

The Cauchy-Lagrange method for 3D-axisymmetric wall-bounded and potentially singular incompressible Euler flows

Tobias Hertel, Nicolas Besse*, Uriel Frisch

Laboratoire J.-L. Lagrange, Observatoire de la Côte d'Azur, Université Côte d'Azur, Bd de l'observatoire CS 34229, 06300 Nice, Cedex 4, France

ARTICLE INFO

Article history:

Available online 13 October 2021

Keywords:

3D axisymmetric incompressible Euler equations on a bounded domain
Finite-time singularity and blow-up
Cauchy-invariants equations
Cauchy-Lagrange method
Semi-Lagrangian and pseudo-spectral methods
Spectral accuracy

ABSTRACT

We treat the incompressible, and axisymmetric Euler equations for a three-dimensional cylindrical domain with boundaries. The equations are solved by the novel Cauchy-Lagrange algorithm (CLA), which uses the time-analyticity of the Lagrangian trajectories of an incompressible Euler flow and computes the time-Taylor coefficients of the Lagrangian map via recursion relations. This semi-Lagrangian algorithm uses a pseudo-spectral type approach in space by approximating the flow fields by Chebyshev-Fourier polynomials. New methods are presented to solve the resulting Poisson problems directly for their second-order space derivatives. The flow fields, known on the Lagrangian trajectories after one time-step, are interpolated back onto the Eulerian grid to start a new recursion cycle. The time-step is only limited by the radius of convergence and, thus, independent of any Courant-Friedrichs-Lewy (CFL) condition. This allows to advance the flow with larger time-steps independently of the mesh. Stationary and swirl-free flows are used to thoroughly test our implementation for the given geometry. In this work, our ultimate goal is to apply the CLA to a flow, which might develop finite time singularities, resulting in a loss of smoothness. This demands either 3D, or 2D with swirl, and may well require the presence of solid boundaries, as indicated in recent numerical work (Luo and Hou (2014) [45]).

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

The setting is that of the basic three-dimensional incompressible Euler equations in a wall-bounded infinite cylinder D , with boundary ∂D ,

$$\begin{aligned} v_t + v \cdot \nabla v &= -\nabla p, \\ \nabla \cdot v &= 0, \quad \text{in } D, \\ v \cdot \nu &= 0, \quad \text{on } \partial D, \\ v|_{t=0} &= v_0, \end{aligned} \tag{1}$$

* Corresponding author.

E-mail addresses: tobias.hertel@oca.eu, hertel.tobias@mail.de (T. Hertel), nicolas.besse@oca.eu (N. Besse), uriel@oca.eu (U. Frisch).

where ν is the unit outward normal vector to ∂D and the velocity field $v = v(a, t)$ is assumed to be axisymmetric and periodic along the cylinder. In the classical sense, the Euler equations are solvable, at least until a finite time of existence $T > 0$, whenever $v_0 \in C^{1,\alpha}$, $\alpha > 0$. In this case, one defines the Lagrangian particle trajectories via the characteristic map X , unique solution to the characteristic equation

$$\begin{aligned} \partial_t X(a, t) &= v(X(a, t), t), \\ X(a, 0) &= a, \end{aligned} \tag{2}$$

for $a \in D$, $t \in [0, T)$.

A *historical remark* is in order here. The above equations still constitute some of the hardest problems in physics and applied mathematics. Especially the Lagrangian view point, using the particle trajectories, has proved to be a useful tool to address those problems. After the postulation by Euler [20] in 1757, the hydro-dynamical part of the equations (without the rigid boundary conditions) were well known to Cauchy in 1815. He proposed rewriting the equations in Lagrangian coordinates, thereby introducing the now called Cauchy-invariants formulation (CIF) [12,25,8]. In 1858, Helmholtz stated remarkable results concerning vorticity conservation laws. Even though these results were expressed in Lagrangian language, the proof was established using Eulerian variables. Shortly after this, Riemann noticed that these results could perhaps be derived directly using Lagrangian variables. Due to Riemann’s suggestion, Goettingen university proposed a prize for giving a direct Lagrangian proof of Helmholtz’s theorems. This prize was awarded to Hankel [62], who proved Helmholtz’s vorticity conservation laws in a purely Lagrangian framework using the Cauchy-invariants formulation, see also [24]. However, it was only in the beginning of the nineteenth century, that a rigorous mathematical proof for the above existence and uniqueness result was proved by Gyunter [33] and Lichtenstein [44] for the case of the whole space, and by Ebin and Marsden [18] in 1970, for the case of bounded domains and manifolds with boundaries (see also [6] for the case of manifolds with boundaries and in the context of the Cauchy-Lagrange approach). At the present moment, one of the most important questions, namely the global existence and uniqueness of an initially smooth solution, is still open. Current mathematical and numerical work (e.g. [19,45,13]) suggests that the presence of boundaries plays a crucial role in answering this question.

Remarkably, the particle trajectories, also called characteristics, are analytic in time, i.e. the characteristic map X is the sum of an absolutely converging Taylor series locally in time [55,56,27,60,31,26,14,7,6]. Thus, we may conveniently write

$$X(a, t; t_0) = \sum_{s=0}^{\infty} X^{(s)}(a) (t - t_0)^s, \quad t_0 \in [0, T), \tag{3}$$

where t must lie within the disc of convergence with radius ϱ around t_0 . In the sequel of this article, we will drop the argument t_0 in eq. (3), because we use $t_0 = 0$ throughout the text. All the referenced proofs show that the temporal analyticity of X holds even under the lowest regularity assumptions on the initial flow, namely $v_0 \in C^{1,\alpha}$, $\alpha > 0$ (or a weak alternative). While most of the proofs in the cited references above make use of singular integral calculus and start directly from the Euler equations, the references [26,7,6] make use of the revived CIF. This reformulation of the incompressible Euler equations¹ is written entirely in terms of the characteristic map and gives rise to a constructive proof of the temporal analyticity of X . As a byproduct, the authors obtain recursion relations for the time-Taylor coefficients $X^{(s)}$ of eq. (3). In [51], it has been shown that this constructive nature of the CIF-approach leads to an effective algorithm for an incompressible Euler-flow in two dimensions with periodic boundary conditions. This numerical scheme, known as the Cauchy-Lagrange algorithm (CLA), belongs to the large class of semi-Lagrangian methods which are very popular in the fields of numerical weather prediction and numerical plasma physics, and also fluid mechanics and gas dynamics. The present work is inspired by the preceding papers and builds on the findings in [7]. In this work, we describe the CLA for wall-bounded Euler flow and investigate numerically a periodic, and axisymmetric incompressible flow in the cylinder² D .

A common numerical method to advance an incompressible Euler flow in time is the Runge-Kutta time-stepping scheme on an underlying Eulerian mesh. A major drawback of this method is the limitation of the time-step size depending on the mesh spacing and the flow itself, well known as the Courant-Friedrichs-Lewy (CFL) condition. This is problematic especially in turbulent flows, where a necessarily high resolution enforces narrow time-steps and, thus, slows down the overall algorithm. The time-step in the CLA is only limited by the convergence radius ϱ , which behaves roughly as $\text{cst.}/\|\omega_0\|_\alpha$ where ω_0 is the vorticity field at t_0 and $\|\cdot\|_\alpha$ denotes the Hölder norm. Therefore, the algorithm is entirely independent of a CFL condition. This advantage makes the CLA a suitable candidate for simulations of potentially singular flows, where high resolutions are needed to accurately represent large developing gradients. A celebrated result connects the blow-up event to the time-integrated vorticity modulus [3,21]. More accurately, if a blow-up occurs at a finite time T^* , then the time-integrated L^∞ -norm of the vorticity ω until T^* becomes infinite and in particular

$$\limsup_{t \rightarrow T^*} \|\omega\|_{L^\infty} = \infty. \tag{4}$$

¹ It is explicitly given in eqs. (5) and (6) in beginning of section 2.

² See Section 3 for a precise definition of the modeled cylindrical domain.

In the spirit of this result, broad numerical studies have been executed in the last decades, e.g. [32,52,39], mainly searching for an initially smooth flow that eventually verifies (4). See also [29] for a thorough inspection of the concerned literature. However, even with steadily growing computational power, it seemed that the blow-up problem could hardly be answered numerically, because earlier claims of finite time singularities were later attributed to numerical artifacts or under-resolution (see e.g. [17,36,10]). Nonetheless, efforts have been continued and now reference [45] provides ample numerical evidence for a blow-up of an axisymmetric flow in a cylinder. To verify their blow-up result, either a rigorous mathematical proof or more simulations of the same singular flow by other, different, numerical schemes have to be established. We believe that the here presented CLA can take on that challenge since almost no parallels to the algorithm in [45] can be drawn, as discussed in Section 10. The absence of a CFL condition makes the CLA a well-suited candidate to approach the strong gradients near the potential blow-up with larger time-steps on arbitrarily fine³ grids. Furthermore, special attention has been employed to solve the appearing Poisson problems directly for the derivatives of the solutions. This minimizes the errors in the computation of the time-Taylor coefficients $X^{(s)}$ and allows to choose larger time-steps in the disc of convergence. The CLA is implemented in Fortran for high performance and is found to be accurate, stable, and fast. Furthermore, a wide parameter palette is available, including different Poisson solving schemes, interpolation methods (with refinement options), truncation and time-stepping settings, limits for conserved quantities, de-aliasing options, etc. The present work is refined from [35], where the implemented CLA and the connected theory is discussed in greater detail.

The article is organized as follows. An overview of the different steps of the CLA is given in Section 2. The technical frame in form of the recursion relations for the time-Taylor coefficients $X^{(s)}$ and the used Helmholtz–Hodge decomposition (HHD) of $X^{(s)}$ in cylindrical coordinates are stated in Sections 3 and 4. Particular attention has been attributed to the appearing Poisson problems in the HHD. Their treatment can be found in Section 5 together with the used spectral Chebyshev-Fourier discretization. How the flow fields (velocity and vorticity) are re-interpolated onto the original Eulerian grid, after they have been computed along the particle trajectories, is described in Section 6. In Section 7, the choice of a possible time-step is discussed, and validation criteria for one iteration are presented in Section 8. Subsequently, the CLA is tested on stationary and swirl-free (non-stationary) solutions in Section 9, before it is applied to the potentially singular solution of [45] in Section 10.

2. Description of the Cauchy-Lagrange algorithm

The main goal of the Cauchy-Lagrange algorithm (CLA) is the computation of the particle trajectory map X as defined in eq. (3). The Cauchy-invariants formulation of the three-dimensional incompressible Euler equations (1), together with the incompressibility condition, reads

$$\sum_{k=1,2,3} \nabla \dot{X}_k(a, t) \times \nabla X_k(a, t) = \omega_0(a), \tag{5}$$

$$\det(DX(a, t)) = 1, \tag{6}$$

in the Cartesian coordinates $a = (a_1, a_2, a_3)^\top$. The symbol ω_0 denotes the initial vorticity and D indicates the total derivative with respect to the spatial variable a , ∇ denotes the nabla operator and the dot-notation has been adopted to indicate temporal derivatives. Equations (5) and (6) give rise to recursion relations for the curl and divergence of the time-Taylor coefficients $X^{(s)}$, such that the latter can be expressed in terms of lower order time-Taylor coefficients $X^{(s')}$, $s' < s$. From [7], we gather those recursion relations, but abbreviate here the appearing sums by calligraphic capital letters \mathcal{F} and \mathcal{G} to simplify the notation. Note that, in the given reference, the recursions are given in terms of the displacement map $\xi = X - \text{Id}$, but since the Taylor coefficients are time derivatives at $t = 0$, we have $\xi^{(s)} = X^{(s)}$, for $s > 0$ and $X^{(0)} = \text{Id}$, the identity map. The first recursion step starts with $X^{(1)} = v_0$, and therefore, $\nabla \cdot X^{(1)} = 0$ and $-\nabla \times X^{(1)} = -\omega_0$. For a time-Taylor coefficient with orders $s > 1$ we have

$$\nabla \cdot X^{(s)} = \mathcal{F} \left((DX^{(s')})_{0 < s' < s} \right) \tag{7}$$

$$-\nabla \times X^{(s)} = \mathcal{G} \left((DX^{(s')})_{0 < s' < s} \right). \tag{8}$$

Subsequently, the time-Taylor coefficient $X^{(s)}$ is expressed by a Helmholtz-Hodge decomposition $X^{(s)} = \nabla \phi + \nabla \times \Psi$, where the potentials result from two Poisson problems, namely $\Delta \phi = \nabla \cdot X^{(s)}$ and $\Delta \Psi = -\nabla \times X^{(s)}$. Those Poisson problems are supplemented with appropriate boundary conditions that are obtained from the boundary defining map $\mathcal{S} : \mathbb{R}^3 \mapsto \mathbb{R}$, for which $\mathcal{S}(X(\partial\Omega, t)) = 0$ holds true for all times $t \geq 0$. In the case of a periodic cylinder, the explicit boundary conditions are specified in Section 4. After a predefined number of recursions S , we may proceed with the time-step insertion by directly computing

³ As far as the computational limits, i.e. working precision, storage, etc., allow.

$$X(a, \Delta t) \approx \sum_{s=0}^S X^{(s)}(a) \Delta t^s, \tag{9}$$

provided a suitable time-step Δt has been found. The only limitation of the latter is the radius of convergence of the time Taylor series. This radius is typically of the form $\varrho = C / \|\omega_0\|_{C^\alpha}$, but the constant $C = C(\Omega, \alpha)$ is difficult to obtain analytically. We have achieved good results in setting the time-step small enough such that the highest order terms do not contribute to the sum anymore for moderate S due to limitations in machine precision. For a detailed inspection of the time-stepping, we refer the reader to Section 7.

Once a time-step has been set, we obtain the velocity field on the trajectories via the characteristic equation

$$u(X(a, \Delta t), \Delta t) = X_t(a, \Delta t) \approx \sum_{s=1}^S sX^{(s)}(a) \Delta t^{s-1}, \tag{10}$$

the vorticity is similarly obtained via the vorticity transport formula (i.e. [46, p. 20])

$$\omega(X(a, \Delta t), \Delta t) = DX(a, \Delta t) \cdot \omega_0(a) \approx \sum_{s=0}^S DX^{(s)}(a) \cdot \omega_0(a) \Delta t^s. \tag{11}$$

Numerically, the characteristics deform the underlying discrete grid and the velocity and vorticity fields are only known on that distorted grid, i.e. at the end points of the particle trajectories. Hence, an interpolation is needed to obtain the values of the advanced flow fields on the Eulerian grid. This three-dimensional scattered interpolation needs, in principal, only to be performed for the vorticity field, but it is better to re-interpolate the velocity field as well, because it appears in the 1st order term in the next time-Taylor series of the characteristics (in $X^{(1)}$). These interpolated flow fields are then taken as the new initial conditions and, therefore, we may set $t_0 = 0$ after each time-step, where t_0 appeared in eq. (3).

In this paper, a three-dimensional axisymmetric flow in a cylindrical domain is modeled to understand the challenges and advantages of this novel algorithm for wall-bounded incompressible Euler flow. The axisymmetry of the problem simplifies both the Poisson solvers and the interpolation procedure to two dimensions. Best results in the interpolation step have been achieved using the cascade interpolation scheme [53].

3. Recursion relations for axisymmetric flow

In this section, we would like to introduce our notations for cylindrical domains and give the recursion relations for our particular setting of an axisymmetric and periodic flow in the cylindrical region

$$D(1, L) := \left\{ a = (a_1, a_2, a_3) \in \mathbb{R} \mid \sqrt{a_1^2 + a_2^2} \leq 1 \text{ and } a_3 \in [0, L] \right\}. \tag{12}$$

Note that the real flow takes place inside of an infinite cylinder, but because of the periodicity we can restrict our computations to the region $D(1, L)$. The Lagrangian particles, however, may leave this region and disperse into the whole cylinder, but they can always be remapped into $D(1, L)$. See Section 6 for a more detailed discussion on that matter. An axisymmetric flow in $D(1, L)$ takes the form

$$v(a, t) = v^r(r, z, t)e_r + v^\alpha(r, z, t)e_\alpha + v^z(r, z, t)e_z, \tag{13}$$

where v and $a = re_r + ze_z$ are represented in the cylindrical basis $e_r = (\cos \alpha, \sin \alpha, 0)^\top$, $e_\alpha = (-\sin \alpha, \cos \alpha, 0)^\top$, $e_z = (0, 0, 1)^\top$. As usual we have, for $a = (a_1, a_2, a_3)^\top$,

$$r = \sqrt{a_1^2 + a_2^2}, \quad \alpha = \mathcal{A}(a_1, a_2), \quad z = a_3,$$

with $(r, \alpha, z) \in [0, 1] \times (-\pi, \pi] \times [0, L]$ and

$$\mathcal{A}(a_1, a_2) := \begin{cases} 0, & \text{if } a_1 = a_2 = 0 \\ \arcsin(\frac{a_2}{r}), & \text{if } a_1 = 0, a_2 \neq 0 \\ \arctan(\frac{a_2}{a_1}), & \text{if } a_1 > 0 \\ \pi - \arcsin(\frac{a_2}{r}), & \text{if } a_1 < 0. \end{cases} \tag{14}$$

The cylindrical vorticity corresponding to eq. (13) reads

$$\omega(a, t) = \omega^r(r, z, t)e_r + \omega^\alpha(r, z, t)e_\alpha + \omega^z(r, z, t)e_z, \tag{15}$$

and is obtained by

$$\omega = \nabla_c \times v = -v_z^\alpha e_r + (v_z^r - v_r^z) e_\alpha + \frac{1}{r} (rv^\alpha)_r e_z, \tag{16}$$

where

$$\nabla_c := e_r \partial_r + \frac{1}{r} e_\alpha \partial_\alpha + e_z \partial_z \tag{17}$$

denotes the cylindrical nabla operator. The components of v and ω are independent of the angular variable α and periodic in $z \in [0, L]$. The α -dependence of the flow fields only lies in the basis vectors e_r and e_α . The same is true for the characteristics $X = X(a, t)$ and the displacement map $\xi := X - \text{Id}$, which are denoted by

$$X(a, t) = X^r(r, z, t) e_r + X^\alpha(r, z, t) e_\alpha + X^z(r, z, t) e_z, \tag{18}$$

$$\xi(a, t) = R(r, z, t) e_r + A(r, z, t) e_\alpha + Z(r, z, t) e_z. \tag{19}$$

We recall the Cauchy-invariants formula in Cartesian coordinates in terms of the displacement map $\xi = (\xi_1, \xi_2, \xi_3)$, namely eqs. (16)-(17) in [7],

$$\nabla \times \dot{\xi} + \sum_{1 \leq k \leq 3} \nabla \dot{\xi}_k \times \nabla \xi_k = \omega_0 \tag{20}$$

$$\nabla \cdot \dot{\xi} + \sum_{1 \leq i < j \leq 3} (\partial_i \xi_i \partial_j \xi_j - \partial_i \xi_j \partial_j \xi_i) + \det(D\xi) = 0. \tag{21}$$

Here, the dot-notation is used to indicate temporal derivatives. An insertion of eq. (19) into eqs. (20) and (21) yields after a slight rearrangement

$$-\nabla_c \times \dot{\xi} = e_r \frac{1}{r} \partial_z (\dot{R}A - \dot{A}R) + e_\alpha [\{\dot{R}, R\} + \{\dot{A}, A\} + \{\dot{Z}, Z\}] + e_z \frac{1}{r} \partial_r (\dot{A}R - \dot{R}A) - \omega_0 \tag{22}$$

$$\nabla_c \cdot \dot{\xi} = \frac{1}{r} (A(\{A, Z\} - A_r) + R(\{R, Z\} - R_r - Z_z)) + \{R, Z\}, \tag{23}$$

where $\{f, g\} := f_z g_r - f_r g_z$ denotes the Poisson bracket for functions $f = f(r, z)$ and $g = g(r, z)$. An insertion of the formal time-Taylor series

$$\Lambda(r, z, t) = \sum_{s \geq 1} \Lambda^{(s)}(r, z) t^s, \quad \dot{\Lambda}(r, z, t) = \sum_{s \geq 1} s \Lambda^{(s)}(r, z) t^{s-1}, \quad \Lambda = R, A, Z, \tag{24}$$

into the expressions of the rotation (eq. (22)) and divergence (eq. (23)) above yields component-wise

$$(-\nabla_c \times \xi^{(s)})^r = -\delta_{1s} \omega_0^r + \sum_{1 \leq m < s} \frac{m}{rs} \partial_z [R^{(m)} A^{(s-m)} - A^{(m)} R^{(s-m)}] \tag{25}$$

$$(-\nabla_c \times \xi^{(s)})^\alpha = -\delta_{1s} \omega_0^\alpha + \sum_{1 \leq m < s} \frac{m}{rs} [\{R^{(m)}, R^{(s-m)}\} + \{A^{(m)}, A^{(s-m)}\} + \{Z^{(m)}, Z^{(s-m)}\}] \tag{26}$$

$$(-\nabla_c \times \xi^{(s)})^z = -\delta_{1s} \omega_0^z + \sum_{1 \leq m < s} \frac{m}{rs} \partial_r [A^{(m)} R^{(s-m)} - R^{(m)} A^{(s-m)}], \tag{27}$$

and

$$\begin{aligned} \nabla_c \cdot \xi^{(s)} = & \\ \{R, Z\}^{(s)} + \sum_{1 \leq m < s} \frac{A^{(m)}}{r} (\{A, Z\}^{(s-m)} - A_r^{(s-m)}) + \frac{R^{(m)}}{r} (\{R, Z\}^{(s-m)} - R_r^{(s-m)} - Z_z^{(s-m)}) \end{aligned} \tag{28}$$

with

$$\{\Lambda, Z\}^{(n)} := \sum_{1 \leq m < n} (\Lambda_z^{(m)} Z_r^{(n-m)} - \Lambda_r^{(m)} Z_z^{(s-m)}), \quad n = 1, 2, \dots,$$

such that $\{\Lambda, Z\} = \sum_{1 \leq s} \{\Lambda, Z\}^{(s)} t^s$ for $\Lambda = R, A$. By convention, empty sums evaluate to zero. Numerically it makes sense to store the time-Taylor coefficients of $\{R, Z\}$ and $\{A, Z\}$. Although this increases storage demand, it speeds up the recurrence computations tremendously for high orders, as the summation terms in eq. (28) consist only of two factors instead of three.

In sight of the envisaged Helmholtz-Hodge decomposition in Section 4 we state the boundary value $R(1, z, t)$ for $t > 0$. As a matter of fact, for a boundary point a , $X(a, t)$ lies on the boundary for all $t > 0$. In the case of our periodic flow in the

cylinder $D(1, L)$, this boundary reduces to the mantle of the cylinder which is described by the kernel of the boundary map $S(a_1, a_2, a_3) = a_1^2 + a_2^2 - 1$. An insertion of $X(a, t)$, $a \in \partial D(1, L)$, yields, in terms of the cylindrical components of ξ ,

$$\begin{aligned} 0 &= \mathcal{S}(X(a, t)) = X_1^2(a, t) + X_2^2(a, t) - 1 \\ &= (R(1, z, t) + 1)^2 + A(1, z, t)^2 - 1 = 2R(1, z, t) + R(1, z, t)^2 + A(1, z, t)^2. \end{aligned} \tag{29}$$

and, thus,

$$R(1, z, t) = -\frac{1}{2} \left(R(1, z, t)^2 + A(1, z, t)^2 \right).$$

An insertion of the power series of the components (24) into the last term, with subsequent grouping of coefficient orders $s \geq 1$, entails

$$R^{(s)}(1, z) = -\frac{1}{2} \sum_{n=1}^{s-1} \left(R^{(n)} R^{(s-n)} + A^{(n)} A^{(s-n)} \right) (1, z). \tag{30}$$

We have stated all the needed recursion formulas for the rotation, divergence and boundary terms of the coefficients $\xi^{(s)}$ (they equate the $X^{(s)}$), which are needed for the Helmholtz-Hodge decomposition that is discussed in the next section. Note that those recursion formulas depend only on $(r, z) \in [0, 1] \times [0, L]$ due to the imposed axisymmetry of the flow, which reduces the computational complexity tremendously when compared to general 3D flow.

4. Helmholtz–Hodge decomposition and computational simplifications

In order to exploit the formerly found recursion relations of the divergence and rotation we state the Helmholtz–Hodge decomposition (HHD) of the trajectory’s time-Taylor coefficients in cylindrical coordinates. In the following, let $s \geq 1$ be fixed. $\xi^{(s)}$ can be expressed by a HHD in the cylindrical domain $D(1, L)$ as

$$\xi^{(s)} = \nabla \phi + \nabla \times \Psi. \tag{31}$$

The potentials ϕ and Ψ should read $\phi^{(s)}$ and $\Psi^{(s)}$, since those potentials depend on s . We omit the superscripts only to increase the readability. The scalar potential ϕ is uniquely described by the Poisson problem

$$\begin{aligned} \Delta \phi &= \nabla \cdot \xi^{(s)}, & \text{in } D(1, L), \\ \partial_\nu \phi &= \nu \cdot \xi^{(s)}, & \text{on } \partial D(1, L), \end{aligned} \tag{32}$$

and the vector potential Ψ by

$$\begin{aligned} \Delta \Psi &= -\nabla \times \xi^{(s)}, \\ \nabla \cdot \Psi &= 0, & \text{in } D(1, L), \\ \Psi \times \nu &= 0, & \text{on } \partial D(1, L), \end{aligned} \tag{33}$$

where ν denotes the unit outward normal vector to the boundary, and in our case we simply have $\nu = e_r$. See [30] for a detailed analysis of the HHD and the above statements for a bounded domain. In principle, we treat a flow inside of an infinite cylinder, which is unbounded in vertical direction, but we do not need to impose a decay of the solution for $z \rightarrow \infty$, because of the periodicity in $[0, L]$. In this particular case, the validity of the above statement is verified without difficulty by a slight adaptation of the arguments in [30]. Note that the boundary conditions for Ψ are different from those in [7]. Indeed, $\Psi = 0$ on the boundary of the domain is not sufficient for a general HHD in a bounded domain, but accounts only for a special situation. We refer the reader to [6], where the boundary value problem eq. (33) from above has been addressed in Remark 6. We reformulate the above problems in cylindrical coordinates $a = (a_1, a_2, a_3) \mapsto (r \cos \alpha, r \sin \alpha, z)$, and keep the symbols for the vector potentials by an abuse of notation. Thus, we are looking for functions $\phi = \phi(r, z)$ and $\Psi = \Psi(r, \alpha, z)$, with

$$\Psi(r, \alpha, z) = \Psi^r(r, z)e_r + \Psi^\alpha(r, z)e_\alpha + \Psi^z(r, z)e_z, \tag{34}$$

defined in the domain $D(1, L)$. Under the assumption of axisymmetry we may fix momentarily $\alpha = 0$ which implies $e_r = (1, 0, 0)^\top$. The boundary conditions for ϕ translate to

$$\partial_r \phi|_{r=1} = e_r \cdot \xi^{(s)}|_{r=1} = R^{(s)}(1, z),$$

and those for Ψ , namely $\Psi|_{r=1} \times e_r = 0$, imply

$$\Psi^z(1, z) = \Psi^\alpha(1, z) = 0. \tag{35}$$

Moreover, the divergence-free condition, $\nabla \cdot \Psi = \frac{1}{r}\Psi^r + \partial_r\Psi^r + \partial_z\Psi^z = 0$, is assumed to hold up to the boundary and delivers the boundary condition on Ψ^r ,

$$\Psi^r(1, z) + \partial_r\Psi^r(1, z) = -\partial_z\Psi^z(1, z) = 0, \quad (36)$$

where eq. (35) is used. Let us define the linear differential operators

$$\mathcal{L} := \frac{1}{r}\partial_r(r \cdot) = \frac{1}{r} + \partial_r, \quad (37)$$

$$\mathcal{L}_1 := \mathcal{L}(\partial_r \cdot) = \frac{1}{r}\partial_r(r\partial_r \cdot) = \partial_r^2 + \frac{1}{r}\partial_r, \quad (38)$$

$$\mathcal{L}_2 := \partial_r(\mathcal{L} \cdot) = \partial_r\left(\frac{1}{r}\partial_r(r \cdot)\right) = \partial_r^2 + \frac{1}{r}\partial_r - \frac{1}{r^2}, \quad (39)$$

$$\mathcal{G} := \mathcal{L}_1 + \partial_z^2 = \partial_r^2 + \frac{1}{r}\partial_r + \partial_z^2, \quad (40)$$

$$\mathcal{H} := \mathcal{L}_2 + \partial_z^2 = \partial_r^2 + \frac{1}{r}\partial_r - \frac{1}{r^2} + \partial_z^2. \quad (41)$$

Subsequently, the translation of eqs. (32) and (33) in cylindrical coordinates entails four partial differential equations

$$\begin{aligned} \mathcal{G}\phi(r, z) &= \left(\nabla \cdot \xi^{(s)}\right)(r, z), \\ \partial_r\phi(0, z) &= 0, \\ \partial_r\phi(1, z) &= R^{(s)}(1, z), \end{aligned} \quad (42)$$

$$\begin{aligned} \mathcal{H}\Psi^r(r, z) &= \left(-\nabla \times \xi^{(s)}\right)^r(r, z), \\ \Psi^r(0, z) &= 0, \end{aligned} \quad (43)$$

$$\Psi^r(1, z) + \partial_r\Psi^r(1, z) = 0,$$

$$\begin{aligned} \mathcal{H}\Psi^\alpha(r, z) &= \left(-\nabla \times \xi^{(s)}\right)^\alpha(r, z), \\ \Psi^\alpha(0, z) &= 0, \\ \Psi^\alpha(1, z) &= 0, \end{aligned} \quad (44)$$

$$\begin{aligned} \mathcal{G}\Psi^z(r, z) &= \left(-\nabla \times \xi^{(s)}\right)^z(r, z), \\ \partial_r\Psi^z(0, z) &= 0, \\ \Psi^z(1, z) &= 0, \end{aligned} \quad (45)$$

for $\phi, \Psi^r, \Psi^\alpha, \Psi^z \in C_L^2([0, 1] \times [0, L])$, where the subscript $(\cdot)_L$ indicates periodicity in $[0, L]$. The radial regularity at the boundary is understood to be a limit. The pole conditions stem from the continuity constraints, as the operators must stay bounded for $r \in [0, 1]$. After solving the above equations, one obtains the coefficients $A^{(s)}, R^{(s)}$ and $Z^{(s)}$ in cylindrical coordinates via eq. (31), i.e.

$$R^{(s)} = \partial_r\phi - \partial_z\Psi^\alpha \quad (46)$$

$$A^{(s)} = \partial_z\Psi^r - \partial_r\Psi^z \quad (47)$$

$$Z^{(s)} = \partial_z\phi + \partial_r\Psi^\alpha + \frac{1}{r}\Psi^\alpha, \quad (48)$$

where all functions only depend on $(r, z) \in [0, 1] \times [0, L]$.

4.1. Computational simplifications for the implementation

In consulting the recursion relations eqs. (25) to (28), the overall algorithm demands the spatial derivatives of the time-Taylor coefficients $R^{(s)}, A^{(s)}$ and $Z^{(s)}$ in radial and vertical direction, i.e. second derivatives on the potentials. Second-order derivatives imply large numerical errors, especially if produced via standard spectral differentiation methods, which potentially limits the possible time-step that can be inserted into the truncated time-Taylor series. Therefore, we aim to directly solve for the needed second-order derivatives, which is discussed in later sections. In the following, we state the derivatives of ϕ and the components of Ψ that need to be found. Fortunately, not all (mixed) second-order derivatives of the potentials need to be solved for, but only a few, which results from the simplifications that are discussed here.

While eqs. (42), (44) and (45) are pure Neumann or Dirichlet type boundary value problems, is eq. (43) a mixed boundary value problem. However, in consulting eqs. (46) to (48), the component Ψ^r is only used for the construction of $A^{(s)}$ and derivatives thereof. In the following we show that it is not necessary to solve eqs. (43) and (45) in the case of our cylindrical geometry. Instead of solving eqs. (43) and (45) for Ψ^r, Ψ^z , we will directly solve for $A^{(s)}$ and its derivatives. As a matter of fact, we have

$$\partial_z A^{(s)} = -(\nabla \times \xi^{(s)})^r \tag{49}$$

$$\mathcal{L}A^{(s)} = \frac{A^{(s)}}{r} + \partial_r A^{(s)} = (\nabla \times \xi^{(s)})^z, \tag{50}$$

stemming from the definition of the curl operator in cylindrical coordinates, namely

$$\nabla_c \times \xi^{(s)} = -A_z^{(s)} e_r + (R_z^{(s)} - Z_r^{(s)}) e_\alpha + \frac{1}{r} (rA^{(s)})_r e_z, \tag{51}$$

with ∇_c defined in eq. (17). The RHSs of eqs. (49) and (50) are given via the recursion relations eqs. (25) and (27), which are computed and saved in the CLA. Integrating the operator $\mathcal{L} = \frac{1}{r} \partial_r(\cdot)$ in eq. (50) directly yields

$$A^{(s)}(r, z) = \frac{1}{r} \int_0^r t (\nabla \times \xi^{(s)})^z(t, z) dt \tag{52}$$

$$A_1^{(s)}(r, z) := \frac{A^{(s)}(r, z)}{r} = \frac{1}{r^2} \int_0^r t (\nabla \times \xi^{(s)})^z(t, z) dt, \tag{53}$$

where we have used that $A^{(s)}(0, z) = 0$ for all $z \in [0, L]$ and $s \in \mathbb{N}_0$. However, our implementation of the CLA solves directly for $\partial_r A^{(s)}$ by calculating $\partial_r \mathcal{L}^{-1}$ in a later defined way. Subsequently, $A_1^{(s)}$ is obtained in using eq. (50), i.e. $A_1^{(s)} = (\nabla \times \xi^{(s)})^z - \partial_r A^{(s)}$, and $A^{(s)} = rA_1^{(s)}$. Furthermore, provided we know $\partial_r R^{(s)}, \partial_z R^{(s)}$ and $R_1^{(s)}$, we compute in a straightforward way

$$\partial_r Z^{(s)} = \partial_z R^{(s)} + (-\nabla \times \xi^{(s)})^\alpha \tag{54}$$

$$\partial_z Z^{(s)} = -R_1^{(s)} - \partial_r R^{(s)} + \nabla \cdot \xi^{(s)}, \tag{55}$$

where we have used again eq. (51) and the definition of the divergence, namely

$$\nabla_c \cdot \xi^{(s)} = \frac{1}{r} (rR^{(s)})_r + Z_z^{(s)}. \tag{56}$$

The RHSs of eqs. (54) and (55) are given by the recursion relations eqs. (26) and (28). Therefore, we only need to compute those expressions of the Helmholtz–Hodge potentials that appear in the following RHSs,

$$\partial_r R^{(s)} = \partial_r^2 \phi - \partial_r \partial_z \Psi^\alpha \tag{57}$$

$$\partial_z R^{(s)} = \partial_z \partial_r \phi - \partial_z^2 \Psi^\alpha \tag{58}$$

$$R_1^{(s)} = \frac{\partial_r \phi}{r} - \frac{\partial_z \Psi^\alpha}{r} \tag{59}$$

$$Z^{(s)} = \partial_z \phi + \partial_r \Psi^\alpha + \frac{1}{r} \Psi^\alpha. \tag{60}$$

In fact, the coefficients $R^{(s)}, A^{(s)}$ and $Z^{(s)}$ are not needed for the recurrence mechanism itself, but for the actual computation of the particle trajectories. This information is mainly needed for the interpolation step after the time-step insertion. The coefficients $R^{(s)}$ and $A^{(s)}$ are simply obtained by multiplying $R_1^{(s)}$ and $A_1^{(s)}$ by r respectively.

5. Discretization in space and Poisson solvers

A crucial part of the investigated algorithm is the accurate computation of the Calderon-Zygmund operators of order zero, namely the double derivatives of an inverse Laplacian, applied to the divergence and rotation of the trajectory’s time-Taylor coefficients. The appearing errors from the second derivatives on top of the errors in the solutions themselves may recursively cascade into higher orders s . This amplification of errors can lead to a lower order scheme and a very limited time-step. Thus, the better the above-mentioned operators are computed, the better the time-step can be chosen and the more accurate the overall algorithm becomes with fewer time-steps. It is desirable to preserve the possible range of a time-step as much as possible by maximally reducing the accumulation of numerical errors.

In this first implementation of the CLA for a domain with boundaries, we refrain from finite difference methods for the Calderon-Zygmund operators, because of their slow convergence and limited accuracy in the derivatives. Instead, a pseudo-spectral approach was chosen and a representation of the trajectory's coefficients by shifted Chebyshev series in the radial dimension and Fourier series in the vertical dimension is applied:

$$X^{(s)}(r, z) = \sum_{|k|=0}^{\infty} \sum_{l=0}^{\infty} X_{kl}^{(s)} T_l^*(r) e^{i\tilde{k}z}, \tag{61}$$

where $T_l^*(r) = T_l(2r - 1)$ are the shifted Chebyshev polynomials and

$$\tilde{k} := \frac{2\pi k}{L}. \tag{62}$$

A truncation to $M + 1, N + 1 \in \mathbb{N}$ modes yields a spectrally discrete approximation of the latter, i.e.

$$X^{(s)}(r, z) \cong (I_{NM} X^{(s)})(r, z) = \sum_{|k|=0}^{M/2} \sum_{l=0}^N X_{kl}^{(s)} T_l^*(r) e^{i\tilde{k}z}. \tag{63}$$

See Appendix A for a basic overview of the shifted Chebyshev polynomials and series.

The Chebyshev-Fourier basis has been chosen due to the underlying numerically fast discrete Fourier transforms, and also to initialize this novel method for wall-bounded incompressible Euler flow in using widely known and well studied spectral representations. The applied methods allow for fast numerical convergence and, thus, for an accurate representation of the solutions to the Helmholtz-Hodge decomposition and the flow itself in low to moderate resolutions. Although the choice for a spectral approach introduces difficulties for higher resolutions, as spectral derivatives become progressively more erroneous and unstable, we have found ways to overcome those pitfalls and find that the overall algorithm is stable, accurate, and fast. This is to a large extent due to sophisticated Poisson solvers, which solve directly for the second-order radial derivatives and divisions by the radial argument. One major advantage of shifted Chebyshev series is that they are obtained from function values known on the shifted Chebyshev extrema, namely $((\cos(i\pi/N) + 1)/2)_{i=0, \dots, N}$. Those are clustered near (and contain) the boundary of the cylindrical domain as well as the pole $r = 0$. This clustering is important, as the numerical studies in [45] indicate a singularity directly at the boundary. In summary, we choose to deal with unstable spectral derivatives in exchange for very good approximation properties and a high point density at the boundary and pole of the cylinder.

5.1. Decoupling of the Poisson problems

Let in the following $u = u(r, z)$ and $w = w(r, z)$ be sufficiently smooth z -periodic functions $[0, 1] \times [0, L] \rightarrow \mathbb{R}$. In sight of eqs. (42) and (44) and the fact that in our approach we will not solve eqs. (43) and (45) explicitly, we will treat the following homogeneous model problems

$$\begin{aligned} \mathcal{G}u &:= \left(\partial_r^2 + \frac{1}{r} \partial_r + \partial_z^2 \right) u(r, z) = f(r, z) \\ \partial_r |_{r=0,1} u(r, z) &= 0, \end{aligned} \tag{64}$$

and

$$\begin{aligned} \mathcal{H}w &:= \left(\partial_r^2 + \frac{1}{r} \partial_r - \frac{1}{r^2} + \partial_z^2 \right) w(r, z) = g(r, z), \\ w(0, z) = w(1, z) &= 0. \end{aligned} \tag{65}$$

As discussed in the discretization Section 5, we use the periodicity in vertical direction and anticipate an approximation of the solutions in truncating the Fourier expanded differential equations. In replacing the appearing functions by truncated Fourier series, i.e.

$$\Lambda^M(r, z) = \sum_{|k|=0}^{M/2} \hat{\Lambda}_k(r) e^{i\tilde{k}z}, \quad \Lambda \in \{u, f, w, g\}, \tag{66}$$

the equations decouple and for each $k \in [0, M/2]$ we are left to solve the ordinary differential equations

$$\begin{aligned} (\mathcal{L}_1 - \tilde{k}^2) \hat{u}_k(r) &= \hat{u}_k''(r) + \frac{1}{r} \hat{u}_k'(r) - \tilde{k}^2 \hat{u}_k(r) = \hat{f}_k(r), \\ \hat{u}_k'(0) = \hat{u}_k'(1) &= 0, \end{aligned} \tag{67}$$

and

$$\begin{aligned} (\mathcal{L}_2 - \tilde{k}^2) \hat{w}_k(r) &= \hat{w}_k''(r) + \frac{1}{r} \hat{w}_k'(r) - \left(\frac{1}{r^2} + \tilde{k}^2 \right) \hat{w}_k(r) = \hat{g}_k(r), \\ \hat{w}_k(0) = \hat{w}_k(1) &= 0, \end{aligned} \tag{68}$$

where \mathcal{L}_1 and \mathcal{L}_2 are defined in eqs. (38) and (39). Since only real valued functions, whose complex modes verify $\hat{u}_k = \overline{\hat{u}_{-k}}$, are treated, we may restrict ourselves to positive modes. Note that eqs. (67) and (68) are commonly known as the modified Bessel ODEs of order 0 and 1 respectively. Their solutions, well known yet hard to compute numerically, are described later in this section.

Remark. To obtain the solution of the inhomogeneous problem eq. (42), with Neumann boundary data $b = b(z)$ at $r = 1$, from a homogeneous problem eq. (64), one simply solves the latter with the modified right hand side $\tilde{f} = \tilde{f}(r, z) = f(r, z) - \mathcal{G}(\frac{r^2}{2}b(z))$ for a solution $\tilde{u} = \tilde{u}(r, z)$. The inhomogeneous solution is then given by $\tilde{u}(r, z) + \frac{r^2}{2}b(z)$. The smoothness requirement for b is neglected here, because the procedure is directly applied to the truncated problem, where all involved functions are C^∞ .

The special case, where $\tilde{k} = 0$ can be treated explicitly, since the operators \mathcal{L} , \mathcal{L}_1 and \mathcal{L}_2 are readily inverted. Note that \mathcal{L} does not appear in the Poisson problems but is needed to solve directly for $A^{(s)}$ and its derivatives as described in Section 4. Suppose $h = h(r)$, $r \in [0, 1]$, is a generic function that is represented by a truncated shifted Chebyshev series

$$h(r) = \sum_{l=0}^N h_l T_l^*(r). \tag{69}$$

The inverses of the linear differential operators are

$$\mathcal{L}^{-1}h(r) = \frac{1}{r} \int_0^r t h(t) dt = \sum_{l=0}^N h_l \left(\frac{1}{r} \int_0^r t T_l^*(t) dt \right), \tag{70}$$

$$\mathcal{L}_1^{-1}h(r) = \int_0^r \frac{1}{t} \int_0^t y h(y) dy dt = \sum_{l=0}^N h_l \left(\int_0^r \frac{1}{t} \int_0^t y T_l^*(y) dy dt \right), \tag{71}$$

$$\mathcal{L}_2^{-1}h(r) = \frac{\int_0^r t \left(c(h) + \int_0^t h(x) dx \right) dt}{r} = \sum_{l=0}^N h_l \left(\frac{\int_0^r t \left(c(T_l^*) + \int_0^t T_l^*(x) dx \right) dt}{r} \right), \tag{72}$$

where

$$c(\cdot) = -2 \int_0^1 t \int_0^t (\cdot)(x) dx dt$$

ensures homogeneity at $r = 1$. Any bounded differential operator can then be applied to the inverses, and, thus, the derivatives of the solutions to the above ODEs can be obtained on a fixed grid via a matrix vector multiplication. For example, the solution of eq. (67), with $k = 0$, subject to the differential operator $D_r \in \{\partial_r, \partial_r^2, \partial_r/r, 1/r\}$ is approximated by

$$(D_r \hat{u}_0(x_i))_{i=0, \dots, N} = (D_r \mathcal{L}_1^{-1} T_l^*(x_i))_{i,l=0, \dots, N} \cdot (\hat{f}_{0l})_{l=0, \dots, N}, \tag{73}$$

on the fixed grid $(x_i)_{i=0, \dots, N}$. Here $(\hat{f}_{0l})_{l=0, \dots, N}$ denote the coefficients of the shifted Chebyshev series that approximate \hat{f}_0 in eq. (67) with $k = 0$. In our CLA code, the matrices $(D_r \mathcal{L}_1^{-1} T_l^*(x_i))_{i,l=0, \dots, N}$ are pre-computed for the needed derivatives and stored for continuous re-application. This method, to directly obtain the derivatives of the solution to an ODE, is very accurate and of usual complexity of $O(N^2)$ for spectral solving methods with full matrices. If $k > 0$, then the operators $\mathcal{L}_1 - \tilde{k}^2$ and $\mathcal{L}_2 - \tilde{k}^2$ are known as the modified Bessel ODEs of order 0 and 1 respectively. Those special operators are also invertible and discussed in Section 5.3. But first we will apply to eqs. (67) and (68), with $k > 0$, well known spectral solvers and propose a slight modification, which allows us to use the here presented technique to solve for the radial derivatives directly.

Notation. The application of a bounded differential operator ∂_r^i/r^j to an inverse operator L^{-1} will be indicated by superscripts of the form $L^{-1,(i)/r^j}$, e.g. $L^{-1,(2)} := \partial_r^2 L^{-1}$ and $L^{-1,(0)/r} := \frac{1}{r} L^{-1}$.

5.2. Chebyshev-Galerkin solver based on a modified Galerkin method

Let us start with the homogeneous Dirichlet problem eq. (68). In [57], the author introduced a Chebyshev-Galerkin scheme for the homogeneous Helmholtz equation in a disc. As a matter of fact, the operators in use are almost identical to ours and, therefore, we may apply his findings to our case. A Galerkin scheme consists of a variational formulation of

a given problem together with a suitable test function space. A finite-dimensional sub-space of the latter, spanned by a complete, sometimes orthogonal, set of basis functions, entails a discrete approximation of the solution to the problem. The Lax-Milgram theorem ensures existence and uniqueness if continuity and coercivity conditions on the variational form hold true. For a complete introduction on Galerkin schemes, we would like to refer the reader to [58]. Here, we aim directly at a suitable formulation for Chebyshev polynomials.

Notation. The method described below is called Chebyshev-Galerkin (CG) method and abbreviated by CG_1^{Dir} , where the super-/subscripts indicate the Dirichlet boundary condition and the used set of basis functions (\mathcal{B}_1 from eq. (80)).⁴ The Neumann case, which treats another operator, will be later referred to as CG^{Neu} .

Starting from eq. (68), the radial interval is expanded via the coordinate transformation $r = (t + 1)/2, t \in I := [-1, 1]$ to give

$$y''(t) + \frac{1}{t+1}y'(t) - \left(\frac{1}{(t+1)^2} + \frac{\tilde{k}^2}{4}\right)y(t) = \frac{1}{4}\hat{g}_k((t+1)/2), \tag{74}$$

with $y(t) := \hat{w}_k((t+1)/2), k \in [0, M/2]$ fixed, which we multiply by $(t+1)$ to obtain

$$\underbrace{(t+1)y''(t) + y'(t)}_{=((t+1)y')'(t)} - \left(\frac{1}{t+1} + \frac{\tilde{k}^2}{4}(t+1)\right)y(t) = h(t), \tag{75}$$

where $h(t) = \frac{1}{4}(t+1)\hat{g}_k((t+1)/2), \tilde{k} = (2\pi k)/L$ as in eq. (62), and $y(0) = y(1) = 0$. A suitable test function space is $Y_0 := H_{0,\omega}^1(I)$, where the subscript ω indicates that the inner product in

$$H_0^1(I) := \{\theta \in H^1(I) = W^{1,2}(I) \mid \theta(-1) = \theta(1) = 0 \text{ in the trace sense}\} \tag{76}$$

is weighted by $\omega(t) = (1 - t^2)^{-1/2}$, i.e. the Chebyshev weight from Appendix A. In the following let $v \in Y_0$, we apply the inner product $(\cdot, v)_\omega = \int_I \cdot v \omega dt$ to both sides of eq. (75), execute an integration by parts in the LHS and multiply by (-1) to obtain

$$\begin{aligned} \int_I (t+1)y'(t)(v\omega)'(t) dt + \int_I \frac{1}{t+1}y(t)v(t)\omega(t) dt + \tilde{k}^2 \int_I \frac{1}{4}(t+1)y(t)v(t)\omega(t) dt \\ = - \int_I h(t)v(t)\omega(t) dt. \end{aligned} \tag{77}$$

The variational formulation to our problem eq. (75) now reads as follows: find $y \in Y_0$, such that eq. (77) holds for each $v \in Y_0$. In defining

$$Y_0^N := \{p \in P_N \mid p(-1) = p(1) = 0\} \subset Y_0, \tag{78}$$

with P_N being the polynomial space of maximal degree $N \in \mathbb{N}$, we may state the Galerkin scheme for our homogeneous Dirichlet problem eq. (75):

$$\left\{ \begin{array}{l} \text{find } y = y^N \in Y_0^N \\ \text{s.t. eq. (77) holds } \forall v = v^N \in Y_0^N, \end{array} \right. \tag{79}$$

in other words, we look for an approximation of the solution $y \in Y_0$ in the finite dimensional space Y_0^N . The advantage of the scheme in [57] (see also [58]) is that it uses special basis functions that fulfill the boundary conditions exactly and allow additionally for simple structured solving matrices. Those basis functions in the Dirichlet case are given by

$$\begin{aligned} \Phi_l &:= T_l - T_{l+2}, \quad l = 0, \dots, N-2, \\ \mathcal{B}_1 &:= \{\Phi_0, \dots, \Phi_{N-2}\}. \end{aligned} \tag{80}$$

The set \mathcal{B}_1 is a basis of Y_0^N and (79) must, therefore, only hold for any $v = \Phi_0, \dots, \Phi_{N-2}$. Thus, we seek a solution in the form

$$y^N(t) = \sum_{l=0}^{N-2} a_l \Phi_l(t), \quad t \in I, \tag{81}$$

⁴ Other sets of basis functions are possible, e.g. $\mathcal{B}_2 := \{(1 - t^2)T_l(t)\}_{l=0, \dots, N-2}$, (see [35] for more details on this choice).

which we insert into eq. (77) to find

$$\begin{aligned} \sum_{l=0}^{N-2} a_l \left(\int_I (t+1)\Phi'_l(t)(\Phi_l\omega)'(t) dt + \int_I \frac{1}{t+1}\Phi_l(t)\Phi_i(t)\omega(t) dt + \tilde{k}^2 \int_I \frac{(t+1)}{4}\Phi_l(t)\Phi_i(t)\omega(t) dt \right) \\ = - \int_I h(t)\Phi_i(t)\omega(t) dt = - \sum_{j=0}^N \hat{g}_{kj} \left(\int_I \frac{1}{4}(t+1)T_j(t)\Phi_i(t)\omega(t) dt \right), \end{aligned} \tag{82}$$

for each $i = 0, \dots, N - 2$. In the last line we have inserted the Chebyshev approximation of g in order to approximate the integral on the RHS. In fact, since we know the values of \hat{g}_k on the shifted Chebyshev nodes, we have $\hat{g}_k(r) = \sum_{j=0}^N \hat{g}_{kj}T_j^*(r)$ and, therefore, $\hat{g}_k((t+1)/2) = \sum_{j=0}^N \hat{g}_{kj}T_j(t)$. To complete the Galerkin scheme CG_1^{Dir} , we solve the following linear system of equations

$$(A + B - \tilde{k}^2 C) \cdot (a_l)_l = -D \cdot (\hat{g}_{kj})_j, \tag{83}$$

where the entries of the solving matrices ($A = (A_{i,l})$) are given by

$$\begin{aligned} A_{i,l} &= \int_{-1}^1 (t+1)\Phi'_l(t)(\Phi_i\omega)'(t) dt, \\ B_{i,l} &= \int_{-1}^1 \frac{1}{t+1}\Phi_l(t)\Phi_i(t)\omega(t) dt, \\ C_{i,l} &= \int_{-1}^1 \frac{1}{4}(t+1)\Phi_l(t)\Phi_i(t)\omega(t) dt, \\ D_{i,j} &= \int_{-1}^1 \frac{1}{4}(t+1)T_j(t)\Phi_i(t)\omega(t) dt, \end{aligned} \tag{84}$$

for $i, l = 0, \dots, N - 2$ and $j = 0, \dots, N$. The above integrals integrate exactly and have to be computed only once. The entries of the first three matrices are stated in [57] and are not repeated here. The entries of D , which are not stated in [57], read

$$D_{i,j} = \begin{cases} (\delta_{i1} + 1)\pi/16, & j = i - 1, \\ (\delta_{i0} + 1)\pi/8, & j = i, \\ \delta_{i0}\pi/16, & j = i + 1, \\ -\pi/8, & j = i + 2, \\ -\pi/16, & j = i + 3, \end{cases}$$

for $i = 0, \dots, N - 2$ and $j = 0, \dots, N$. The system (83) is solved for a solution in the form (81) and transformed to the solution \hat{w}_k^N of eq. (68) in form of a shifted Chebyshev series, namely $\sum_{j=0}^N \hat{w}_{kj}T_j^*$, in using the variable transform $t = 2r - 1, r \in [0, 1]$ and the coefficient transform

$$\hat{w}_{kj} = a_j - a_{j-2}, \quad j = 0, \dots, N,$$

where we set $a_{-2} = a_{-1} = a_{N-1} = a_N = 0$.

Let us come to the Poisson problem with Neumann boundary conditions (67). The approach is identical to the Dirichlet case and, hence, abbreviated. The same procedure as before turns eq. (67) into

$$\begin{aligned} ((\cdot + 1)z')'(t) - \frac{\tilde{k}^2}{4}(t+1)z(t) = l(t), t \in I, \\ z'(-1) = z'(1) = 0, \end{aligned} \tag{85}$$

with $z(t) = \hat{u}_k((t+1)/2)$ and $l(t) = \frac{1}{4}(t+1)\hat{f}_k((t+1)/2)$, $k \in [0, M/2]$ fixed. A variational formulation of the latter reads

$$\int_I (t+1)z'(t)(v\omega)'(t) dt + \tilde{k}^2 \int_I \frac{1}{4}(t+1)z(t)v(t)\omega(t) dt = - \int_I h(t)v(t)\omega(t) dt, \tag{86}$$

for $v \in Y := \{\theta \in H^1_\omega(I) \mid \partial\theta(-1) = \partial\theta(1) = 0\}$. We look for an approximation z^N , $N \in \mathbb{N}$, of the solution z in a finite dimensional polynomial subspace $Y^N := \{p \in P_N \mid p'(-1) = p'(1) = 0\} \subset Y$, that is spanned by the basis functions

$$\Psi_l := T_l - \gamma_l T_{l+2}, \quad l = 0, \dots, N - 2, \tag{87}$$

where

$$\gamma_l = \frac{l^2}{(2+l)^2}.$$

Inserting the basis functions together with the representations $z(t) = \sum_{l=0}^{N-2} z_l \Psi_l(t)$ and $\hat{f}_k((t+1)/2) = \sum_{j=0}^N \hat{f}_{kj} T_j(t)$ into eq. (86) entails the system of linear equations

$$(E + \tilde{k}^2 F) \cdot (z_l)_{l \leq N-2} = G \cdot (\hat{f}_{kj})_{j \leq N}, \tag{88}$$

with matrices $E, F \in \mathbb{R}^{N-2 \times N-2}$ and $G \in \mathbb{R}^{N-2 \times N}$, whose entries are defined by

$$\begin{aligned} E_{i,l} &= \int_{-1}^1 (t+1) \Psi_i'(t) (\Psi_l \omega)'(t) dt, \\ F_{i,l} &= \int_{-1}^1 \frac{1}{4} (t+1) \Psi_i(t) \Psi_l(t) \omega(t) dt, \\ G_{i,l} &= \int_{-1}^1 \frac{(t+1)}{4} T_j(t) \Psi_i(t) \omega(t) dt. \end{aligned}$$

Once eq. (88) is solved, we find the Chebyshev series representation of the approximated solution $\hat{u}_k^N = \sum_{j=0}^N \hat{u}_{kj} T_j^*$ in using the coefficient transform

$$\hat{u}_{kj} = z_j - \gamma_{j-2} z_{j-2}, \quad j = 0, \dots, N, \tag{89}$$

with $\gamma_j = 0$ for $j = -2, -1, N - 1, N$. It is rather tedious to find the exact values of the matrix entries, yet they can also be computed via Chebyshev-Gauss integration. This quadrature rule is accurate for polynomials with a maximal degree of $2n - 1$, where n is the number of nodes on which the integrands are evaluated (see eq. (190) in Appendix A). As a matter of fact, the above expressions are weighted integrals over polynomials $(t+1)T_i T_j = \frac{1}{2}(t+1)(T_{i+j} + T_{|i-j|})$. Therefore, those polynomials are of maximal degree $i + j + 1$ and, for the application of the Chebyshev-Gauss quadrature, it suffices to evaluate the integrands on $n = (i + j)/2 + 1$ Chebyshev nodes (i.e. on Chebyshev zeros). The matrices have to be computed only once and to make sure that the values are correctly stored and to minimize rounding errors, the matrices in the present implementation are computed in quadruple precision and rounded to working precision. Also, the programming effort reduces to a minimum, because this method can be applied to all the preceding matrices in this section. Moreover, once computed for a high number of rows and columns, the matrices can be saved to disc in quadruple precision and read in for an extraction of any inferior or equal (Chebyshev-)dimension.

A slight **modification** of the Chebyshev-Galerkin solvers is undertaken in order to solve directly for the second vertical derivative of the periodic solution $u = u(r, z)$, resp. $w = w(r, z)$. By the help of this modification, one may, subsequently, use the methods of inverted operators from further above to solve directly for radial derivatives. Instead of solving for the modes \hat{u}_k , the solution of eq. (67), we will solve directly for the terms $-\tilde{k}^2 \hat{u}_k$, which are the Fourier modes of the second vertical derivative of u . This is readily achieved in dividing the solving matrix by $-\tilde{k}^2$. More precisely, for the Neumann problem we will solve, instead of eq. (88),

$$Q \cdot (\tilde{a}_l)_l = G \cdot (\hat{f}_{kj})_{j \leq N}, \tag{90}$$

where $Q = -(E + \tilde{k}^2 F)/\tilde{k}^2$ and $\tilde{a}_l = -\tilde{k}^2 a_l$. Note that only the solving matrix is modified and not the RHS of eq. (67). Once we have solved eq. (90) for all $k \in [0, M/2]$, the second vertical derivative of u may be back-transformed to physical space and subtracted from the RHS function to entail

$$\mathcal{L}_1 u^M(r, z) = f^M(r, z) - \partial_z^2 u^M(r, z) =: \tilde{f}^M(r, z). \tag{91}$$

The latter equation may be accurately solved for radial derivatives and divisions by means of the methods $\mathcal{L}_1^{-1, (2)}$ and $\mathcal{L}_1^{-1, (1)/r}$ from before. Exactly the same procedure may be applied to the Dirichlet boundary problem eq. (68) with the associated linear system eq. (88). One solves

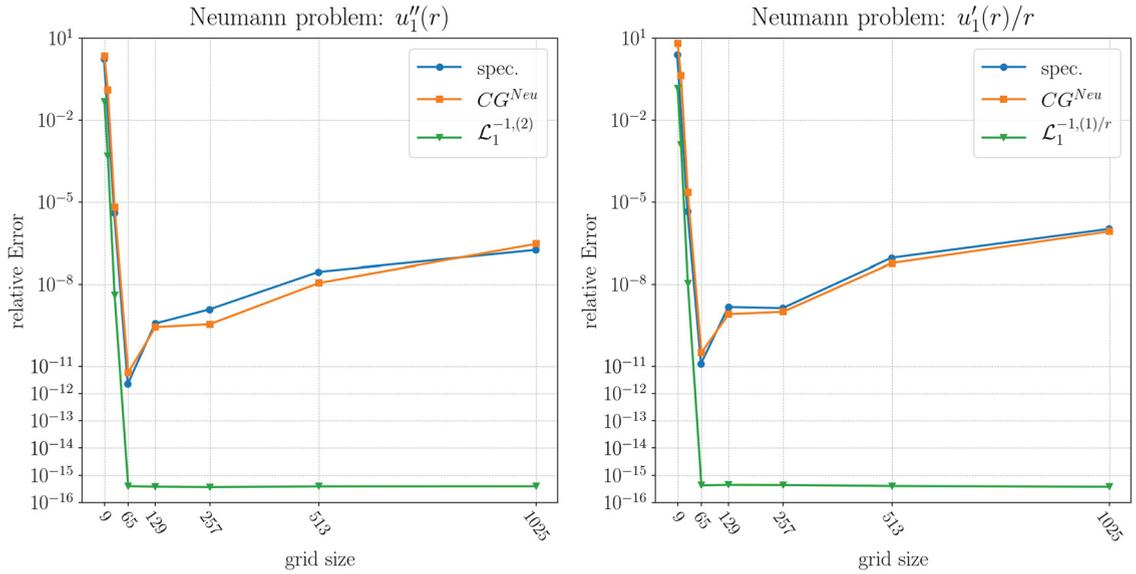


Fig. 1. Convergence and accuracy verification for the methods $\mathcal{L}_1^{-1,(2)}$ and $\mathcal{L}_1^{-1,(1)/r}$ with the test function $u_1(r) = e^{1-\cos(2\pi r)} - 1$. The RHS function is analytically calculated as $f_1(r) = \mathcal{L}_1 u_1(r)$ and implemented exactly, the relative errors are computed in the l^2 -norm on the fixed grid. The legend references the graphs by the method that was used to obtain them. Here, the CG^{Neu} method was used to solve $\mathcal{L}_1 u_1 = f_1$ for the solution itself, to which standard spectral differentiation and division methods (see Appendix A) were applied. Those spectral differentiation and division methods, applied directly to the implemented function u_1 (no solving is involved, u_1 is directly projected into the finite dimensional Chebyshev-Fourier space), are referenced with spec. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\left(-\tilde{k}^{-2}(A + B - \tilde{k}^2 C)\right) \cdot (\tilde{z}_l)_{l \leq N-2} = -D \cdot (\hat{g}_{kj})_j, \tag{92}$$

for $\tilde{z}_l = -\tilde{k}^2 z_l$, and obtains the radial divisions and derivatives via

$$\mathcal{L}_2 w^M(r, z) = g^M(r, z) - \partial_z^2 w^M(r, z) =: \bar{g}^M(r, z). \tag{93}$$

Note that the term $\tilde{k} = 2\pi k/L$ may become very large, especially for large grid sizes and small lengths of periodicity $L > 0$, and it may quickly dominate eqs. (67) and (68). Therefore, a back-transform of the second vertical derivative to physical space before subtraction from the RHS function is recommended. As a matter of fact, the condition number of the solving matrix is not altered by the division and, hence, the system is solved as accurately as for the solution itself.

Notation. The methods that divide the solving matrices by $-\tilde{k}^2$ and, subsequently, apply the methods of inverted operators for the treatment of eqs. (91) and (93) will be referred to by $CG^{Dir,(1)}$, $CG^{Dir,(0)/r}$, $CG^{Neu,(2)}$ and $CG^{Neu,(1)/r}$.

Fig. 1 shows a comparison between the Chebyshev-Galerkin solver CG^{Neu} from this section and the method of inverted operators for \mathcal{L}_1^{-1} in the particular case $\tilde{k} = 0$. Of course, in this particular case, there is no division by $-\tilde{k}^2$ in the solving matrices.

As visualized in the referenced figure, the solving method for \mathcal{L}_1^{-1} produces the needed expressions close to machine precision. This implies for eq. (91) that one obtains the radial derivatives and division by r close to the accuracy of the formerly found second vertical derivative, which is in principle as accurate as the solution without the division by $-\tilde{k}^2$, because of the unaltered condition number.

5.3. Bessel-Chebyshev solver based on exact solutions to truncated problems

Initially, set the focus on the homogeneous Neumann Problem (64). Here, we will state the exact analytical solution for an approximated RHS and propose an algorithm that will give rise to a very precise approximation to the discrete solution and its derivatives. Moreover, as we have the analytical solutions at hand, we can get a formula for the second derivatives in a straightforward way and obtain a precise approximation of the latter. The resulting algorithm for this method is of complexity $O(N^2)$ for each (relevant) Fourier mode, but its precision is striking, and efficient parallelization of the code assures the applicability of the method.

Let us depart directly from the Fourier truncated problem eq. (67), namely $(\mathcal{L}_1 - \tilde{k}^2)\hat{u}_k(r) = \hat{u}_k''(r) + \frac{1}{r}\hat{u}_k'(r) - \tilde{k}^2\hat{u}_k(r) = \hat{f}_k(r)$, $\hat{u}_k'(0) = \hat{u}_k'(1) = 0$, where $\tilde{k} = 2\pi k/L$, $k \in [0, M/2]$.

The case where $k = 0$, has been treated further above, so let us fix $k > 0$ in the following. If $k > 0$, then eq. (67) is well known as the modified Bessel ODE of order 0. Its fundamental solutions are the modified Bessel functions of order 0, namely $I_0(\tilde{k} \cdot)$ and $K_0(\tilde{k} \cdot)$.

Notation. In order to increase readability, for $j = 0, 1$, we will abbreviate

$$\begin{aligned} \mathcal{J}[I_j h](\tilde{k}, r) &:= \int_0^r t I_j(\tilde{k}t) h(t) dt, \\ \mathcal{J}[K_j h](\tilde{k}, r) &:= \int_0^r t K_j(\tilde{k}t) h(t) dt, \end{aligned} \tag{94}$$

for a generic function h . If $h = h(x)$ is a power function with exponent n , then we will write x^n in place of h , i.e. $\mathcal{J}[I_j x^n]$ and $\mathcal{J}[K_j x^n]$.

By an application of the method of variation of parameters (e.g. see [40]), under consideration of the homogeneous boundary conditions, one obtains the complete analytical solution to problem (64) as

$$\hat{u}_k(r) = I_0(\tilde{k}r) \cdot \mathcal{J}[K_0 \hat{f}_k](\tilde{k}, r) - K_0(\tilde{k}r) \cdot \mathcal{J}[I_0 \hat{f}_k](\tilde{k}, r) + c_k I_0(\tilde{k}r). \tag{95}$$

The solution can be verified by a direct insertion into the differential equation. The exact first order derivatives read

$$\hat{u}'_k(r) = \tilde{k} I_1(\tilde{k}r) \left(\mathcal{J}[K_0 \hat{f}_k](\tilde{k}, r) + c_k \right) + \tilde{k} K_1(\tilde{k}r) \mathcal{J}[I_0 \hat{f}_k](\tilde{k}, r) \tag{96}$$

$$\begin{aligned} \hat{u}''_k(r) = \frac{1}{2} \tilde{k} \left(-\tilde{k} (K_0(\tilde{k}r) + K_2(\tilde{k}r)) \mathcal{J}[I_0 \hat{f}_k](\tilde{k}, r) + \tilde{k} (I_0(\tilde{k}r) + I_2(\tilde{k}r)) \left(\mathcal{J}[K_0 \hat{f}_k](\tilde{k}, r) + c_k \right) \right. \\ \left. + 2r \hat{f}_k(r) \left(I_0(\tilde{k}r) K_1(\tilde{k}r) + I_1(\tilde{k}r) K_0(\tilde{k}r) \right) \right). \end{aligned} \tag{97}$$

Using eq. (96), one easily shows that

$$c_k = -\mathcal{J}[K_0 \hat{f}_k](\tilde{k}, 1) + \frac{K_1(\tilde{k})}{I_1(\tilde{k})} \cdot \mathcal{J}[I_0 \hat{f}_k](\tilde{k}, 1)$$

ensures the Neumann homogeneity. Subsequently, we approximate the Fourier mode \hat{f}_k by a truncated shifted Chebyshev series,

$$\hat{f}_k(r) = \sum_{l=0}^N \hat{f}_{kl} T_l^*(r), \tag{98}$$

and insert it into the precedingly found formulas. Since the expressions for the solution, and derivatives thereof, are linear in \hat{f}_k , we may pull out the sum and conveniently write

$$\hat{u}_k(r) = \sum_{l=0}^N \hat{f}_{kl} \mathcal{N}^{(0)}(\tilde{k}, l, r), \tag{99}$$

where

$$\mathcal{N}^{(0)}(\tilde{k}, l, r) := I_0(\tilde{k}r) \cdot \mathcal{J}[K_0 T_l^*](\tilde{k}, r) - K_0(\tilde{k}r) \cdot \mathcal{J}[I_0 T_l^*](\tilde{k}, r) + c_k^* I_0(\tilde{k}r), \tag{100}$$

and

$$c_k^* = -\mathcal{J}[K_0 T_l^*](\tilde{k}, 1) + \frac{K_1(\tilde{k})}{I_1(\tilde{k})} \cdot \mathcal{J}[I_0 T_l^*](\tilde{k}, 1).$$

Similarly, we find expressions for the derivatives, namely

$$\hat{u}'_k(r) = \sum_{l=0}^N \hat{f}_{kl} \mathcal{N}^{(1)}(\tilde{k}, l, r), \tag{101}$$

$$\hat{u}''_k(r) = \sum_{l=0}^N \hat{f}_{kl} \mathcal{N}^{(2)}(\tilde{k}, l, r), \tag{102}$$

with

$$\mathcal{N}^{(1)}(\tilde{k}, l, r) := \tilde{k} \left(I_1(\tilde{k}r) \cdot \mathcal{J}[K_0 T_l^*](\tilde{k}, r) - K_1(\tilde{k}r) \cdot \mathcal{J}[I_0 T_l^*](\tilde{k}, r) + c_k^* I_1(\tilde{k}r) \right), \tag{103}$$

$$\begin{aligned} \mathcal{N}^{(2)}(\tilde{k}, l, r) := & \frac{1}{2} \tilde{k} \left(-\tilde{k}(K_0(\tilde{k}r) + K_2(\tilde{k}r)) \mathcal{J}[I_0 T_l^*](\tilde{k}, r) \right. \\ & + \tilde{k}(I_0(\tilde{k}r) + I_2(\tilde{k}r)) \left(\mathcal{J}[K_0 T_l^*](\tilde{k}, r) + c_k^* \right) \\ & \left. + 2r T_l^*(r) \left(I_0(\tilde{k}r) K_1(\tilde{k}r) + I_1(\tilde{k}r) K_0(\tilde{k}r) \right) \right). \end{aligned} \tag{104}$$

To obtain the expression $\hat{u}'_k(r)/r$, eq. (103) and eq. (101) are directly divided by r . We extend the definitions of the $\mathcal{N}^{(j)}$ by the $k=0$ terms in using the exact form of \mathcal{L}_1^{-1} in eq. (71) (here in an integrated form):

$$\begin{aligned} \mathcal{N}^{(0)}(0, l, r) &:= \ln(r) \left(\int_0^r y T_l^*(y) dy \right) - \int_0^r y T_l^*(y) \ln(y) dy \\ \mathcal{N}^{(1)}(0, l, r) &:= \frac{\int_0^r y T_l^*(y) dy}{r} \\ \mathcal{N}^{(2)}(0, l, r) &:= T_l^*(r) - \frac{\int_0^r y T_l^*(y) dy}{r^2}. \end{aligned}$$

For the first derivative divided by r we define furthermore

$$\mathcal{N}^{(1)/r} := \frac{1}{r} \mathcal{N}^{(1)}. \tag{105}$$

Hence, given a Chebyshev-Fourier approximation of the RHS of (64) in the domain $[0, 1] \times [0, L]$, with modes \hat{f}_{kl} , the derivatives of the solution $u^{MN} = u^{MN}(r, z)$ to the spectrally truncated Neumann problem take the form

$$\partial_r^j u^{MN}(r, z) = \sum_{|k|=0}^M \sum_{l=0}^N \hat{f}_{kl} \mathcal{N}^{(j)}(\tilde{k}, l, r) e^{ikz}, \quad j = 0, 1, 2. \tag{106}$$

Equally, the first derivative divided by r is given by

$$\frac{1}{r} \partial_r u^{MN}(r, z) = \sum_{|k|=0}^M \sum_{l=0}^N \hat{f}_{kl} \mathcal{N}^{(1)/r}(\tilde{k}, l, r) e^{ikz}. \tag{107}$$

Notation. These particular methods of matrix-vector multiplications to obtain $\partial_r^2 u_k^{MN}(r, z)$ and $\partial_r u_k^{MN}(r, z)/r$ will be referred to as the Bessel-Chebyshev (BC) method for the Neumann problem and denoted by $\mathcal{N}^{(2)}$ and $\mathcal{N}^{(1)/r}$ respectively. Other expressions do not need to be computed in the current implementation of the CLA.

Let us come to the Dirichlet problem (65), for which we proceed in the same manner as for the Neumann problem before. We recall the truncated problem eq. (68), for all $k \in [0, M/2]$: $(\mathcal{L}_2 - \tilde{k}^2) \hat{w}_k(r) = \hat{w}'_k(r) + \frac{1}{r} \hat{w}'_k(r) - (\frac{1}{r^2} + \tilde{k}^2) \hat{w}_k(r) = \hat{g}_k(r)$, $\hat{w}_k(0) = \hat{w}_k(1) = 0$, where we further assume that $\hat{g}_k(r)$ is approximated by a truncated Chebyshev series $\sum_{l=0}^N \hat{g}_{kl} T_l^*(r)$.

• **For $k=0$** , the operator $\mathcal{L}_2 = \partial_r (\frac{1}{r} \partial_r (\cdot))$ integrates exactly, as already seen in eq. (72), to give

$$\begin{aligned} \hat{w}_0(r) &= \frac{\int_0^r t \left(c + \int_0^t \hat{g}_0(x) dx \right) dt}{r} = \sum_{l=0}^N \hat{g}_{0l} \frac{\int_0^r t \left(c_l + \int_0^t T_l^*(x) dx \right) dt}{r} \\ &=: \sum_{l=0}^N \hat{g}_{0l} \mathcal{D}^{(0)}(0, l, r), \\ \partial_r \hat{w}_0(r) &= \sum_{l=0}^N \hat{g}_{0l} \left(-\frac{\int_0^r t \left(c_l + \int_0^t T_l^*(x) dx \right) dt}{r^2} + c_l + \int_0^r T_l^*(x) dx \right) \end{aligned} \tag{108}$$

$$\begin{aligned}
 &=: \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(1)}(0, l, r), \\
 \partial_r^2 \hat{w}_0(r) &= \sum_{l=0}^N \hat{g}_{0l} \frac{1}{r} \left(\frac{2 \int_0^r t \left(c_l + \int_0^t T_l^*(x) dx \right) dt}{r^2} - c_l - \int_0^r T_l^*(x) dx \right) + T_l^*(r) \\
 &=: \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(2)}(0, l, r),
 \end{aligned} \tag{109}$$

where

$$c = -2 \int_0^1 t \int_0^t \hat{g}_0(x) dx dt \quad \text{and} \quad c_l = -2 \int_0^1 t \int_0^t T_l^*(x) dx dt .$$

• **For $k > 0$,** eq. (68) is the modified Bessel ODE of first kind, whose fundamental solutions are I_1 and K_1 . The particular solutions to the homogeneous Dirichlet problem eq. (68) are, by the method of variation of parameters,

$$\hat{w}_k(r) = I_1(\tilde{k}r) \left(\mathcal{J}[K_1 \hat{g}_k](\tilde{k}, r) + d_k \right) - K_1(\tilde{k}r) \mathcal{J}[I_1 \hat{g}_k](\tilde{k}, r), \tag{110}$$

with

$$d_k = -\mathcal{J}[K_1 \hat{g}_k](\tilde{k}, 1) + \frac{K_1(\tilde{k})}{I_1(\tilde{k})} \cdot \mathcal{J}[I_1 \hat{g}_k](\tilde{k}, 1)$$

to enforce homogeneity at $r = 1$. An insertion of the shifted Chebyshev representation of $\hat{g}_k(r)$ and subsequent simplifications, which exploit the linearity of the involved integrals, give

$$\hat{w}_k(r) = \sum_{l=0}^N \hat{g}_{kl} (I_1(\tilde{k}r) \left(\mathcal{J}[K_1 T_l^*](\tilde{k}, r) + d_k^* \right) - K_1(\tilde{k}r) \mathcal{J}[I_1 T_l^*](\tilde{k}, r)) =: \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(0)}(\tilde{k}, l, r), \tag{111}$$

with

$$d_k^* = -\mathcal{J}[K_1 T_l^*](\tilde{k}, 1) + \frac{K_1(\tilde{k})}{I_1(\tilde{k})} \cdot \mathcal{J}[I_1 T_l^*](\tilde{k}, 1) .$$

The same procedure applied to the derivatives of the analytic solution produces

$$\begin{aligned}
 \partial_r \hat{w}_k(r) &= \sum_{l=0}^N \hat{g}_{kl} \left(\frac{\tilde{k}}{2} ((I_0(\tilde{k}r) + I_2(\tilde{k}r)) \left(\mathcal{J}[K_1 T_l^*](\tilde{k}, r) + d_k^* \right) \right. \\
 &\quad \left. + (K_0(\tilde{k}r) + K_2(\tilde{k}r)) \mathcal{J}[I_1 T_l^*](\tilde{k}, r) \right) \\
 &=: \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(1)}(\tilde{k}, l, r) \\
 \partial_r^2 \hat{w}_k(r) &= \sum_{l=0}^N \hat{g}_{kl} \frac{\tilde{k}}{4} \left[-\tilde{k} (3K_1(\tilde{k}r) + K_3(\tilde{k}r)) \mathcal{J}[I_1 T_l^*](\tilde{k}, r) \right. \\
 &\quad \left. + \tilde{k} (3I_1(\tilde{k}r) + I_3(\tilde{k}r)) \left(\mathcal{J}[K_1 T_l^*](\tilde{k}, r) + d_k^* \right) \right. \\
 &\quad \left. + 2r T_l^*(r) (I_1(\tilde{k}r) K_0(\tilde{k}r) + I_1(\tilde{k}r) K_2(\tilde{k}r) \right. \\
 &\quad \left. + I_0(\tilde{k}r) K_1(\tilde{k}r) + I_2(\tilde{k}r) K_1(\tilde{k}r)) \right] \\
 &=: \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(2)}(\tilde{k}, l, r) .
 \end{aligned}$$

To obtain the expression $\hat{w}_k(r)/r$, eq. (111) is directly divided by r , such that we may define

$$\mathcal{D}^{(0)/r} := \frac{1}{r} \mathcal{D}^{(0)}. \tag{112}$$

We have found the representations of the exact derivatives of the solution $w^{MN} = w^{MN}(r, z)$ to the spectrally discretized version of eq. (65) in the form

$$\partial_r^j w^{MN}(r, z) = \sum_{|k|=0}^M \sum_{l=0}^N \hat{g}_{kl} \mathcal{D}^{(j)}(\tilde{k}, l, r) e^{ikz}, \quad j = 0, 1, 2, \tag{113}$$

where $\mathcal{D}^{(0)/r}$ replaces $\mathcal{D}^{(j)}$ to obtain $w^{MN}(r, z)/r$.

Notation. We will refer to the particular methods of matrix-vector multiplication to obtain $\partial_r w^{MN}(r, z)$ and $w^{MN}(r, z)/r$ as $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(0)/r}$ respectively. We will refer to them as Bessel-Chebyshev (BC) methods for the Dirichlet problem. Other expressions do not need to be computed in the current implementation of the CLA.

Computation of the solving matrices Above we have stated the exact solutions and their derivatives, eqs. (106) and (113), to the truncated problems eqs. (67) and (68) respectively. In order to calculate them numerically, the exact values of the expressions $\mathcal{N}^{(\cdot)}$ and $\mathcal{D}^{(\cdot)}$ are needed, where (\cdot) can be any of (0), (1), (2), (1)/r and (0)/r as defined above. Given a discrete set of points $\{r_0, \dots, r_N\} \subset [0, 1]$, we aim at the computation of the 3-dimensional cubic matrices $N^{(\cdot)}$ and $D^{(\cdot)}$ with entries

$$\begin{aligned} N_{kli}^{(\cdot)} &:= \mathcal{N}^{(\cdot)}\left(\frac{2\pi}{L}k, l, r_i\right), \\ D_{kli}^{(\cdot)} &:= \mathcal{D}^{(\cdot)}\left(\frac{2\pi}{L}k, l, r_i\right), \end{aligned} \tag{114}$$

for $k = 0, \dots, M/2$ and $l, i = 0, \dots, N$, where $M + 1$ and $N + 1$ are the Fourier and Chebyshev dimensions respectively. However, not all matrices have to be computed, if only a particular derivative is needed. Also, inferior orders of differentiation may be obtained by integration of a higher derivative if more orders are needed. The latter only works provided a supplementary function value in the interval is known. For instance, one could integrate the second derivative of the Neumann solution to obtain the first derivative, in knowing that the last must vanish at $r = 0$ to keep the operator \mathcal{L}_1 bounded. Once the matrices $N^{(\cdot)}, D^{(\cdot)}$ are computed, one may save them to the disc and reuse them any number of times. We will see in the following that the numerical computation of the above entries is by no means straightforward or low in complexity and a non-negligible effort has been employed to obtain them. Therefore, the needed matrices have been computed for a moderately high number of Chebyshev-Fourier modes on the nested grid $\{r_i\}_{0 \leq i \leq N}$ of the shifted Chebyshev-extrema. This enables us to use the matrix for $M/2 = N = 512$ modes (513^3 entries) to solve on all $(2^n + 1) \times 2^m$ with $n \leq 9, m \leq 10$, physical Chebyshev-Fourier sub-grids of 513×1024 points. Note that the odd numbers $2^n + 1$ have been chosen deliberately because the underlying fast Chebyshev transform on the Chebyshev-extrema grid, namely the DCT-I in FFTW3, shows the best performance for these numbers (see [23,22] for details on the FFTW3).

The most involved part of the matrix computations is the evaluation of the integrals in (94). The modified Bessel functions expose such extreme behavior at $r = 0$ and $r = 1$ that any standard quadrature formula even in quadruple precision is insufficient to find accurate values. The functions $K_j, j = 0, 1, \dots$ possess (logarithmic) singularities at the pole $r = 0$ and $I_j, j = 0, 1, \dots$ grow exponentially for large arguments ($\tilde{k}r$). Also, these integrals need to be evaluated to a very large number of digits, because the summands in the exact analytical solutions and their derivatives show violent cancelations. To be sure to calculate all entries correctly to more than 36 digits, a precision of roughly $\frac{4}{5} \frac{2\pi}{L} k$ digits is needed, being especially costly for small heights L or large k , but the cancelations do dependent on the actual values of $\frac{2\pi}{L} k, l$ and r so that the working precision may be adapted. The amount of needed computational power and time seems to be in no proportion to the seemingly standard goal of inverting a Calderon-Zygmund operator of order zero. For very accurate computations of incompressible Euler flow, with a possible goal of detecting finite-time singularities, however, this one-time effort might be appropriate.

Because of steep singularities at $r = 0$ and exponential growth at $r = 1$, the integrals in (94) are not amenable to treatment by standard quadrature schemes and we will thus proceed in mathematical theory, which will allow for a direct evaluation of the latter integrals in terms of Hypergeometrical and Meijer-G functions.

Lemma 1. *The following holds for $\tilde{k} = \frac{2\pi}{L}k, n \in \mathbb{N}_0$ and $r \in [0, 1]$,*

$$\begin{aligned} \mathcal{J}[I_0 x^n](\tilde{k}, r) &= \int_0^r I_0(\tilde{k}t) t^{n+1} dt \\ &= \frac{r^{n+2} {}_1F_2\left(\frac{n}{2} + 1; 1, \frac{n}{2} + 2; \frac{\tilde{k}^2 r^2}{4}\right)}{n + 2}, \end{aligned} \tag{115}$$

$$\begin{aligned} \mathcal{J}[K_0 x^n](\tilde{k}, r) &= \int_0^r K_0(\tilde{k}t)t^{n+1} dt \\ &= \frac{r^{n+2}}{(n+2)^2} \left[(n+2)K_0(\tilde{k}r) {}_1F_2 \left(1; \frac{n}{2} + 1, \frac{n}{2} + 2; \frac{\tilde{k}^2 r^2}{4} \right) \right. \\ &\quad \left. + \tilde{k}r K_1(\tilde{k}r) {}_1F_2 \left(1; \frac{n}{2} + 2, \frac{n}{2} + 2; \frac{\tilde{k}^2 r^2}{4} \right) \right], \end{aligned} \tag{116}$$

$$\begin{aligned} \mathcal{J}[I_1 x^n](\tilde{k}, r) &= \int_0^r I_1(\tilde{k}t)t^{n+1} dt \\ &= \left[\frac{t^{n+1}}{\tilde{k}} I_0(\tilde{k}t) \right]_{t=0}^{t=r} - (n+1)\mathcal{J}[I_0 x^n](\tilde{k}, r) \end{aligned} \tag{117}$$

$$= \frac{1}{4} \tilde{k} r^{n+3} \frac{\Gamma\left(\frac{n+3}{2}\right)}{\Gamma\left(\frac{n+5}{2}\right)} {}_1F_2 \left(\frac{n+3}{2}; 2, \frac{n+5}{2}; \frac{\tilde{k}^2 r^2}{4} \right), \tag{118}$$

$$\begin{aligned} \mathcal{J}[K_1 x^n](\tilde{k}, r) &= \int_0^r K_1(\tilde{k}t)t^{j+1} dt \\ &= \frac{1}{4} r^{n+2} G_{1,3}^{2,1} \left(\frac{\tilde{k}r}{2}, \frac{1}{2}; -\frac{1}{2}, \frac{1}{2}, -\frac{n}{2} - 1 \right) \end{aligned} \tag{119}$$

$$\stackrel{(n \geq 0)}{=} \left[-\frac{t^{n+1}}{\tilde{k}} K_0(\tilde{k}t) \right]_{t=0}^{t=r} + (n+1)\mathcal{J}[K_0 x^n](\tilde{k}, r), \tag{120}$$

$$\mathcal{J}[K_1 x^0](\tilde{k}, r) = \frac{\pi r (L_0(kr)K_1(kr) + L_1(kr)K_0(kr))}{2k}, \tag{121}$$

where ${}_1F_2$ is a generalized hypergeometric function, G the Meijer-G function and L the Struve-L function, see [50] for a comprehensive introduction to these special functions.

Proof. The identities eqs. (117) and (120) are obtained by partial integration. Equation (121) is obtained from eq. (6.561-4) on page 676 in [37] by setting $a = kr$ and a change of variables. Equations (115) and (118) are adapted from eq. (84) on page 198 in [2] again via a change of variables. Equations (116) and (119) originate from eq. (6.592-2) on page 690 in [37] and references therein, see also [49] page 365. These identities have additionally been verified by Mathematica's symbolic integration and via its arbitrary precision integration and evaluation for arbitrarily chosen values of k, r and n . \square

We may now use the explicit polynomial expression (183) of a shifted Chebyshev polynomial $T_l^*(r) = T_l(2r - 1) = \sum_{j=0}^l a_{j,l} r^j$, where

$$a_{j,l} = \begin{cases} 1, & l = 0, j = 0, \\ \frac{4^j l! (-1)^{l-j} (j+l-1)!}{(2j)! (l-j)!}, & l > 0, j = 0, \dots, l, \end{cases}$$

and find the explicit formulas for the integrals in (94) by inserting I_0, I_1, K_0 and K_1 into

$$\mathcal{J}[\cdot T_l^*](\tilde{k}, r) = \sum_{j=0}^l a_{j,l} \mathcal{J}[\cdot x^j](\tilde{k}, r). \tag{122}$$

The cubic matrices $N^{(\cdot)}$ and $D^{(\cdot)}$ from eq. (114) can then be obtained as a sum of special functions which are evaluated to a high enough precision. The working precision for these computations has to be large enough to provide for the range of the cancelations. The evaluation of the involved special functions is very costly and can only be achieved by modern arbitrary precision packages, which have the needed functions implemented. An efficient parallelization is also mandatory for moderate to high resolutions. To reduce the computational amount, eqs. (117) and (120) are implemented and re-use the entries of $\mathcal{J}[I_0 x^n](\tilde{k}, r)$ and $\mathcal{J}[K_0 x^n](\tilde{k}, r), n \geq 1$. Currently, eq. (119) is implemented for $\mathcal{J}[K_0 x^n](\tilde{k}, r), n = 0$, but eq. (121) is faster and the involved Struve-L functions are easier to find as components of arbitrary precision libraries. The Library

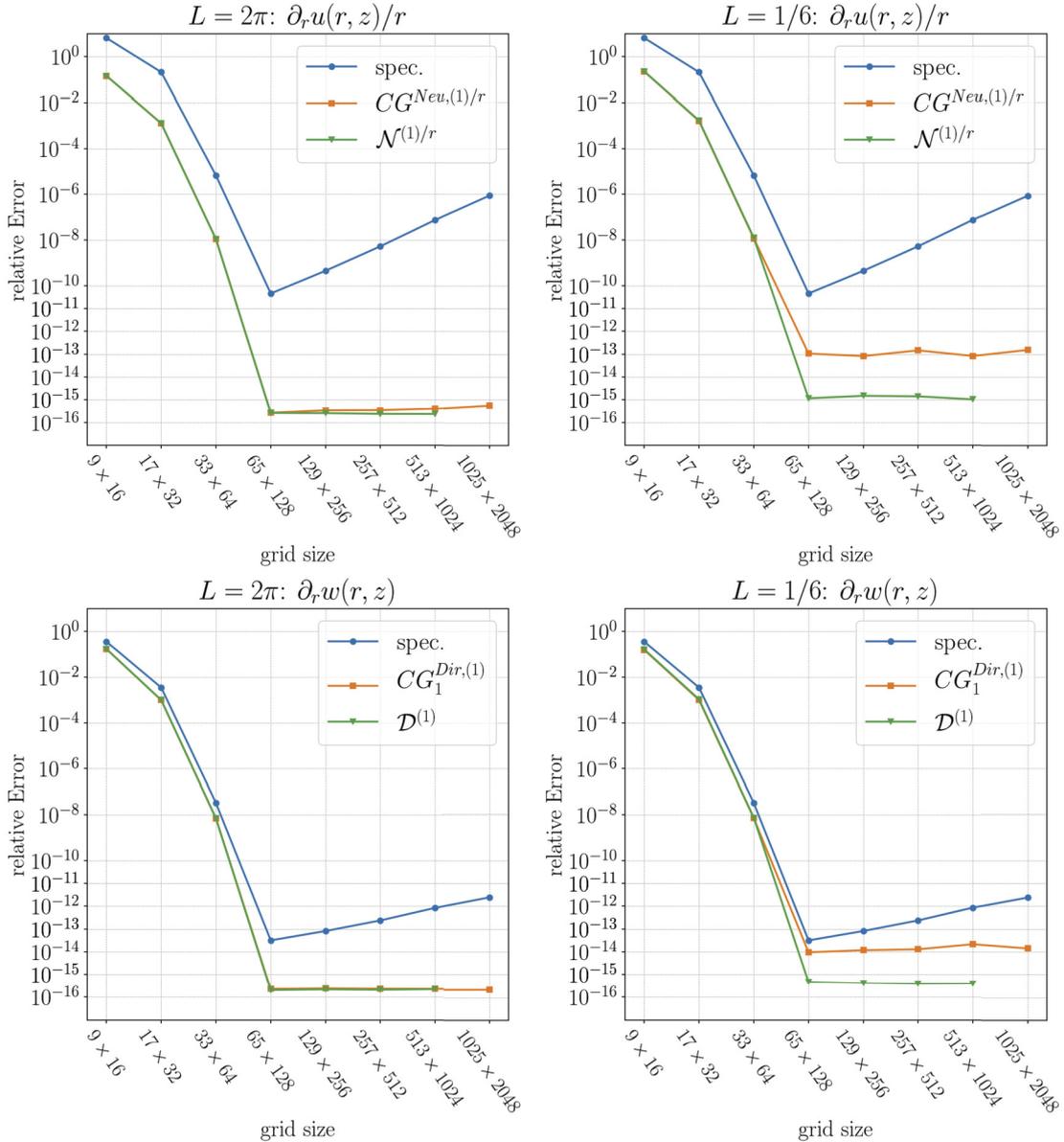


Fig. 2. Derivative of the approximated solution to eq. (64) divided by r and derivative of the approximated solution to eq. (65) for $L = 2\pi$ and $L = 1/6$. The analytic solutions are $u(r, z) = w(r, z) = u_1(r) e^{\sin(2\pi z/L)}$, with u_1 as in Fig. 1. The relative errors are computed with the l^2 -norm on the fixed Chebyshev-Fourier grid.

that is used to compute the values of the solving matrices $N^{(\cdot)}$ and $D^{(\cdot)}$ is *ARBLIB* [38]. This C library uses ball arithmetic and has a large number of special functions implemented.

Fig. 2 shows an example test and comparison of the methods $CG_1^{Neu,(1)/r}$ and $\mathcal{N}^{(1)/r}$, as well as $CG_1^{Dir,(1)}$ and $\mathcal{D}^{(1)}$. The analytic test function is chosen to have several Chebyshev-Fourier modes whose modulus are above machine epsilon, it is given in the caption of the figure. All the presented solving methods prove to be stable and accurate for $L = 2\pi$. In this case the resulting relative errors almost overlap. If $L = 1/6$, then the accuracy levels shift higher but the methods stay stable and comparatively accurate. The blue-dotted curves in Fig. 2, labeled by ‘spec.’ are the result of standard spectral projection of the solutions themselves with no solving involved, i.e. the relations in eqs. (191) and (192) are applied to the Chebyshev modes directly. In this test scenario, the BC methods $\mathcal{N}^{(\cdot)}$ and $\mathcal{D}^{(\cdot)}$ give the most accurate results, their relative errors stay close to machine epsilon even for $L = 1/6$, where k becomes large. In summary, both, the CG and BC methods are far superior to the standard spectral methods for the derivative and division in the radial argument. For further tests of the presented Poisson solving methods, we would like to refer the reader to [35].

Remark. By the virtue of Lemma 1, the presented methods for the computation of the analytical solutions to truncated problems may also be applied to other polynomial basis functions. Instead of Chebyshev polynomials, one may use Lagrange, Laguerre, Jacobi, or any other useful polynomials in the approximation of the RHS function. Also, the choice of evaluation points may be deliberately chosen and can differ from the grid points that are related to the basis functions. The CLA code uses Chebyshev polynomials, mainly because of their connection to Fourier transforms, which are computed by the FFT algorithm.

6. Flow fields along the trajectories and spatial interpolation

Once sufficiently many coefficients $X^{(s)}$ have been computed, a suitable time-step can be inserted into the truncated time-Taylor series in order to approximate the new particle positions via

$$X(a, t) \cong \sum_{s=0}^S X^{(s)}(a) t^s = \sum_{s=0}^S \left(R^{(s)}(r, z) e_r + A^{(s)}(r, z) e_\alpha + Z^{(s)}(r, z) e_z \right) t^s, \quad (123)$$

where $S \in \mathbb{N}$ is the truncation order. Clearly, the cylindrical components of X are approximated by their respective truncated time-Taylor series, e.g.

$$X^r(r, z, t) \cong \sum_{s=0}^S R^{(s)}(r, z) t^s, \quad R^{(0)} = r, \quad (124)$$

and X^α, X^z alike. Due to the assumed axisymmetry of the underlying flow, it is sufficient to compute the time-Taylor coefficients for particles that lie initially on a slice of the cylinder $D(1, L)$. The discrete grid points on that slice, seen as distinct particles, disperse into the cylinder with time. Some particles may also leave the region $D(1, L)$ since the actual flow takes place in an infinite cylinder along the vertical axis. The periodicity property allows us to re-map those leaving particles back inside the modeled region. It is important to note, that, again, due to the axisymmetric, the angular displacements of the particles from a chosen slice can be neglected and only the radial and vertical movements of the particles are relevant. Therefore, a scattered two-dimensional interpolation has to be executed to interpolate the flow fields, known only at the endpoints of the particle trajectories, onto the regular Chebyshev-Fourier grid.

The velocity field along the trajectories is given by the characteristic equation $\partial_t X(a, t) = v(X(a, t), t)$, which reads in cylindrical coordinates

$$\partial_t (\tilde{R} e_{\tilde{R}} + \tilde{Z} e_z) = v(\tilde{R}, \tilde{\Theta}, \tilde{Z}, t) = v^r(\tilde{R}, \tilde{Z}, t_1) e_{\tilde{R}} + v^\alpha(\tilde{R}, \tilde{Z}, t_1) e_{\tilde{\Theta}} + v^z(\tilde{R}, \tilde{Z}, t_1) e_z. \quad (125)$$

Here, we have used the representation of the characteristic map (18) in its local basis

$$X(r, \alpha, z, t) = \tilde{R}(r, z, t) e_{\tilde{R}} + \tilde{Z}(r, z, t) e_z \quad (126)$$

where $e_{\tilde{R}} = (\cos \tilde{\Theta}, \sin \tilde{\Theta}, 0)$ and $e_{\tilde{\Theta}} = (-\sin \tilde{\Theta}, \cos \tilde{\Theta}, 0)$ with

$$\tilde{R} := \sqrt{(X^r)^2 + (X^\alpha)^2}, \quad \tilde{\Theta} := \mathcal{A}(X^r, X^\alpha) + \alpha, \quad \tilde{Z} := X^z, \quad (127)$$

and \mathcal{A} as in eq. (14). With these definitions we obtain the correspondences

$$v^r(\tilde{R}, \tilde{Z}, t) = \dot{X}^r \cos(\mathcal{A}(X^r, X^\alpha)) + \dot{X}^\alpha \sin(\mathcal{A}(X^r, X^\alpha)), \quad (128)$$

$$v^\alpha(\tilde{R}, \tilde{Z}, t) = -\dot{X}^r \sin(\mathcal{A}(X^r, X^\alpha)) + \dot{X}^\alpha \cos(\mathcal{A}(X^r, X^\alpha)), \quad (129)$$

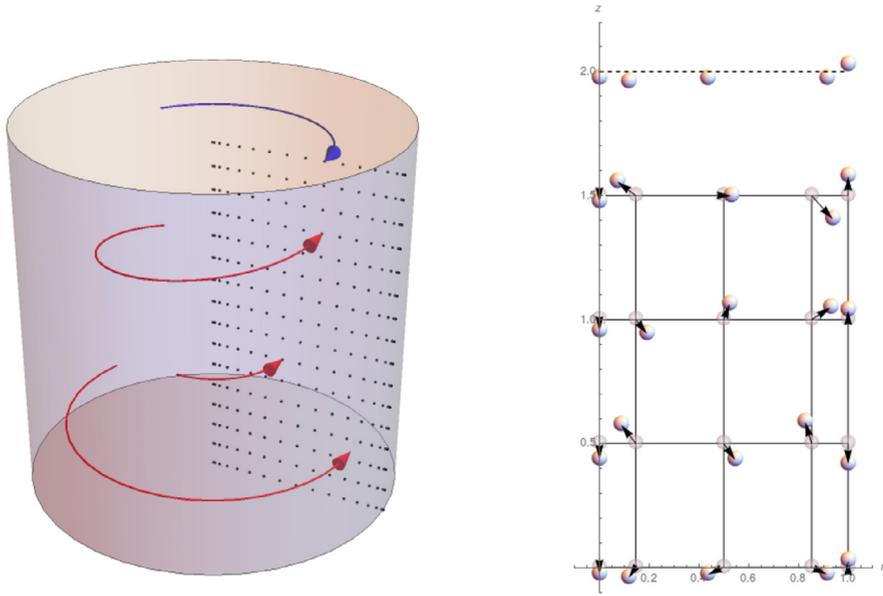
$$v^z(\tilde{R}, \tilde{Z}, t) = \dot{X}^z. \quad (130)$$

The vorticity $\omega(r, \alpha, z, t) = \omega^r(r, z, t) e_r + \omega^\alpha(r, z, t) e_\alpha + \omega^z(r, z, t) e_z$ is interpolated in the same way once it is obtained via the vorticity transport formula $\omega(X(a, t), t) = DX(a, t) \cdot \omega_0(a)$, which entails

$$\begin{aligned} \omega^r(\tilde{R}, \tilde{Z}, t) &= \cos(\mathcal{A}(X^r, X^\alpha)) \left(-\frac{X^\alpha \omega_0^\alpha}{r} + X_r^r \omega_0^r + X_z^r \omega_0^z \right) \\ &\quad + \sin(\mathcal{A}(X^r, X^\alpha)) \left(X_r^\alpha \omega_0^r + X_z^\alpha \omega_0^z + \frac{X^r \omega_0^\alpha}{r} \right), \end{aligned} \quad (131)$$

$$\begin{aligned} \omega^\alpha(\tilde{R}, \tilde{Z}, t) &= -\sin(\mathcal{A}(X^r, X^\alpha)) \left(-\frac{X^\alpha \omega_0^\alpha}{r} + X_r^r \omega_0^r + X_z^r \omega_0^z \right) \\ &\quad + \cos(\mathcal{A}(X^r, X^\alpha)) \left(X_r^\alpha \omega_0^r + X_z^\alpha \omega_0^z + \frac{X^r \omega_0^\alpha}{r} \right), \end{aligned} \quad (132)$$

$$\omega^z(\tilde{R}, \tilde{Z}, t) = \omega_0^r X_r^z + \omega_0^z X_z^z. \quad (133)$$



(a) Particles move onto the slice that contains the Chebyshev-Fourier grid.

(b) Particle displacement on the (r, z) -plane on a 5×4 grid. The upper row of particles have been added by periodicity.

Fig. 3. The modeled slice of the cylindrical domain $D(1, 2)$ with the underlying Chebyshev-Fourier grid is shown. Only the radial and vertical movements of the particles are relevant. The top particle positions in (b) are obtained by periodicity of the flow from those that correspond to the first horizontal grid line.

The displacement inside the cylinder and the relevant radial and vertical movements inside of the slice in the cylinder are illustrated in Fig. 3 (a). After a time-step insertion and the computation of the components X^r, X^α and X^z , we seek to interpolate the components of the flow fields from the scattered Lagrangian points $(\tilde{R}(r_i, z_j, t_1), \tilde{Z}(r_i, z_j, t_1))_{i,j}$ back onto the fixed grid $(r_i, z_j)_{i,j}$. Clearly, the difficulty here lies in the fact that one has to interpolate from an irregular grid onto a regular grid. Furthermore, particles may leave the convex hull of the Chebyshev-Fourier grid points in vertical direction or show a contraction, such that there are grid points that do not lie in the convex hull of the scattered particles. In the latter case one could perform an extrapolation, but it is better to use the periodicity of the flow and simply include more particles from above and below the first and last horizontal grid lines as visualized in Fig. 3 (b).

Several methods for the needed two-dimensional scattered interpolations have been tested, such as radial basis functions, fitting of bi-cubic splines, as well as Delaunay tessellation with subsequent linear barycentric interpolation. However, we found that the best results by far are achieved with the cascade interpolation [53]. This method was developed explicitly for the setting of Lagrangian displacements in a fluid flow and reduces the two-dimensional problem to a sequence of one-dimensional problems. Those one-dimensional problems may then be tackled by any other (scattered) interpolation method, e.g. Lagrange or B-spline interpolations, which are both implemented. We refer to [53,51,35] for details on the cascade interpolation.

For brevity, we recall the Lagrange polynomial interpolation here and refer the reader to [15] for an introduction to B-splines and the involved interpolation schemes. For a number $n \in \mathbb{N}$ of strictly increasing sites $x_0 < x_1 < \dots < x_{n-1}$, the Lagrange basis polynomials of order n (and degree $n - 1$) are defined by

$$I_l(x) := \prod_{\substack{0 \leq i \leq n-1 \\ i \neq l}} \frac{x - x_i}{x_l - x_i}, \quad l = 0, \dots, n - 1. \tag{134}$$

A function y , with function values $y(x_0), \dots, y(x_{n-1})$, may be interpolated by a Lagrange polynomial as

$$y(x) \cong \sum_{l=0}^{n-1} y(x_l) I_l(x). \tag{135}$$

The fact that this method does not involve a linear system, which needs to be solved, makes it very fast, and numerous implementations can be found in public libraries.⁵ In the implemented Lagrange routine, the sites are chosen such that the interpolation point is in their center, if possible, i.e. a centered Lagrange polynomial is computed with the same amount of sites to one side as to the other side of the interpolation point. In vertical direction, this can always be fulfilled, because of the periodicity of the interpolated functions. In radial direction, especially close to the boundary or pole, a periodicity, or parity with respect to the interval end points, is not known and, thus, no valid extension outside of the interval $[0, 1]$ can be constructed. In this case, the interpolation polynomial is still constructed as above with sites that lie closest to the boundary or pole. In this scenario, the Chebyshev extrema (also their zeros) are well known to reduce the Runge phenomenon [9,15], such that the Lagrange interpolation is stable even if an un-centered site distribution is used.

In the B-spline interpolation, one has to solve for the coefficients α_i in

$$\sum_{i=0}^{n-1} \alpha_i N_{i,p-1}(x_j) = y(x_j), \tag{136}$$

where $N_{i,p-1}$ are the B-splines of order p (and degree $p - 1$) based on $n + p$ non-decreasing knots $u_j, j = 0, \dots, n + p - 1$. The x_j are the strictly increasing interpolation points from before. A B-spline interpolation module, which is based on routines from [15], is publicly available on Github.⁶ Parts of this module were extracted and included in the CLA code after an adaptation to our needs. In these routines, the $n + p + 1$ knots u_i of the B-splines from above are distributed in the interpolation interval, such that $2p$ knots are placed on the interval endpoints, p on each end point, and the rest is placed on the data points starting from $x_{p/2}$, if p is even, and in between the data sites, if p is odd, starting after $x_{(p-1)/2}$. See [15] for more information on knot placing.

7. Time-stepping

A central question to the CLA is how to obtain the radius of convergence of eq. (3) that limits the possible time-step in an accurate way, or, if that's not possible, how to choose an appropriate time-step. Unfortunately, there is currently no explicit formula for the convergence radius available. Thus, the choice of the time-step becomes a challenging aspect of this algorithm. In principle, it would be possible to obtain an explicit formula for the convergence radius from the constructive proof in [7]. However, the estimates that are used in [7] involve constants that originate from Schauder estimates of the Newton potentials. Those constants are in general hard to compute and slight variations in their determined value might have a strong influence on the value of the later computed radius of convergence. As a result, in the space $C^{1,\alpha}, \alpha > 0$, the convergence radius $\varrho_\alpha > 0$ for (3) can be bound by

$$\varrho_\alpha \geq \frac{C}{\|\omega_0\|_\alpha}, \tag{137}$$

where $C = C(\Omega, \alpha)$ is a positive constant that depends only on the geometry of the domain Ω and $\|\cdot\|_\alpha$ denotes the Hölder norm.

Classically, there are other ways to approximate the radius of convergence, e.g. via the Cauchy-Hadamard formula

$$\frac{1}{\varrho} = \limsup_{s \rightarrow \infty} \|X^{(s)}\|^{1/s}, \tag{138}$$

or via the ratio rule

$$\varrho = \lim_{s \rightarrow \infty} \frac{\|X^{(s-1)}\|}{\|X^{(s)}\|}, \tag{139}$$

if the limit exists. The norms in the above expressions can be those of $C^{1,\alpha}$, but also of $H^s, s > 5/2$ or simply of an L^p space, in short, from any space where the convergence needs to be established. The Cauchy-Hadamard formula eq. (138) is less useful in numerical computations, because of the strong growth of the coefficients $X^{(s)}$ and the difficulty that arises when computing the root of large orders s . A straightforward attempt to use the ratio rule can be made in computing a large number S of time-Taylor coefficients and use extrapolation techniques to approximate the limits in eq. (138) or (139). This method is emphasized here for a stationary flow from [43]. The regarded stationary vorticity in the periodic cylinder $D(1, 2\pi)$ reads

⁵ E.g. https://people.sc.fsu.edu/~jburkardt/f_src/lagrange_interp_1d/lagrange_interp_1d.html.

⁶ The author of this module is Jacob Williams, it can be obtained from https://github.com/jacobwilliams/bspline-fortran/blob/master/src/bspline_sub_module.f90.

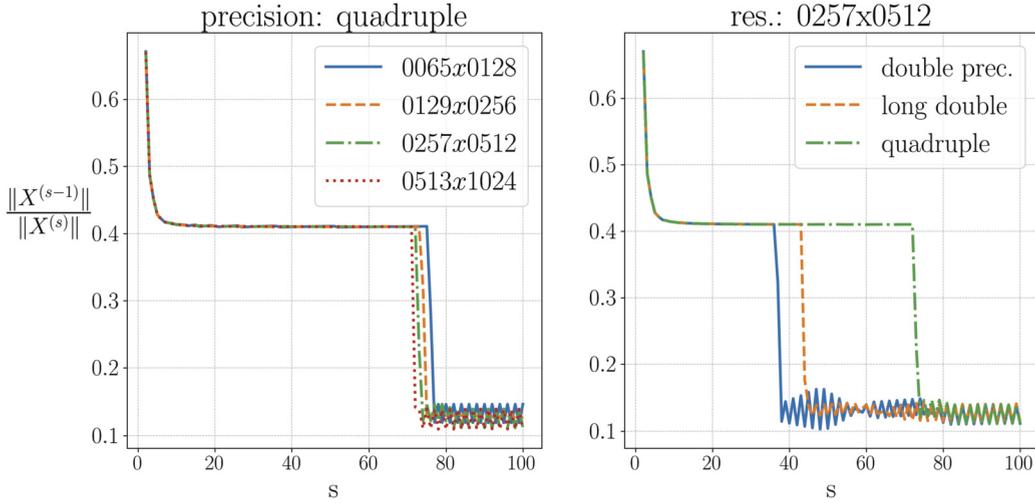


Fig. 4. Approximation of the limit in eq. (139) in the l^∞ -norm on the underlying grid. The curves overlap to at least $s = 30$, from which a possible limit at around 0.4 can be inferred. The sudden drop in values indicates that the working precision is insufficient to accurately represent the time-Taylor coefficients.

$$\begin{aligned} \omega^r(r, z) &= \sqrt{c_1^2 + 1} \sin(z) J_1(rc_1), \\ \omega^\alpha(r, z) &= (c_1^2 + 1) \cos(z) J_1(rc_1), \\ \omega^z(r, z) &= c_1 \sqrt{c_1^2 + 1} \cos(z) J_0(rc_1), \end{aligned}$$

where $r \in [0, 1]$, $z \in [0, 2\pi]$ and c_1 is the first positive root of the Bessel function J_1 . The coefficients $X^{(s)}$, $0 \leq s \leq 100$ for this flow have been computed via the recursion relations stated in Section 3. Fig. 4 shows an application of the ratio rule, which seems to suggest a bound of the convergence radius in L^∞ around 0.4. At this point, we would like to remind the reader that a time-step below an approximated radius of convergence stays valid regardless of the underlying resolution, because this radius is independent of resolution. These computations are, however, not very practical, because many coefficients need to be computed accurately, which slows down the CLA.

Here we would like to present a different approach, which is based on a self-adapting time-stepping scheme. In this method, the convergence radius is not computed in the running program, but only once in the beginning to give a valid initial time-step. More precisely, let the truncated time-Taylor series of $X^{(s)}$, evaluated at t_1 , be abbreviated by

$$b_s(t_1) = \sum_{\tilde{s}=0}^s X^{(\tilde{s})} t_1^{\tilde{s}} \tag{140}$$

and define

$$F(s, t_1) := \frac{\|b'_{s-1}(t_1) - b'_s(t_1)\|_\infty}{\|b'_{s-1}(t_1)\|_\infty} \tag{141}$$

where b'_s denotes the temporal derivative of b_s . Then after each time-step insertion the following property is checked:

$$\varepsilon_m > F(S - 1, t_1) > F(S, t_1), \tag{142}$$

where ε_m is machine epsilon, i.e. $\sim 10^{-16}$ in double precision computations. In case this property is not fulfilled by a certain time-step t_1 , then this time-step may be adapted accordingly. Condition (142) is meant to assure that additional terms of order higher than the truncation order S would not alter the result of the truncated time-Taylor sum in the working precision. Surely this approach is only valid, if the coefficients show a point-wise monotone growth, which must be checked before hand, i.e. by computing a larger number of coefficients in certain iteration intervals. In fact, this method also accounts for alternating signs in the entries of $X^{(s)}(a)$, which is internally given by a two dimensional array. The derivative b'_s is used in definition eq. (141), because it represents the approximation of the velocity field by the characteristic equation, and it implies that the same property (eq. (142)) holds with b_s instead of b'_s in the definition of $F(s, t_1)$.

The method adjusts a suitable and initially fixed time-step at the event of an insufficient number of coefficients. But it can also be used to implement a fully adaptive time-stepping scheme. In this scenario, with a fixed truncation order S , a time-step is set to fulfill

$$F(S, t_1) \cong \varepsilon_m. \tag{143}$$

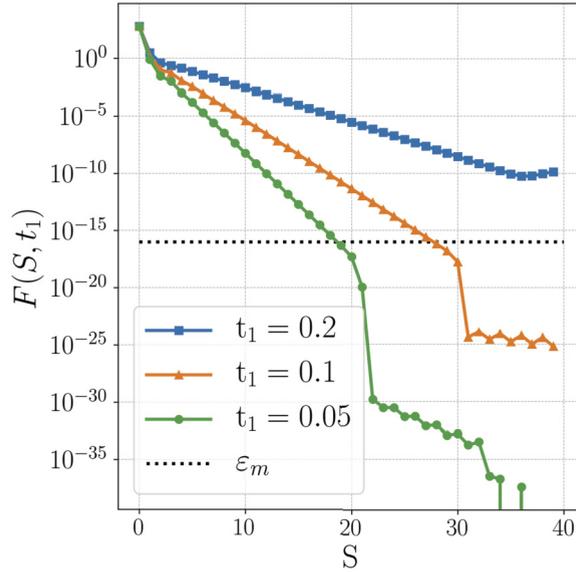


Fig. 5. Plot of the function F defined in (142) for different time-steps. The recursion mechanism may be stopped, when the values of F drop below machine epsilon ϵ_m , here in double precision and indicated by a dotted line in the plot. All higher order terms $X^{(s)}$ after the criterion has been met give the same summation result in the time-Taylor series in the working precision. A break in the linear behavior of F indicates the depletion of the number range in double precision in the coefficients $X^{(s)}$.

This can be achieved by slowly reducing an initial value of t_1 until eq. (143) is met. In this scenario, higher numbers of S imply a larger time-step due to the growth behavior of the coefficients: It holds $\|X^{(s)}\|_\infty < cst./\varrho^s$ for s large enough, where ϱ is the minimal radius of point-wise convergence. The time-step is chosen such that the time-Taylor sum and its first temporal derivative, truncated to S coefficients, is not altered by the last terms. This implies a bound on the last term in the Taylor sum of the form $|X^{(S)}|t^S < \epsilon$, for some $\epsilon > 0$. Thus, the bound from before allows choosing t closer to ϱ for larger S .

8. Conserved quantities and testing of the implementation

This section is intended to recall the de facto standard methods for validating numerical simulations of the (axisymmetric) incompressible Euler equations and adds original methods for the CLA. To date, there is no reliable method to verify the validity of a non-stationary incompressible Euler flow. There are, however, many propositions to approach this problem. A non-exhaustive list of numerical validation techniques can be found in [45]. The most prominent conserved quantities to watch in an incompressible Euler simulation are the total kinetic energy⁷ (E) the total helicity (H), as well as the total angular momentum (I_1) and total angular helicity (H_1). They are given respectively by the following expressions

$$E = \frac{1}{2} \int_0^1 \int_0^L ((v^r)^2 + (v^\alpha)^2 + (v^z)^2) r \, dr dz \tag{144}$$

$$H = \int_0^1 \int_0^L (v^r \omega^r + v^\alpha \omega^\alpha + v^z \omega^z) r \, dr dz \tag{145}$$

$$I_1 = \int_0^1 \int_0^L v^\alpha r^2 \, dr dz \tag{146}$$

$$H_1 = \int_0^1 \int_0^L v^\alpha \omega^\alpha r \, dr dz. \tag{147}$$

⁷ In the sequel of this article we shall refer to this quantity as the total energy, or simply energy.

If the velocity field v solves the incompressible Euler equations and ω is the corresponding vorticity field, then the above quantities are conserved for all times. The converse is clearly not true. The check for this conservation can, therefore, only serve as a control mechanism to verify that the computed solution is not too far from the actual solution. Furthermore, the conservation of such quantities should not be trusted too much. Indeed, tests with stationary solutions show that the relative energy loss does not correlate well to the actual relative errors in the flow fields, which could give a false impression on the validity of the computation.

In the literature, the majority of Euler flow simulations are carried out on an Eulerian grid. The Lagrangian viewpoint allows in return to carry out additional quality measurements regarding an implementation. Note, however, that since we discuss a semi-Lagrangian algorithm, we only have access to a set of characteristics departing from a fixed grid within one time-step. This set of characteristics is different from one time-step to another. The following validation criteria are thus only applicable within one time-step and not with respect to the initial state (as it would be in a purely Lagrangian scheme without interpolation). One of those criteria is the *movement of the boundary particles*. Theoretically, they stay on the boundary, but numerically, an error in the distance to the boundary after every iteration can be measured. In our cylindrical domain, this is easily achieved in computing

$$\max_{z \in [0, L]} |1 - \sqrt{R(1, z, t_1)^2 + A(1, z, t_1)^2}|, \tag{148}$$

where t_1 is a time-step. In the aftermath, this error is usually very close to the error in the energy conservation, even though the current implementation of the CLA resets by default the boundary value to 1 after every iteration.

Unique to the implemented Cauchy-Lagrange formalism is also the possibility to *verify directly the Cauchy invariants equation* (CIE), eq. (5), together with the *incompressibility condition* eq. (6) after every time-step insertion. Instead of checking only some necessary conditions for a solution, this method verifies directly the CIE and thus, the Euler equations. On the contrary, in a purely Eulerian setting, the verification of the Euler equations appears to be tricky and impractical. Note that the initial vorticity in eq. (5) is replaced with the current vorticity after the time-step on the grid, hence, the CIE is not verified with the initial vorticity at $t = 0$, but with that at time t_{n-1} , as mentioned before.

Another conserved quantity is the *circulation* about a closed material curve C that moves with the fluid ([42,4]). Kelvin's circulation theorem states that

$$\partial_t \Gamma(t) = 0, \tag{149}$$

with

$$\Gamma(t) = \int_{X(C,t)} v(x,t) \cdot d\sigma(x) = \int_0^1 v(X(C(s), t), t) \cdot \partial_s X(C(s), t) ds, \tag{150}$$

if C is parametrized by $s \in [0, 1]$, such that $C(0) = C(1)$. Differently formulated, the quantity

$$\int_C v(a, t) \cdot d\sigma(a) \tag{151}$$

is constant under the material derivative $D_t = \partial_t + (v \cdot \nabla)$. Numerically, this conservation is difficult to verify for arbitrary closed curves, but is in principle possible with spectral representations and the ability to evaluate them anywhere in the covered domain. An Evaluation of a non-trivial closed curve in the cylindrical domain $D(1, L)$ would need to be done very thoroughly in order to let the integral converge accurately to its conserved value. A considerable numerical complexity and large errors seem to be implied, which practically rules out this circulation method for general curves in ambitious Euler flow simulations, i.e. with the purpose of finding finite time singularities. In return, for simple circular curves around the axis, this task is readily feasible. Such a curve is of the form

$$\bar{r}e_r + \bar{z}e_z, \quad \text{for fixed } \bar{r} \in [0, 1], \bar{z} \in [0, L], \tag{152}$$

and is parametrized by $\alpha \in [0, 2\pi]$. With the notation of eq. (126), Γ in cylindrical coordinates takes the form

$$\int_0^{2\pi} v(\tilde{R}, \tilde{\Theta}, \tilde{Z}, t) \cdot \partial_\alpha (\tilde{R}e_{\tilde{R}} + \tilde{Z}e_z) d\alpha = \int_0^{2\pi} v(\tilde{R}, \tilde{\Theta}, \tilde{Z}, t) \cdot \tilde{R}e_{\tilde{\Theta}} d\alpha \tag{153}$$

$$\begin{aligned} &= \int_0^{2\pi} (v^r(\tilde{R}, \tilde{Z}, t)e_{\tilde{R}} + v^\alpha(\tilde{R}, \tilde{Z}, t)e_{\tilde{\Theta}} + v^z(\tilde{R}, \tilde{Z}, t)e_z) \cdot \tilde{R}e_{\tilde{\Theta}} d\alpha \\ &= 2\pi \tilde{R} v^\alpha(\tilde{R}, \tilde{Z}, t). \end{aligned} \tag{154}$$

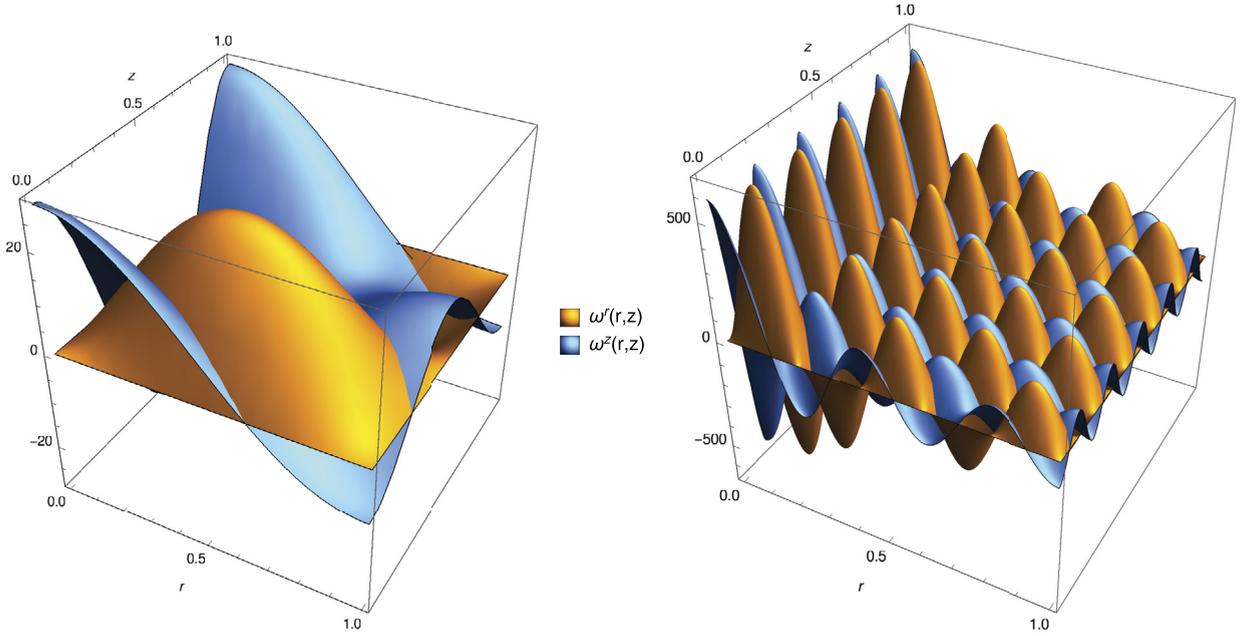


Fig. 6. Stationary flow components. Left: radial and vertical vorticity for $\kappa = 1$ and with c_1 (first root). Right: same as left with $\kappa = 5$ and c_5 (fifth root).

Kelvins circulation theorem tells us that Equation (154) stays constant, which can be easily verified in every time iteration via

$$\frac{\|rv^\alpha(r, z, t_i) - \tilde{R}(r, z, t_{i+1})v^\alpha(\tilde{R}(r, z, t_{i+1}), \tilde{Z}(r, z, t_{i+1}), t_{i+1})\|_{L^2(D^\#)}}{\|rv^\alpha(r, z, t_i)\|_{L^2(D^\#)}}, \tag{155}$$

where t_i is the time after i iterations and $D^\#$ the underlying grid. This verification method is quite simple, low in numerical complexity and implemented in the CLA code. Note that in eq. (155) one cannot use the initial velocity $v_0^\alpha = v_0^\alpha(r, z, 0)$, because, in the CLA implementation, $\tilde{R}(r, z, t_{i+1})$ is the spatial displacement with respect to the underlying grid within one time-step and not within the whole integration period $[0, t_{i+1}]$. In order to compare to the initial velocity v_0^α , one would need to track the particle positions $(\tilde{R}(r, z, t), \tilde{Z}(r, z, t))$ from $t = 0$ to the time where the conservation of eq. (155) is checked.

9. Testing of the CLA

9.1. Tests on stationary solutions

The stationary velocity field, which is the principal test flow for the CLA and taken from [43], reads component-wise

$$\begin{aligned} v^r(r, z) &= \kappa J_1(c_j r) \sin(\kappa z), \\ v^\alpha(r, z) &= \sqrt{c_j^2 + \kappa^2} J_1(c_j r) \cos(\kappa z), \\ v^z(r, z) &= c_j J_0(c_j r) \cos(\kappa z). \end{aligned} \tag{156}$$

The J_k denote the k -th order Bessel functions of first kind, c_j is the j -th positive root of J_1 , and κ can be any natural number. In taking the cylindrical curl of the velocity field, one obtains the corresponding vorticity components, namely

$$\begin{aligned} \omega^r(r, z) &= \kappa \sqrt{c_j^2 + \kappa^2} J_1(rc_j) \sin(\kappa z), \\ \omega^\alpha(r, z) &= (c_j^2 + \kappa^2) J_1(rc_j) \cos(\kappa z), \\ \omega^z(r, z) &= c_j \sqrt{c_j^2 + \kappa^2} J_0(rc_j) \cos(\kappa z). \end{aligned} \tag{157}$$

See Fig. 6 for an illustration of this initial vorticity field with two different values of κ and roots c_j . One of the advantages of knowing such stationary solutions, is the fact that we are able to compute analytically the time-Taylor coefficients to some order. The details on this method are stated in Appendix B.

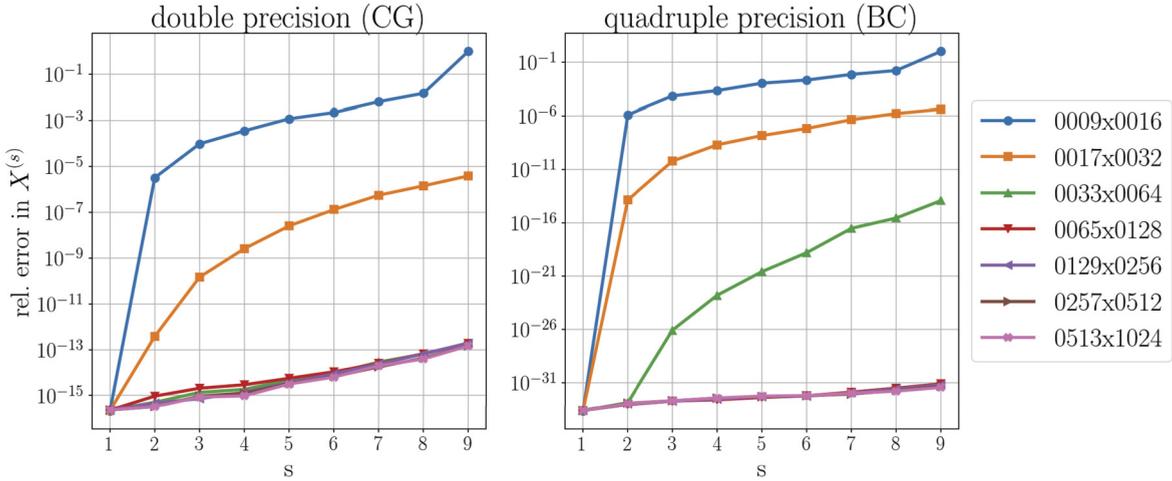


Fig. 7. Relative errors (as defined in (158)) in the computed coefficients $X^{(1)}, \dots, X^{(9)}$, which belong to the flow in eq. (156) with $\kappa = 1$ and c_1 . The Poisson problems were solved in double precision via the Chebyshev-Galerkin solver (left) and in quadruple precision via the Bessel-Chebyshev solver (right). The true coefficients were obtained via symbolic differentiation inside a computer algebra system by the method described in Appendix B.

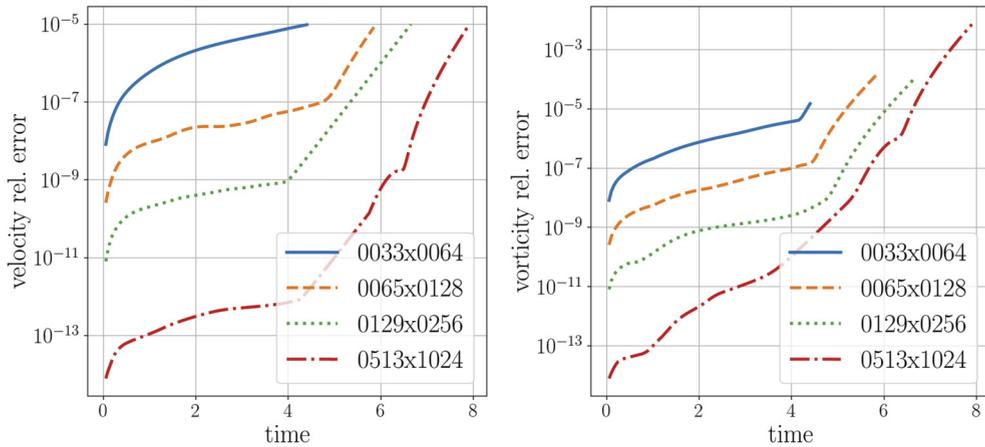


Fig. 8. A visualization of relative errors in the velocity (left) and vorticity (right) fields due to different resolutions. The fixed run-time parameters are: $S = 25$, time-step 0.05, 5-th order B-spline cascade interpolation.

In the following, if not stated otherwise, then the relative errors in vector-fields are computed on the underlying Chebyshev-Fourier grid $D^\#$ in the l^2 -norm. More precisely, for a two-dimensional array g , which is approximated by \tilde{g} , we compute

$$\frac{\|g - \tilde{g}\|_{l^2(D^\#)}}{\|g\|_{l^2(D^\#)}}. \tag{158}$$

The l^2 -norm returns a scalar with the square root of the sum of all elements squared, and if g equals zero, then only the difference $g - \tilde{g}$ is measured with the given norm (absolute error). The relative error in vector fields with higher dimensions is defined as the maximum over the l^2 -relative errors in all the (cylindrical) components. Fig. 7 shows the relative errors in the first nine time-Taylor coefficients of X , belonging to the stationary flow from above, and computed via the recursion mechanism stated in Sections 3 and 4. The figure shows that the coefficients are very accurately computed for sufficiently large grid sizes. The effect of the resolution becomes further apparent in Fig. 8, where the relative errors in the analytically known velocity field are visualized. Even low resolutions deliver good results due to the approximation qualities of the Chebyshev-Fourier basis.

From our investigations in Section 7 (and in particular Fig. 4) we assume a valid time-step for the given stationary flow below 0.4. It was further analyzed in Fig. 5 that a time-step should lie even below 0.1 to assure accuracy in a double precision computation. For a larger time-step there would be many more coefficients needed, but the Real64 number format in the arrays is ultimately not able to contain large order coefficients $X^{(s)}$, as their norms grow steeply with the order s . This limitation can be observed in Fig. 5 as a sudden drop of values in the ratio formula. See in particular the relation between

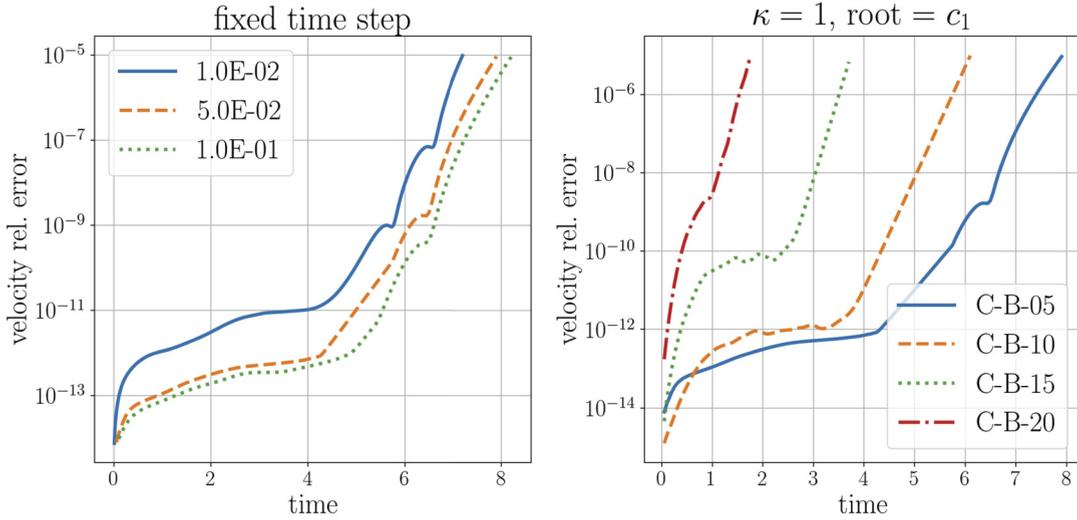


Fig. 9. The effect of different time-steps (left) and orders in the B-Spline cascade interpolation (right). Larger time-steps and lower interpolation orders yield the best results. The run-time parameters are as in Fig. 8.

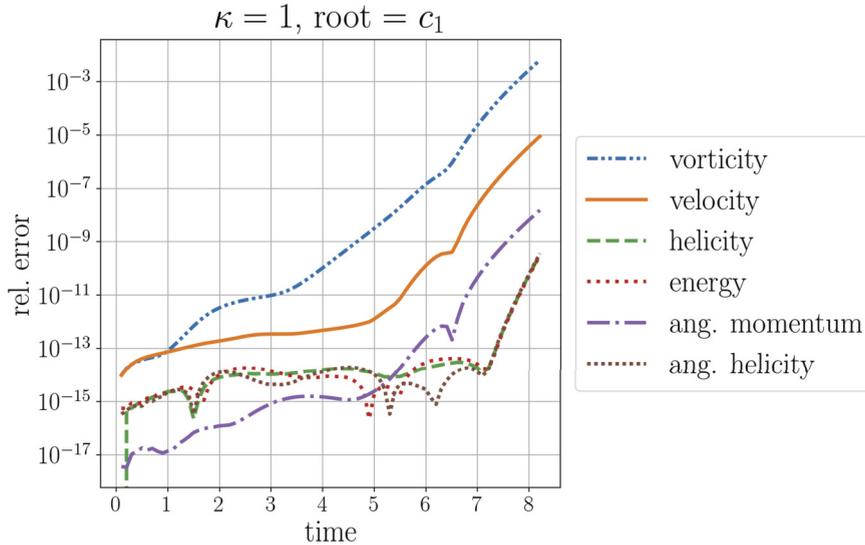


Fig. 10. Test flow and standard conserved quantities, computed on 513×1024 points, with $S = 40$, time-step $\Delta t = 0.1$, Chebyshev-Galerkin solver and 5th order B-spline cascade interpolation.

the critical orders and the employed working precision in Fig. 5. Therefore, we have tested the time-steps 0.1, 0.05 and 0.02 for a truncation order $S = 25$. The time-step is maintained as long as eq. (142) is satisfied. If the errors in the computation grow such that eq. (142) is not verified anymore, then the time-step is halved. The results are illustrated in the left panel of Fig. 9. We clearly see that larger time-steps yield better results, which is a common property of semi-Lagrangian schemes and stems from the fact that fewer interpolations are carried out to arrive to a certain point of time. In the right panel of Fig. 9 we show the dependence of the relative errors in the velocity field to the order of B-spline functions inside the cascade interpolation. Lower to moderate orders produce the best relative errors over the majority of the integration time span, also if Lagrange interpolation is used instead of B-spline interpolation. This can be explained by the amplification of errors, namely that high order B-splines, or polynomials in the Lagrange interpolation, amplify rounding and approximation errors.

Stationary solutions are good candidates for testing the CLA, because one can directly compare the true relative error in the flow fields to other validation criteria, for instance, those that were discussed in Section 8. If a good agreement between the relative error in a flow field and a relative change in a conserved quantity is found, then this correlation might help to assert the quality of a simulation once the flow fields are not known analytically, i.e. for non-stationary flows. In Fig. 10, we observe that the relative changes in most of the classical conserved quantities provide poor correlations to the actual relative errors in the flow fields. Only the total angular momentum (or first-order Casimir) aligns roughly with the error

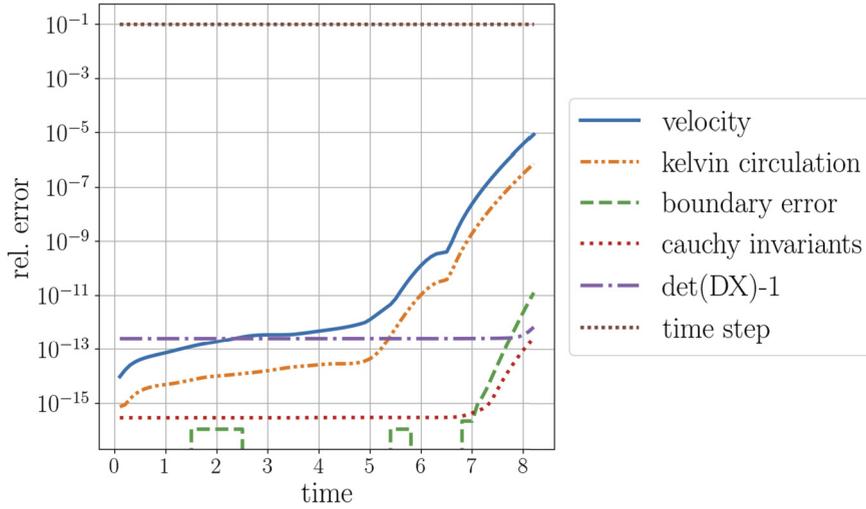


Fig. 11. Test flow and conserved quantities that are particular to the CLA; same run-time parameters as in Fig. 10.

in the velocity field. A better alignment is found in the preservation of Kelvin circulation along perfect circles in planes perpendicular to the vertical axis, see Fig. 11. Other validation criteria that are particular to the CLA, such as the movement of the boundary particles, the Cauchy-invariants formula (CIF), or the incompressibility condition, do neither show good indications of the validity of a simulation. The limiting factor here is that those expressions are checked only within one time-step and not in the whole integration period. The initial vorticity is replaced by that of the proceeding time-step before a new recursion cycle starts. To verify the CIF with the initial vorticity, one would also need to know the characteristics since the starting time and the derivatives thereof. Such verification is not yet implemented in the current CLA code.

9.2. Tests on a non-stationary flow without swirl

Another possibility to further test the CLA is given by *swirl-free* flows, whose angular component stays exactly zero at all times. As the here discussed swirl-free flows are non-stationary, one cannot directly measure a relative error in the flow fields, but fortunately they are better understood compared to general three-dimensional flows. For instance, it is shown in [59] that the axisymmetric Euler equations in a (wall-bounded) cylindrical domain Ω are well-posed globally in time for swirl-free initial data in $H^s(\Omega)$ for $s \geq 3$. It is shown furthermore that $s > 5/2$ is sufficient in case $\Omega = \mathbb{R}^3$. See also [61,54,59,46] for connected results in \mathbb{R}^3 with $s > 7/2$. Moreover, we can verify numerically, next to the fact that the angular component stays zero at all times, that the pseudo-vorticity ω^α/r stays constant along the trajectories and verifies a maximum principle (see e.g. [46]). More precisely, in the notation of eq. (126), we have

$$\frac{\omega^\alpha(\tilde{R}(r, z, t), \tilde{Z}(r, z, t), t)}{\tilde{R}(r, z, t)} = \frac{\omega^\alpha(r, z, 0)}{r}. \tag{159}$$

In order to verify these facts, we simulate a swirl- and divergence-free flow with initial components

$$\begin{aligned} v_0^r(r, z) &= \frac{2\pi}{L}(1-r)^n r^{a+2} \cos(2\pi z/L), \\ v_0^\alpha(r, z) &= 0, \\ v_0^z(r, z) &= -(1-r)^{n-1} r^{a+1} \sin(2\pi z/L) (a-r(a+n+3)+3), \end{aligned}$$

where $a \geq 1$ and $n \in \mathbb{N}$. As $\omega = \nabla \times v$, and in using eq. (16), the radial and vertical component of the vorticity field vanish identically and the initial angular vorticity component reads

$$\begin{aligned} \omega_0^\alpha(r, z) &= \frac{1}{L^2}(1-r)^{n-2} r^a \sin(2\pi z/L) (L^2((a+1)(a+3)+r^2(a+n+1)(a+n+3) \\ &\quad - r(2a^2+2a(n+4)+5n+6)) - 4\pi^2(r-1)^2 r^2). \end{aligned}$$

As a matter of fact, there holds

$$v^\alpha(r, z, t) = \omega^r(r, z, t) = \omega^z(r, z, t) = 0 \tag{160}$$

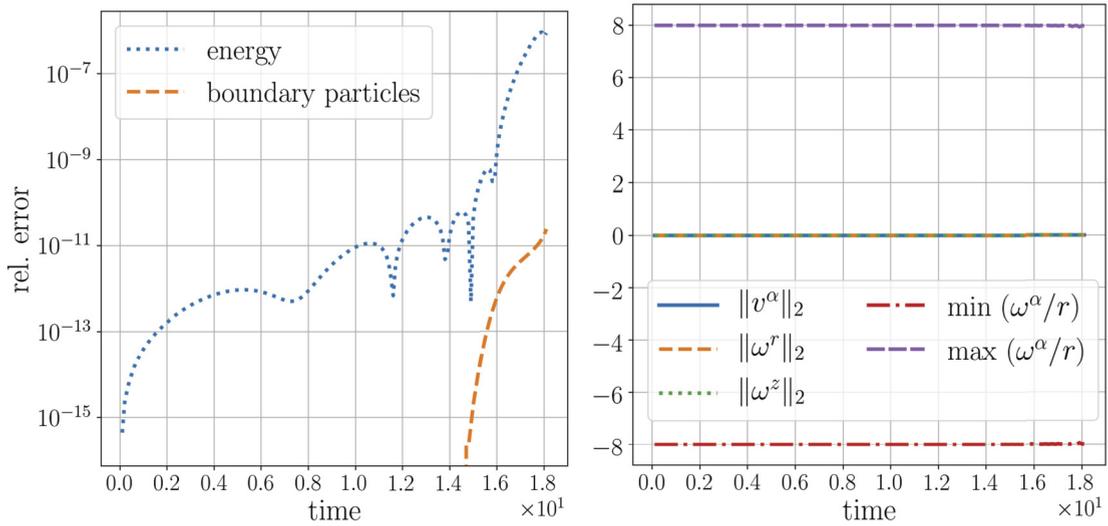


Fig. 12. Test results for the given swirl-free flow with $a = 1$ and $n = 2$, and run-time parameters: $S = 20$, $\Delta t = 0.1$, Chebyshev-Galerkin solver, 5th order B-spline cascade interpolation, no dealiasing, on a 513×1024 Chebyshev-Fourier grid. Left: kinetic energy conservation and movement of the boundary particles in radial direction, all the conserved quantities, which are not plotted here, stayed identically zero in the simulation; Right: verification of the min/max principle for $\omega^\alpha/r|_{r \neq 0}$, and verification of eq. (160).

for all times t . These identities are exactly preserved by our implementation of the CLA, which is visualized exemplary in Fig. 12, right panel. Therefore, all the conserved quantities which only depend on those components, such as Kelvin circulation etc., stay exactly zero in the computation. Also, the helicity evaluates exactly to zero, since for its computation $v^r \omega^r + v^\alpha \omega^\alpha + v^z \omega^z = 0$ is integrated over the (r, z) -plane. As for the stationary solution in Section 9.1, the results for the swirl-free flow depend on resolution, the interpolation order, and the chosen time-step. Higher resolutions, lower to moderate interpolation orders, and larger time-steps increase the integration period until a critical energy loss occurs. The time-step $\Delta t = 0.1$ was maintained over the whole simulation and the run was stopped, when energy conservation exceeded 10^{-6} . The total kinetic energy, in Fig. 12 left panel, seems to be well preserved until $t = 1.5$. However, we know from the tests with the stationary solution (see Fig. 10 in particular), that the actual error in the flow fields can behave differently to the error in the energy conservation. The min/max principle, in Fig. 12 right panel, is well demonstrated too, even though a final shift in the min/max values on the fixed grid of around 10^{-5} is observed. This shift, however, might simply stem from the fact that the real extrema do not lie on the grid and move towards, or away from the closest grid points. Higher resolutions yield better results also with respect to this point, although very large resolutions imply other inaccuracies, such as accumulating rounding errors.

10. Application to a potentially singular flow

In [45], the authors compute an axisymmetric and incompressible Euler Flow in the lower quarter of the cylindrical region $D(1, 1/6)$, and claim a finite time singularity of the maximal vorticity modulus for the initial swirl-only velocity field with $v_0^r = v_0^z = 0$ and

$$v_0^\alpha(r, z) = 100 r e^{-30(1-r^2)^4} \sin(2\pi z/L), \quad L = 1/6. \tag{161}$$

A 3×10^8 -fold increase of the quantity $\|\omega\|_{L^\infty}$ is reported in [45], where the maximum of $|\omega|$ is attained on the boundary $\partial D(1, 1/6)$ (at $r = 1$) and more precisely, on the symmetry axis at $z = 0$. Their investigations depart from a totally different formulation of the incompressible Euler equations than that in the CLA. In using a transform of the stream-vorticity formulation of the incompressible Euler equations, one obtains

$$v_{1,t} + v^r v_{1,r} + v^z v_{1,z} = 2v_1 \psi_{1,z}, \tag{162}$$

$$\omega_{1,t} + v^r \omega_{1,r} + v^z \omega_{1,z} = (v_1^2)_z, \tag{163}$$

$$-\left[\partial_r^2 + (3/r) \partial_r + \partial_z \right] \psi_1 = \omega_1, \tag{164}$$

with

$$v_1 = v^\alpha / r, \quad \omega_1 = \omega^\alpha / r, \quad \psi_1 = \psi^\alpha / r, \tag{165}$$

Table 1

The approximate maximal vorticity values in this table are taken from [45] and [1]. They will serve as orientation points for our simulations and are marked as black crosses in the maximal vorticity plots. At least three algorithms [45,1,41] have good agreement on the value at $t = 3.1 \cdot 10^{-3}$.

| time | 0.0 | $3.0 \cdot 10^{-3}$ | $3.1 \cdot 10^{-3}$ | $3.4 \cdot 10^{-3}$ | $3.5 \cdot 10^{-3}$ | $3.505 \cdot 10^{-3}$ |
|---------------------|----------------------|---------------------|-------------------------|---------------------|---------------------|------------------------|
| $\ \omega\ _\infty$ | $3.76999 \cdot 10^3$ | $9.0847 \cdot 10^4$ | $1.54276901 \cdot 10^5$ | $4.3127 \cdot 10^6$ | $5.8413 \cdot 10^9$ | $1.2401 \cdot 10^{12}$ |

where ψ^α is the angular component of the vector stream function, satisfying $-\Delta \psi = \omega = \nabla \times v$. The sub-scripts, different from 1, indicate partial derivatives as usual. See [45,46] for further details on the construction of the above formulation. The radial and vertical components of the velocity field are then given by

$$v^r = r\psi_{1,z}, \quad v^z = 2\psi_1 + r\psi_{1,r}. \tag{166}$$

The Euler equations in this form are numerically convenient, because they avoid divisions by the radial argument r . Note that the CLA obtains the terms A_1 and R_1 in eqs. (53) and (59) also without divisions by r in its effective implementation.

In [45], eq. (164) is solved via a 6th order B-Spline based Galerkin solver, and a 4th order Runge-Kutta time-stepping scheme is subsequently applied to eqs. (162) and (163); the appearing derivatives are obtained by a 6th order centered difference formula. However, the most important part of their method is an adaptive moving mesh, which keeps shifting grid points into the region with peak vorticity values. In [45], this adaptive mesh allows to reach an effective maximal resolution of around $(10^{12})^2$ per unit square near the location of the potential singularity. Another important facet of their implementation is the use of the preserved symmetry properties of the initial velocity field, which allows to only model the flow in the quarter cylinder $D(1, L/4)$ and start with a much higher point density. The CLA has not yet been adapted to exploit those flow symmetries, it always computes the flow in the full cylindrical domain $D(1, L)$, here with $L = 1/6$. This can be seen as a draw-back of our current implementation because a much higher grid size has to be employed to obtain similar initial local resolutions. However, even with this drawback, the computational results are convincing, which highlights the potential of the CLA as a whole.

As in Section 9, we will investigate different parameter settings to show their influence on the outcome of a simulation. In all the presented runs, the computations were aborted if the time-step dropped below 5×10^{-8} . The time-step is halved, whenever the precision criterion in eq. (142) is not fulfilled at the truncation order S , i.e. if $F(S, \Delta t) > \varepsilon_m$ for a time-step Δt . This halving takes place until criterion (142) is matched or the code aborted due to insufficient energy or helicity conservation.⁸ The results here are mainly measured by means of the conservation of helicity, energy, and Kelvin circulation. Energy and helicity have not proven to be particularly good indicators for correctness, at least for the stationary solution examined in Section 9.1, but they are very common in computational fluid dynamics, and in using them, the results become more comparable to other numerical methods. However, if the limit of 10^{-5} in relative energy or absolute⁹ helicity change is reached, then the solution is likely to be inaccurate. Fortunately, some maximal vorticity values $\|\omega\|_\infty$ at specific time instances are stated in [45] and are, therefore, directly comparable. The list of those maximal values is given in Table 1 and is appended by an additional value taken from [1]. The author of [1] has repeated the simulation of the same equations as in [45], i.e. system (162)-(164), with a spectral Chebyshev-Fourier solver on a fixed grid with 769 radial and 2048 vertical points and a Runge-Kutta time-stepping scheme (for eqs. (162) and (163)). It is important to note that [1] also models the equations on the quarter cylinder $D(1, L/4)$. It is further stated in [1] that one additional maximal vorticity value was communicated by one of the authors of [45]. This additional value at $t = 0.0031$, given in the third column of Table 1, is verified in [1] with a relative error of 1.9×10^{-8} . Also very recently the authors of [41] develop a Chebyshev-Fourier pseudospectral method with a fourth-order Runge-Kutta time-stepping scheme to solve equations (162)-(164), and to study the Luo-Hou initial condition [45]. The results of [41] confirm the simulations of [1,45] at least until the time $t = 0.0031$. In addition, using an analyticity-strip method and an extrapolation method based on a linear least-square regression, the authors of [41] obtain an estimate of the time at which the (potential) singularity occurs, and this time value is consistent with that of [45]. Therefore, there are at least three independent simulations that agree on the leading digits of a maximal vorticity value at a time instance not too far away from the reported temporal singularity. In the following plots of the maximal vorticity, the second, third, and fourth column values in Table 1 are indicated by black crosses to show qualitatively how far the CLA can approach them in those test runs with various parameter settings. These approximations are given quantitatively in Table 2.

Fig. 13 shows the behavior of the error in total kinetic energy, helicity, and Kelvin circulation, as described in Section 8, as well as the vorticity growth. As seen in the figure, the result strongly depends on the chosen resolution, when all other run-time parameters stay fixed. The curves of the Kelvin circulation error, which will henceforth only be referred to as the Kelvin conservation, allow to easily distinguish the low and moderate grid sizes. The helicity conservation also shows noticeable differences, while the energy conservation curves are less clearly attributable to the different resolutions. Also, all curves show very similar behavior with respect to their measured quantity. The appearing sudden drops in energy and

⁸ I.e. when those conserved quantities show an error above 10^{-5} .

⁹ We use $|H(0) - H(t)|$ instead of $|H(0) - H(t)|/|H(0)|$, since the initial helicity $H(0)$ is zero for this flow.

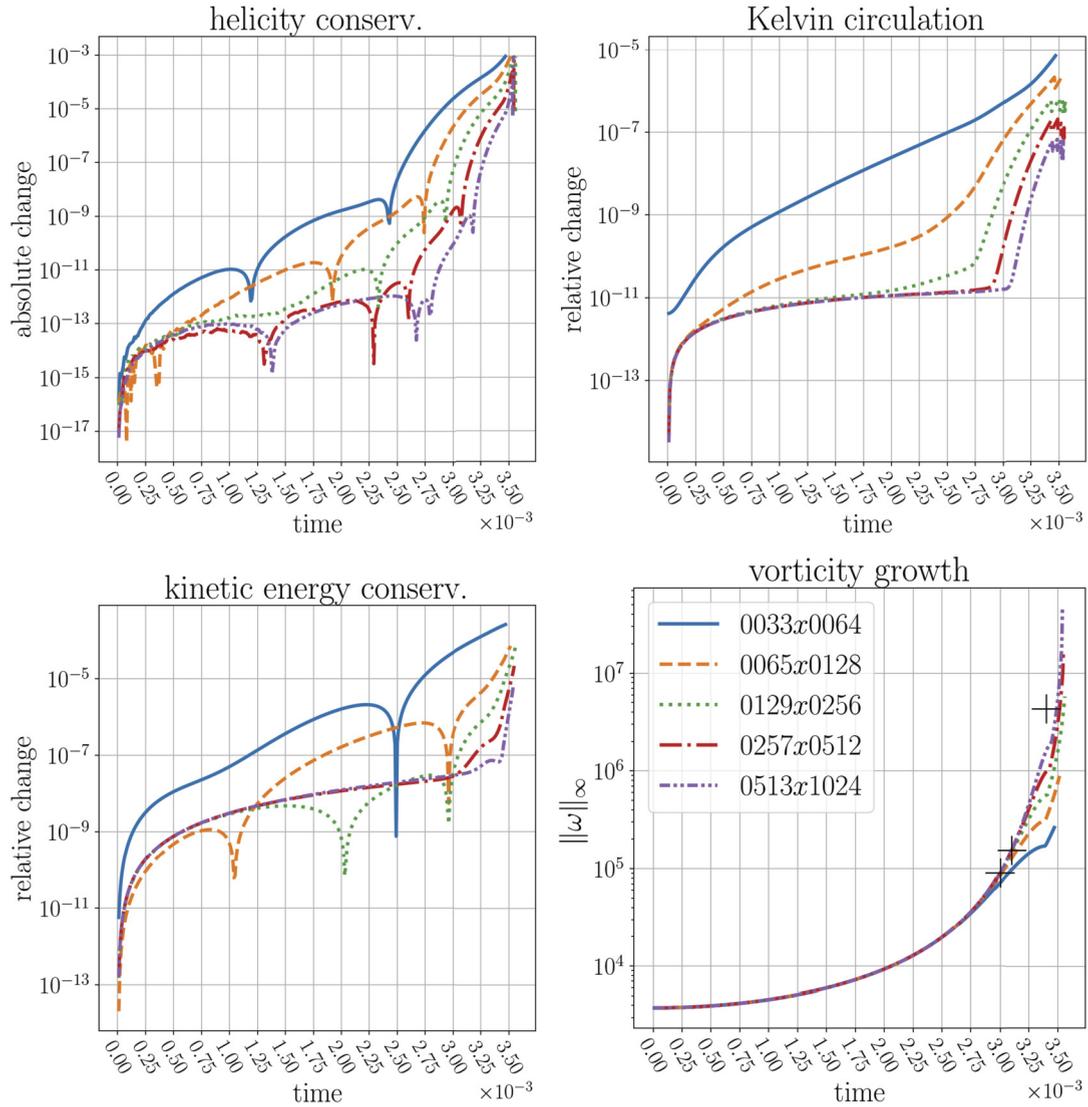


Fig. 13. Resolution study with fixed parameters: $S = 15$, $\Delta t = 10^{-5}$, Bessel-Chebyshev solver, 5th order B-spline interpolation method. The legend in the vorticity growth plots applies to all other panels. Here and in the following plots, the black crosses in the vorticity plot are the exact coordinates from Table 1 at times $t = 0.003$, $t = 0.0031$ and $t = 0.0034$. Also the helicity panel displays absolute errors, because the initial helicity vanishes for this flow.

helicity conservation are usually present, but their cause is not yet known to the authors of this work. The maximal vorticity values that were computed on the largest grid agree qualitatively with the first values in Table 1, but their exact fulfillment will be studied later together with runs in higher resolutions. Note that even for the largest grid in Fig. 13, only 256 vertical grid points lie in the modeled region $D(1, L/4)$ of [45] and [1]. For all the curves in Fig. 13, the Bessel-Chebyshev solver (BC) was used, but the Chebyshev-Galerkin (CG) method produces very similar results. Regarding the second and fourth panel of Fig. 13, the maximal vorticity seems to fall off from its strong growth after a sudden change in the Kelvin conservation. This could indicate that the breaks in the Kelvin conservation and the fall-off in the vorticity growth are connected and yield inaccurate computations from these points on.

Concerning the time-steps, the method that was used to obtain the approximate convergence radius for the stationary solution in Fig. 4 fails for the given initial velocity field in eq. (161). The initial time-Taylor coefficients $X^{(s)}$ cannot be accurately represented for $s \gtrsim 20$ even in quadruple precision, due to a strong growth of their l^∞ -norms. The early development of the quotient $X^{(s-1)}/X^{(s)}$ seems to point to 10^{-4} , see [35, Fig. 5.4]. However, time-steps between 5×10^{-6} and 5×10^{-5} have proven to provide good results. The time-step dependence of the outcome is plotted in Fig. 14. Remarkably, the absolute helicity change is lowest for the largest time-step, while the conservation of kinetic energy and Kelvin circulation seems better with the smallest time-step. Also, if we consider the vorticity growth, then large time-steps and their corresponding vorticity growth falls off later than the growth with smaller time-steps. When searching for sudden breaks in the Kelvin conservation, then it appears that those happen later for larger time-steps, and the same is true for the energy conservation

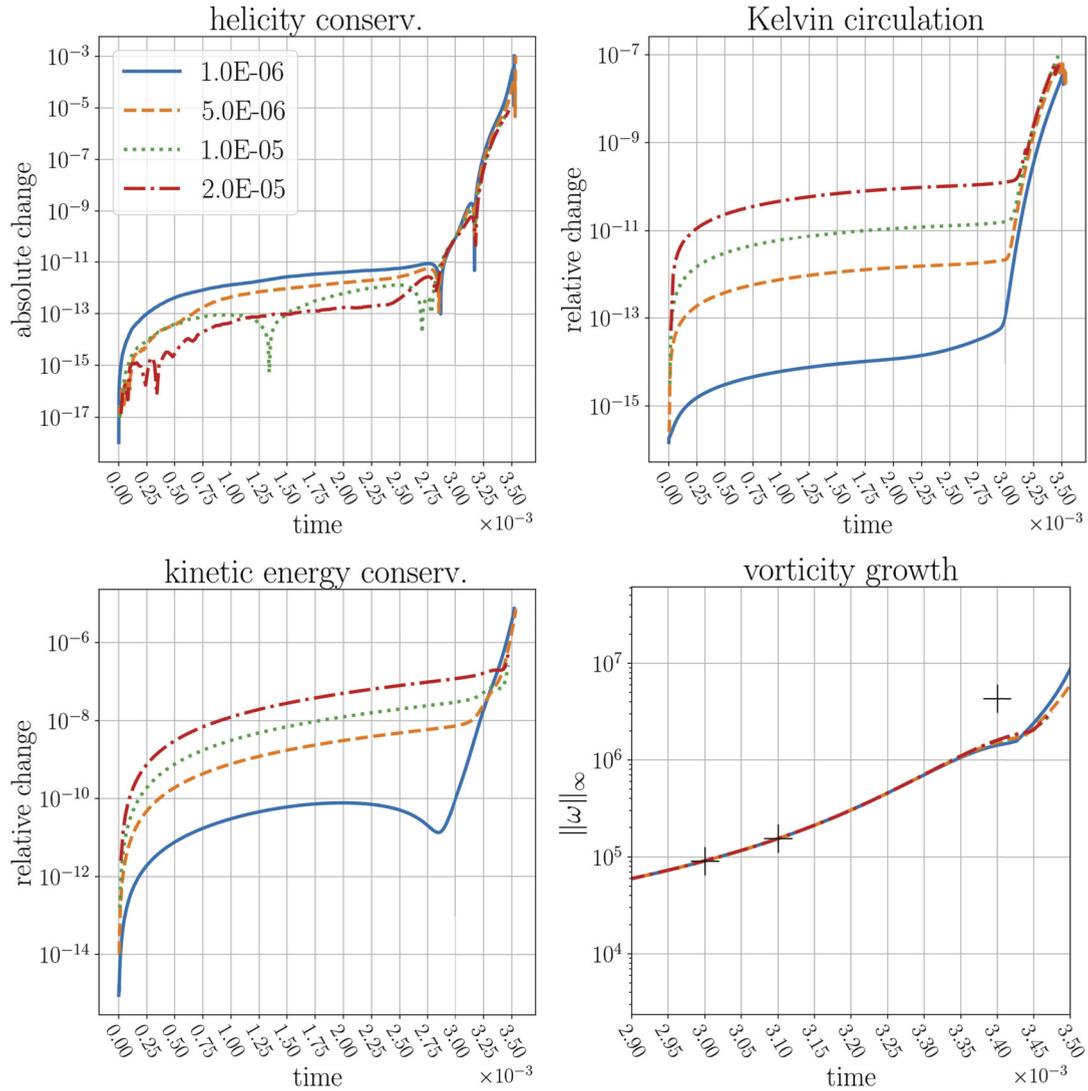


Fig. 14. Several fixed time-steps are compared on a 513×1024 grid; same run-time parameters, other than time-step, as in Fig. 13. Larger time-steps deliver good and fast results. The legend in the first panel is valid for the others too. Note that the fourth panel does not show the whole integration period.

curves. The breaks from a rather regular behavior of the conservation curves seem to indicate a loss of accuracy and, thus, we interpret a later appearance of those breaks as a sign for better stability. Moreover, the runs with larger time-steps take a fraction of the total execution time compared to smaller time-steps, while still providing adequate, if not better, results. The latter property is quite common among semi-Lagrangian methods and helps us here to achieve very large resolutions (see further below). The initial time-steps are not maintained throughout the computation, they are adapted to allow for precision and conservation criteria as mentioned above. However, these adaptations do not happen very often and one time-step can usually be maintained over many iterations.

The adaptive time-stepping procedure discussed in the end of Section 7 is examined in Fig. 15 for the truncation orders $S = 8, 10, 12, 15$. Again, higher truncation orders, thus larger time-steps, show a break out in the Kelvin conservation slightly later and seem to maintain the strong vorticity growth slightly longer than their lower order competitors. The helicity conservation curves are very close to each other, but the high orders give lower absolute changes than the small orders. The energy and Kelvin conservation curves behave seemingly opposite to the helicity behavior. There, the relative errors are lower for lower truncation orders, but those values only depend on the velocity field and are measured by a quotient, not a difference.

Extensive tests have been carried out, involving higher working precision, different interpolation orders, the use of de-aliasing techniques, as well as refinement strategies in the interpolation step. We would like to refer to [35, Chapter 7.2] for illustrations and discussions of those results. In summary, a higher working precision does not necessarily lead to better

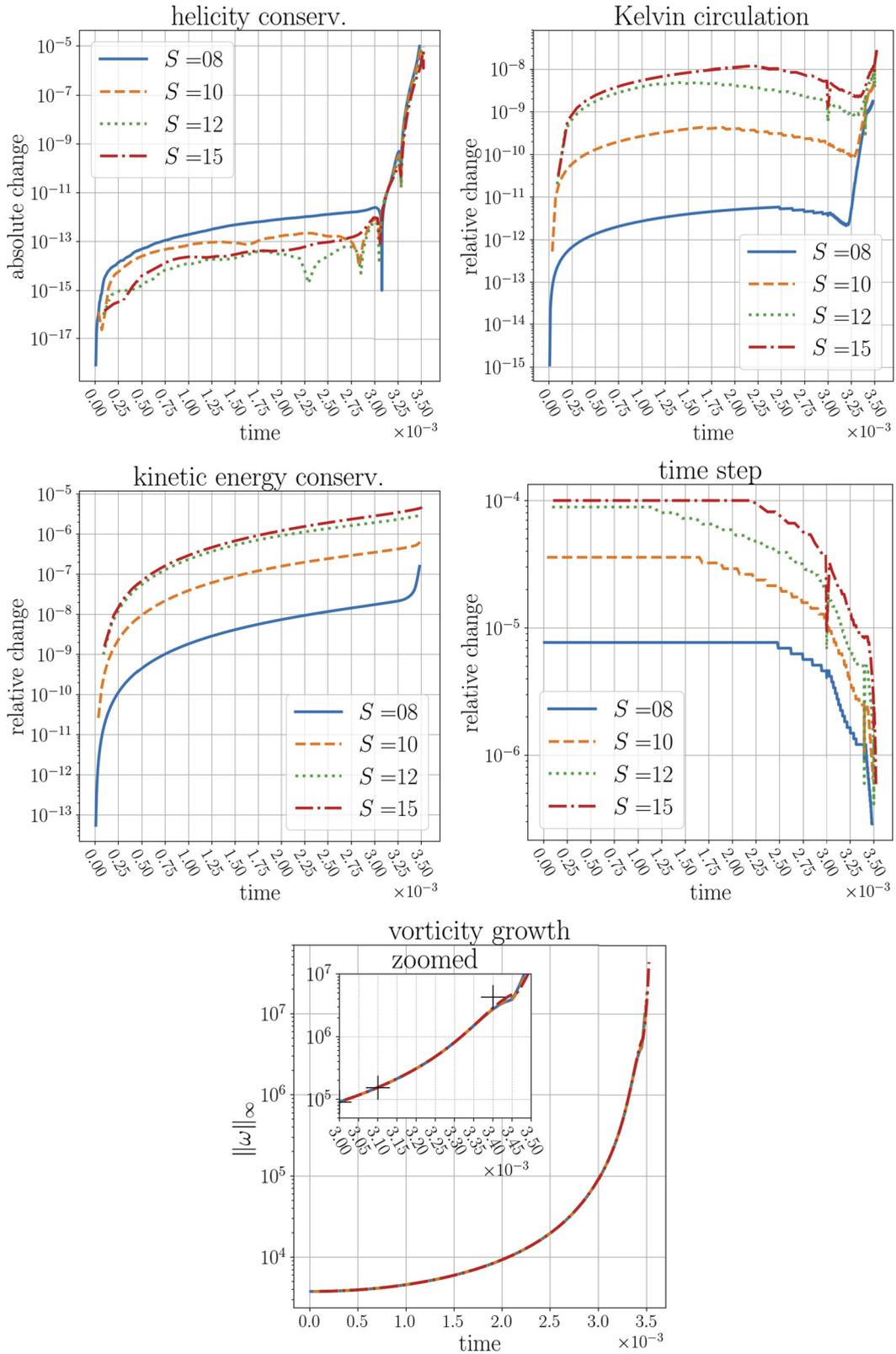


Fig. 15. The effect of different truncation orders in the adaptive time-stepping. Higher orders imply larger time-steps and fewer iterations. These runs were executed on a 513×3072 grid, the other parameters are the same as in the plots from above. Sudden drops in the size of the time-steps in the lower left panel at $t = 0.003$ (and selected later times) are forced time-steps to arrive exactly at that time for a comparison to the values in Table 1.

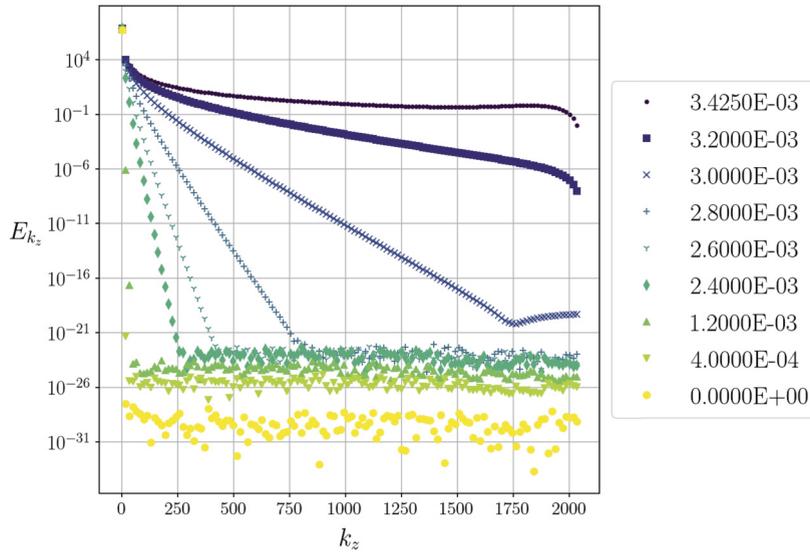


Fig. 16. Energy spectrum at the boundary $r = 1$ for various time instances. Small modes fill up rapidly and the flow can soon not be resolved accurately anymore. The underlying grid size was fixed to 513×4096 Chebyshev-Fourier points. All other fixed run time parameters are given in the text.

Table 2

The body of this table states the computed maximal vorticity at selected times and grid sizes. The values in the last row are those of Table 1 and come from the given references.

| time | $3.0 \cdot 10^{-3}$ | $3.1 \cdot 10^{-3}$ | $3.4 \cdot 10^{-3}$ |
|---------------------|-------------------------|-------------------------|-------------------------|
| 513×1024 | $9.08451276 \cdot 10^4$ | $1.54167635 \cdot 10^5$ | $1.60528443 \cdot 10^6$ |
| 513×4096 | $9.08466826 \cdot 10^4$ | $1.54277249 \cdot 10^5$ | $3.13597705 \cdot 10^6$ |
| 1025×8192 | $9.08466912 \cdot 10^4$ | $1.54277317 \cdot 10^5$ | $3.82533611 \cdot 10^6$ |
| 1025×10240 | $9.08466912 \cdot 10^4$ | $1.54277318 \cdot 10^5$ | $3.94749487 \cdot 10^6$ |
| ref. [45,1] | $9.0847 \cdot 10^4$ | $1.54276901 \cdot 10^5$ | $4.3127 \cdot 10^6$ |

results.¹⁰ The reason for that simply lies in the fast evolution of the turbulent behavior of the flow. Ever smaller Fourier-scales fill up rapidly and the flow quickly becomes under-resolved, as seen in Fig. 16. Such an under-resolution cannot be dealt with by increasing the working precision, but only by increasing the resolution. Regarding the interpolation orders, lower to moderate orders give better performance, as for the stationary flow, due to lesser error amplification. De-aliasing methods¹¹ seem not to improve the results and refinement methods in the interpolation fall short also because of the under-resolution.

Clearly, the strongest influence is given by the resolution and will be further investigated. Fig. 17 shows a comparison between a few higher resolutions. The largest grid size, which has been plotted, is 1025×10240 . The grid spacing at the boundary of this flow is $2.35 \cdot 10^{-6}$ in radial direction (due to the clustering of the Chebyshev extrema) and $1.63 \cdot 10^{-5}$ in vertical direction. Our mesh spacings are of course still very far from the mesh spacings that were achieved in [45] close to the reported singularity, but they are comparable to those in [1]. For higher resolutions, the curves belonging to conserved quantities stay longer in a regular shape and break their behavior later, when compared to lower resolution runs, which is also visible in Fig. 13. The later those breaks appear, the better is the approach of the maximal vorticity to the value at $t = 0.0034$ given in [45] and recalled in Table 1. This qualitative observation can be examined more closely, together with other values from Table 1, by comparing the exact values in the following Table 2.

Our maximal vorticity values show good agreement with those from the references for $t = 3 \cdot 10^{-3}$ and $t = 3.1 \cdot 10^{-3}$. The value at $3.4 \cdot 10^{-3}$ is approached but only agrees in magnitude. Those are exceptional results, in particular, if the execution time of the simulations is considered. All of the computations presented in Table 2 and Fig. 17 were carried out on a single CPU¹² and finished within one day. Typically, implementations of CFL bound algorithms need multiples of that time span on a comparable CPU and execute thousands of time-steps. The run of the CLA on the highest resolution in Table 2 reached $t = 3.1 \cdot 10^{-3}$ in only 310 iterations and $t = 3.4 \cdot 10^{-3}$ in 344. Even larger grid sizes and runs with a higher

¹⁰ Here, we mean concerning a larger integration period together with improved conservation of the discussed quantities.

¹¹ Mainly the 2/3-rule and double length methods.

¹² The cluster in use provides Intel Xeon Gold 6148 CPUs with 20 cores and 40 possible threads, which are individually supplemented by 192GB of memory (RAM).

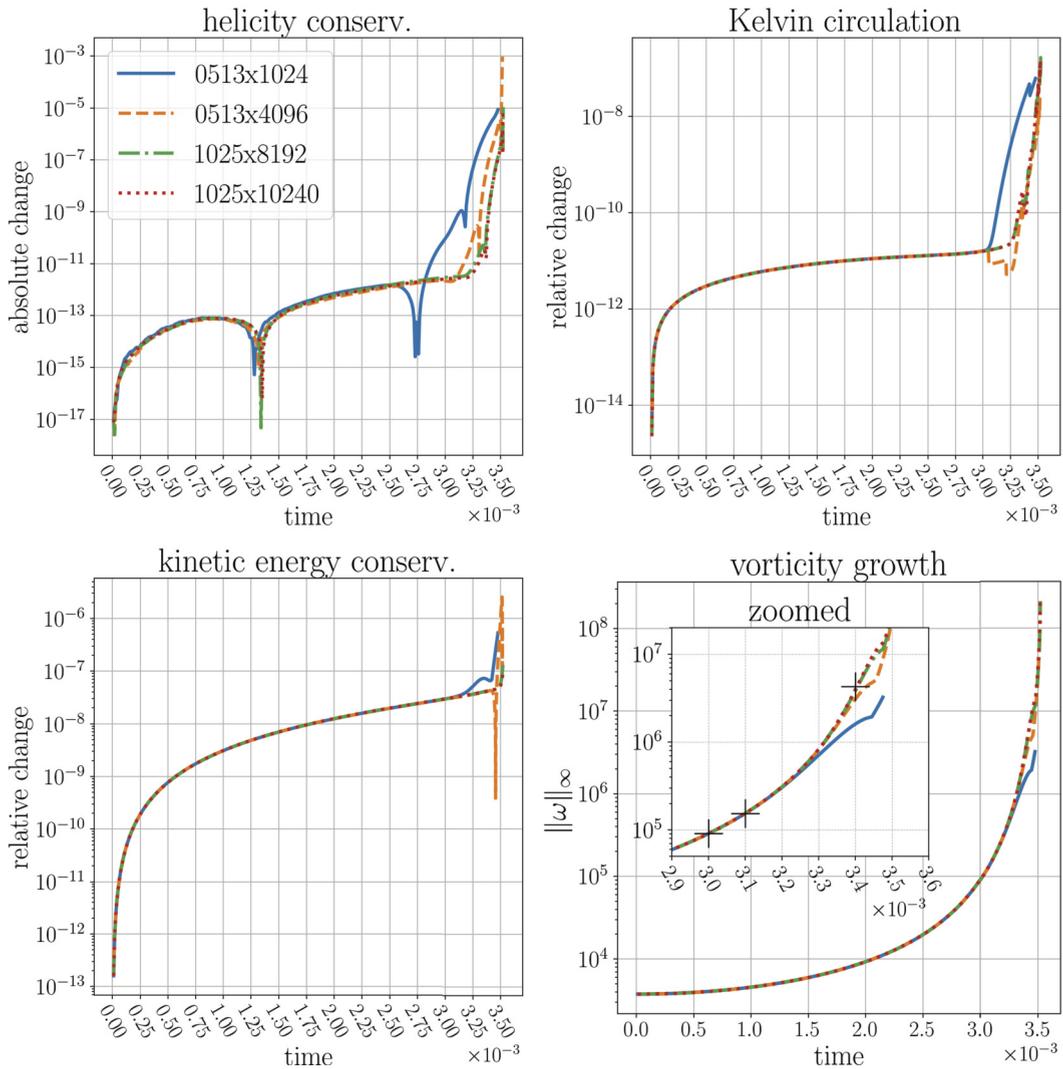


Fig. 17. Higher resolution runs are shown. Here, the truncation order was set to $S = 18$ to maintain the initial time-step 10^{-5} longer. The interpolation was done by the 5th order B-spline cascade; no dealiasing is used here. Sudden breaks in the conserved quantities arrive much later than in the previous plots, indicating a lasting stability with increased resolution.

working precision at those high resolutions are possible. However, larger grid sizes or a higher working-precision certainly increase the memory consumption of the CLA. The current implementation is particularly memory demanding, because all the components of the coefficients, as well as their spatial derivatives and divisions by r , up to the truncation order S , are stored for direct insertion into the recurrence relations. In return, this allows us to avoid numerical errors from standard (spectral) differentiation schemes. Surely, for the particular flow that is investigated here and in the given references, namely a blow-up candidate for the incompressible Euler equations, it is much better and more efficient to exploit the underlying symmetries, as in [45] and [1], which allows modeling the flow in only a quarter of the periodicity domain. In the case of the CLA, such symmetry considerations would cut the memory demands tremendously and would thus allow to achieve highly resolved Euler simulations in a fraction of the time of CFL bound algorithms for the same problem. As the presented work gives a first realization of the CLA in a wall-bounded domain, it was intended to provide a code that can simulate any axisymmetric, periodic flow, which does not necessarily possess special symmetries. Fig. 18 displays the profiles of the flow field components at $r = 1$. The vertical derivatives of the angular and vertical velocity components seem to become singular. The radial velocity component should stay zero at $r = 1$ for all times, due to the no-slip boundary condition, but errors build up especially at the symmetry axes of the flow. Better preservation of the no-slip boundary condition may indicate a higher validity of the simulation and is tracked in our implementation of the CLA.

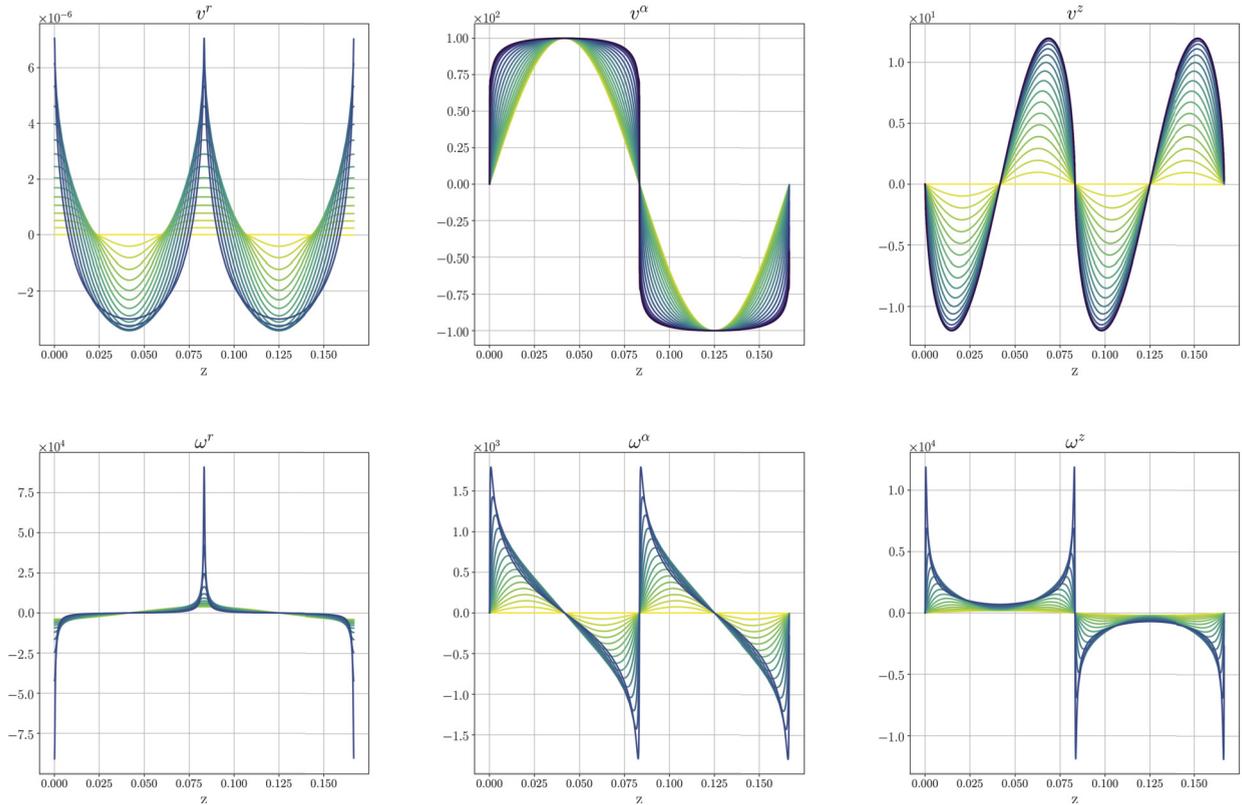


Fig. 18. Upper row: Plot of the velocity components at $r = 1$ until $t = 3.51 \times 10^{-3}$. The values were obtained from a simulation with 513×4096 grid points, the other run time parameters were the same as in the preceding figures. The velocity component v^r in the top left panel should stay zero at $r = 1$ due to the no-flow boundary condition, but large errors appear eventually at the symmetry axes. Lower row: Plot of the vorticity components at $r = 1$. Here, the evolution of the components is only displayed until $t = 3.0 \times 10^{-3}$, the growth of the local extrema is too strong to be conveniently visualized. For surface plots of the flow field components, we would like to refer the reader to [35].

11. Conclusions

In this work, we have implemented and examined the Cauchy-Lagrange algorithm (CLA) for an axisymmetric flow in a cylindrical domain with boundaries. We have systematically investigated several methods for the resulting Poisson problems, the choice of suitable time-steps, interpolation of the flow fields from the distorted Lagrangian grid back onto the regular Eulerian grid, as well as validation criteria for the computed flow. Special attention has been given to the Calderon-Zygmund operators of order zero, namely the second-order derivatives of an inverse Laplacian. The accuracy in those derivatives of the solutions to the Poisson problems, which are boundary value problems, have strong implications on the overall quality of a simulation with the CLA. In this respect, we have avoided error-prone standard differentiation methods in spectral space and developed effective techniques to directly solve for the needed derivatives. Also, common to other semi-Lagrangian methods, the interpolations of the flow fields introduce larger errors and, thus, larger time-steps yield better results, as fewer interpolations have to be carried out. In the CLA, the time-steps are only bound by the convergence radius of the time-Taylor series of the trajectories, and, therefore, they are independent of any CFL condition. This means that, for a given initial velocity field, we can use a certain time-step, which is a fraction of the convergence radius, regardless of the underlying mesh. However, due to the finite working precision in a computation,¹³ the time-steps cannot be chosen too close to the radius of convergence, as demonstrated in Fig. 5. In our tests, time-steps with the size of around one fourth of the convergence radius in a double precision computation produced stable and accurate results. Larger fractions of the convergence radius for a time-step may be achieved, using a higher working precision. Our current implementation of the CLA allows to choose between double, extended double, or quadruple precision, whenever the Bessel-Chebyshev (BC) solver is selected. Unfortunately, the BC method is currently restricted to 513×1024 Chebyshev-Fourier points with reasons explained in Section 5.3. The Chebyshev-Galerkin (CG) method only works in double precision, but it can also be adapted to work in multiple precision modes.¹⁴

¹³ Here, we refer to double (8 Byte floats), extended double (10 Byte floats), or quadruple (16 Byte floats) precision computations.

¹⁴ The appearing linear systems in the CG method are internally solved with the LAPACK package, which is restricted to double precision computations.

Regarding the simulation of the potentially singular flow treated in [45], we have seen that a high resolution is crucial to catch the strong small-scale growth of the vorticity field. Using high resolutions, we were able to achieve good agreement with the maximal vorticity values given in [45] at times, which are not too far from the potential singularity. Currently, our CLA implementation only works on a fixed product grid, but a (locally) adaptive mesh, as used in [45], seems to be mandatory to reproduce their extreme maximal vorticity values. Furthermore, the CLA results presented here compute the flow in the whole cylinder $D(1, L)$, instead of in the quarter cylinder $D(1, L/4)$ (as in [45,1]), which can be seen as a draw-back in computing this particular flow. Future work may consist of restricting the CLA to the quarter cylinder in combination with a locally adaptive mesh, where grid points are shifted into the region of high vorticity values. For this, however, the Chebyshev-Fourier series representation of the flow-fields must be dropped, which could result in a slower algorithm due to the non-applicability of fast Fourier transforms. A solution, like the one performed in [5], could be to replace the Chebyshev-Fourier representation by a multi-wavelets representation to obtain a local adaptive multi-scales representation of the solution on a non-uniform dyadic grid. Indeed, in the adaptive multiresolution semi-Lagrangian method designed in [5] the authors use interpolating wavelets (interpolets) which are built on polynomial spaces (Lagrange, Legendre, Chebyshev, ...) and enjoy a fast wavelet transform (FWT) like the FFT for the Chebyshev-Fourier representation.

An easier alternative approach to spatial adaptivity is to implement a global refinement method via uniform dyadic grids. This can readily be achieved in our implementation, provided we stay in the full cylinder. More precisely, in exploiting the nesting property of our Chebyshev-Fourier grid of sizes $(2^N + 1) \times 2^M$, one may double the grid size, whenever the spectral dimension is insufficient to accurately represent the flow fields, i.e. when the smallest modes become non-zero. This refinement mechanism is described in detail in [35, Section 4.2], but it is yet only implemented in the CLA to improve the interpolation of the flow fields back onto the Eulerian grid.¹⁵ In using a global refinement method with the mentioned dyadic grids, we can approach a potentially singular time of a flow much faster and with less computer resources, because we can start a simulation on a coarse grid and refine it when needed. Additional perspectives for future work can be the extension of the presented CLA to fully three-dimensional non-axisymmetric flow and to flows in other domains with boundaries. In this spirit, it is crucial to simulate other potentially singular flows in bounded domains and check numerically the location of blowing-up gradients in the velocity field. Moreover, in the fully three-dimensional case without symmetries and for potential singularities living on a low-dimensional subspace (typically a point or a one-dimensional curve) an adaptive multiresolution Cauchy-Lagrange method, based on a multi-wavelets representation, should be essential. The presence of boundaries may, after all, be a driving factor for the appearance of finite-time singularities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We are very grateful to Claude Bardos, Sai Swetha Venkata Kolluru and Rahul Pandit for fruitful discussions. This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the EURATOM research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A. Chebyshev polynomials and series representation

Introductions to Chebyshev polynomials can be found plentiful in the literature (e.g. [47], [9], [28]). Here, we will only recall from the given references the definitions and properties that are used in our computations.

Definition. The Chebyshev polynomial of the first kind of order n is defined as

$$T_n(x) = \cos(n \arccos(x)), \quad x \in I, \quad n \in \mathbb{N}_0, \quad (167)$$

where $I := [-1, 1]$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\} = \{0, 1, 2, \dots\}$.

Setting $x = \cos(\vartheta)$, $\vartheta \in [0, \pi]$, we find

$$T_n(\cos(\vartheta)) = \cos(n\vartheta), \quad (168)$$

from which all the properties in the following may be derived. The Chebyshev polynomials satisfy the three term recurrence relation

¹⁵ For the results in this work, we have not applied the mentioned refinement methods in the interpolations.

$$\begin{aligned}
 T_0(x) &= 1, \\
 T_{n+1}(x) &= 2xT_n(x) - T_{|n-1|}(x), \quad \forall n \in \mathbb{N}_0.
 \end{aligned}
 \tag{169}$$

The Chebyshev extrema, i.e. the local extrema of $T_n(x)$ in the interval I , are given by

$$x_i = \cos\left(\frac{i\pi}{n}\right), \quad i \in \{0, \dots, n\},
 \tag{170}$$

such that $T_n(x_i) = (-1)^i$, the zeros are given by

$$\tilde{x}_i = \cos\left(\frac{2i+1}{2n}\pi\right), \quad i \in \{0, \dots, n-1\}.
 \tag{171}$$

Without difficulty, one verifies the nesting property of the Chebyshev extrema, that is

$$\left\{ \cos\left(\frac{i\pi}{n}\right) \right\}_{0 \leq i \leq n} \subset \left\{ \cos\left(\frac{i\pi}{2n}\right) \right\}_{0 \leq i \leq 2n}, \quad \forall n \in \mathbb{N},
 \tag{172}$$

or, in other words, the extrema of T_n are also extrema of T_{2n} . Furthermore, there holds the orthogonality relation

$$\int_{-1}^1 T_l(t)T_k(t)\omega(t) dt = \int_{-1}^1 \frac{T_l(t)T_k(t)}{\sqrt{1-t^2}} dt = c_l \frac{\pi}{2} \delta_{lk},
 \tag{173}$$

where

$$c_l := \delta_{l0} + 1 = \begin{cases} 2, & \text{if } l = 0, \\ 1, & \text{if } l > 0, \end{cases}
 \tag{174}$$

and $\delta_{(\cdot)}$ denotes the Kronecker symbol. A discrete version of the latter relation, based on the Chebyshev extrema $(x_i)_{0 \leq i \leq N \in \mathbb{N}}$ from above, reads

$$\sum_{i=0}^N \frac{1}{c_i c_{N-i}} T_l(x_i)T_k(x_i) = \frac{c_l c_{N-l}}{2} N \delta_{lk},
 \tag{175}$$

for $l, k = 0, \dots, N$. In the literature the above summation $\sum_{i=0}^N 1/(c_i c_{N-i})$, indicating the halving of the first and last terms, is often denoted by \sum'' and $\sum_{i=0}^N 1/c_i$ by \sum' . Here we intend to avoid primed sums, the appearing primes in the text represent derivatives with respect to the sole argument of a function.

A multiplication of two Chebyshev polynomials entails

$$2T_n T_m = T_{n+m} + T_{|n-m|}, \quad \forall n, m \in \mathbb{N}_0,
 \tag{176}$$

with the special case

$$xT_n = \frac{1}{2}(T_{n+1}(x) + T_{|n-1|}(x)).
 \tag{177}$$

The derivatives of the Chebyshev polynomials fulfill the relations

$$\begin{aligned}
 T_0 &= T'_1, \quad 4T_1 = T'_2, \\
 2T_n &= \left(\frac{T_{n+1}'}{n+1} - \frac{T_{n-1}'}{n-1} \right), \quad n \geq 2,
 \end{aligned}
 \tag{178}$$

as well as

$$T_n'' = \sum_{\substack{k=1 \\ k+n \text{ even}}}^{n-2} \frac{1}{c_k} n(n^2 - k^2) T_k.
 \tag{179}$$

The *shifted Chebyshev polynomial* of the first kind of order $n \in \mathbb{N}_0$ is obtained from the ordinary Chebyshev polynomial by a change of the variable, namely $t = 2r - 1$ with $r \in [0, 1]$, and is denoted by

$$T_n^*(r) := T_n(2r - 1), \quad r \in [0, 1].
 \tag{180}$$

The shifted Chebyshev polynomials satisfy the three term recurrence relation

$$\begin{aligned} T_0^*(r) &= 1, \\ T_n^*(r) &= 2(2r - 1)T_{n-1}^* - T_{|n-2|}^*, \quad n \in \mathbb{N}, \end{aligned} \tag{181}$$

which implies

$$rT_n^* = \frac{1}{4} \left(T_{|n-1|}^* + 2T_n^* + T_{n+1}^* \right), \quad \forall n \in \mathbb{N}_0. \tag{182}$$

An explicit representation of the shifted Chebyshev polynomial of order $n > 0$ is given by the polynomial sum

$$T_n^*(r) = \sum_{j=0}^n n \frac{4^j (-1)^{n-j} (j+n-1)!}{(2j)!(n-j)!} r^j. \tag{183}$$

From eq. (178), in using eq. (180), we directly obtain the derivative relations

$$2T_0^* = T_1^{*\prime}, \quad 8T_1^* = T_2^{*\prime}, \quad 4T_n^* = \left(\frac{T_{n+1}^{*\prime}}{n+1} - \frac{T_{n-1}^{*\prime}}{n-1} \right), \quad n \geq 2, \tag{184}$$

and with it the integral relations

$$\begin{aligned} \int T_0^*(r) dr &= \frac{T_1^*}{2}, \quad \int T_1^*(r) dr = \frac{T_2^*}{8}, \\ \int T_n^*(r) dr &= \frac{1}{4} \left(\frac{T_{n+1}^*}{n+1} - \frac{T_{n-1}^*}{n-1} \right), \quad n \geq 2, \end{aligned} \tag{185}$$

which are valid up to an additive constant of integration. The next integral relation follows readily from eqs. (182) and (185),

$$\begin{aligned} \tau(n, r) &:= \int rT_n^*(r) dr = \int \frac{1}{4} \left(T_{|n-1|}^*(r) + 2T_n^*(r) + T_{n+1}^*(r) \right) dr \\ &= \begin{cases} \frac{1}{4}T_1^*(r) + \frac{1}{16}T_2^*(r), & n = 0 \\ \frac{1}{16}T_1^*(r) + \frac{1}{16}T_2^*(r) + \frac{1}{48}T_3^*(r), & n = 1 \\ -\frac{1}{8}T_1^*(r) + \frac{1}{24}T_3^*(r) + \frac{1}{64}T_4^*(r), & n = 2 \\ \frac{1}{16} \left(-\frac{T_{n-2}^*(r)}{n-2} + \frac{2T_{n+1}^*(r)}{n+1} - \frac{2T_{n-1}^*(r)}{n-1} + \frac{T_{n+2}^*(r)}{n+2} \right), & n \geq 3, \end{cases} \end{aligned} \tag{186}$$

where the constant of integration has again been omitted. In the following we assume to be given function values $u_i^* = u(x_i^*)$, $i = 0, 1, \dots, N$, of a continuous function $u : [0, 1] \rightarrow \mathbb{R}$ on the shifted Chebyshev extrema

$$x_i^* := \frac{x_i + 1}{2} \in [0, 1], \tag{187}$$

with x_i as in eq. (170). Note that the shifted Chebyshev extrema fulfill the same nesting property (172) as the non-shifted Chebyshev extrema. An approximation u^N of u is given by the shifted Chebyshev series

$$u^N(r) = \sum_{l=0}^N u_l T_l^*(r), \tag{188}$$

such that $u^N(x_i^*) = u_i^*$. The series coefficients u_l are given by

$$u_l = \frac{1}{c_l c_{N-l}} \frac{1}{N} \left(u_0^* + (-1)^l u_N^* + 2 \sum_{i=1}^{N-1} u_i^* \cos \left(\frac{il\pi}{N} \right) \right). \tag{189}$$

The bracketed term is commonly known as the type-I discrete cosine transform, or DCT-I, of the vector $(u_i^*)_{0 \leq i \leq N}$; it is implemented in numerous open source software.¹⁶ The transform from function values based on the Chebyshev zeros to the Chebyshev coefficients u_l is defined by a DCT-II, whose back-transform is a DCT-III. However, for our numerical needs, namely the inclusion of interval end-points (boundary points of the cylinder) and the nesting property (172) that is not fulfilled by the Chebyshev zeros, we use the shifted Chebyshev extrema throughout this work. However, the zeros may be used for integration purposes in form of the Gauss-integration scheme, which is reviewed here briefly as well. See [11,48] for the following statements and proofs thereof.

¹⁶ We use FFTW 3.3.9, see http://www.fftw.org/fftw3_doc/What-FFTW-Really-Computes.html for a definition of their implemented transforms.

Gauss integration for Chebyshev polynomials. Let w_0, \dots, w_{n-1} be the solution of the linear system

$$\sum_{j=0}^{n-1} (\tilde{x}_j)^k w_j = \int_{-1}^1 x^k \omega(x) dx, \quad 0 \leq k \leq n-1,$$

where $\tilde{x}_0 < \tilde{x}_1 < \dots < \tilde{x}_{n-1}$ are the roots of the n -th Chebyshev polynomial from eq. (171). Then there holds

$$w_j = \frac{\pi}{n}, \quad \text{for } j = 0, \dots, n-1,$$

and

$$\sum_{j=0}^{n-1} p(\tilde{x}_j) w_j = \int_{-1}^1 p(x) w(x) dx, \quad \text{for all } p \in \mathbb{P}_{2n-1}. \tag{190}$$

If a function $g = g(r)$ is given by a shifted Chebyshev series with coefficients $\{g_l^{(0)}\}_{l=0, \dots, N}$, then the first-order derivative can be expressed by a shifted Chebyshev series as well, with coefficients $\{g_l^{(1)}\}_{l=0, \dots, N-1}$. The latter coefficients are obtained via the relation

$$c_{l-1} g_{l-1}^{(1)} = g_{l+1}^{(1)} + 4l g_l^{(0)}, \quad l = N, N-1, \dots, 1, \tag{191}$$

with $g_{N+1}^{(1)} = g_N^{(1)} = 0$ and c_l as in (174). Higher order derivatives can be obtained by a re-application of this relation.

Also the division $g(r)/r$ can be given by a shifted Chebyshev series with coefficients $\{a_l\}_{l=0, \dots, N}$, in solving the linear system of equations

$$A \cdot (a_l)_l = (g_l^{(0)})_l, \tag{192}$$

where

$$A = \begin{pmatrix} 1/2 & 1/4 & 0 & 0 & \dots \\ 1/2 & 1/2 & 1/4 & 0 & \\ 0 & 1/4 & 1/2 & 1/4 & \\ 0 & 0 & 1/4 & 1/2 & \\ \vdots & & & & \ddots \end{pmatrix}. \tag{193}$$

Appendix B. Explicit computation of the time-Taylor coefficients for stationary solutions

We depart from the characteristic equation and leave the dependence on the arguments out for convenience (the following is true in any coordinate system),

$$\dot{X}(t) = V(X(t)), \quad X(0) = \text{Id}, \tag{194}$$

for a classical stationary solution $V = V(a)$ to the three dimensional incompressible axisymmetric Euler equations. If $X(t)$ is analytic in time, then so is the velocity along the trajectories $V(X(t))$ due to eq. (194). Let $g(t) := V(X(t))$, then

$$g(t) = \sum_{s \geq 0} g^{(s)} t^s, \quad \text{with } g^{(s)} = \frac{1}{s!} \partial_t^s V(X(t))|_{t=0}. \tag{195}$$

One finds

$$\sum_{s \geq 1} s X^{(s)} t^{s-1} = \sum_{s \geq 0} g^{(s)} t^s, \tag{196}$$

and after collecting powers of t ,

$$X^{(s)} = g^{(s-1)}, \quad s \geq 1. \tag{197}$$

The first coefficients are, thus, given by $X^{(0)} = \text{Id}$, $X^{(1)} = V$, $(2!)X^{(2)} = DV \cdot X^{(1)} = DV \cdot V$, $(3!)X^{(3)} = X^{(1)\top} \cdot D^2V \cdot X^{(1)} + DV \cdot X^{(2)} = V^\top \cdot D^2V \cdot V + DV \cdot (DV \cdot V), \dots$

Despite the fact that the general form may be given by a recursion formula and the formula of Faa di Bruno [16,34], here, we are mostly interested in the explicit calculation of the time Taylor coefficients in cylindrical coordinates. This is possible by hand in the case of the vortex, but rather exhausting in the case of more intricate solutions as presented in [43]. We will therefore make use of computer algebra systems, but first state the recursion in cylindrical coordinates.

References

- [1] D. Barkley, A fluid mechanic's analysis of the teacup singularity, *Proc. R. Soc. A* 476 (2020) 20200348.
- [2] H. Bateman, *Tables of Integral Transforms*, Ed. A. Erdélyi, Vol. II, McGraw-Hill Book Company, 1954.
- [3] J.T. Beale, T. Kato, A. Majda, Remarks on the breakdown of smooth solutions for the 3-D Euler equations, *Commun. Math. Phys.* 94 (1) (1984) 61–66.
- [4] A. Bennett, *Lagrangian Fluid Dynamics*, Cambridge University Press, 2006.
- [5] N. Besse, E. Deriaz, É. Madaule, Adaptive multiresolution semi-Lagrangian discontinuous Galerkin methods for the Vlasov equations, *J. Comput. Phys.* 332 (2017) 376–417.
- [6] N. Besse, Regularity of the geodesic flow of the incompressible Euler equations on a manifold, *Commun. Math. Phys.* 375 (3) (2020) 2155–2189.
- [7] N. Besse, U. Frisch, A constructive approach to regularity of Lagrangian trajectories for incompressible Euler flow in a bounded domain, *Commun. Math. Phys.* 351 (2) (2017) 689–707.
- [8] N. Besse, U. Frisch, Geometric formulation of the Cauchy invariants for incompressible Euler flow in flat and curved spaces, *J. Fluid Mech.* 825 (2017) 412–478.
- [9] J.P. Boyd, *Chebyshev and Fourier Spectral Methods*, Courier Corporation, 2001.
- [10] M.D. Bustamante, R.M. Kerr, 3D Euler about a 2D symmetry plane, *Phys. D: Nonlinear Phenom.* 237 (14–17) (2008) 1912–1920.
- [11] C. Canuto, et al., *Spectral Methods: Fundamentals in Single Domains*, Springer Science & Business Media, 2007.
- [12] A.-L. Cauchy, Sur l'état du fluide à une époque quelconque du mouvement, *Mém. Extraits Recueils Acad. Sci. Inst. Fr., Sci. Math. Phys.* 1 (1827) 33–73.
- [13] J. Chen, T.Y. Hou, Finite time blowup of 2D Boussinesq and 3D Euler equations with $C^{1,\alpha}$ velocity and boundary, arXiv preprint, arXiv:1910.00173, 2019.
- [14] P. Constantin, V. Vicol, J. Wu, Analyticity of Lagrangian trajectories for well posed inviscid incompressible fluid models, *Adv. Math.* 285 (2015) 352–393.
- [15] C. De Boor, *A Practical Guide to Splines*, Applied Mathematical Sciences, vol. 27, Springer, 2001.
- [16] F. Di Bruno, Note sur une nouvelle formule de calcul différentiel, *Pure Appl. Math. Q.* 1 (1857) 359–360.
- [17] W. E, C.-W. Shu, Small-scale structures in Boussinesq convection, *Phys. Fluids* 6 (1) (1994) 49–58.
- [18] D.G. Ebin, J.E. Marsden, Groups of diffeomorphisms and the motion of an incompressible fluid, *Ann. Math.* 92 (1) (1970) 102–163.
- [19] T.M. Elgindi, I.-J. Jeong, Finite-time singularity formation for strong solutions to the axis-symmetric 3D Euler equations, *Ann. PDE* 5 (16) (2019).
- [20] L. Euler, Principes généraux du mouvement des fluides, *Mém. Acad. Sci. Berlin* 11 (1757) 274–315.
- [21] A.B. Ferrari, On the blow-up of solutions of the 3-D Euler equations in a bounded domain, *Commun. Math. Phys.* 155 (2) (1993) 277–294.
- [22] M. Frigo, S.G. Johnson, FFTW3, 2006, available from <http://www.fftw.org/#documentation>, 2007.
- [23] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, *Proc. IEEE* 93 (2) (2005) 216–231.
- [24] U. Frisch, G. Grimberg, B. Villone, A contemporary look at Hermann Hankel's 1861 pioneering work on Lagrangian fluid dynamics, *Eur. Phys. J. H* 42 (2017) 537–556.
- [25] U. Frisch, B. Villone, Cauchy's almost forgotten Lagrangian formulation of the Euler equation for 3D incompressible flow, *Eur. Phys. J. H* 39 (3) (2014) 325–351.
- [26] U. Frisch, V. Zheligovsky, A very smooth ride in a rough sea, *Commun. Math. Phys.* 326 (2) (2014) 499–505.
- [27] P. Gamblin, Système d'Euler incompressible et régularité microlocale analytique, *Ann. Inst. Fourier* 44 (5) (1994) 1449–1475.
- [28] W. Gautschi, *Orthogonal Polynomials*, Oxford University Press, 2004.
- [29] J.D. Gibbon, The three-dimensional Euler equations: where do we stand?, *Phys. D: Nonlinear Phenom.* 237 (14–17) (2008) 1894–1904.
- [30] V. Girault, P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms*, Vol. 5, Springer Science & Business Media, 2012.
- [31] O. Glass, F. Sueur, T. Takahashi, Smoothness of the motion of a rigid body immersed in an incompressible perfect fluid, *Ann. Sci. Éc. Norm. Supér.* 45 (1) (2012) 1–51.
- [32] R. Grauer, T.C. Sideris, Numerical computation of 3D incompressible ideal fluids with swirl, *Phys. Rev. Lett.* 67 (25) (1991) 3511–3514.
- [33] N. Gyunter, Über ein Hauptproblem der Hydrodynamik, *Math. Z.* 24 (1926) 448–499.
- [34] H. Gzyl, Multidimensional extension of Faa di Bruno's formula, *J. Math. Anal. Appl.* 116 (2) (1986) 450–455.
- [35] T. Hertel, The Cauchy–Lagrange algorithm for axisymmetric incompressible Euler flow in a wall bounded cylindrical domain, PhD thesis, Université de la Côte d'Azur, 2020.
- [36] T.Y. Hou, R. Li, Dynamic depletion of vortex stretching and non-blowup of the 3-D incompressible Euler equations, *J. Nonlinear Sci.* 16 (6) (2006) 639–664.
- [37] A. Jeffrey, D. Zwillinger, *Table of Integrals, Series, and Products*, Elsevier, 2007.
- [38] F. Johansson, Arb: efficient arbitrary-precision midpoint-radius interval arithmetic, *IEEE Trans. Comput.* 66 (8) (2017) 1281–1292.
- [39] R.M. Kerr, Evidence for a singularity of the three-dimensional, incompressible Euler equations, *Phys. Fluids A, Fluid Dyn.* 5 (7) (1993) 1725–1746.
- [40] A.C. King, J. Billingham, S.R. Otto, *Differential Equations: Linear, Nonlinear, Ordinary, Partial*, Cambridge University Press, 2003.
- [41] V.S.S. Kolluru, P. Sharma, R. Pandit, A pseudospectral study of a potentially singular solution of the three-dimensional axisymmetric incompressible Euler equation: tygers and thermalization, arXiv preprint, arXiv:2012.14182, 2020.
- [42] S.H. Lamb, *Hydrodynamics*, 6th ed., in: R. Caflich (Ed.), Cambridge Mathematical Library, Cambridge University Press, 1975.
- [43] N. Leprovost, B. Dubrulle, P.-H. Chavanis, Dynamics and thermodynamics of axisymmetric flows: theory, *Phys. Rev. E* 73 (4) (2006) 046308.
- [44] L. Lichtenstein, Über einige Existenzprobleme der Hydrodynamic, *Math. Z.* 281 (1928) 387–415.
- [45] G. Luo, T.Y. Hou, Toward the finite-time blowup of the 3D axisymmetric Euler equations: a numerical investigation, *Multiscale Model. Simul.* 12 (4) (2014) 1722–1776.
- [46] A.J. Majda, A.L. Bertozzi, *Vorticity and Incompressible Flow*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2001.
- [47] J.C. Mason, D.C. Handscomb, *Chebyshev Polynomials*, CRC Press, 2002.
- [48] B. Mercier, *Analyse numérique des méthodes spectrales*, Tech. Rep., 1981.
- [49] F. Oberhettinger, *Tables of Bessel Transforms*, Springer Berlin Heidelberg, 1972.
- [50] F.W.J. Olver, et al., *NIST Handbook of Mathematical Functions* Hardback and CD-ROM, Cambridge University Press, 2010.
- [51] O. Podvigina, V. Zheligovsky, U. Frisch, The Cauchy–Lagrangian method for numerical analysis of Euler flow, *J. Comput. Phys.* 306 (2016) 320–342.
- [52] A. Pumir, E.D. Siggia, Development of singular solutions to the axisymmetric Euler equations, *Phys. Fluids A, Fluid Dyn.* 4 (7) (1992) 1472–1491.
- [53] R.J. Purser, L.M. Leslie, An efficient interpolation procedure for high-order three-dimensional semi-Lagrangian models, *Mon. Weather Rev.* 119 (10) (1991) 2492–2498.
- [54] X. Saint Raymond, Remarks on axisymmetric solutions of the incompressible Euler system, *Commun. Partial Differ. Equ.* 19 (1–2) (1994) 321–334.
- [55] P. Serfati, Étude mathématique de flammes infiniment minces en combustion. Résultats de structure et de régularité pour l'équation d'Euler incompressible, PhD thesis, Thèse de Doctorat de l'Université Paris 6, 1992.
- [56] P. Serfati, Structures holomorphes à faible régularité spatiale en mécanique des fluides, *J. Math. Pures Appl.* 74 (Jan. 1995) 95–104.
- [57] J. Shen, Efficient spectral-Galerkin methods III: polar and cylindrical geometries, *SIAM J. Sci. Comput.* 18 (6) (1997) 1583–1604.
- [58] J. Shen, T. Tang, L.-L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Vol. 41, Springer Science & Business Media, 2011.

- [59] T. Shirota, T. Yanagisawa, Note on global existence for axially symmetric solutions of the Euler system, Proc. Jpn. Acad., Ser. A, Math. Sci. 70 (10) (1994) 299–304.
- [60] A. Shnirelman, On the analyticity of particle trajectories in the ideal incompressible fluid, arXiv preprint, arXiv:1205.5837, 2012.
- [61] M.R. Ukhovskii, V.I. Yudovich, Axially symmetric flows of ideal and viscous fluids filling the whole space, J. Appl. Math. Mech. 32 (1) (1968) 52–62.
- [62] B. Villone, C. Rampf, Hermann Hankel's "On the general theory of motion of fluids", Eur. Phys. J. H 42 (2017) 557–609.