Asymptotic Ultimate Regime of Homogeneous Rayleigh-Bénard Convection on Logarithmic Lattices - Supplementary Materials

1 Numerical scheme

Simulations on log-lattices introduce a variety of interesting challenges, among which the numerical integration of viscosity. Since the grid's maximum wave vector scales as $k_{\max} = k_{\min} \cdot \lambda^N$ for a grid of size N, a "classical" integration scheme for $\partial_t u = A(t) - \nu k^2 u$ becomes exponentially slow with N !

A classical way to resolve this issue is to use an implicit temporal scheme by evaluating the viscosity term at $t+\mathrm{dt}$ rather than at t, which yields $u(t+\mathrm{dt}) = \dfrac{u(t) + A(t) \cdot \mathrm{dt}}{1 + \nu k^2 \cdot \mathrm{dt}}$. However, this form is not compatible with any of the common ODE solvers, which do not disclose the time step dt (and have good reasons for doing so, as hacking naively the implicit equation in an ODE solver will yield significant mathematical artifacts.)

To run ODEs on Log-lattices, we developed our own solver, which is a simple variation on usual multistep ODE solvers. We simply compute the equation without the viscous terms using the multistep solver, then we retroactively add a viscous correction.

For the sake of reproducibility, let's look at an example based on the solver we chose, DOPRI5, a multistep method based on Runge-Kutta algorithms (this trivially generalizes to most multistep solvers). The unmodified DOPRI5 method we use is defined in the python library Scipy, and is based on **?**. Consider an observable `y` and its ODE update function `f(y, t)->dy/dt(t)=`$A(t) - k^2 y$. Each simulation step, `DOPRI5` evaluates `f` several times, determines the new time step `dt`, then updates `t` and `y` accordingly: `DOPRI5(y, f, t)->y(t+dt), t+dt, dt`. Our modification consists in decoupling the explicit and implicit step for the solver: the solver first calls DOPRI5 with $f = A(t)$, which yields a new field y_1: `DOPRI5(y, A, t)->y1, t+dt, dt`. We then update y_1 implicitly with the viscosity: $y(t + \mathrm{dt}) = \dfrac{y_1}{1 + k^2 \cdot \mathrm{dt}}$.

This way, we get to keep the flexibility of multi-stepping, while integrating viscous terms in a trivial time.

In practice, we decouple HRB equation in the following fashion: on each equation the top line (red) is explicit and multistepped, the bottom line (blue) is implicit and does not *seemingly* take part in multistepping (to be exact, it still takes part under the hood since, in the original DOPRI5 implementation, the value of `y'` at step `t` is used in the computation of `y` at time `t+1`. Since we modify the value of `y` and thus the true value of `y'`, it is necessary to recompute the full derivatives matrix each step. This makes this version of DOPRI5 slightly slower than its original version.)

$$\partial_t \omega_i = -\omega_j \partial_j u_i - u_j \partial_j \omega_i + \theta [\boldsymbol{\nabla} \times \boldsymbol{z}]_i + \sqrt{\frac{\mathrm{Ra}}{\mathrm{Pr}}} \Delta \omega_i - f \omega_i \delta_{k \sim k_{min}}$$
$$\partial_t \theta = -u_i \partial_i \theta + u_z + \frac{\Delta \theta}{\sqrt{\mathrm{Ra}\,\mathrm{Pr}}} - f \theta \delta_{k \sim 2\pi}$$

It is important to note that the stability of this method comes at the cost of an apparent hyperviscosity if $k_{\mathrm{max}}^2 \,\mathrm{dt} \ll 1$ is not verified anymore, which can impact the expected conservation laws.