

ROUGHNESS EFFECTS ON TWO-DIMENSIONAL TURBULENT CONVECTION: HEAT TRANSPORT, FLOW REVERSALS, AND MACHINE LEARNING ANALYSIS

V. Kavitha

Department of Mathematics
Vardhaman College of Engineering
India
vkavithaou@gmail.com

Seelam Sunitha Devi

Department of Mathematics
Koneru Lakshmaiah Education Foundation
India
sunithamallakula@yahoo.com

Balla Chandra Sekhar

Department of Mathematics
Vardhaman College of Engineering
India
shekmaths10@gmail.com

Article history:

Received 18.07.2024, Accepted 20.12.2024

Abstract

In two-dimensional turbulent convection, the effects of roughness configurations on heat transport and flow reversal are examined in this work. The impact of five distinct rough models on the Nusselt number (Nu) as a function of Rayleigh number (Ra) is investigated and analysed. All the rough models show reduced heat transport at low Ra ; the model with locally compact roughness elements shows the most significant reduction in heat transport. As Ra increases, the normalized Nu generally increases, with differences observed between models with sparsely distributed and locally compact roughness. Flow reversals in 2DRB convection are also explored, with the presence or absence of reversals categorized among the rough models. Flow reversal processes are identified using angular momentum analysis. The study reveals chaotic oscillations in the flow field and Nu for certain models, indicating the influence of roughness on the Large-Scale Circulation (LSC). Sparse models with widely spaced rough elements exhibit more active correlations between the cavity's fluid and LSC, leading to enhanced heat transfer. The scaling relationship between Nu and Ra is investigated, showing distinct scaling regimes for different Ra ranges. The distribution of roughness elements and the relative contributions of the majority of the surface and boundary-layer areas to thermal dissipation influence scalar behaviour. Machine learning techniques, including Convolutional Auto-encoders (CAEs) and Gated Recurrent Units (GRUs), are employed to compress and predict snapshots of turbulent convection data. These techniques offer a promising approach to analyse complex turbulence data and facilitate sequence analysis and prediction. Overall, this work delivers valuable insights into the role of roughness configurations in two-dimensional turbulent convection, shedding light on heat transport,

flow reversals, and scaling relationships. The use of machine learning models enhances the understanding and prediction of complex turbulence behaviour.

Key words

Nusselt number, Large-Scale Circulation, Convolutional Auto-encoders, Gated Recurrent Units, Angular Momentum.

1 Introduction

For heat-driven flows, which are common in both organic and industrial environments, Rayleigh–Bénard convection (RBC) offers a well-researched paradigm [Ecke et al., 2023]. Temperature gradients [Plumley et al., 2019] and their influence on heat transport investigate buoyancy effects that eventually lead to fluid dynamics instability. One excellent area of study with important implications for science is the regulation of RBC [Shishkina et al., 2021]. Furthermore, in many applications, controlling heat transport and/or preventing, reducing, or enhancing such instabilities are essential. Processes for crystal growth, such as those used to make silicon wafers, are examples [beintema et al., 2020]. A frequent occurrence in nature, buoyancy-driven turbulent fluid motion is influenced by rotation and has numerous industrial uses. Rayleigh-Benard convection (RBC) [Liu, et al., 2020], a well-researched laboratory implementation of turbulent convection, involves a fluid contained within a convection cell featuring adiabatic vertical walls, a cooled top, and a heated bottom. Under these circumstances, cooperative plume motion gives rise to a large-scale circulation (LSC) that is a crucial aspect of turbulent RBC [Ecke et al., 2023]. Heat transport is improved by Ekman pumping [Shishkina et al., 2021], [beintema et al., 2020], [Liu, et al., 2020], [Favier et al.,

2019], [Wang et al., 2019], [[Zhang, et al., 2023], [Cheng et al., 2022], [Horn et al., 2022] numerical assessments of vorticity and temperature fluctuations in the bulk are significantly impacted [Xu et al., 2022], [Huang et al., 2022], [Schneide, et al., 2022], [[Madonia, et al., 2023], [[Boot, et al., 2021], [[Zhang, et al., 2023], [[Song, et al., 2023] and heat plumes change into vortices of heat when rotation about a vertical axis is added. These changes result in a distinct kind of convection. Suppressing the LSC of nonrotating convection is an essential part of rotation for rotation rates that are fast enough [Huang et al., 2022], [Schneide, et al., 2022], [[Madonia, et al., 2023], [[Boot, et al., 2021], [[Zhang, et al., 2023], [[Song, et al., 2023], [[Vogt, et al., 2021], [[Tagawa, et al., 2023]. The nature of this suppression seems to be influenced by the aspect ratio of diameter to height $\Gamma = \frac{D}{H}$ [Chong, et al., 2023] In RBC geometries where $\frac{1}{2} \leq \Gamma \leq 2$, the LSC typically covers the cell in a roll-like movement of size H . With increasing rotation rate, rotating convection exhibits a reduced naturally occurring linear vortex separation scale [Plumley et al., 2019], [kashanj, et al., 2023]. This implies that lowering the geometric aspect ratio, or $\Gamma < 1$, could be achieved while preserving a lateral cell size to linear scale ratio is large [5]. Several new experiments are utilising these convection cells [Abbate, et al., 2024]. The existence of a global circulation that significantly affects the internal state of the system and contributes to a significant amount of global heat transport is therefore a crucial question concerning rotating convection in thin cylindrical cells. Using $\Gamma = 1$ inverse Rossby number $\frac{1}{Ro} = 2.78$ Rayleigh number $Ra = 10^9$ and Prandtl number $Pr = 6.4$ (Ro, Ra, and Pr defined below), a core region of anti-cyclonic circulation, located close to the cylinder sidewall, was surrounded by a cyclonic azimuthal velocity boundary-layer flow, as demonstrated by direct numerical simulations (DNS) of rotating convection[23]in cylindrical geometry. The outcomes were clarified by means of active Ekman layers powering sidewall Stewartson layers at the top and bottom of the cell [?], [Kannan, et al., 2024].

2 GOVERNING EQUATIONS AND PARAMETERS

To facilitate a clear understanding of the physical modeling involved in rotating Rayleigh–Bénard convection (RRBC), we first define all the variables and parameters used in the governing equations. The fluid motion is described in a rotating frame with a constant angular velocity vector $\Omega = \Omega_{(e_z)}$ where Ω is the rotation rate and e_z is the unit vector in the vertical direction. The primary field variables are the velocity vector $u = (u, v, w)$ the temperature T , and the pressure p . The equations are written under the Oberbeck–Boussinesq approximation, which assumes that all fluid properties are constant except for the density variation in the buoyancy term. The reference density is denoted by ρ_0 the kinematic viscosity by ν the thermal diffusivity by κ the thermal expansion coefficient by α and the gravita-

tional acceleration by $g = -g(e_z)$. The Rayleigh number $Ra = \frac{g \alpha \Delta T H^3}{\nu \kappa}$ characterizes the ratio of buoyancy to viscous and thermal diffusion effects, where ΔT is the temperature difference across the fluid layer and H is the vertical height of the domain. The Prandtl number $Pr = \frac{\nu}{\kappa}$ expresses the relative importance of momentum diffusion to thermal diffusion. The centrifugal force appears as $-\Omega \times (\Omega \times \mathbf{r})$ and the Coriolis force as $-2\Omega \times \mathbf{u}$ where $\mathbf{r}$ is the position vector. The gradient operator is ∇ and the Laplacian is ∇^2 . Additionally, e_y is used to denote the vertical unit vector in the 2D case. The governing equations include the incompressible continuity equation, the Navier-Stokes momentum equation modified by rotation and buoyancy terms, and the thermal energy equation accounting for convection and diffusion.

When transitioning from a fixed coordinate system to a rotating one with an angular velocity of $\vec{\Omega} = \Omega \hat{e}_z$ you introduce extra Coriolis $-2\vec{\Omega} \times \vec{u}$ and centrifugal $-\vec{\Omega} \times (\vec{\Omega} \times \vec{r})$ accelerations, which manifest as supplementary efficient forces within the momentum equation (1) is given below.

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{\nabla P}{\rho} + \nu \nabla^2 \vec{u} + \vec{g} - 2\vec{\Omega} \times \vec{u} - \vec{\Omega} \times (\vec{\Omega} \times \vec{r}) \quad (1)$$

(The sidebar labelled Dimensional Characteristics of RRBC contains information on notational conventions). For $\vec{\Omega} \equiv \Omega \hat{e}_z$ one obtains $-\vec{\Omega} \times (\vec{\Omega} \times \vec{r}) = \Omega^2 r \hat{e}_r$.

In the most basic Oberbeck–Boussinesq (OB) approximation that includes the effects of buoyancy as described by Oberbeck in 1879 and Boussinesq in 1903, all of the fluid’s characteristics are taken to stay constant, with an exemption of the buoyancy term’s density. This density is considered to have a linear dependency on temperature, expressed as, $\rho \approx \rho_0 [1 - \alpha(T - T_0)]$. By introducing a reduced pressure term, denoted as $p \equiv P + \rho_0 g z \hat{e}_z - \frac{\rho_0}{2} \Omega^2 r^2 \hat{e}_r$ and considering $|\alpha(T - T_0)| \leq \frac{\alpha \Delta}{2} \ll 1$ which is a valid assumption within the OB approximation, we can derive the relation $\frac{P}{\rho} \approx \left(\frac{P}{\rho_0}\right) [1 + \alpha(T - T_0)]$. This also holds true for the momentum equation within the Oberbeck–Boussinesq approximation, as described by Becker et al. (2006) [REF]. The equation (2) is given as

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{\nabla p}{\rho_0} + \nu \nabla^2 \vec{u} - 2\Omega \hat{e}_z \times \vec{u} - \alpha(T - T_0) \Omega^2 r \hat{e}_r + \alpha(T - T_0) g \hat{e}_z \quad (2)$$

Which, in addition to the heat equation is given as eq. (3) and the continuity equation, $\nabla \cdot \vec{u} = 0$

$$\frac{\partial T}{\partial t} + (\vec{u} \cdot \nabla) T = \kappa \nabla^2 T \quad (3)$$

the governing equations where u is velocity, T is temperature, p is pressure, ρ_0 is reference density, ν is kinematic viscosity, κ is thermal diffusivity, g is gravitational acceleration, α is thermal expansion coefficient, Ω is angular rotation vector, Ra is Rayleigh number, Pr is Prandtl number, ∇ is gradient operator, and ∇^2 is Laplacian operator within the context of OB rotating Rayleigh-Bénard convection. Both gravitational buoyancy and buoyancy are present in RRBC, as can be seen from the two final variables in Equation 2, acting vertically (similar to non-rotating Rayleigh-Bénard convection), and centrifugal buoyancy, exerting itself radially. The centrifugal term can only be disregarded when the angular velocity Ω is relatively small. It is important to note that, the centrifugal term is assessed assuming a constant density. This allows for the integration of the complete centrifugal phrase into the decreased pressure, resulting in a formula that is similar to equation 2 but lacks the final term. Nevertheless, when studying the centrifugal effects within the Oberbeck-Boussinesq (OB) approximation, the complete momentum. The standard boundary conditions for Equations 2 and 3 consist of no-slip conditions for velocity $u = 0$ at all boundaries, isothermal temperature at the bottom T^+ and a lower temperature $T^- < T^+$ at the top, as well as adiabatic conditions $\frac{\partial T}{\partial n} = 0$ at the lateral boundaries.

3 NUMERICAL METHOD

We conduct DNS simulations of turbulent Rayleigh-Bénard convection in 2D containers featuring distinct arrangements of triangular rough elements located on the horizontal walls. The container's height and horizontal length are both consistently set at 1 unit. We explore the effects of different spatial distributions of these rough elements on Rayleigh-Bénard convection using five computational models, denoted as model1, model2, and so forth (refer to Fig. 1 for visualization). These models are selected to maintain the same total contact area for the rough elements.

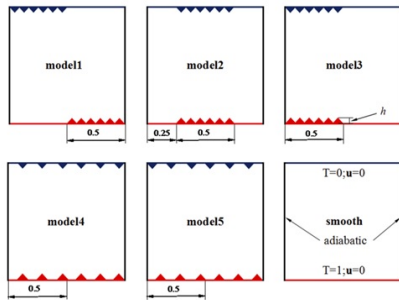


Figure 1. Rayleigh-Bénard convection using five computational models

The figure illustrates that models 1 through 3 have a dense concentration of rough elements on certain por-

tions of the horizontal walls, while models 4 and 5 showcase a sparser and more uniform distribution. We classify these models into two categories based on a sparsity ratio, denoted as Γ_r , which represents the proportion of rough component width to vertex spacing. Compact models, including model 1 through 3, have $\Gamma_r = 1$ whereas sparse models, represented by model 4 and model 5, feature $\Gamma_r = 2$.

Since the height of each rough element is constant at $h = \frac{1}{24}$ across every structure, and with a vertex angle of 90 degrees, all rough elements are isosceles right triangles, it does not add to variations in the measured Nusselt number (Nu). $\lambda = \frac{1}{12}$ for compact models and $1/6$ for sparse models is the wavelength, which is expressed as the ratio of vertex spacing to cell height. A variable that improves the rough contact area between the upper and lower surfaces of $\frac{\sqrt{2}+1}{2}$ in the current configurations.

The horizontal (vertical) direction in relation to the centre and central symmetry regarding the cavity centre. Model 1 possesses central symmetry, designed to amplify large-scale circulation. Model 2 features all three symmetries, potentially leading to a splitting of the large-scale structure. Since Model 3 only has vertical symmetry, the large-scale circulation may be diminished by the placement of rough elements on one side. Both model 4 and model 5 share the characteristic of vertical symmetry and central symmetry, with the expectation that these models will enhance plume emission.

The thermal convection system is described by a set of equations governing the conservation of mass, momentum, and energy, which incorporates the Oberbeck-Boussinesq approximation. We use the height of the box (H) to scale the imposed temperature difference Δ time $\frac{H^2}{\kappa}$ and velocity $\frac{\kappa}{H}$ to make these equations dimensionless (4-6) as a result:

$$\nabla \cdot \mathbf{u} = 0 \quad (4)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + Pr \nabla^2 \mathbf{u} + Ra Pr T \hat{e}_y \quad (5)$$

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \nabla^2 T \quad (6)$$

The velocity field (u), temperature field (T), pressure field (p), and the vertical direction e^y are key components of the system. At every solid wall, we enforce no-slip boundary conditions ($u=0$). The top and bottom plates are kept at constant dimensionless temperatures, $T=0$ and 1 , respectively, while the side walls are heat isolated $\frac{\partial T}{\partial n} = 0$. Fischer et al.'s open-source spectral element code Nek5000 [Fischer, et al., 2006] is used in our simulations. This code has a strong track record in

accurately simulating turbulent RB convection and convection involving rough surfaces.

In our simulations, the Prandtl number (Pr) remains constant at 0.7, while the Rayleigh number varies from 10^6 to 10^9 . We collect data for a duration exceeding 500 times the free-fall time $\tau_f = \sqrt{Ra Pr}$. To ensure the smallest turbulence scales are resolved, for distinct Rayleigh numbers, we employ various amounts of spectral elements (SE). These elements are concentrated near solid walls to effectively capture the boundary layers. After further decomposition, every component is divided into a grid with P Gauss-Lobatto-Legendre (G-L-L) quadrature points; P = 11 is used in the present article. Notably, as Wagner and Shishkina [Wagner, et al., 2006] point out, at $Ra=109$, approximately seven spots are located inside the boundary layers, and in the bulk region, the dimension of the grid is sufficiently small to figure out the Batchelor and Kolmogorov scales [Hurtan, et al., 2006].

4 RESULTS

4.1 Heat transmission and flow patterns

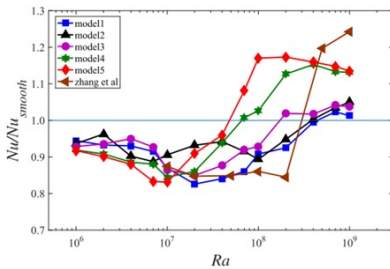


Figure 2. Nusselt number with models

Initially, examine how heat transport changes with Ra for our five different rough models. We present the results in Figure 2, where we normalize the Nusselt number (Nu) by the smooth cell's value to highlight the effects of improvement or decrease. By comparing this Nusselt number ratio, we can evaluate the effect of the rough models. All five rough models clearly show reduced heat transport at low Ra. At approximately $Ra = 2107$ in model 1, compared to the smooth case, Nu is decreased by 17.7% the largest reduction in heat transfer occurs. The normalised Nu typically rises with Ra at higher Ra. When comparing models with locally compact roughness (model 1, model 2, and model 3) to those with sparsely distributed rough elements (models 4 and 5), the increase in Nu/Nu_{smooth} happens at smaller Ra. As a result, Heat transfer improvement is depicted in the figure 3 is attained for compact models with $\Gamma_r = 1$ (models 1-3) at a comparatively high $Ra > 10^8$ and regarding sparse approaches with $\Gamma_r = 2$ (models 4 and

5), at a relatively low Ra ($10^7 < Ra < 10^8$). It is stated that the distinct roughness distribution on the boundaries results in increased heat transfer above a critical Rayleigh number, Ra_c which is established by the roughness distribution [32]. When Ra is less than the crucial value, the temperature boundary layer is larger than the average roughness dimensions. Enrichment is observed when the thermal boundary layer is smaller than the roughness size. Additionally, we include the results of roughness covering all the plate models from Zhang et al, [Zhang, et al., 2024]. for comparison. It is noticeable that the value of the crucial Ra_c is comparable to that of the smaller models (models 1-3); but these models exhibit more of an increase in heat transfer when Ra exceeds the critical level.

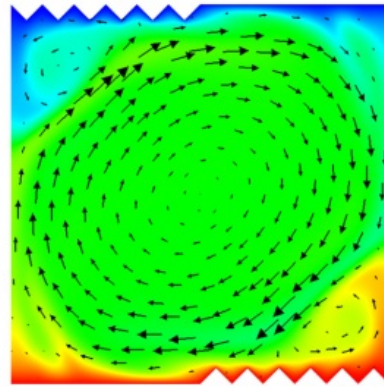


Figure 3. Heat transfer improvement

In 2DRB convection, it's known that the flow field experiences a reversal around $Ra = 10^7$ for medium Prandtl numbers. It is highly desirable to comprehend how roughness configurations affect these flow reversals. In our work, we categorize the rough models into two groups when Ra is less than 10^7 . One group consists of model 3, model 4, and model 5, in which flow reversal is observed, while the other group consists of model 1 and model 2, where our simulations show that there is no flow reversal. We can identify the reversal process by examining the net angular momentum $L(t)$, calculated as $L(t) = \langle -(y - 0.5)u(x, y, t) + (x - 0.5)v(x, y, t) \rangle_V$ where $\langle \cdot \rangle_V$ denotes averaging over the entire region. A change in the sign of L indicates a flow reverse. In Figure 3, we display the Nu time series and immediate flow field for model 1 at $Ra = 10^7$ which displays the temperature and velocity fields. There are chaotic oscillations with a dominant vortex in a combination of the field of flow and the immediate Nu that is measured.

Similar features can be seen in the field of flow in model 2 at a similar Ra. In models 3, 4, and 5, flow reversals are detectable over the Rayleigh number range of $4 \times 10^6 \leq Ra \leq 10^7$. Figures 4 and 5 depict, respectively, the time-varying Nu and L as well as two typical

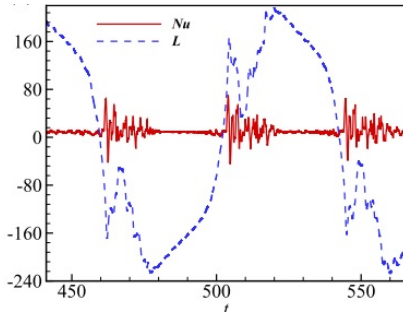


Figure 4. Time-varying Nu and L

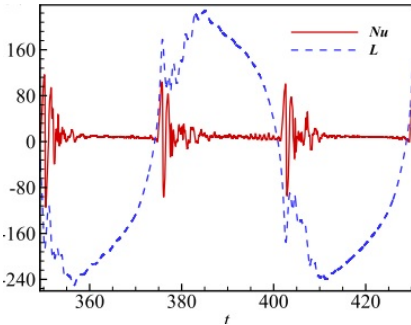


Figure 5. Time-varying Nu and L

immediate flow fields at $Ra = 107$ that are derived from models 3 and 5. It should be noted that because the rough element arrangements in models 4 and 5 are similar, the reversal processes in the two models are nearly identical.

When comparing Nu's time series in Figs. 4 and 5, it is evident that the value fluctuates significantly during reversals. L's sign changes during reversals, as we can see in the interim. We compute the flow reversal frequencies for $Ra = 107$, which have the reverse data per free-fall time as the definition. Models 3, 4, and 5 have respective values of 0.022, 0.025, and 0.026. These models have very similar reversal frequencies, however model 5's Nu(t) variation amplitude is substantially larger. Following reversal, Nu fluctuations diminish rapidly, and in model 5, more evenly and sparsely distributed rough elements even cause the flow to become quasi-steady. According to the average temperature near the two horizontal plates, the roughness arrangements of Models 1, 2, and 3 are more likely to trap hot or cold fluid in the cavity areas among the closest rough elements. The trapped fluid is unable to mix thoroughly as a result, which hinders the cavities' ability to transfer heat globally. On the other hand, it will generate a hot plume if the boundary layer in model 1–model 3 is sufficiently thin at a higher Ra number. Ra will therefore be encouraged when it surpasses the critical value for heat transfer. However, the models 4 and 5 are completely distinct from the first three. The great separation between neighbouring rough elements makes the LSC's penetration into the cavity regimes, as indicated by the flow visualisations. As a result, the LSC and the fluid inside the cavities com-

municate more actively. It is also simpler to generate a secondary vortex that is exactly the identical size as the roughness, and more plume emissions can be easily triggered by communications among the rough element tips and the secondary vortex. Examining the instantaneous fields reveals that in both models 4 and 5, nearly every rough element tip on the top and bottom plates experiences a quick excitation and release of hot or cold plumes. The local temperature oscillation is related to the characteristic plume emission frequency. For models 1 through 5, the oscillation frequencies are 0.0264, 0.051, 0.05, 0.076, and 0.075, respectively. We have recorded the temperature at (0.5, 0.1). As a result, models 4 and 5 have significantly higher plume emissions, which raises the efficiency of heat transport. The flow field cannot, however, directly reveal the significant difference in the measured Nu among both of these models for symmetric and anti-symmetric rough element arrangements, as seen in Fig. 2.

4.2 $Nu \sim Ra$ scaling

Besides impacting the Nu magnitude, rough surfaces also serve a crucial function in altering the scaling relationship between Nu and Ra. Figure 6 displays the determined Nu values as a function of Ra on a logarithmic scale. In the case of smooth surfaces, the best-fitting power-law curve for Nu versus Ra yields an exponent of approximately 0.31, consistent with prior findings.

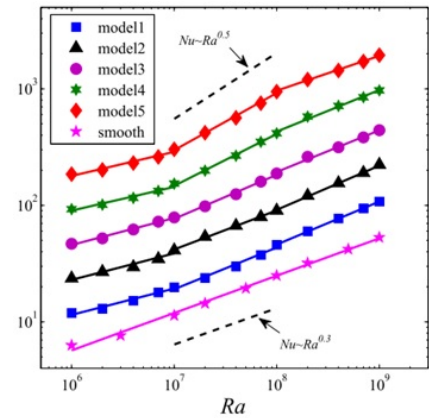


Figure 6. Nu values as a function of Ra on a logarithmic scale

However, with rough surface models, the behavior of Nu for Ra becomes more complex. Three distinct scaling regimes emerge, specifically for Ra within the ranges 10^6 to 10^7 , 10^7 to 10^8 , and 10^8 to 10^9 . The five rough models have scaling exponents ranging from 0.2 to 0.24 in the minimal Ra range (10^6 to 10^7), which closely resemble the 1/4 scaling observed in the inadequate Rayleigh numbers in a smooth-wall model.

Ra increases more quickly as it approaches the moderate range of 10^7 to 10^8 , and the scaling exponent fluctuates.

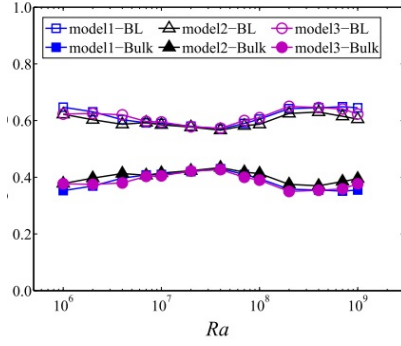


Figure 7. Scaling exponent for Ra with Nu

tuates between 0.335 and 0.49. These numbers are significantly higher than the smooth surface model's. It is important to note that, contrary to Zhu et al. [Zhu, et al., 2018], [Plotnikoy, et al., 2024], the highest possible system is not yet reached by the flow state, even though the highest scaling exponent for models 4 and 5 in this range is nearly equal to that of the ultimate regime.

The scaling-law range does not continue to expand with additional increases in Ra. The scaling exponent for Ra between 10^8 and 10^9 is between 0.305 and 0.379, as shown in Figure 7. This suggests a tendency for the scaling relationship to revert.

For Ra values between 10^7 and 10^9 , the scaling exponents for compact models (model 1-model 3) exhibit only slight variations with increasing Ra; nevertheless, the results show notable differences for sparse models (models 4 and 5). This variation is probably caused by the different ways that the rough elements are arranged.

The Nu-Ra relation is significantly influenced by the rate of thermal dissipation. If the bulk contribution is dominant for the current variable covers of Ra and Pr, the scaling exponent should be approximately $\frac{1}{2}$; if the boundary layer is linear in form, then the scaling coefficient should be roughly $\frac{1}{3}$ if the boundary layer role is dominant. Zhu et al. [34], have also demonstrated such a shift from a system dominated by bulk to the system dominated by boundary layer. We directly calculated the thermal dissipation rates, $\varepsilon_\theta = \left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2$, for each model to determine whether this transition can account for the observed variations of the scaling exponent. In Fig. 7, the relative contributions to the total ε_θ of the bulk and boundary-layer regions are plotted as functions of Ra. Observe that we use the global estimation $\frac{H}{2Nu}$ to determine the thermal boundary-layer thickness. It is evident that for all Ra studied for compact models (models 1 through 3), the boundary layer dominates the thermal dissipation rate. Nonetheless, for the sparse models (models 4 and 5), the bulk contribution exceeds that of the boundary-layer region for the moderate Ra ($10^7 < Ra < 10^8$) while the global thermal dissipation rate is dominated by the boundary layer for the high Ra ($\sim 10^9$) and low Ra ($\sim 10^6$). This provides a good explanation for the various scaling exponents for

various models measured in various Ra regimes. Compared to Zhu et al [34], $Ra_c > 10^9$ with $\frac{\lambda}{h} = 1$ for $h = 0.05, 0.1$, and 0.15 , respectively, for the current instances, the crucial Ra for the scaling transitions is much reduced in size. The apparent reason for this is most likely the two studies' disparate rough element arrangements.

4.3 $Re \sim Ra$ scaling

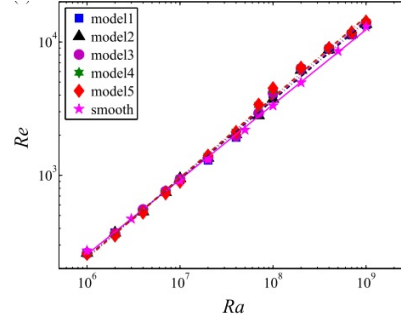


Figure 8. a logarithmic plot for correlation among the measured Reynolds number (Re) and the Rayleigh number (Ra) for various models

The Reynolds number can be used to assess the impact of roughness on flow strength Re . In this case, we compute $Re = \frac{U_{rms}H}{\nu}$, where $U_{rms} = \sqrt{\langle v_x^2 + v_y^2 \rangle_{A,t}}$. In Figure 8, a logarithmic plot displays the correlation among the measured Reynolds number (Re) and the Rayleigh number (Ra) for various models. Notably, it is apparent that the scaling exponents for all models are approximately 0.6, indicating that the arrangement of roughness elements in these models does not significantly affect Re. Clearly, the buoyancy forces responsible for driving the large-scale flow appear to be unaffected by the boundary geometry. In Figure 8, plots are presented that compensate for Re by multiplying it by Ra raised to the power of -0.6. For all of the rough models, we can see local maxima around Ra values of about 10^8 . These maxima may be linked to variations in the correlation among the Nusselt number (Nu) and Ra.

5 DATA FROM TWO-DIMENSIONAL TURBULENT CONVECTION SIMULATION

5.1 The Turbulent Convection Data

Turbulent convection data in the two-dimensional case are produced through a DNS (Direct Numerical Simulation) employing the nek5000 spectral element solver. The simulation focuses on solving the dimensionless Boussinesq equations (equations 7 to 8), which describe the interaction between velocity components (u_x and u_z) and temperature (T), within a closed rectangular cell characterized by an aspect ratio of $L/H=6$.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (7)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \sqrt{\frac{\text{Pr}}{\text{Ra}}} \frac{\partial^2 u_i}{\partial x_j^2} + T \delta_{iz} \quad (8)$$

$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} = \frac{1}{\sqrt{\text{Ra Pr}}} \frac{\partial^2 T}{\partial x_j^2} \quad (9)$$

The pressure field is represented as 'p,' with 'i' and 'j' belonging to the set x, z. 'x' signifies the horizontal coordinate, while 'z' signifies the vertical coordinate. The dimensionless Rayleigh number, denoted as 'Ra,' quantifies the intensity of convective turbulence and has been set to a value of 10^7 in this context. The Prandtl number, 'Pr,' a dimensionless quantity that quantifies the ratio of momentum to thermal diffusion, has been maintained at '7,' corresponding to its value for thermal convection in water. The specific parameter values are as follows equation (10):

$$\text{Ra} = \frac{g\alpha\Delta TH^3}{\nu\kappa} \quad \text{and} \quad \text{Pr} = \frac{\nu}{\kappa} \quad (10)$$

In this context, several variables are defined: g represents the gravitational acceleration, α stands for the thermal expansion coefficient, ν is the kinematic viscosity, κ denotes the thermal diffusivity, and ΔT indicates the temperature difference between the upper and lower plates. Establishing with the cell height H renders every formula undefined, the free fall velocity U_f is calculated as $\sqrt{g\alpha\Delta TH}$ and it's important to note that ΔT is greater than zero. Also, the temperature T is constrained within the range of 0 to 1, which implies that ' θ ' lies between -0.5 and 0.5.

The simulation domain is divided into a grid of 48×16 spectral elements, where every component represents a region in the simulation. Polynomials of order 11 for every spatial dimension for the four fields characterise these elements. The data for these fields can be acquired by spectral interpolation on a uniform grid of $N_x \times N_z = 320 \times 60$ points for the machine learning analysis. This process results in 2400 snapshots, yielding a total of $2400 \times 320 \times 60$ data points. These pictures were collected at intervals of 0.125 free fall time units, which is H/U_f . Comprising the probabilistic perturbation initialization stage, the entire DNS (Direct Numerical Simulation) data generation process required roughly 123 CPU-hours.

5.2 Producing a compressed snapshot image

One type of machine learning model is an auto-encoder that comprises two main components: an encoder and a

decoder. The encoder's main responsibility is the input data to be compressed into a space of latent information that has been trained. Data in this latent representation is taken by the accompanying decoder, which reconstructs it to look like the original input data. It's important to note that autoencoders are considered self-supervised, meaning that the network uses its input as both the input and the expected output during training, eliminating the need for additional labelling efforts for a training dataset. In concept, an autoencoder takes input data I and encodes it into a representation $c = \text{encode}(f_0)(I)$, where $c \in \mathbb{R}^l$. Then, it decodes c back into an approximation of the original input, denoted as $\hat{I} = \text{decode}(f_0)(c)$. In actuality, though, there might be some variations, resulting in $\hat{I}$ being an approximation of I , as shown in Figure 9. To train the autoencoder, an L_2 norm is used as the objective function, and the training process utilizes a gradient descent method, such as the adaptive momentum-based optimization algorithm known as Adam.

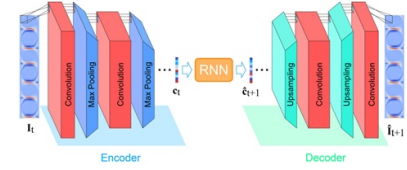


Figure 9. Auto encoder Architecture

Layers with convolution are utilised instead of fully connected layers, in a Convolutional Autoencoder (CAE) is particularly well-suited for handling complex high-dimensional input data. To introduce nonlinearity, a standard convolutional layer incorporates an activation function with a convolution operation on the input data using learnable kernels. Consider input data that is three dimensional, denoted as $I \in \mathbb{R}^{C_{in} \times N_x \times N_z}$, where C_{in} represents the number of input channels. In this specific case, $C_{in} = 1$, as we are loading only the heat flux field entering the network convectively, so $I \in \mathbb{R}^{N_x \times N_z}$. This input is convolved with a kernel, k_m , as defined in Equation (9), where $M = [1, C_{out}] \subset \mathbb{N}$, and C_{out} is the number of channels for output. Additionally, b_m represents the bias term associated with kernel k_m , and ψ is a nonlinear activation function. Normally, zero-padding is introduced to the layer's input to avoid the loss of information at the outer borders of the input data. The Equation (11) is given as

$$\text{Conv}(m, I) = \psi \left(b_m + \sum_{i=1}^{C_n} k_m * I_i \right), \quad m \in M \quad (11)$$

In the encoder, there are naturally several convolutional layers, which are usually subsequent by pooling layers. These pooling layers use a configurable window

size and a sliding step across the input. This process aggregates the input within utilising a selected aggregation function, combine every window into a single output element. For instance, the conventional max-pooling layer only retains the greatest value for each window. The step size of the pooling layer is usually selected to reduce the input's dimensionality. As a result, the encoder's final output is in a lower-dimensional latent space denoted as RI , where I is much smaller than $N \times N \times Z$.

The decoder of the Convolutional Autoencoder (CAE) follows a similar design, with the primary difference being the use of up sampling layers as a substitute of max-pooling layers to restore the data to its original input size. Up sampling, or unpooling, effectively doubles the data's dimensions. In this work, nearest-neighbour linear interpolation is employed for this purpose. A deep architecture with multiple convolutional layers and matching up sampling layers is required to process complex turbulence data efficiently and preserve information with the least amount of loss.

After training every network in a complete scenario, we use the CAE's encoder to compress a given snapshot into a compressed form ct . This compressed snapshot is then used as input for a subsequent Recurrent Neural Network (RNN) to predict the following diminished representation $c(t+1)$. This predicted representation is subsequently backward-decoded into the initial input domain using the CAE's decoder unit.

5.3 Gated Recurrent Unit

As previously mentioned in Section I, we are also investigating the utilization of a gated recurrent unit (GRU) within an encoder-decoder architecture, treating it as a recurrent neural network (RNN) for tackling intricate sequence analysis and prediction tasks. The GRU represents an advanced type of RNN cell, employing gating mechanisms to regulate the inclusion of information into its ongoing cell state and to determine which previously acquired data should be discarded. This system effectively addresses the issues of vanishing and exploding gradients that are typically encountered when training RNNs with lengthy sequences. Equation (12) is given as

$$\lambda_t = \sigma \left(W^{(\lambda)} \cdot [h_t, c_t] \right) \quad (12)$$

And equation (13) as follows

$$\omega_t = \sigma \left(W^{(\omega)} \cdot [h_t, c_t] \right) \quad (13)$$

where ω_t represents the update gate vector at time step t and λ_t signifies the reset gate vector; The equivalent weight matrices $W^{(\lambda)}$ and $W^{(\omega)}$ are applied to a vector created by combining the hidden state vector h_t with the input vector c_t at time step t . An output range of 0 to 1 is guaranteed by the sigmoid activation function σ .

The output is utilised to calculate how much of the value of another vector element to retain through an element-wise vector multiplication. In particular, when determining the updated intermediate state $\hat{h}_t$, λ_t is multiplied element-wise by h_t to "reset" individual values of the previous state. The equation (14) is shown below

$$\hat{h}_t = \tanh(W \cdot [\lambda_t \odot h_t, c_t]) \quad (14)$$

where $\tanh$ is the hyperbolic tangent activation function defined as $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and W is a supplementary weight matrix. In (15), the element-wise Hadamard product is represented by $\odot$. Likewise, ω_t is multiplied element-wise by h_t and $(1 - \omega_t)$ by $\hat{h}_t$ to incorporate the latest input data into the modified cell state $h_{(t+1)}$, which is also the cell's output, as shown by equation (15)

$$h_{t+1} = (1 - \omega_t) \odot h_t + \omega_t \odot \hat{h}_t \quad (15)$$

In order to predict the next compressed snapshot, $\hat{c}_{t+1}$, after that, the output is passed through one last fully connected layer with a linear activation. As the encoder constructs the hidden state, which processes a sequence of compressed input snapshots, resulting in a hidden version of the input. The decoder receives this latent representation after which it forecasts the subsequent sequence's compressed picture, $\hat{c}_{t+1}$, using the data that has been encoded up to the previous time step. Once an external step is executed, the decoder proceeds in an auto-regressive manner. It forecasts the next output, $\hat{c}_{t+2}$, by calculating an updated latent representation using the earlier iteration's output, $\hat{c}_{t+1}$ as an input.

6 RESULTS AND DISCUSSION

It is necessary to provide some further information about the DNS data records before moving on to the training, validation, and testing phases. We used 2400 snapshots in total, broken down into three different sets. The primary set, exclusively used for training and referred to as the "training set," comprised 1000 snapshots. The second set, known as the "validation set," consisted of 500 snapshots, and it served the purpose of evaluating the model's performance and tuning hyperparameters during runtime to ensure an unbiased estimation of training. This validation step occurred immediately after training with each chosen set of hyperparameters, helping to address over- and under-fitting and find the ideal η for the next testing stage. The final 900 pictures were in the third set and was designated as the "test dataset." This dataset was independent, unseen, and not used in training. The final model's evaluation was based on this test data, offering an unbiased assessment on a completely new dataset. It's worth noting that we trained the Convolutional Autoencoder (CAE) and Gated Recurrent

Unit (GRU) on NVIDIA GeForce GTX 1060 and RTX 2080TI GPUs, respectively, while the Echo State Network (ESN) was trained on a CPU with 16GB of memory.

6.1 Convolutional Autoencoder Training

Initially, a 12-layer Convolutional Autoencoder (CAE) was created for both the encoder and decoder. To restore the original data dimensions of 320x60, the decoder included an extra cropping layer, which employed an activation of the sigmoid to guarantee a normalised output. Bayesian optimisation (BO) was used to optimise the network hyperparameters. A 5x5 kernel size was used for the convolutions, resulting in a larger receptive field. Regarding the shared latent representation of the encoder and decoder, a 40-element size was determined to be the most suitable for our application, yielding a 40-dimensional latent space. A Glorot uniform distribution was used to initialise all network weights. After acquiring the optimized parameters, we initiated the training of the CAE and performed validation on the model after each epoch. The training process typically converged around the 30-epoch mark, but we continued it until it met the early stopping criteria. Notably, both training and validation errors consistently decreased throughout, indicating the model's robustness when dealing with previously unknown validation information.

6.2 Training of the echo state network

We first obtained a pre-trained Convolutional Autoencoder (CAE) that yields compressed representations of the input data, and then we trained the Echo State Network (ESN) to predict the latent space temporal variations in convection flow. We used the same dataset and data splitting strategy as was previously mentioned. Minimising the mean squared error (MSE) between the forecasted and actual ground truth modes was our goal for ESN training. In order to evaluate training effectiveness, we also determined the mean square error (MSE) among the forecasted values and the initial heat flux from turbulent convection.

Table 1. Hyperparameters used for the encoder-decoder GRU training

Parameter	Search Range	Optimized Value
Kernel size	(1x1)-(5x5)	5x5
Latent vector size		40
Learning rate	0.0001-0.001	0.00058
Batch size		16

To achieve this, the predicted test modes were continuously fed into the decoder and gathered into ensembles. Moreover, we applied Bayesian optimisation (BO) to optimise the hyperparameters of the network. For this particular example, we optimised the regularisation keeping the other hyperparameters unchanged and adjusting parameter 'b'. The red curve in these figures represents the true (unknown and black-box) objective function. It is evident that the objective function is quite complex, and a grid-search approach would be ineffective if there is insufficient grid resolution to capture the ideal values. For our optimization, we utilized the mean squared error (MSE) as the cost function and aimed to maximize the negative of the MSE.

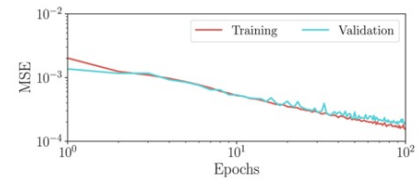


Figure 10. plot of MSE vs epochs

Table 2. Optimised parameters for the ESN that were acquired from BO. Here, the prior was initialised with five randomly selected points, the BO was iterated 50 times, and k=1

Reservoir Size	100-5000	2992
Spectral radius	0.90-0.99	0.97
Reservoir density	0.05-0.20	0.09
Scaling	True, False	False
Leakage Rate	0.5-0.9	0.50
Regularization parameter	0-600	4.89

Table 3. Hyperparameters used for the encoder–decoder GRU training

Parameter	Search Range	Optimized Value
Initial learning rate	0.006, 0.003, 0.001, 0.0006, 0.0003, 0.0001, 0.00006, 0.00003, 0.00001	0.001
Batch Size	32, 64, 128, 256, 512	128
Hidden state size	128, 256, 384, 512, 1024	512

7 CONCLUSION

In this study, we conducted a comprehensive analysis of a complex dataset generated through Direct Numerical Simulation (DNS) of two-dimensional turbulent convection. The dataset consisted of 2400 snapshots, and we divided it into three distinct sets: a training set with 1000 snapshots, a validation set with 500 snapshots, and a test dataset with 900 snapshots. This careful separation of data allowed us to train, validate, and test machine learning models in a rigorous and unbiased manner. We employed various machine learning techniques to analyze and predict the behavior of turbulent convection. The Convolutional Autoencoder (CAE) was used for data compression and dimensionality reduction, resulting in a 40-dimensional latent space. Bayesian optimization was utilized to fine-tune the CAE's hyperparameters, ensuring optimal performance. The training process for the CAE demonstrated robust convergence, with both training and validation errors consistently decreasing. Subsequently, we introduced the Echo State Network (ESN), which was trained to forecast temporal changes in convection flow within the latent space. The training objective was to minimize the mean squared error (MSE) between predicted modes and ground truth modes, as well as between predicted values and original turbulent convective heat flux. Bayesian optimization was employed once again to optimize ESN hyperparameters, achieving a highly effective model. In addition, we incorporated a Gated Recurrent Unit (GRU) in an encoder-decoder architecture to capture sequence dependencies within the data. The GRU was trained with carefully selected hyperparameters, resulting in a powerful model for sequence analysis and prediction. Overall, our approach demonstrated the effectiveness of machine learning in analyzing complex fluid dynamics data. By

training, validating, and testing these models on separate datasets, we ensured the reliability and generalizability of our results. This work paves the way for improved understanding and prediction of turbulent convection and holds promise for a wide range of applications in fluid dynamics and beyond.

References

- Abbate, J.A. (2024). *Laboratory-Theoretical Investigations of Rotating Convection in Planetary Interiors*. University of California, Los Angeles.
- Beintema, G., Corbetta, A., Biferale, L., Toschi, F. (2020) *Controlling Rayleigh–Bénard convection via reinforcement learning*. Journal of Turbulence, vol.21, no.9–10, pp.585–605.
- Boot, W. J. M., M. Madonia, AJ Aguirre Guzmán, R. P. J. Kunnen (2021). *Wall and bulk modes in confined turbulent rotating convection*.
- Cheng, H., Jiang, H., Chong, K. L., Zhou, Q., Liu, Y., Lu, Z. (2022). *The effect of surface roughness on the Lagrangian coherent structures in turbulent Rayleigh–Bénard convection*. Physics of Fluids, vol.24, no.11.
- Chong, K.L., Wagner, S., Kaczorowski, M., Shishkina, O., Xia, K.Q. (2018). *Effect of Prandtl number on heat transport enhancement in Rayleigh–Bénard convection under geometrical confinement*. Physical Review Fluids, vol.3, no.1, pp. 013501.
- Ecke, R.E., Shishkina, O. (2023). *Turbulent rotating rayleigh–bénard convection*. Annual Review of Fluid Mechanics, vol.55, pp.603–638.
- Favier, B., Purseed, J., Duchemin, L. (2019). *Rayleigh–Bénard convection with a melting boundary*. Journal of Fluid Mechanics, vol.858, pp.437–473.
- Fischer, P., Merzari, E., Min, M., Kerkemeier, S., Lan, Y.H., Phillips, M., Warburton, T. (2021). *Highly optimized full-core reactor simulations on Summit*. arXiv preprint arXiv:2110.01716.
- Horn, S., Schmid, P. J., Aurnou, J. M. (2022). *Unravelling the large-scale circulation modes in turbulent Rayleigh–Bénard convection (a)*. Europhysics Letters, vol.136, no.1, pp. 14003.
- Huang, M., Wang, Y., Bao, Y., He, X. (2022). *Heat transport and temperature boundary-layer profiles in closed turbulent Rayleigh–Bénard convection with slippery conducting surfaces*. Journal of Fluid Mechanics, vol.943, pp. A2.
- Hurtán, E., Monteiro, C., Jofre, M., Casals-Terré, J., Jofre, L. (2024). *Data-informed characterization of spatio-temporal scales in experiments of microconfined high-pressure transcritical turbulence*. Experimental Thermal and Fluid Science, vol.159, pp.111282.
- Kannan, V., Swaminathan, N., Davidson, P.A. (2024). *Effects of sidewall thermal condition on the formation of poloidal circulation in rotating three-dimensional convection*. Physics of Fluids, vol.36, no.3.

- Kashanj, S. (2025). *Experimental Study on the Role of Flow Large-Scale Circulating Structures in Poiseuille and Rayleigh-Bénard Convection*.
- Kurmyshev, E.V. and Lopez-Reyes, L.J. (2013). *Resonance instability of nonlinear vibrations of a string under harmonic heating*. Cybernetics and Physics, vol.2, no.3, pp.165-168.
- Liu, W. (2008). *Magnetized Ekman layer and Stewartson layer in a magnetized Taylor-Couette flow*. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, vol.77, no.5, pp. 056314.
- Liu, S., Jiang, L., Chong, K.L., Zhu, X., Wan, Z.H., Verzicco, R., Sun, C. (2020). *From Rayleigh-Bénard convection to porous-media convection: how porosity affects heat transfer and flow structure*. Journal of fluid mechanics, vol.895, no.A18.
- Madonia, M., Guzmán, A.J.A., Clercx, H.J., Kunnen, R.P. (2023). *Reynolds number scaling and energy spectra in geostrophic convection*. Journal of fluid mechanics, vol.962, pp. A36.
- Plotnikov, S.A. (2024). *Conditions Of Yakubovich Oscillatory For Nonlinear Systems Under Disturbances*. Cybernetics and Physics, vol.13, no.1, pp.57-61.
- Plumley, M., Julien, K. (2019). *Scaling laws in Rayleigh-Benard convection*. Earth and Space Science, vol.6, no.9, pp.603-638.
- Schneide, C., Vieweg, P.P., Schumacher, J., Padberg-Gehle, K. (2022). *Evolutionary clustering of Lagrangian trajectories in turbulent Rayleigh-Bénard convection flows*. Chaos: An Interdisciplinary Journal of Nonlinear Science, vol.32, no.1.
- Shishkina, Olga. (2021). *Rayleigh-Bénard convection: The container shape matters*. Physical Review Fluids, vol.6, no.9, pp.090502.
- Song, J., Shishkina, O., Zhu, X. (2023). *Scaling regimes in rapidly rotating thermal convection at extreme Rayleigh numbers*. arXiv preprint arXiv:2304.14854.
- Tagawa, T. (2023). *Numerical Analysis of Linear Traveling Wave in Rotating Rayleigh-Bénard Convection with an Adiabatic Sidewall*. Fluids, vol.8, no.3, pp. 96.
- Velmisov, P.A., Ankilov, A.V. (2021). *Mathematical modeling in problems about dynamics and stability of elastic elements of wing profiles*. Cybernetics and Physics, vol.10, no.3, pp.201-212.
- Vogt, T., Horn, S., Aurnou, J. M. (2021). *Oscillatory thermal-inertial flows in liquid metal rotating convection*. Journal of Fluid Mechanics, vol.911, pp. A5.
- Wagner, S., Shishkina, O. (2015). *Heat flux enhancement by regular surface roughness in turbulent thermal convection*. Journal of Fluid Mechanics, vol.763, pp.109-135.
- Wang, Q., Chong, K. L., Stevens, R. J., Verzicco, R., Lohse, D. (2020) *From zonal flow to convection rolls in Rayleigh-Bénard convection with free-slip plates*. Journal of fluid mechanics, vol.905, pp.A21.
- Xu, F., Zhang, L., Xia, K. Q. (2022). *Three-dimensional properties of the viscous boundary layer in turbulent Rayleigh-Bénard convection*. Journal of Fluid Mechanics, vol.947, pp. A15.
- Zhang, X., Van Gils, D. P., Horn, S., Wedi, M., Zwirner, L., Ahlers, G., Shishkina, O. (2020). *Boundary zonal flow in rotating turbulent Rayleigh-Bénard convection*. Physical review letters, vol.124, no.8, pp.084505.
- Zhang, X., Reiter, P., Shishkina, O., Ecke, R.E. (2023). *Wall modes and the transition to bulk convection in rotating Rayleigh-Bénard convection*. arXiv preprint arXiv:2310.19044.
- Zhang, S., Zhao, B., Liu, Y. (2024). *Coupling effect of plastic deformation and surface roughness on friction behaviors of rough line contact for ground metal surfaces*. Tribology International, vol.193, pp.109338.
- Zhu, X., Mathai, V., Stevens, R. J., Verzicco, R., Lohse, D. (2018). *Transition to the ultimate regime in two-dimensional Rayleigh-Bénard convection*. Physical review letters, vol.1, pp.109338.