parareal with spatial coarsening. Comput. Vis. Sci. **19**(1–2), 31–44 (2018)
29. Matsui, H., Heien, E., Aubert, J., Aurnou, J.M., Avery, M., Brown, B., Buffett, B.A., Busse, F., Christensen, U.R., Davies, C.J., et al.: Performance benchmarks for a next generation numerical dynamo model. Geochem. Geophys. Geosyst. **17**(5), 1586–1607 (2016)
30. McKenzie, D.P., Roberts, J.M., Weiss, N.O.: Convection in the earth's mantle: towards a numerical simulation. J. Fluid Mech. **62**(3), 465–538 (1974)
31. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci. **5**(2), 265–301 (2010). https://doi.org/10.2140/camcos.2010.5.265
32. Mound, J.E., Davies, C.J.: Heat transfer in rapidly rotating convection with heterogeneous thermal boundary conditions. J. Fluid Mech. **828**, 601–629 (2017). https://doi.org/10.1017/jfm.2017.539
33. Ruprecht, D.: Wave propagation characteristics of parareal. Comput. Vis. Sci. **19**(1), 1–17 (2018). https://doi.org/10.1007/s00791-018-0296-z
34. 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. Comput. Phys. Commun. **221**, 19–27 (2017)
35. Samuel, H.: Time domain parallelization for computational geodynamics. Geochem. Geophys. Geosyst. **13**(1) (2012)
36. Schumacher, J., Bandaru, V., Pandey, A., Scheel, J.D.: Transitional boundary layers in low-Prandtl-number convection. Phys. Rev. Fluids **1**(8), 084402 (2016)
37. Shishkina, O., Stevens, R.J., Grossmann, S., Lohse, D.: Boundary layer structure in turbulent thermal convection and its consequences for the required numerical resolution. N. J. Phys. **12**(7), 075022 (2010)
38. Siggia, E.D.: High Rayleigh number convection. Annu. Rev. Fluid Mech. **26**(1), 137–168 (1994)
39. Staff, G.A., Rønquist, E.M.: Stability of the parareal algorithm. In: Kornhuber, R. 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). https://doi.org/10.1007/3-540-26825-1_46
40. 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)
41. Stevens, R.J., Verzicco, R., Lohse, D.: Radial boundary layer structure and Nusselt number in Rayleigh–Bénard convection. J. Fluid Mech. **643**, 495–507 (2010)
42. Verzicco, R., Camussi, R.: Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell. J. Fluid Mech. **477**, 19–49 (2003)
43. Zhu, X., Mathai, V., Stevens, R.J., Verzicco, R., Lohse, D.: Transition to the ultimate regime in two-dimensional Rayleigh–Bénard convection. Phys. Rev. Lett. **120**(14), 144502 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.