Asymptotic Self-Similar Blow-Up Profile for Three-Dimensional Axisymmetric Euler Equations Using Neural Networks

Y. Wang,[1] C.-Y. Lai,[1] J. Gómez-Serrano,[2,3,4] and T. Buckmaster[5,6,*]

[1]*Department of Geosciences, Princeton University, Princeton, New Jersey 08544, USA*
[2]*Department of Mathematics, Brown University, Kassar House, 151 Thayer Street, Providence, Rhode Island 02912, USA*
[3]*Departament de Matemàtiques i Informàtica, Universitat de Barcelona,*
Gran Via de les Corts Catalanes, 585, 08007, Barcelona, Spain
[4]*Centre de Recerca Matemàtica, Edifici C, Campus Bellaterra, 08193 Bellaterra, Spain*
[5]*School of Mathematics, Institute for Advanced Study, Princeton, New Jersey 08540, USA*
[6]*Department of Mathematics, Princeton University, Princeton, New Jersey 08544, USA*

(Received 24 June 2022; revised 30 December 2022; accepted 25 April 2023; published 16 June 2023)

Whether there exist finite-time blow-up solutions for the 2D Boussinesq and the 3D Euler equations are of fundamental importance to the field of fluid mechanics. We develop a new numerical framework, employing physics-informed neural networks, that discover, for the first time, a smooth self-similar blow-up profile for both equations. The solution itself could form the basis of a future computer-assisted proof of blow-up for both equations. In addition, we demonstrate physics-informed neural networks could be successfully applied to find unstable self-similar solutions to fluid equations by constructing the first example of an unstable self-similar solution to the Córdoba-Córdoba-Fontelos equation. We show that our numerical framework is both robust and adaptable to various other equations.

DOI: 10.1103/PhysRevLett.130.244002

A celebrated open question in fluids is whether or not from smooth initial data the 3D Euler equations may develop finite-time singularities (the inviscid analog of the Navier-Stokes millennium-prize problem). For nonsmooth $C^{1,\alpha}$ initial data with $0 < \alpha \ll 1$, finite-time self-similar blow-up was proven in the groundbreaking work of Elgindi [1] (cf. [2]). The question of finite-time blow-up from *smooth* initial data remains unresolved.

In the presence of a cylindrical boundary, Luo and Hou [3] (cf. Ref. [4]) performed compelling numerical simulations proposing a scenario—sharing similarities with Pumir and Siggia [5] (cf. Refs. [6,7])—for finite-time blow-up of the axisymmetric 3D Euler equations. They simulated the time-dependent problem and observed a dramatic growth in the maximum of vorticity (by a factor of 3×10^8), strongly suggesting formation of a singularity. The work is also suggestive of asymptotic self-similarity.

To confirm the existence of the finite-time singularity in the Luo-Hou scenario, and find its self-similar structure, we need to solve the self-similar equations associated with the axisymmetric 3D Euler equations in the local coordinates near the singularity, which poses an extreme challenge to classical numerical methods as explained later. In this Letter, we develop a new numerical strategy, based on physics-informed neural networks (PINNs) that can solve the self-similar equations in a simple and robust way. This new method allows us, for the first time, to find the smooth asymptotic self-similar blow-up profile for the Luo-Hou scenario. To the best of our knowledge, the solution is also the first truly multidimensional smooth backward self-similar profile for an equation from fluid mechanics.

Singularity formation for 3D Euler equations with a cylindrical boundary is intrinsically linked to the same problem for the 2D Bousinessq equations (cf. Refs. [4,8–10]), another fundamental question in fluid mechanics, mentioned in Ref. [11]. The mechanism for blow-up for the two equations is believed to be identical. The 2D Boussinesq equations take the form

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = (0, \theta),$$
$$\partial_t \theta + \mathbf{u} \cdot \nabla \theta = 0, \qquad \operatorname{div} \mathbf{u} = 0, \tag{1}$$

where the 2D vector $\mathbf{u}(\mathbf{x}, t)$ is the velocity and the scalar $\theta(\mathbf{x}, t)$ is the temperature. We consider the spatial variable $\mathbf{x} = (x_1, x_2)$ to be taken on the half plane $x_2 \geq 0$ and impose a nonpenetration boundary condition at $x_2 = 0$ (x_1 axis), namely, $u_2(x_1, 0) = 0$.

To search for singularity formation for the Boussinesq equations (1), we look for *backward* [12] self-similar solutions of the form $\mathbf{u} = (1-t)^{\lambda}\mathbf{U}(\mathbf{y})$ and $\theta = (1-t)^{-1+\lambda}\Theta(\mathbf{y})$, where we define the self-similar coordinates as $\mathbf{y} = (y_1, y_2) = (x_1, x_2)/(1-t)^{1+\lambda}$, with $\lambda > -1$ yet to be determined. Under such a self-similar ansatz, the Eq. (1) become

$$-\lambda \mathbf{U} + [(1+\lambda)\mathbf{y} + \mathbf{U}] \cdot \nabla \mathbf{U} + \nabla P = (0, \Theta),$$
$$(1-\lambda)\Theta + [(1+\lambda)\mathbf{y} + \mathbf{U}] \cdot \nabla \Theta = 0, \qquad \operatorname{div} \mathbf{U} = 0. \tag{2}$$

The corresponding solution is expected to have infinite energy [13]; however, we impose that the solution to Eq. (2) has mild growth at infinity (see cost functions in Supplemental Material) [14] which is an essential requirement for such a solution to be *cut off* to produce an asymptotic self-similar solution with a finite energy [16].

Setting $\Omega = \mathrm{curl}\,\mathbf{U} = \partial_{y_1} U_2 - \partial_{y_2} U_1$, $\Phi = \partial_{y_1}\Theta$, and $\Psi = \partial_{y_2}\Theta$, we rewrite Eq. (2) in vorticity form:

$$\begin{aligned} \Omega + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Omega &= \Phi, \\ (2 + \partial_{y_1} U_1)\Phi + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Phi &= -\partial_{y_1} U_2 \Psi, \\ (2 + \partial_{y_2} U_2)\Psi + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Psi &= -\partial_{y_2} U_1 \Phi, \\ \mathrm{div}\;\mathbf{U} &= 0. \end{aligned} \tag{3}$$

To help find the solution, we impose the following symmetries: (U_1, Φ, Ω) are odd and (U_2, Ψ) are even in the y_1 direction. In addition, we impose $U_2(y_1, 0) = 0$ (the nonpenetration boundary condition). To guarantee the uniqueness of the solution (removing scaling symmetry), we constrain $\partial_{y_1}\Omega(0,0) = -1$. Finally, to rule out extraneous solutions, we impose that $\nabla\mathbf{U}$, Φ, and Ψ all vanish at infinity.

To describe the Luo-Hou scenario for the 3D Euler blowup in the presence of boundary, we write the axisymmetric 3D Euler equations:

$$\begin{aligned} (\partial_t + u_r\partial_r + u_3\partial_{x_3})\left(\frac{\omega_\theta}{r}\right) &= \frac{1}{r^4}\partial_{x_3}(ru_\theta)^2, \\ (\partial_t + u_r\partial_r + u_3\partial_{x_3})(ru_\theta) &= 0, \end{aligned} \tag{4}$$

where (u_r, u_θ, u_3) is the velocity in cylindrical coordinates and ω_θ is the angular component of the vorticity. We introduce a cylindrical boundary at $r = 1$, and restrict to the exterior domain $\{(r, x_3) \in r \geq 1, x_3 \in \mathbb{R}\}$. By imposing the self-similar ansatz $(u_{x_3}, u_r) = (1-t)^\lambda \mathbf{U}(\mathbf{y}, s)$, $\omega_\theta = (1-t)^{-1}\Omega(\mathbf{y}, s)$, $\partial_r(ru_\theta)^2 = (1-t)^{-2}\Psi(\mathbf{y}, s)$, and $\partial_{x_3}(ru_\theta)^2 = (1-t)^{-2}\Phi(\mathbf{y}, s)$ for self-similar coordinates $\mathbf{y} = (y_1, y_2) = (x_3, r-1)/(1-t)^{1+\lambda}$ and $s = -\log(1-t)$, the 3D Euler equation (4) becomes

$$\begin{aligned} (\partial_s + \Omega) + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Omega &= \Phi + \mathcal{E}_1, \\ (\partial_s + 2 + \partial_{y_1} U_1)\Phi + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Phi &= -\partial_{y_1} U_2 \Psi, \\ (\partial_s + 2 + \partial_{y_2} U_2)\Psi + [(1+\lambda)\mathbf{y} + \mathbf{U}]\cdot\nabla\Psi &= -\partial_{y_2} U_1 \Phi, \\ \mathrm{div}\;\mathbf{U} &= \mathcal{E}_2, \end{aligned} \tag{5}$$

where the errors $\mathcal{E}_1$ and $\mathcal{E}_2$ are given by the expressions

$$\begin{aligned} E_1 &= -y_2(y_2 e^{-(1+\lambda)s} + 2)(y_2^2 e^{-2(1+\lambda)s} + 2y_2 e^{-(1+\lambda)s} + 2)\Phi e^{-(1+\lambda)s}/(1 + y_2 e^{-(1+\lambda)s})^4 \\ E_2 &= -U_2 e^{-(1+\lambda)s}/(1 + y_2 e^{-(1+\lambda)s}) \end{aligned}$$

We look for solutions which are asymptotically self-similar: i.e., in self-similar coordinates they converge to a stationary state as $s \to \infty$. For such solutions, at any fixed $\mathbf{y}$, we have $\mathcal{E}_1, \mathcal{E}_2 = O(e^{-(1+\lambda)s})$ and thus the errors decay exponentially fast in self-similar coordinates assuming that $\lambda > -1$. Thus, the self-similar equations (5) for Euler converge to Bousinessq (3) as $s \to \infty$. Namely, the self-similar solution for Boussinesq is identical to the asymptotic self-similar blow-up profile to Euler with cylindrical boundary.

A key difficulty of solving Eq. (3) lies in the unknown parameter λ that needs to be solved simultaneously. We search for smooth nontrivial solutions to Eq. (3), which exist for discrete λ values. This problem is extremely challenging using classical evolution (time-dependent) based numerical methods (cf. Refs. [3,4,17–21]).

Physics-informed neural networks were recently developed [22,23] as a new class of numerical solver for partial differential equations (PDEs) and have been widely used in science and engineering [24]. In PINNs, neural networks approximate the solution to a PDE by searching for a solution in a continuous domain that approximately satisfies the physics constraints (e.g., equations and solution constraints). PINNs have been successfully used to solve not only forward problems but also inverse problems (e.g., identifying the Reynolds number from a given flow and the Navier-Stokes equation [22]), demonstrating the capacity of PINNs to invert for unknown parameters in the governing equations. Here, we use a PINN to find not only the self-similar solution profile but also the unknown self-similarity exponent λ. To guarantee the success of PINN, it is critical to understand the key symmetries of the problem, its spurious solutions, as well as intuition of the qualitative properties of the solution (e.g., its geometry and asymptotics).

To find the self-similar solution for the Bousinessq equation, we represent each of U_1, U_2, Ω, Φ, or Ψ in the Bousinessq equations (3) by an individual fully connected neural network with y_1 and y_2 as its inputs. We use 6 hidden layers with 30 units in each hidden layer for each network and use the hyperbolic tangent function tanh as the activation function. We impose the symmetry of each variable $(U_1, U_2, \Omega, \Phi, \Psi)$ by constructing the function form $q_{\mathrm{odd}} = [\mathrm{NN}_q(y_1, y_2) - \mathrm{NN}_q(-y_1, y_2)]/2$ and $q_{\mathrm{even}} = [\mathrm{NN}_q(y_1, y_2) + \mathrm{NN}_q(-y_1, y_2)]/2$, where NN_q is the neural network created for the variable q.

To train the neural network, we need a cost function and an optimization algorithm. For PINNs, the cost function is composed of two types of loss. The first is the *condition loss*, which evaluates the residue of the solution condition, where the residue here is defined as the difference between the neural network approximated condition and the true solution condition. The condition loss can be written as

$$\mathrm{loss}_c^{(j)} = \frac{1}{N_c^{(j)}} \sum_{i=1}^{N_c^{(j)}} g_{(j)}^2(\mathbf{y}_i, \hat{q}(\mathbf{y}_i)), \tag{6}$$

where $g_{(j)}(\mathbf{y}_i, \hat{q}(\mathbf{y}_i))$ indicates the residue of the jth boundary condition at the ith position $\mathbf{y}_i = (y_1, y_2)_i$ and $\hat{q}(\mathbf{y}_i)$ indicates the neural network prediction of the variable q. The parameter $N_c^{(j)}$ indicates the total number of points used for evaluating the jth boundary condition.

The second type of loss is known as the *equation loss*, which evaluates the residue of the governing equation averaged over a set of collocation points over the domain. The residue of the equation $f_{(k)}$ is defined as the error in the equation calculated with the neural network predictions. The equation loss can be written as

$$\mathrm{loss}_f^{(k)} = \frac{1}{N_f^{(k)}} \sum_{i=1}^{N_f^{(k)}} f_{(k)}^2(\mathbf{y}_i, \hat{q}(\mathbf{y}_i)), \tag{7}$$

where $f_{(k)}(\mathbf{y}_\mathrm{i}, \hat{q}(\mathbf{y}_i))$ indicates the residue of the kth equation evaluated at the ith collocation point. The parameter $N_f^{(k)}$ denotes the total number of collocation points used for the kth equation. The residues of the boundary conditions $g_{(j)}$ and equations $f_{(k)}$ involved in the cost function are listed in the Supplemental Material [14].

We stress that all equations are local, which is an advantage of our method versus alternate methods that require a careful consideration of nonlocality in infinite domains. In our implementation, to approximate an infinite domain, we introduce the coordinates $z = (z_1, z_2) = (\sinh^{-1}(y_1), \sinh^{-1}(y_2))$ [see Fig. 1(a)] and consider a domain $z \in [-30, 30]^2$. In the y coordinates, this corresponds to a domain $\approx[-5 \times 10^{12}, 5 \times 10^{12}]^2$. The equations written in z coordinates are given in the Supplemental Material [14].

The equations and conditions provided so far can only find the unique solution to Eq. (3) for a specified λ. Figure 1 in the Supplemental Material shows the PINN solution to Eq. (3) for $\lambda = 3$ [14]. The large equation residue at the origin indicates the nonsmoothness of the solution at the origin (see Supplemental Material). To search for the right λ that guarantees the smoothness of solutions, i.e., avoiding the local peaks in the equation residue, we impose the *smoothness constraint* to penalize the gradient of equation residues around the nonsmooth point (origin).

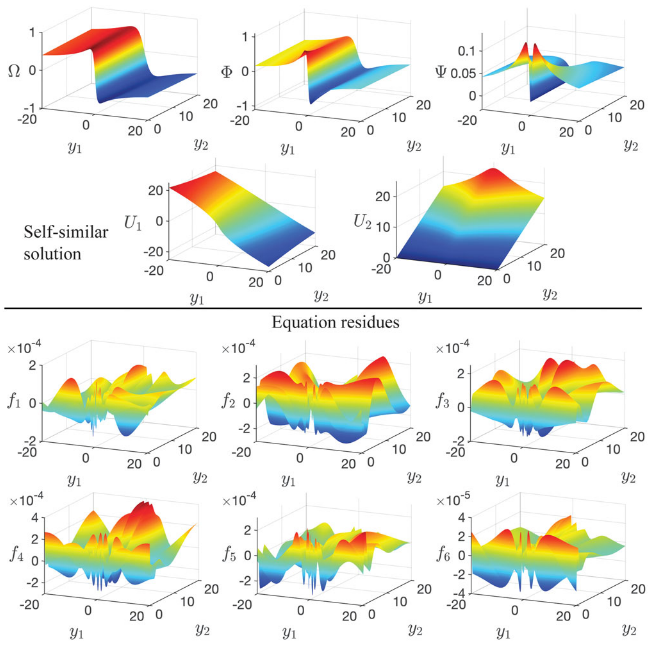

FIG. 1. Solution for Eq. (3) found by PINN. f_i indicate the residues, which are of 5 orders of magnitude smaller than the solution. The inferred value of λ is 1.917 ± 0.002 after systematic test (see Supplemental Material [14]).

$$\mathrm{loss}_s^{(k)} = \frac{1}{N_s^{(k)}} \sum_{i=1}^{N_s^{(k)}} |\nabla f_{(k)}(\mathbf{y}_i, \bar{q}(\mathbf{y}_i))|^2, \tag{8}$$

where $N_s^{(k)}$ indicates the total number of collocation points close to the origin. The sum of Eqs. (6)–(8) defines the final cost function,

$$J(\mathbf{y}, \mathbf{w}) = \frac{1}{n_b} \sum_{j=1}^{n_b} \mathrm{loss}_c^{(j)} + \gamma \left(\frac{1}{n_e} \sum_{k=1}^{n_e} \mathrm{loss}_f^{(k)} + \frac{1}{n_e} \sum_{k=1}^{n_e} \mathrm{loss}_s^{(k)} \right),$$

where $n_b = 8$ and $n_e = 6$ are the total number of solution conditions and governing equations used (see Supplemental Material). The constant γ is a hyperparameter of PINNs, known as the *equation weight* [25], which balances the contribution of the condition loss and equation loss in the final cost function $J(\mathbf{y}, \mathbf{w})$. For Boussinesq, we choose $\gamma = 0.1$ for optimal training performance.

The common optimization methods used for PINN training are Adam [26] and L-BFGS [27]. Despite the fact that no optimization method guarantees the convergence to a global minimum, our empirical experience, consistent with prior studies [28], shows that Adam performs better at avoiding local minima, while L-BFGS has a faster convergence rate throughout the training. Thus, we use Adam first for 100 000 iterations and then L-BFGS for 250 000 iterations to search for the self-similar solution for the Bousinessq equations. Figure 1(b) shows the convergence

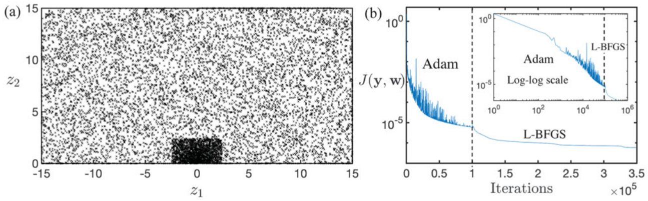

FIG. 2. (a) Spatial distribution of the collocation points. Here $(z_1,\ z_2)$ are the rescaled coordinates: $y_1 = \sinh(z_1)$ and $y_2 = \sinh(z_2)$. 10 000 collocation points in total are used for training. (b) Decrease of the total loss over the training iterations. The inset shows the loss curve in a log-log scale.

of the cost function $J(\mathbf{y}, \mathbf{w})$ throughout the training iterations.

Figure 2 shows the approximate solutions to Eq. (3), along with their corresponding equation residues. The equation residues are approximately 5 orders of magnitude smaller than that of the solution found. With the smoothness constraint Eq. (8) the inferred exponent for the smooth solution is $\lambda \approx 1.917$. In the Supplemental Material we show the robustness of the PINN prediction with different random initialization and normalization condition [14]. We also demonstrate convergence of the inferred λ with domain size. The solutions found by the PINN are in agreement with the asymptotics of the time-dependent solutions found by Luo and Hou [3,4]. Extrapolating from Ref. [4], the work is suggestive of a self-similarity exponent of $\lambda \approx 1.9$, in agreement with the exponent found by the PINN. Similarly to Luo and Hou, the trajectories corresponding to the self-similar velocity follow the geometry of a hyperbolic point at the origin.

The spatial distribution of the collocation points plays a critical role for the success of PINN training. To guide the neural network to find the correct self-similar solution for Bousinessq (3), we train the neural network to prioritize the equation constraints around the origin. Toward this goal, we divide the domain into two regions, one close to the origin and one far; in each region the collocation points are uniformly distributed. We increase the number of collocation points surrounding the origin. Otherwise, the neural network prediction would likely be trapped in a local minimum during the training.

The PINN-based scheme for finding self-similar blow-up offers advantages in terms of both universality and efficiency. For universality, the above PINN scheme can be generally applied to solving various self-similar equations without the requirement of prior knowledge of specific structure. For efficiency, the smooth self-similar solution was in fact found by PINN throughout one single training. There is no continuation scheme or time evolution required for the training, largely reducing the computational cost of the method. An additional major advantage of the PINN scheme, as we will later demonstrate, is its ability to find *unstable* self-similar solutions which would be incredibly difficult, if not impossible, to find via traditional methods.

To validate our approach, we compare self-similar solutions obtained to known results in the literature (Supplemental Material [14]). We apply the PINN scheme to find nonsmooth solutions to the Boussinesq equation, which are in agreement with the explicit approximate solutions of Chen and Hou [10] (see also Sec. 1.5 of the Supplemental Material).

One of the simplest PDEs exhibiting self-similar blow-up is the 1D Burgers equation, which can be solved analytically. The equation provides an excellent sandbox to test and refine the PINN. Section 2 of the Supplemental Material shows that the PINN scheme can find stable, unstable, and nonsmooth self-similar solutions to the Burgers equation [14]. A common numerical strategy to finding self-similar solutions is to introduce time dependence into the problem: while this is straightforward in the stable case, instabilities in the unstable case make finding unstable self-similar profiles comparatively more difficult, if not impossible. The PINN scheme does not suffer this drawback and thus presents itself as a great method for finding unstable smooth self-similar solutions. This latter fact will be reinforced below where we demonstrate that the PINN is successful in finding new unstable self-similar solutions that have applicability to an important open problem in mathematical fluid dynamics.

The generalized De Gregorio equation [29] is given by

$$\omega_t + au\omega_x = \omega u_x, \quad \text{where } u = \int_0^x H\omega = \Lambda^{-1}\omega,$$

and H is the Hilbert transform. The equation is a generalization of the De Gregorio equation ($a = 1$) [30] and has been proposed as a one-dimensional model for an equation for which there is nontrivial interaction of advection and vortex stretching (modeling behavior of the 3D Euler equations).

The case $a = 0$, in the absence of advection, is known in the literature as the Constantin-Lax-Majda equation. In this simple case, exact self-similar blow-up solutions can be constructed [31]. The case $a = -1$ [known as the Córdoba-Córdoba-Fontelos (CCF) model] was proposed as a model of the surface quasigeostrophic equation and it also develops finite-time singularities [32]. In the case $a < 0$, advection and vortex stretching work in conjunction leading to finite-time singularities [33]. The case $a > 0$ leads to the competition of the two terms. By a clever expansion in a, smooth self-similar profiles were constructed in Ref. [34] for small, positive a, leading to finite-time blow-up. Via a computer-assisted proof, Chen, Hou, and Huang [35] proved blow-up for the De Gregorio equation ($a = 1$).

Lushnikov *et al.* [36] represent the most thorough numerical study of the generalized De Gregorio equation to date. In Ref. [36], self-similar solutions were found in the

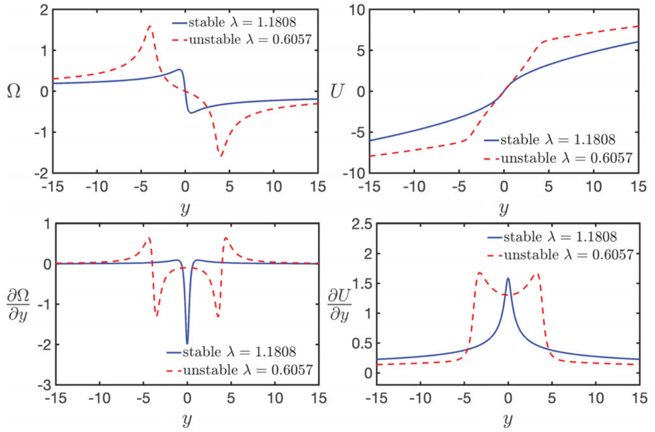

FIG. 3. Comparison between stable and unstable solutions with associated λ for CCF equation.

whole range $a \in [-1, 1]$ and beyond. We use their reported parameters as benchmarks for our results. In Ref. [37], we show that PINN can accurately reproduce the findings of Ref. [36].

Returning to the specific case of CCF ($a = -1$), an interesting question is to add fractional dissipation $(-\Delta)^{\alpha/2}$ and ask for which values of α do singularities occur. Blow-up is known to occur for $0 \leq \alpha < \frac{1}{2}$, whereas for $\alpha \geq 1$ the problem is global well posed [38–40]. The behavior in the range $\frac{1}{2} \leq \alpha < 1$ remains an important open problem.

Assuming the ansatz $\omega = (1-t)^{-1}\Omega(x/(1-t)^{1+\lambda}, s)$ for $s = -\log(1-t)$, then in the self-similar evolution equation for Ω, the dissipative term takes the form $e^{[(1+\lambda)\alpha-1]s}(-\Delta)^{\alpha/2}\Omega$. Analogous to how self-similar Boussinesq solutions can be used to construct asymptotic self-similar solutions to Euler with boundary, self-similar inviscid CCF solutions satisfying the condition $(1+\lambda)\alpha - 1 < 0$ may be employed to construct asymptotic self-similar solutions to dissipative CCF. Since $\lambda \approx 1.18078$ for the stable self-similar solution to inviscid CCF, such a solution is ill suited to prove blow-up in the parameter range $\frac{1}{2} \leq \alpha < 1$. Motivated by known work on the Burgers equation [41,42] and compressible Euler [43,44], one could conjecture the existence of a discrete hierarchy of unstable solutions with decreasing λ. By windowing the parameter λ, including additional derivatives of the governing equation in our residues, and using the constraint $\Omega(0.5) = 0.05$ to renormalize, the PINN discovers an unstable self-similar solution corresponding to $\lambda \approx 0.60573$ (see Fig. 3). Such a solution would allow us to prove blow-up for dissipative CCF for the range $\alpha < 1/(1+\lambda) \approx 0.61$. Moreover, such a result is suggestive of a possible strategy of addressing the Navier-Stokes millennium prize [45], i.e., via unstable self-similar solutions to 3D Euler (the same strategy has proved successful for dissipative Burgers and compressible Navier-Stokes [44,46,47]). One expects that the PINN may be adapted to find higher order unstable solutions to CCF as well as unstable solutions to the Boussinesq equation—this is a subject of future work.

The authors were supported by the NSF Grants No. DMS-1900149 and No. DMS-1929284, ERC Grant No. 852741, the Spanish State Research Agency, through the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (CEX2020-001084-M), Dean for Research Fund at Princeton University, the Schmidt DataX Fund at Princeton University, a Simons Foundation Mathematical and Physical Sciences Collaborative Grant, and grant from the Institute for Advanced Study. The simulations presented in this Letter were performed using the Princeton Research Computing resources at Princeton University and School of Natural Sciences Computing resources at the Institute for Advanced Study.

[*] Author to whom correspondence should be addressed. tbuckmaster@math.ias.edu

[1] T. Elgindi, Ann. Math. **194**, 647 (2021).

[2] T. Elgindi, T.-E. Ghoul, and N. Masmoudi, arXiv:1910.14071.

[3] G. Luo and T. Y. Hou, Proc. Natl. Acad. Sci. U.S.A. **111**, 12968 (2014).

[4] G. Luo and T. Y. Hou, Multiscale Model. Simul. **12**, 1722 (2014).

[5] A. Pumir and E. D. Siggia, Phys. Rev. Lett. **68**, 1511 (1992).

[6] S. Childress, Phys. Fluids **30**, 944 (1987).

[7] A. Pumir and E. D. Siggia, Phys. Fluids A **4**, 1472 (1992).

[8] T. M. Elgindi and I.-J. Jeong, Ann. PDE **6**, 5 (2020).

[9] T. M. Elgindi and I.-J. Jeong, Ann. PDE **5**, 16 (2019).

[10] J. Chen and T. Y. Hou, Commun. Math. Phys. **383**, 1559 (2021).

[11] V. I. Yudovich Mosc. Math. J. 711 (2003).

[12] Here *backwards* self-similar solution refers to a solution that *forms* a self-similar solution in finite time. Conversely, a *forward* self-similar solution refers to a solution that starts from singular initial data and exists for all later time. This latter type of solution is, for example, relevant for studying the nonuniqueness of Leray-Hopf solutions to the Navier-Stokes equations.

[13] D. Chae, Commun. Math. Phys. **273**, 203 (2007).

[14] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.130.244002 for cost functions. Specifically, we impose that the gradient of **U** vanishes at infinity, which includes Ref. [15].

[15] R. Bilato, O. Maj, and M. Brambilla, Adv. Comput. Math. **40**, 1159 (2014).

[16] Ideally, blow-up should be proven in the context of finite-energy solutions. To make use of an infinite-energy self-similar solution, one smoothly truncates the self-similar solution and then proves a form of stability. This procedure is standard. The enforced decay prohibits spurious solutions such as $(\mathbf{U}, \Theta) = (y, 0)$.

[17] D. Córdoba, M. A. Fontelos, A. M. Mancho, and J. L. Rodrigo, Proc. Natl. Acad. Sci. U.S.A. **102**, 5949 (2005).

[18] R. K. Scott and D. G. Dritschel, Phys. Rev. Lett. **112**, 144505 (2014).

[19] R. K. Scott and D. G. Dritschel, J. Fluid Mech. **863**, R2 (2019).

[20] J.-G. Liu and R. L. Pego, arXiv:2108.00445.

[21] J. Eggers and M. A. Fontelos, Nonlinearity **22**, R1 (2009).

[22] M. Raissi, P. Perdikaris, and G. Karniadakis, J. Comput. Phys. **378**, 686 (2019).

[23] M. Raissi, A. Yazdani, and G. E. Karniadakis, Science **367**, 1026 (2020).

[24] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Nat. Rev. Phys. **3**, 422 (2021).

[25] R. van der Meer, C. W. Oosterlee, and A. Borovykh, J. Comput. Appl. Math. **405**, 113887 (2021).

[26] D. P. Kingma and J. Ba, arXiv:1412.6980.

[27] D. C. Liu and J. Nocedal, Math. Program. **45**, 503 (1989).

[28] S. Markidis, Front. Big Data **4** (2021).

[29] H. Okamoto, T. Sakajo, and M. Wunsch, Nonlinearity **21**, 2447 (2008).

[30] S. D. Gregorio, Math. Models Methods Appl. Sci. **19**, 1233 (1996).

[31] P. Constantin, P. D. Lax, and A. Majda, Commun. Pure Appl. Math. **38**, 715 (1985).

[32] A. Córdoba, D. Córdoba, and M. A. Fontelos, Ann. Math. **162**, 1377 (2005).

[33] A. Castro and D. Córdoba, Adv. Math. **225**, 1820 (2010).

[34] T. M. Elgindi and I.-J. Jeong, Arch. Ration. Mech. Anal. **235**, 1763 (2019).

[35] J. Chen, T. Hou, and D. Huang, Commun. Pure Appl. Math. **74**, 1282 (2021).

[36] P. M. Lushnikov, D. A. Silantyev, and M. Siegel, J. Nonlinear Sci. **31**, 82 (2021).

[37] In Sec. 3 of the Supplemental Material [14] we reproduce the findings of Ref. [36] for the generalized De Gregorio equation.

[38] D. Li and J. Rodrigo, Adv. Math. **217**, 2563 (2008).

[39] H. Dong, J. Funct. Anal. **255**, 3070 (2008).

[40] A. Kiselev, Math. Model. Nat. Phenom. **5**, 225 (2010).

[41] J. Eggers and M. A. Fontelos, Nonlinearity **22**, R1 (2009).

[42] J. Eggers, Phys. Rev. Fluids **3**, 110503 (2018).

[43] F. Merle, P. Raphaël, I. Rodnianski, and J. Szeftel, Ann. Math. **196**, 567 (2022).

[44] T. Buckmaster, G. Cao-Labora, and J. Gómez-Serrano, arXiv:2208.09445.

[45] C. L. Fefferman *et al.*, The millennium prize problems, American Mathematical Society (2000), p. 57.

[46] S.-J. Oh and F. Pasqualotto, arXiv:2107.07172.

[47] F. Merle, P. Raphaël, I. Rodnianski, and J. Szeftel, Ann. Math. **196**, 779 (2022).