Physics-constrained data-driven derivation of governing equations for convection

Christopher J. Wareing, Alasdair Roy, Steven M. Tobias

School of Mathematics, University of Leeds, Leeds LS2 9JT, UK. *C.J.Wareing@leeds.ac.uk*

Summary

We have used the Dedalus[1] framework v3.0 for spectrally solving differential equations in order to generate an extended time-series of two- and three-dimensional DNS data at various Rayleigh numbers for Rayleigh-Bénard convection and planar convective Couette flow. To this data, we have applied the Sparse Identification of Non-linear Dynamics (SINDy)[2] and the Sparse Physics-Informed Discovery of Empirical Relations (SPIDER)[3,4] data-driven machine-learning algorithms. Using the weak formulation of these algorithms in combination with mixed-integer optimisation[5] and physics-constrained libraries of possible terms, we have been able to converge the methods and recover the governing equations, constraints (i.e. the continuity condition) and boundary conditions of these convection problems purely from data[7].

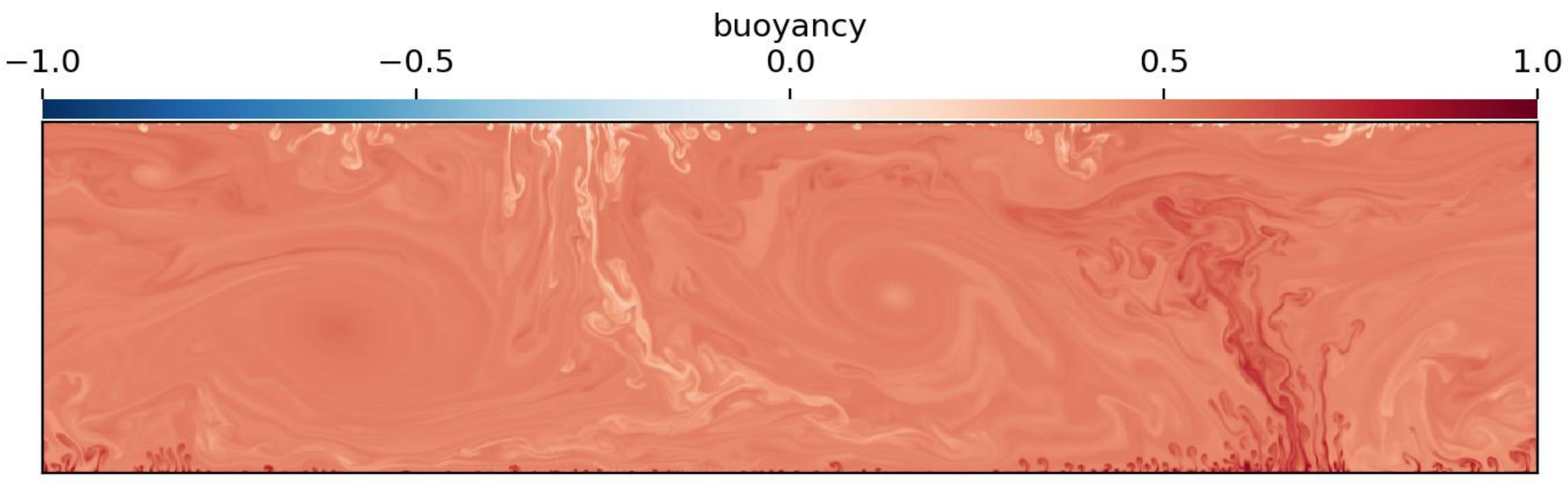
Physical Model & Governing Equations

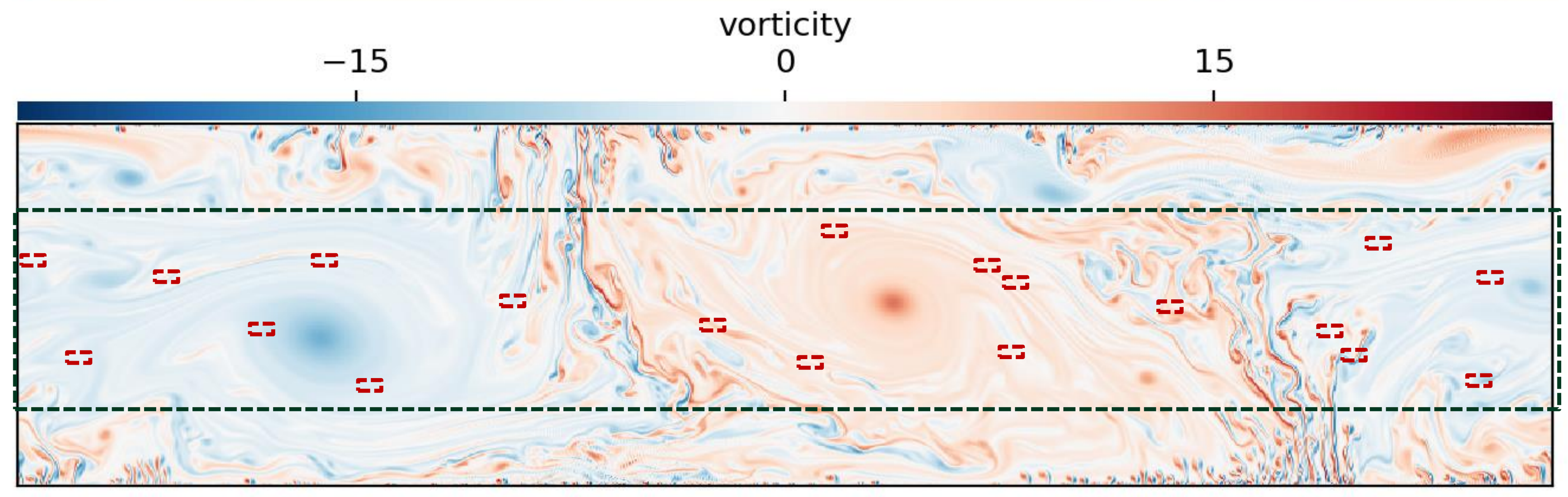
- 2D horizontally-periodic Rayleigh-Bénard convection.
- Non-dimensionalised using the box height and freefall time.
- Rayleigh numbers, $R = 10^6, 10^8, 10^{10}$ & 10^{12}. Prandtl number, $P = 1$.
- $R = 10^4, 10^5, 10^6$ & 10^7 in 3D.
- Stress-free boundary condition at top $z = 0$ and bottom $z = 1$.
- Boussinesq approximation.
- Temperature $T(z=0)=1$, $T(z=1)=0$.
- 2D and 3D Rayleigh-Bénard convection: $\boldsymbol{u}(z=0) = \boldsymbol{u}(z=1) = 0$.
- 2D planar convective Couette Flow: $u_x(z=0) = u_x(z=1) = 0$; $u_z(z=0) = -U_0$ & $u_z(z=1) = U_0$, where $U_0 = 0.5, 1.0$ at $R = 10^6, 10^8$ & 10^{10}.

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\,\boldsymbol{u} = -\boldsymbol{\nabla} p + T\,\hat{\boldsymbol{z}} + \sqrt{\frac{P}{R}}\nabla^2 \boldsymbol{u}\,, \qquad (1)$$

$$\frac{\partial T}{\partial t} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\,T = \frac{1}{\sqrt{R\,P}}\nabla^2 T\,, \qquad (2)$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0\,, \qquad (3)$$





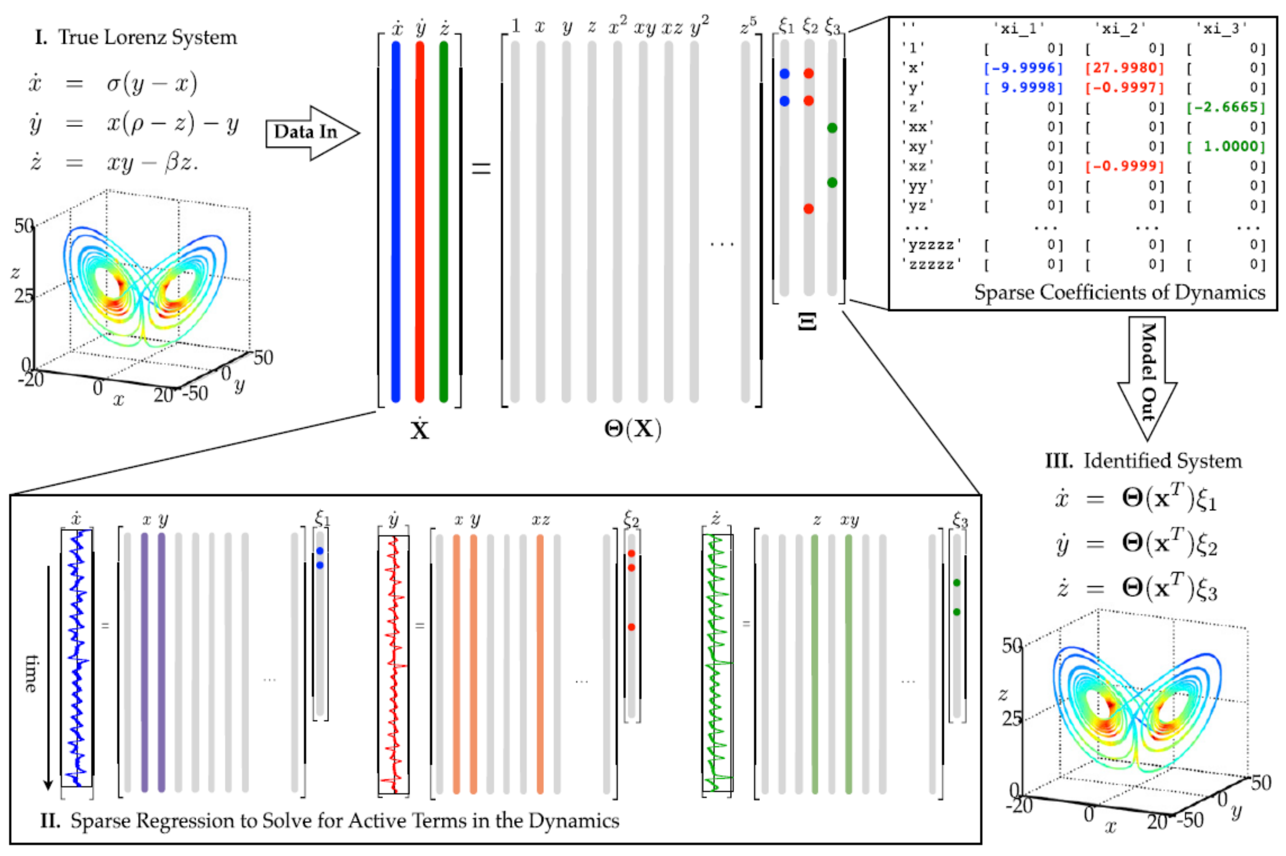
- Snapshot at t=48.5 of resolved 2D convection, R = 10^{10}.

Numerical technique

- Dedalus v3 flexible framework for solving PDEs using spectral methods[1].
- Open-source, widely-used, well supported.
- 1024 real space Fourier points (N_x) in the x direction, $L_x = 4$.
- 384 real space Chebyshev points (N_z) in the z direction, $L_z = 1$.
- RK443 - 3rd-order, 4-stage DIRK+ERK timestepping scheme.

SINDy

- Demonstrative overview:



Brunton et al., Fig.1 [2]

- Construct the complete library of 104 terms:-
  - Data fields: $u_x(x,z)$, $u_z(x,z)$, $p(x,z)$, $T(x,z)$ (4 library terms);
  - First order, *O(1)*, partial derivatives (8 terms);
  - Second order, *O(2)*, partial derivatives (12 terms);
  - u_x, u_z, p and T multiplied by every *O(1)* derivative (32 terms);
  - u_x, u_z, p and T multiplied by every *O(2)* derivative (48 terms).
- Physically constrain the library:-
  - Remove terms linked by incompressibility constraint: $\partial u_x/\partial x = -\partial u_z/\partial z$.
  - Enforce vector coefficient symmetry between four equations.
  - Diffusion via Laplacian operator only, limiting possible *O(2)* derivatives.
- Optimisation
  - Exact formulation of SINDy using mixed-integer optimisation[5].
  - Gurobi modern optimisation solver [6].

SINDy results

- Applying SINDy in this manner can robustly identify equations (1) and (2) up to $R = 10^{10}$. At $R = 10^{12}$, the diffusion term is not easily recovered.

```
(U)' = 0.00099995 U_22 + -0.99981655 P_1 + 0.00099995 U_11 + -0.99979113 WU_2 + -0.99979113 UU_1
(W)' = 0.99981780 T + -0.99981655 P_2 + 0.00099995 W_22 + 0.00099995 W_11 + -0.99979113 WW_2 + -0.99979113 UW_1
(P)' = 0.00000000
(T)' = 0.00100058 T_22 + 0.00100058 T_11 + -0.99920030 WT_2 + -0.99920030 UT_1
```

- Unconstrained fitting is extremely sensitive to data selection, the random initialisation and the size and number of subdomains, leading to difficulties in ensuring a converged machine-learning process.

Conclusions and future work

- SPIDER starts with more physical intuition to define a library.
- Differences in the performance of the algorithms – SPIDER performs better at high R – can be influenced by normalisation and optimisation choices.
- In future, we will apply these methods to the recovery of subgrid-scale turbulence closure models.

SPIDER

- SPIDER is similar to SINDy, in that both perform a sparse regression problem on a weak formulation of PDEs, but different in other ways:-
  - Physical assumption of smoothness, locality and symmetry;
  - Scalar ***and vector*** equation recovery;
  - Construction of library is not a brute force combinatorial problem. Instead, library definition comes from the combination of available fields with the differential operators $\partial/\partial t$ ($\equiv \partial_t$) and $\boldsymbol{\nabla}$.
- Scalar library:-

$$\mathcal{L}_0 = \{1, p, T, \nabla\cdot\boldsymbol{u}, \partial_t p, p^2, p^3, \partial_t T, T^2, T^3, \boldsymbol{u}^2, \boldsymbol{u}\cdot\nabla p, \boldsymbol{u}\cdot\nabla T, \nabla^2 p, \nabla^2 T, p\partial_t p, T\partial_t p, p\partial_t T, T\partial_t T, \partial_t^2 p, \partial_t^2 T, p(\nabla\cdot\boldsymbol{u}), T(\nabla\cdot\boldsymbol{u}), \boldsymbol{u}^2 p, \boldsymbol{u}^2 T, \boldsymbol{u}\cdot\partial_t\boldsymbol{u}\}$$

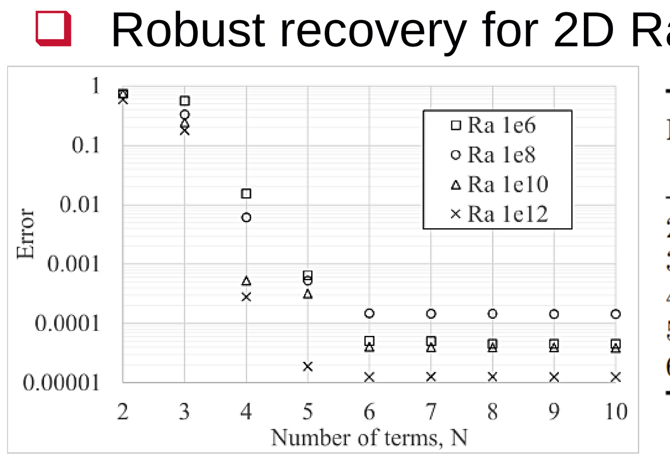
- Vector library:-

$$\mathcal{L}_1 = \{\boldsymbol{u}, \partial_t\boldsymbol{u}, \nabla p, \nabla T, p\boldsymbol{u}, T\boldsymbol{u}, (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}, \nabla^2\boldsymbol{u}, \partial_t^2\boldsymbol{u}, u^2\boldsymbol{u}, p^2\boldsymbol{u}, T^2\boldsymbol{u}, \partial_t\nabla p, \partial_t\nabla T, p\nabla p, p\nabla T, T\nabla P, T\nabla T, \boldsymbol{u}(\nabla\cdot\boldsymbol{u}), \boldsymbol{u}\cdot(\nabla\boldsymbol{u}), \nabla(\nabla\cdot\boldsymbol{u}), p\partial_t\boldsymbol{u}, T\partial_t\boldsymbol{u}, \boldsymbol{u}\partial_t p, \boldsymbol{u}\partial_t T\}.$$

- Weak form of the PDEs makes the regression more robust.
- Derivatives are shifted from the data to a weight function via integration by parts.
- M spatio-temporal domains of size H_x x H_z x H_t, constructs matrix.
- Iterative greedy algorithm identifies multiple term relations with residual errors.
- Selecting a final relation from the comparison of reduced sequences formed by repeating the process and dropping one term each time, is based on the choice between simplicity (i.e. number of terms N) and the accuracy quantified by the residual error.

SPIDER results

- Robust recovery for 2D Rayleigh-Benard convection, in detail at $R = 10^{10}$.



N	Error	Coefficients of terms[b] $\frac{\partial \boldsymbol{u}}{\partial t}$	$(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}$	∇p	$T\,\hat{\boldsymbol{z}}$	$\nabla^2\boldsymbol{u}$	$\frac{\partial\nabla p}{\partial t}$
2	0.740	-	-	0.579[c]	-1[c]	-	-
3	0.242	-	0.882	0.960	-1	-	-
4	5.261×10^{-4}	1	0.999	0.999	-0.999	-	-
5	3.219×10^{-4}	0.999	1	1	-0.999	-1.023×10^{-5}	-
6	4.048×10^{-5}	1	1	1	-0.999	-1.006×10^{-5}	-5.112×10^{-4}

- Spider is also able to identify the continuity constraint, $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$.
- Define normal $\boldsymbol{n}$ and tangential libraries, using $P_\perp = \boldsymbol{nn}$ and $P_\parallel = \mathbb{1} - \boldsymbol{nn}$, as:-

$$\mathcal{L}_\parallel = \{P_\parallel\,\boldsymbol{u},\, P_\parallel\nabla(\boldsymbol{u}\cdot\boldsymbol{n}),\, P_\parallel(\boldsymbol{n}\cdot\nabla)\boldsymbol{u}\},$$

$$\mathcal{L}_\perp = \{\boldsymbol{n}\cdot\boldsymbol{u},\, 1,\, \boldsymbol{n}\cdot\nabla(\boldsymbol{u}\cdot\boldsymbol{n}),\, \boldsymbol{n}\cdot(\boldsymbol{n}\cdot\nabla)\boldsymbol{u}\},$$

- With these libraries, SPIDER can successfully retrieve boundary conditions.

References:

[1] Burns, Vasil, Oishi, Lecoanet & Brown, *Physical Rev. Res.* **2** (2) 023068, 2020.
[2] Brunton, Proctor & Kutz, *PNAS* **113 (15)**, 3932-3937, 2016.
[3] Golden et al., *Sci. Adv.*, **9**, eabq6120, 2023.
[4] Gurevich et al., ArXiv: 2105.00048v3
[5] Bertsimas & Gurnee, *Nonlinear Dyn.*, 111, 6585-6604, 2023.
[6] Gurobi Optimization LLC, Gurobi Optimiser Reference Manual 2023; v. 9.1.0
[7] Wareing, Roy & Tobias, *GAFD special iss.*, in prep.

Funded by the European Union

European Research Council
Established by the European Commission

Acknowledgements: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. D5S-DLV-786780). The calculations for this paper were performed on the University of Leeds ARC4 facility, hosted and enabled through the ARC HPC resources and support team at the University of Leeds, to whom we extend our grateful thanks.

UNIVERSITY OF LEEDS