Journal of Computational Physics 518 (2024) 113306

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Bifurcation tracking on moving meshes and with consideration of azimuthal symmetry breaking instabilities

Christian Diddens *, Duarte Rocha

Physics of Fluids group, Department of Science and Technology, Mesa+ Institute, Max Planck Center for Complex Fluid Dynamics and J. M. Burgers Centre for Fluid Dynamics, University of Twente, P.O. Box 217, Enschede, 7500 AE, the Netherlands

A B S T R A C T

We present a black-box method to numerically investigate the linear stability of arbitrary multi-physics problems. While the user just has to enter the system's residual in weak formulation, e.g. by a finite element method, all required discretized matrices are automatically assembled based on just-in-time generated and compiled *C* codes. Based on this method, entire phase diagrams in the parameter space can be obtained by bifurcation tracking and continuation at low computational costs. Particular focus is put on problems with moving domains, e.g. free surface problems in fluid dynamics, since a moving mesh introduces a plethora of complicated nonlinearities to the system. By symbolic differentiation before the code generation, however, these moving mesh problems are made accessible to bifurcation tracking methods. In a second step, our method is generalized to investigate symmetry-breaking instabilities of axisymmetric stationary solutions by effectively utilizing the symmetry of the base state. Each bifurcation type is validated on the basis of results reported in the literature on versatile fluid dynamics problems, for which we subsequently present novel results as well.

1. Introduction

Many physical systems exhibit spontaneous symmetry breaking or fundamental changes in behavior upon subtle changes in a control parameter. Classical examples are e.g. the onset of convection in a Rayleigh-Bénard system, any self-organized pattern formation [1] as e.g. the Turing instability [2], neuronal dynamics as e.g. the FitzHugh-Nagumo model [3] and more.

Despite of the complexity and the inherent nonlinearities in these systems, a promising way to gain insight into the dynamics is the investigation of the linear dynamics around the stationary states close to a critical parameter threshold, i.e. by a linear stability analysis. Thereby, one can calculate at which critical parameter value the system undergoes a transition and additionally obtain the corresponding eigenfunction, from which further understanding of the intrinsic dynamics close to the instability can be extracted. This approach can be applied on systems that can be modeled by a system of ordinary differential equations as well as a system continuous in space, i.e. described by partial differential equations. In particular in the latter case, however, an analytical treatment of the linear stability analysis is often hampered by the nonlinearities, which generically prevent exact analytical solutions for nontrivial stationary solutions. Furthermore, a nontrivial geometry of the system also hampers the applicability of analytical methods.

For these cases, one has to fall back to the numerical solution of the stationary solution and the corresponding eigenvalue problem to find the dominant eigenvalue λ_c of perturbations proportional to $e^{\lambda t}$, i.e. the eigenvalue with the highest real part, and subsequently determine the corresponding critical parameter value when its real part becomes zero, i.e. the bifurcation point at which $\Re(\lambda_c) = 0$ holds. This can be achieved rather quickly by the bisection method, however, this method fails e.g. for fold bifurcations, since the stationary solution branch ceases to exist beyond the critical parameter. Instead, one can virtually jump directly on the bifurcation point by solving an augmented system of equations to simultaneously solve for the stationary solution, the critical parameter value

* Corresponding author.
E-mail addresses: c.diddens@utwente.nl (C. Diddens), d.rocha@utwente.nl (D. Rocha).

https://doi.org/10.1016/j.jcp.2024.113306
Received 16 December 2023; Received in revised form 3 June 2024; Accepted 23 July 2024

Available online 29 July 2024

and the corresponding critical eigenfunction. Once achieved, the bifurcation branch can be tracked along another parameter, e.g. by (pseudo-)arclength continuation. This approach is known as bifurcation tracking and it or related methods have been implemented in a variety of software packages, e.g. PDE2PATH [4], AUTO-07P [5], OOMPH-LIB [6] or LOCA [7]. A detailed overview on numerical stability and bifurcation analysis, with particular focus on fluid dynamics, can be found in Ref. [8]. Also, section 2 briefly introduces the topic, the relevant terminology and the mathematical notation.

From the user perspective, it is desired to provide a simple interface to enter the system of equations to be investigated and subsequently automatize the bifurcation analysis in a black-box manner. However, the assembly of the required Jacobian and mass matrices for the eigenproblem and the bifurcation tracking can be challenging, since these matrices must be treated in a monolithic manner to consider all intrinsic couplings of the problem to obtain the exact eigenproblem and its solution. In particular, if moving domains are considered as e.g. in fluid-structure interaction or free surface flows implemented in an sharp-interface moving mesh approach, the influence on the moving geometry on the discretized equations has to be considered as well. Calculating the required first and second order derivatives of the discretized equations with respect to the mesh coordinates by finite difference is in principle possible, but computationally expensive and usually not sufficiently accurate to ensure convergence of the solution. Similar aspects are relevant in shape optimization problems, for which recently automatic differentiation approaches have been successfully implemented [9,10].

In this article, we present a black-box bifurcation tracking method for arbitrary multi-physics problems on moving meshes on the basis of our finite element framework PYOOMPH,[1] which is based on OOMPH-LIB [6] and GiNaC[11]. For the transient integration of multi-physics problems, our framework has been successfully applied in multiple publications on various topics, e.g. in Refs. [12–16]. For bifurcation tracking presented in this article, we automatically perform symbolical differentiation of the entered system of equations to obtain the monolithic Jacobian, the mass matrix, parameter derivatives thereof, and the Hessian, which are subsequently assembled by generated highly performant C codes. The system of equations is entered by the user in weak formulation, which is subsequently discretized by the finite element method. For all considered bifurcation types, we present a representative example and validate our implementation on the basis of literature results.

Since the method can still be too expensive for full three-dimensional settings, we also present a reduction of the dimensions for stationary solutions that possess a symmetry. In particular, we show how the method can be applied to assess azimuthal symmetry breaking bifurcations of axisymmetric base states at very low computational costs.

While all these steps can be manually coded in frameworks as e.g. OOMPH-LIB [6], the framework presented in this article automatizes the most cumbersome and time-consuming steps. With short PYTHON scripts, complicated problems are made accessible to bifurcation tracking, with moving meshes and/or for the investigation of azimuthal instabilities, yet with a computational efficiency on par with manually coded approaches. This article is mainly on validation and performance comparison of our method. However, we also present novel results of particular physical problems, namely the influence of thermal Marangoni flow of a droplet detaching under gravity and the influence of the height of a centered constriction in a Hele-Shaw cell on the stability of a bubble transported therein, i.e. generalizing the results investigated in Refs. [17–21]. Moreover, a novel bifurcation tracking method for the onset of azimuthal instabilities is presented.

The article is structured as follows: In section 2, the general mathematical notation as well as the used bifurcation tracking systems are introduced. In section 3, the complications for bifurcation tracking on moving meshes are discussed, followed by details on our particular implementation in section 4. After validation of our approach in section 5, the generalization to the azimuthal stability analysis of axisymmetric base states is discussed and validated in section 6. The article ends with a conclusion, including potential applications of this method.

2. Mathematical description

2.1. Notation

In the following, we consider real-valued multi-physics problems which are governed by a set of partial differential equations in space and time t. Numerically, such systems can be solved, along with suitable initial and boundary conditions, by discretizing the spatial domain by a grid or a mesh and performing the temporal integration by a discrete time stepping scheme. Promising numerical approaches to perform such a discretization and the subsequent solution are e.g. the finite element method [22] or the finite volume method [23]. If we only apply the spatial discretization, i.e. keep the temporal dependency symbolically, the spatially degrees of freedom of the spatially discretized fields can be arranged in an N-dimensional vector of time-dependent unknowns, $\mathbf{U}(t)$. Spatial differential operators will lead to a coupling of the temporal dynamics of the entries in $\mathbf{U}(t)$. Thereby, we obtain a potentially nonlinear system of coupled ordinary differential equations, which can be most generally written implicitly, i.e.

$$\mathbf{R}^*(\mathbf{U}, \dot{\mathbf{U}}, p) = \mathbf{0}\,. \tag{1}$$

Here, $\dot{\mathbf{U}}$ denotes the first-order time derivative of $\mathbf{U}$ and the residual vector $\mathbf{R}^*$ is a nonlinear real-valued function with N components, describing the coupled temporal dynamics. For a list of symbols used throughout this article, please refer to the supplementary information. $\mathbf{R}^*$ might depend on further parameters, but it is sufficient to focus on a single parameter p here and keep potential

[1] Available as free software at https://pyoomph.github.io.

further parameters fixed. Higher order time derivatives can be considered by introducing auxiliary components in $\mathbf{U}$ and modifying $\mathbf{R}^*$ accordingly, i.e. the typical conversion of higher order ordinary differential equations to a first order system.

If it exists at the given parameter p, a stationary solution $\mathbf{U}^0$ is obtained by solving the stationary residual

$$\mathbf{R}(\mathbf{U}, p) = \mathbf{R}^*(\mathbf{U}, \mathbf{0}, p) = \mathbf{0}\,, \tag{2}$$

for $\mathbf{U}$. Since $\mathbf{R}$ can be nonlinear, Newton's method provides a good approach, where one iteratively updates a reasonable guess $\mathbf{U}^{\text{guess}}$ for $\mathbf{U}$ by

$$\mathbf{U}^{\text{guess}} \hookleftarrow \mathbf{U}^{\text{guess}} - \mathbf{J}^{-1}(\mathbf{U}^{\text{guess}}, p)\mathbf{R}(\mathbf{U}^{\text{guess}}, p) \tag{3}$$

until the maximum norm $\|\mathbf{R}(\mathbf{U}^{\text{guess}}, p)\|_\infty$ falls below a given threshold. Here, $\mathbf{J}$ is the Jacobian, i.e. the derivatives of the stationary residual $\mathbf{R}$ with respect to $\mathbf{U}$. The direct usage of (3) requires that the Jacobian contains the fully coupled system, i.e. the derivatives of all residual entries with respect to all unknowns, and is therefore called a monolithic strategy. Alternatives are partitioned (segregated) solvers, which solve smaller subsystems by disregarding cross-coupling terms and yet can converge within a Newton iteration loop. A comparison between these approaches can be found in Ref. [24].

Once the stationary solution $\mathbf{U}^0$ has been found, the linear stability can be assessed by determining the temporal evolution of a perturbation $\mathbf{V}$ added to $\mathbf{U}^0$. To that end, the ansatz $\mathbf{U}^0 + \epsilon\mathbf{V}\exp(\lambda t)$ is plugged into $\mathbf{R}^*$, which results for small perturbations ($\epsilon \ll 1$) in the generalized eigenvalue problem:

$$\lambda\, \mathbf{M}|_{(\mathbf{U}^0,p)}\, \mathbf{V} = -\,\mathbf{J}|_{(\mathbf{U}^0,p)}\, \mathbf{V}\,. \tag{4}$$

The Jacobian $\mathbf{J}$ and mass matrix $\mathbf{M}$ and are defined by

$$M_{ij} = \left.\frac{\partial R_i^*}{\partial \dot{U}_j}\right|_{\dot{\mathbf{U}}=\mathbf{0}} \qquad \text{and} \qquad J_{ij} = \frac{\partial R_i}{\partial U_j}\,. \tag{5}$$

The linear stability is determined by solving (4) for the up to N eigenvalues λ with the corresponding eigenvectors $\mathbf{V}$. If there exists at least a single eigenvalue with a positive real part, $\Re(\lambda) > 0$, the stationary solution $\mathbf{U}^0$ is linearly unstable. If all eigenvalues satisfy $\Re(\lambda) < 0$, the stationary solution is linearly stable. When gradually changing the parameter p, one can continue the stationary solution $\mathbf{U}^0$ in the parameter p to obtain a stationary solution branch. Usually, also the eigenvalues vary along the branch, i.e. the linear stability of the solution can change. Points on the branch where this happens are called bifurcations, which can be determined by bifurcation tracking, as discussed later in section 2.2.

A monolithic assembly of $\mathbf{J}$ and $\mathbf{M}$, i.e. including all cross-couplings, is obviously essential for this method. However, in particular when the residuals stem from the spatial discretization of partial differential equations defined on a moving mesh, such a monolithic treatment can become cumbersome, as outlined later in section 3.3. This demanding task is fully automatically handled within our framework.

For completeness, we also introduce the Hessian $\mathbf{H}$, which is required in the bifurcation tracking later on:

$$H_{ijk} = \frac{\partial J_{ij}}{\partial U_k} = \frac{\partial R_i}{\partial U_j \partial U_k}\,. \tag{6}$$

2.2. Bifurcation tracking

Instead of iteratively solving the generalized eigenvalue problem (4) and adjusting the parameter p until $\Re(\lambda) = 0$ holds, e.g. by bisection, bifurcation tracking solves for the critical parameter p_c, the corresponding eigenvector $\mathbf{V}$ and the potentially parameter-dependent stationary solution $\mathbf{U}^0$ simultaneously. This is achieved by the same steps as above, i.e. by solving Newton's method (3), but the unknowns and the residual vector have to be augmented. Depending on the bifurcation type, the particular augmentation must be chosen differently, which is elaborated in the following. The reader is also referred to Refs. [25–29] within this context.

2.2.1. Fold bifurcation

In a fold bifurcation (or saddle-node bifurcation), the stationary solution $\mathbf{U}^0$ ceases to exist at the critical parameter p_c and an unstable and a stable branch of $\mathbf{U}^0$ meet in the bifurcation point. The corresponding eigenvalue has no imaginary part and hence the eigenvalue problem (4) reduces to $\mathbf{JV} = 0$. This equation is solved together with the original residual $\mathbf{R}$. As a further constraint, the trivial solution $\mathbf{V} = 0$ must be prevented, which can be enforced by solving additionally $\mathbf{V}\cdot\mathbf{C} = 1$ for some nontrivial vector $\mathbf{C}$, which can be chosen arbitrarily as long as it is not orthogonal to $\mathbf{V}$, e.g. by using the initial guess of $\mathbf{V}$. As additional unknowns for this $N+1$ additional equations, the eigenvector $\mathbf{V}$ and the bifurcation parameter p are used. Hence, the original system is augmented as follows [30]:

$$\mathbf{U}^{\text{aug}} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \\ p \end{pmatrix} \qquad \mathbf{R}^{\text{aug}} = \begin{pmatrix} \mathbf{R} \\ \mathbf{JV} \\ \mathbf{V}\cdot\mathbf{C} - 1 \end{pmatrix} \qquad \mathbf{J}^{\text{aug}} = \begin{pmatrix} \mathbf{J} & \mathbf{0} & \partial_p\mathbf{R} \\ \mathbf{HV} & \mathbf{J} & (\partial_p\mathbf{J})\mathbf{V} \\ \mathbf{0} & \mathbf{C} & 0 \end{pmatrix}. \tag{7}$$

After solving this system with a good starting guess, e.g. providing a normalized eigenvector solution near the bifurcation from the generalized eigenvalue problem (4), with the Newton method (3), one obtains the critical stationary solution $\mathbf{U}^0$, the corresponding eigenvector $\mathbf{V}$ and the critical parameter p_c, i.e. the location of the fold bifurcation.

2.2.2. Pitchfork bifurcation

Pitchfork bifurcations generically appear in systems with assumed symmetry, where they are located at the intersection of a symmetry-preserving and a symmetry-breaking branch of solutions. At the bifurcation point, the symmetry-preserving branch changes stability and the symmetry-breaking branch emerges. To track a pitchfork bifurcation, the following augmented system must be solved [7]:

$$\mathbf{U}^{\text{aug}} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V} \\ p \\ \epsilon \end{pmatrix} \qquad \mathbf{R}^{\text{aug}} = \begin{pmatrix} \mathbf{R} + \epsilon \mathbf{S} \\ \mathbf{J}\mathbf{V} \\ \mathbf{V} \cdot \mathbf{C} - 1 \\ \mathbf{U} \cdot \mathbf{S} \end{pmatrix} \qquad \mathbf{J}^{\text{aug}} = \begin{pmatrix} \mathbf{J} & \mathbf{0} & \partial_p \mathbf{R} & \mathbf{S} \\ \mathbf{H}\mathbf{V} & \mathbf{J} & (\partial_p \mathbf{J})\mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & 0 & 0 \\ \mathbf{S} & \mathbf{0} & 0 & 0 \end{pmatrix}. \tag{8}$$

The unknown ϵ is a slack variable and $\mathbf{S}$ is an antisymmetric vector with respect to the assumed symmetry. $\mathbf{S}$ and $\mathbf{C}$ can be set to the initial guess for the eigenvector $\mathbf{V}$, which is obtained from an explicit solution of the eigenvalue problem (4) close to the bifurcation. For a solution of this system, the eigenvector $\mathbf{V}$ is hence breaking the symmetry, i.e. is orthogonal to $\mathbf{U}$ and hence the value of ϵ is small. If the mesh does not reflect the symmetry that is broken at the bifurcation, the discrete orthogonality relation $\mathbf{U} \cdot \mathbf{S} = 0$ usually cannot be satisfied. When $\mathbf{S}$ is initialized by the eigenvector close to the pitchfork bifurcation, the system discretized on a nonsymmetric mesh will at most exhibit an imperfect pitchfork bifurcation, i.e. the pitchfork breaks up into a continuous branch and a fold bifurcation [31]. Numerically, demanding $\mathbf{U} \cdot \mathbf{S} = 0$ on systems discretized by such non-conforming meshes therefore usually prevents convergence. However, as we show later in section 5.2, one can achieve good convergence when formulating this symmetry constraint as an integral constraint, i.e. by integrating over the product of finite element interpolations of $\mathbf{U}$ and $\mathbf{S}$ in the continuous domain. The convergence of pitchfork bifurcation tracking can also be improved by using projection operators, are discussed in Refs. [32,26].

2.2.3. Hopf bifurcation

A system undergoes a Hopf bifurcation if, when a parameter crosses a critical threshold p_c, the system becomes unstable by a self-excited oscillation. This bifurcation is associated with a pair of complex conjugated eigenvalues crossing $\Re(\lambda) = 0$ with a non-vanishing imaginary part. The eigenvalues and eigenvectors can therefore be written as $\lambda = \lambda_r \pm i\lambda_i$ and $\mathbf{V} = \mathbf{V}_r \pm i\mathbf{V}_i$, respectively. A Hopf bifurcation hence generally occurs when $\lambda_r = 0$ and $\lambda_i \neq 0$ holds at the critical parameter p_c. The augmented system will have $3N + 2$ unknowns [33], and it can be solved through:

$$\mathbf{U}^{\text{aug}} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V}_r \\ \mathbf{V}_i \\ p \\ \lambda_i \end{pmatrix} \qquad \mathbf{R}^{\text{aug}} = \begin{pmatrix} \mathbf{R} \\ -\mathbf{J}\mathbf{V}_r + \lambda_i \mathbf{M}\mathbf{V}_i \\ -\mathbf{J}\mathbf{V}_i - \lambda_i \mathbf{M}\mathbf{V}_r \\ \mathbf{V}_r \cdot \mathbf{C} - 1 \\ \mathbf{V}_i \cdot \mathbf{C} \end{pmatrix}$$

$$\mathbf{J}^{\text{aug}} = \begin{pmatrix} \mathbf{J} & \mathbf{0} & \mathbf{0} & \partial_p \mathbf{R} & \mathbf{0} \\ -\mathbf{H}\mathbf{V}_r + \lambda_i(\partial_{\mathbf{U}}\mathbf{M})\mathbf{V}_i & \mathbf{J} & \lambda_i \mathbf{M} & -(\partial_p \mathbf{J})\mathbf{V}_r + \lambda_i(\partial_p \mathbf{M})\mathbf{V}_i & \mathbf{M}\mathbf{V}_i \\ -\mathbf{H}\mathbf{V}_i - \lambda_i(\partial_{\mathbf{U}}\mathbf{M})\mathbf{V}_r & \lambda_i \mathbf{M} & \mathbf{J} & -(\partial_p \mathbf{J})\mathbf{V}_i - \lambda_i(\partial_p \mathbf{M})\mathbf{V}_r & -\mathbf{M}\mathbf{V}_r \\ 0 & \mathbf{C} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{C} & 0 & 0 \end{pmatrix}. \tag{9}$$

Once again, the constraint vector $\mathbf{C}$, preventing the trivial eigensolution $\mathbf{V}_r = \mathbf{V}_i = 0$, can be the real part of the initial guess for the eigenvector $\mathbf{V}$. The second constraint, $\mathbf{V}_i \cdot \mathbf{C} = 0$, is required to select a unique phase of the complex-valued eigenvector. This equation is associated with the additional unknown Hopf frequency λ_i. Provided that the initial guess is near the bifurcation, the Newton method (3) is generally able to capture the Hopf bifurcation's critical stationary solution $\mathbf{U}^0$, its critical parameter p_c, the corresponding real and imaginary parts of the eigenvector $\mathbf{V}$ and the corresponding imaginary eigenvalue λ_i.

3. Spatial discretization via the finite element method

In order to use the aforementioned bifurcation tracking approaches on spatially continuous domains, it is necessary to obtain the discretized residual vector including the time dynamics, i.e. $\mathbf{R}^*$, from a spatially continuous residual functional $\mathcal{R}^*$. Here, the finite element method is used for this step, but in principle, any other spatial discretization method can be applied. Particular focus is put on the complications that arise when a moving mesh is considered, i.e. the mesh coordinates are part of the unknowns.

3.1. Finite element method for a static mesh

The finite element method provides a flexible and accessible approach to discretize arbitrary coupled equation in space. In particular, it can be applied to complicated geometries and moving meshes. In order to keep the equations brief, we will discuss it on

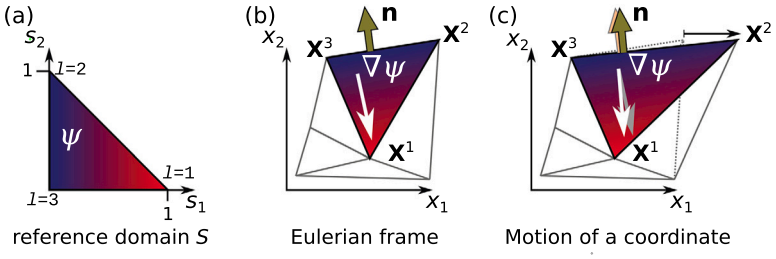

Fig. 1. (a) Spatial integrations are carried out element-wise, where the integrals are evaluated in the reference domain S. The value of the shape function ψ corresponding to the node at $\mathbf{s} = (1, 0)$ is indicated by the color gradient (blue: $\psi = 0$, red: $\psi = 1$). (b) By the Eulerian coordinates $\mathbf{X}^l$ of the element nodes, the integrals and derivatives are mapped into the Eulerian domain. (c) When a mesh node moves, the integral transformation, i.e. the area of the element, and also Eulerian gradients of shape functions $\nabla\psi^l$ and potentially normal vectors $\hat{\mathbf{n}}$ at interfaces change accordingly. This gives plenty of additional contributions in the Jacobian and Hessian. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the basis of a simple Poisson equation, although it does not show any bifurcations. Of course, it is textbook material, but it sets the basis to discuss all nonlinearities arising due to the moving mesh.

The Poisson equation on a domain Ω with boundary $\Gamma = \Gamma_\mathrm{D} \cup \Gamma_\mathrm{N}$ reads

$$\begin{aligned} \nabla^2 u &= h \quad \text{on } \Omega \\ u &= u_\mathrm{D} \quad \text{on } \Gamma_\mathrm{D} \\ -\hat{\mathbf{n}} \cdot \nabla u &= j \quad \text{on } \Gamma_\mathrm{N}, \end{aligned} \tag{10}$$

i.e. with source term h, and Dirichlet and Neumann boundary conditions u_D and j, respectively. The generic weak formulation reads

$$\int_\Omega (\nabla u \cdot \nabla \chi + h\chi)\,\mathrm{d}\Omega + \int_{\Gamma_\mathrm{N}} j\chi\,\mathrm{d}\Gamma = 0 \quad \text{with} \quad u = u_\mathrm{D} \quad \text{on} \quad \Gamma_D\,, \tag{11}$$

which has to hold for all appropriate choices of the test function χ that fulfill $\chi|_{\Gamma_\mathrm{D}} = 0$.

The central idea of the finite element method is the choice of a discrete amount of basis functions for the test function χ and simultaneously also expand the unknown function u in such a finite amount of basis functions. For the Galerkin approach, both sets of basis functions are the same, i.e. we introduce the shape functions ψ^l for discrete values of u, i.e. for $l = 1, \ldots, N_u^\mathrm{dof} + N_u^\mathrm{D}$. Here, the first N_u^dof unknown degrees of freedom are located in the bulk of Ω, whereas the remaining N_u^D are known values on the Dirichlet boundary Γ_D. Next, u is expanded as $u \approx u^l \psi^l$, into discrete amplitudes u^l. Throughout this article, we use summation convention, i.e. here summation over $l = 1, \ldots, N_u^\mathrm{dof} + N_u^\mathrm{D}$. On Γ_D, the corresponding value u^l is known from the Dirichlet condition u_D and it is not considered as degree of freedom. Subsequently, one sets $\chi = \psi^k$ for $k = 1, \ldots, N_u^\mathrm{dof}$, i.e. only for the degrees of freedom, not for the Dirichlet values. Thereby, one obtains the discrete residual vector $\mathbf{R}$ via

$$R_k = \int_\Omega \left(u^l \nabla\psi^l \cdot \nabla\psi^k + h\psi^k\right) \mathrm{d}\Omega + \int_{\Gamma_\mathrm{N}} j\psi^k\,\mathrm{d}\Gamma \tag{12}$$

On a static mesh, it is trivial to obtain the Jacobian by differentiation with respect to u_l

$$J_{kl} = \int_\Omega \nabla\psi^l \cdot \nabla\psi^k \mathrm{d}\Omega \tag{13}$$

and the Hessian $\mathbf{H}$ vanishes here due to the linearity of the Poisson equation.

3.2. Calculating the spatial integrals

To calculate the spatial integrals occurring e.g. in (12) and (13), one usually splits the integrals in a sum over all elements, separated in the discretized bulk domain Ω^h and the discretized Neumann boundary Γ_N^h,

$$R_k = \sum_{\mathcal{T}\in\Omega^h} R^\mathcal{T}_{\Omega^h,k} + \sum_{\mathcal{I}\in\Gamma_\mathrm{N}^h} R^\mathcal{I}_{\Gamma^h,k} = \sum_{\mathcal{T}\in\Omega^h} \int_\mathcal{T} \left(u^l \nabla\psi^l \cdot \nabla\psi^k + h\psi^k\right) \mathrm{d}\Omega + \sum_{\mathcal{I}\in\Gamma_\mathrm{N}^h} \int_\mathcal{I} j\psi^k\,\mathrm{d}\Gamma \tag{14}$$

and likewise for the Jacobian J_{kl} and potentially the Hessian H_{klm}. The occurring integrals over the elements only have support in terms of the shape functions ψ^k and ψ^l if the corresponding degrees of freedom are parts of the current element $\mathcal{T}$ (or $\mathcal{I}$ on the Neumann boundary Γ_N). This allows the calculation of the elemental integrals by iterating the index k (and l and potentially m for the Jacobian and Hessian, respectively) over the degrees of freedom associated with each element only.

The integration within each element is carried out in a fixed reference domain S, i.e. the local coordinate system $\mathbf{s}$ of the element (see Fig. 1). For the bulk contribution $R^\mathcal{T}_{\Omega^h,k}$, the integration is hence calculated by

$$R^{\mathcal{T}}_{\Omega^h,k} = \int_S \left(u^l \nabla\psi^l \cdot \nabla\psi^k + h\psi^k\right) \sqrt{\det \mathbf{g}}\, \mathrm{d}^{n_\mathrm{e}} s\,. \tag{15}$$

Here, n_e is the elemental dimension and the shape functions ψ are directly evaluated in the reference domain S. The Eulerian derivatives $\nabla\psi$, however, must be transformed accordingly. For that, the covariant metric tensor $\mathbf{g}$ ($n_\mathrm{e} \times n_\mathrm{e}$) of the coordinate transformation to the Eulerian coordinate $\mathbf{x} = \mathbf{x}(\mathbf{s})$ with the entries

$$g_{\alpha\beta} = \mathbf{t}_\alpha \cdot \mathbf{t}_\beta \tag{16}$$

is used. The covariant base vectors $\mathbf{t}_\alpha$ can be obtained by knowing that the Eulerian position $\mathbf{x}$ can also be expanded into shape functions $\bar{\psi}^{\bar{I}}$, i.e. by

$$\mathbf{x} = \mathbf{X}^{\bar{I}} \bar{\psi}^{\bar{I}}(\mathbf{s}) \tag{17}$$

with the Eulerian coordinates $\mathbf{X}^{\bar{I}}$ of the nodes and the shape functions of the position space $\bar{\psi}^{\bar{I}}$. In general, the shape functions $\bar{\psi}^{\bar{I}}$ of the Eulerian position $\mathbf{x}$ can be different from the shape functions ψ^I of u, e.g. second order Lagrange basis functions for the position $\mathbf{x}$ and first order for u. In particular, also the range of $\bar{I}$ and I can be different. We use the convention that the shape functions $\bar{\psi}$ of the Eulerian position space as the corresponding indices $\bar{I}$ are augmented with an overline, whereas the shape functions ψ of other unknown fields (here u) and their corresponding indices I are written without an overline. The covariant basis vectors can then be calculated from the nodal positions:

$$\mathbf{t}_\alpha = \frac{\partial \mathbf{x}}{\partial s^\alpha} = \mathbf{X}^{\bar{I}} \frac{\partial \bar{\psi}^{\bar{I}}}{\partial s^\alpha}\,. \tag{18}$$

Note that the dimension n_dim of the Eulerian coordinate vector $\mathbf{x}$ can be higher than the element dimension n_e, e.g. on the interface elements $\mathcal{I}$.

The Eulerian derivatives of the shape functions of u, i.e. $\nabla\psi$, in (15) are transformed according to

$$\partial_{x_i} \psi^I = \left(\nabla\psi^I\right)_i = g^{\alpha\beta} t_{\alpha,i} \frac{\partial \psi^I}{\partial s^\beta}\,, \tag{19}$$

where $g^{\alpha\beta}$ is the contravariant metric tensor. On interface elements, this convention automatically yields the appropriate surface gradient operator.

For interface elements $\mathcal{I}$, also the facet normal $\hat{\mathbf{n}}$ is calculated from the covariant base vectors (18), where in some cases, the derivatives must actually be performed with respect to all nodal coordinates of the attached bulk element, e.g. for a one-dimensional interface element ($n_\mathrm{e} = 1$) attached to a two-dimensional bulk element ($n_\mathrm{e} = 2$) embedded in a three-dimensional Eulerian space ($n_\mathrm{dim} = 3$).

Eventually, the integration is carried out by a Gauss-Legendre quadrature on the reference domain S including all these transformations.

3.3. Additional nonlinearities stemming from a moving mesh

If a moving mesh is considered, the Eulerian positions $X_i^{\bar{I}}$ of the mesh nodes are parts of the unknown vector $\mathbf{U}$. For the Poisson equation (10) defined on such a moving mesh, the vector of unknowns $\mathbf{U}$, the residual vector $\mathbf{R}$ and the Jacobian $\mathbf{J}$ can hence be split into components of u and the moving mesh coordinates $\mathbf{X}$:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_u \\ \mathbf{U}_\mathbf{X} \end{pmatrix} \qquad \mathbf{R} = \begin{pmatrix} \mathbf{R}_u \\ \mathbf{R}_\mathbf{X} \end{pmatrix} \qquad \mathbf{J} = \begin{pmatrix} \mathbf{J}_{uu} & \mathbf{J}_{u\mathbf{X}} \\ \mathbf{J}_{\mathbf{X}u} & \mathbf{J}_{\mathbf{XX}} \end{pmatrix} \tag{20}$$

The residual for the mesh motion is usually calculated on a fixed reference mesh, i.e. one describes the motion with respect to some Lagrangian coordinates, e.g. the initial mesh positions. Furthermore, we assume here for simplicity that the mesh motion is independent of u, i.e. $\mathbf{J}_{\mathbf{X}u} = \mathbf{0}$. Any Eulerian integral or derivative contribution to the mesh position residuals $\mathbf{R}_\mathbf{X}$ or any feedback of u on $\mathbf{X}$, i.e. $\mathbf{J}_{\mathbf{X}u} \neq \mathbf{0}$, must be treated equally as discussed in the following on the basis of the Jacobian block $\mathbf{J}_{u\mathbf{X}}$.

While the Jacobian block $\mathbf{J}_{uu}$ coincides with the one of a static mesh, plenty of additional terms arise in the block $\mathbf{J}_{u\mathbf{X}}$. The derivative of the k-th elemental residual contribution (15) of u from the bulk with respect to the coordinate position $X_j^{\bar{I}}$ reads

$$\begin{aligned} J^{\mathcal{T}}_{u^k X_j^{\bar{I}}} = \int_S \Big[& u^l \left(\partial_{X_j^{\bar{I}}} \left(\nabla\psi^l\right) \cdot \nabla\psi^k + \nabla\psi^l \cdot \partial_{X_j^{\bar{I}}} \left(\nabla\psi^k\right) \right) \sqrt{\det \mathbf{g}} \\ & + \left(u^l \nabla\psi^l \cdot \nabla\psi^k + h\psi^k\right) \partial_{X_j^{\bar{I}}} \left(\sqrt{\det \mathbf{g}}\right) \Big] \mathrm{d}^{n_\mathrm{e}} s \end{aligned} \tag{21}$$

To solve bifurcation tracking problems via Newton's method, also the Hessian is required. Entries from the Hessian block $\mathbf{H}_{u\mathbf{X}u}$ can simply be calculated from (21) by deriving with respect to u^l. The elemental Hessian contribution from the bulk $\mathbf{H}_{u\mathbf{XX}}$ is obtained by deriving (21) again with respect to a potentially other nodal coordinate $X_{j'}^{\bar{I}'}$. This gives an even longer expression than (21), involving second order derivatives of the shape function gradients and the functional determinant $\sqrt{\det \mathbf{g}}$. On interface elements $\mathcal{I}$,

the considered equation potentially also includes the normal vector $\hat{\mathbf{n}}$, which first and second derivative with respect to the nodal positions is also required. In our implementation, we precalculate some quantities to efficiently calculate all these highly nonlinear derivatives up to the second order. These exact relations can be found in the supplementary information.

The example of a simple Poisson equation solved on a moving mesh illustrates the complexity that arises in the Jacobian and even more in the Hessian. Therefore, for more complicated problems, a sophisticated implementation to accurately calculate these derivatives automatically is required. This is discussed in the next section.

4. Implementation

All the complications discussed in the previous section can be circumvented if derivatives with respect to the moving mesh coordinates are evaluated numerically by finite differences, i.e. the entry $J_{u^k X_j^{\bar{l}}}$ of the Jacobian block $\mathbf{J}_{u\mathbf{X}}$ is just calculated by

$$J_{u^k X_j^{\bar{l}}} = \frac{R_{u^k}(\mathbf{U}_u, \mathbf{U}_\mathbf{X} + \epsilon \mathbf{e}_j^{\bar{l}}) - R_{u^k}(\mathbf{U}_u, \mathbf{U}_\mathbf{X})}{\epsilon} \tag{22}$$

for some small value of ϵ which perturbs the nodal coordinate $X_j^{\bar{l}}$ only, i.e. $\mathbf{e}_j^{\bar{l}}$ is zero everywhere except at the equation index of $X_j^{\bar{l}}$, where it is unity. For problems without bifurcation tracking, i.e. without the requirement of the Hessian, it works reasonably well. When the Hessian is required for bifurcation tracking problems, our numerical experiments have proven that finite differences of second order do not provide sufficient numerical accuracy (cf. Table 1 later). Newton's method for the augmented bifurcation tracking system usually does not converge, and if it does, it converges poorly. Also, the assembly of the augmented Jacobian is typically computationally expensive when using finite differences.

One can of course improve the convergence of finite difference methods by using symmetric finite differences instead of the forward-finite difference formula (22). This approach, however, increases the computational costs by the additional evaluation of the (augmented) residual vector at the perturbation by $-\epsilon$. The optimal value of the finite-difference step ϵ is usually problem-specific, depends on the mesh resolution and may be different for first or second order derivatives. However, the authors of Ref. [19] were able to extract even third-order derivatives based on such centred finite differences, provided that the Jacobian is filled symbolically.

A promising way to overcome the lack of accuracy of finite differences is automatic differentiation, but this requires the augmentation of all mathematical operations to account for dual numbers [34]. While even third or higher order derivatives can be calculated quite efficiently and accurately up to machine precision by automatic differentiation, it can be complicated to apply this on the transformation from the local coordinate $\mathbf{s}$ to the Eulerian coordinate $\mathbf{x}$, which is rather hard-coded in many existing finite element frameworks. Furthermore, it is not trivial to reuse already calculated subexpressions that appear multiple times in the full Jacobian or Hessian during automatic differentiation. In general, the overhead due to automatic differentiation can increase the computational costs for assembly by a factor of up to two compared to hand-coded routines filling the Jacobian and/or Hessian [8].

In our implementation, we therefore opted for symbolical differentiation, i.e. indeed performing the steps discussed in the previous section, but entirely assisted and automatized by symbolic computer algebra. Only the residual must be defined in weak formulation and, subsequently, all symbolical derivatives up to second order, also with respect to the mesh coordinates, can be calculated. For performance reasons, all of these expressions are written in efficient C code which is automatically compiled and linked back into the running program. For the symbolic differentiation and the code generation, we use the efficient, accurate and flexible open-source framework GiNaC [11]. During the compilation, common subexpression elimination can speed up the process. Furthermore, particular quantities appearing multiple times in the residual can be explicitly marked by the user to be calculated and derived in beforehand, just by wrapping the particular symbolical expression in a `subexpression` call (cf. supplementary material). For complicated expressions, this can lead to a considerable performance gain for the assembly.

The performance of the generated code is on par with hand-coded implementations and, typically, for multi-physics problems, even run up to twice as fast as the handwritten implementations of oomph-lib, as discussed in the supplementary material. Our approach of treating the entered equations symbolically, i.e. not by pure automatic differentiation, also has benefits in automatically deriving the corresponding forms for azimuthal symmetry breaking, as discussed later in section 6.

As finite element framework, oomph-lib is employed [6,29]. It already offers several methods for bifurcation tracking and arclength continuation, monolithically treated moving meshes and support for multi-domain and multi-physics problems including spatial adaptivity, etc. The default way of calculating the Jacobian and Hessian, however, is by finite differences, unless the user explicitly codes the corresponding symbolical expressions by hand. Our automatic code generation fills exactly this gap, which is otherwise very cumbersome. oomph-lib also allows to access the low-level of the finite element method, i.e. the transformation from the local coordinate $\mathbf{s}$ to the Eulerian coordinate $\mathbf{x}$. Therefore, it provides the ideal framework for our purposes here.

Inspired by the framework FEniCS [35], we wrapped the core of oomph-lib into python, so that equations, problems and meshes can easily be assembled in python, but still use the full computational efficiency of the compiled oomph-lib core and the generated C code for the residual vector, the mass and Jacobian matrices, the Hessian and parameter derivatives of the former three. Arbitrary multi-physics problems can be formulated with a few lines of python code, where the entered residual form is directly converted to a GiNaC expression tree within our C++ core. Rather arbitrary combinations of finite element spaces (including discontinuous Galerkin spaces) for scalar, vectorial and tensorial quantities can be used. Integral constraints, (fields of) Lagrange multipliers, potentially only defined at interfaces, and error estimators for mesh adaptivity can be incorporated directly. At shared interfaces between two multi-physics domains, the interface and bulk fields and gradients thereof can be accessed on both sides. Once the residual of the system is formulated that way, the monolithic Jacobian and Hessian can be quickly assembled on the basis of the generated C code. A more

Table 1
Averaged assembly times for one Newton step (in seconds, on a single Intel i7-4790K thread/core) and number of Newton iterations (in brackets) for different methods the different bifurcation tracking examples discussed in the following. *dnc* means *does not converge*, i.e. Newton's method fails. (a) all derivatives symbolically calculated. (b) Hessian $\mathbf{H}$ by finite differences from the symbolical Jacobian, all other derivatives symbolical. (c) Like (b), but all derivatives (also first order) with respect to the mesh coordinate by finite differences. (d) All derivatives by finite differences. Some configurations using finite differences are not available (*N/A*), since the required filling via finite differences is not implemented (yet). For the different pitchfork cases, cf. section 5.2.

	(a)	(b)	(c)	(d)
fold (with Stokes flow)	0.85 (6)	2.73 (6)	4.63 (dnc)	7.24 (dnc)
fold (with Young-Laplace equation)	0.77 (3)	1.31 (4)	1.64 (dnc)	1.53 (dnc)
pitchfork (symmetric mesh)	8.32 (2)	9.43 (2)	18.98 (dnc)	23.37 (dnc)
pitchfork (nonsymm. mesh, dofwise symmetry)	8.38 (dnc)	9.19 (dnc)	18.68 (dnc)	22.66 (dnc)
pitchfork (nonsymm. mesh, weak symmetry)	10.08 (3)	N/A	N/A	N/A
Hopf	8.72 (4)	9.48 (4)	N/A	N/A

detailed overview of our framework and some example codes of our validation cases are discussed in the supplementary information to illustrate the straightforward formulation of complicated problems.

5. Validation

To validate our implementation, we compare all types of bifurcation trackers with problems discussed in the literature in the following. Often, we use pseudo-arclength continuation in another parameter to obtain entire bifurcation diagrams, which usually only takes a few minutes with our framework.

Before the discussion of the individual validation cases, however, the reader is referred to Table 1. For all cases, we compared the assembly time of the augmented Jacobian $\mathbf{J}^{\mathrm{aug}}$ and the convergence of the Newton method for different assembly methods. Our symbolical method is always the fastest in assembly time and ensures good convergence for all considered cases. When calculating the second order derivatives in the Hessian by finite differences, but all first order derivatives symbolically, the convergence is mainly the same, but the assembly is slower. If any of the first order derivatives are calculated by finite differences, we could not achieve reliable convergence in any case. These results highlight the necessity of accurate derivatives in bifurcation tracking and the strengths of our symbolical code generation.

5.1. Fold bifurcation

To validate the fold bifurcation tracking on moving meshes, we consider a droplet hanging on the bottom of a vertical plate. Gravity will pull the droplet down and deform the droplet to deviate from a spherical cap shape according to the Young-Laplace equation, but at some critical volume V or sufficiently strong gravitational force compared to the capillary force, the droplet will detach, i.e. the stationary hanging droplet solution will vanish in a fold bifurcation. While this problem can easily be solved by applying e.g. a shooting method on a simple ordinary differential equation [36], we implement it via bifurcation tracking instead. This allows for straightforward generalizations later on, e.g. investigating the influence of thermal Marangoni flow.

In nondimensionalized quantities, we set the volume to unity, and use the Bond number Bo as bifurcation parameter. The Bond number can be calculated from the dimensional quantities as $\mathrm{Bo} = \rho g V^{2/3}/\sigma$, with the mass density ρ, the gravitational acceleration g and the surface tension σ. As additional parameter, the nondimensionalized contact line radius r_{cl}, which is assumed to be fixed (pinned contact line) is introduced. Since the hanging solution corresponds to vanishing velocity, it is sufficient to solve the Stokes equations for the flow. The onset of detachment is also independent of the viscosity of the droplet, so that we use a coefficient of unity for the viscous term in the nondimensional equations. These assumptions obviously only hold for the bifurcation, not for the full detachment and pinch-off process, where inertia and viscous forces definitely play an important role [37].

In total, we hence solve the following system in axisymmetric coordinates:

$$\nabla \cdot \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right] = \nabla p - \mathrm{Bo}\,\mathbf{e}_z \tag{23}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{24}$$

$$\left[\dot{\mathbf{X}} - \mathbf{u}\right] \cdot \hat{\mathbf{n}} = 0 \tag{25}$$

$$\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right] \cdot \hat{\mathbf{n}} - p\hat{\mathbf{n}} = \kappa \hat{\mathbf{n}} \tag{26}$$

$$\int_{\mathrm{drop}} 2\pi r \mathrm{d}r\mathrm{d}z = 1 \tag{27}$$

Here, $\kappa = -\nabla_S \cdot \hat{\mathbf{n}}$ is the curvature and the volume constraint (27) is enforced by the constant of the pressure nullspace with respect to an additive constant. The only time derivative appears is the motion of the mesh coordinate with the fluid velocity in normal direction, i.e. in the kinematic boundary condition (25). The mesh dynamics can be chosen arbitrarily, as long as the mesh is following the physics of the fluid and does not influence the dynamics in a nonphysical way, i.e. despite entering via the curvature κ in the dynamic

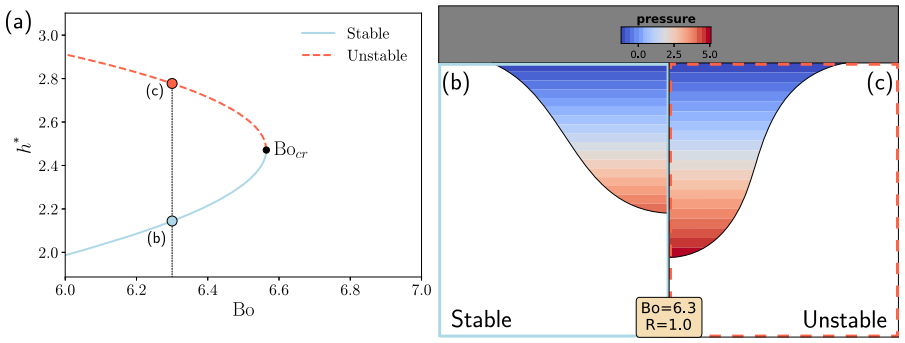

Fig. 2. Fold bifurcation for a hanging droplet, (a), with stable and unstable parts of the solution branch. For $\mathrm{Bo} = 6.3$, the droplet shapes and pressure profiles are represented for the stable (b) and unstable (c) parts of the solution branch.

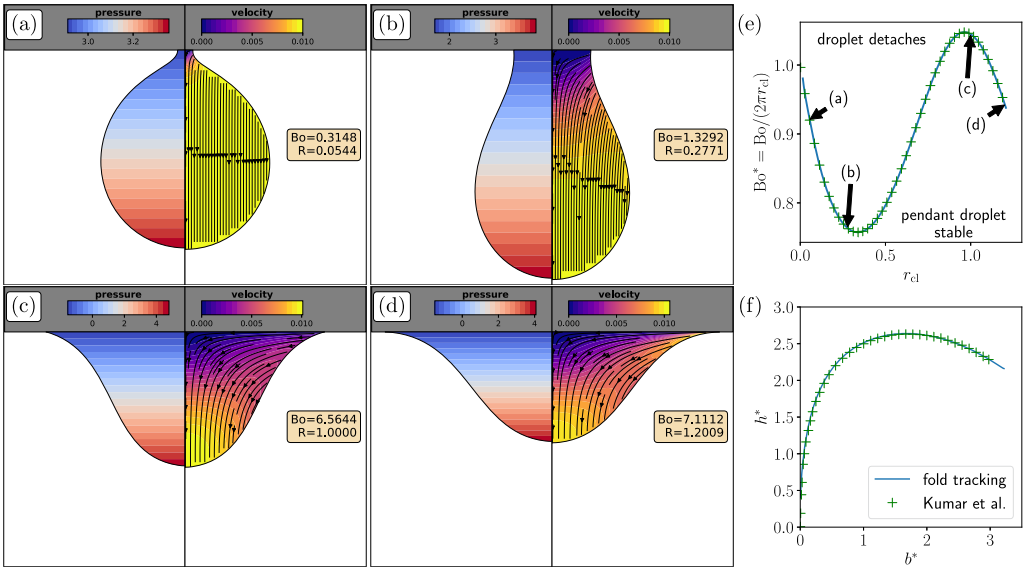

Fig. 3. Validation of the fold bifurcation tracking based on a hanging droplet detaching due to gravity. (a-d) snapshots of solutions on the bifurcation curve with critical solution (pressure, left) and critical eigendynamics (velocity and displacement, right) (e) Bifurcation curve expressed in Bo^* as function of the nondimensional contact radius r_{cl}. The markers stem for the fold bifurcation tracking of the Young-Laplace equation. (f) Validation with literature results extracted from Kumar et al. [38,39], where $b^* = r_{cl}\sqrt{\mathrm{Bo}}$ and $h^* = h\sqrt{\mathrm{Bo}}$.

boundary condition. Here, we just chose a Laplace-smoothed mesh, but comparison with a mesh that deforms like an elastic body instead gave virtually the same results, i.e. identical bifurcation locations up to at least the fifth significant digit. The weak formulation of this problem is available in the supplementary information. Also, the simple and convenient way to express this system within our code framework is illustrated within the supplementary information.

Fig. 2 shows representative solutions of the system. Up to a critical Bond number, two stationary solutions coexist, which vanish in a fold bifurcation when increasing Bo towards the critical value Bo_{cr}. Upon small perturbations, the unstable solution depicted in Fig. 2(c) will either retract to the stable solution (Fig. 2(b)) or detach. Beyond Bo_{cr}, the solution of a hanging droplet ceases to exist.

In Fig. 3, the results of the fold bifurcation tracking are shown. The left sides of Fig. 3(a-d) show some representative solutions at the critical Bond number at the given nondimensional contact line radius r_{cl}. The dynamics of the instability, extracted from the critical eigenfunction, are shown on the right half of each plot. While droplets with small r_{cl} just start to fall down as spherical object, flow at near the contact line becomes more important for high values of r_{cl}.

Fig. 3(e) shows a rescaled critical Bond number $\mathrm{Bo}^* = \mathrm{Bo}/(2\pi r_{cl})$. This definition gives the ratio of gravitational force by the droplet mass to the capillary force acting on the circumference of the contact line. In the limit $r_{cl} \to 0$, it converges to unity as expected, i.e. the gravity by the entire droplet mass must be balanced by the capillary force at the contact line of the small connection with the wall. In total, however, it shows some nontrivial curve, which was determined by the fold bifurcation tracking combined with continuation in r_{cl} in a few minutes. Occasionally, the mesh had to be reconstructed during this scan to prevent large mesh deformations. By interpolating the current critical solution and eigenfunction from the old mesh to the newly constructed mesh, the continuation can carry on after each mesh reconstruction. Whenever the mesh has to be reconstructed, we solve again at the current parameter set after interpolation, which usually takes three Newton iterations or less for the bifurcation tracker to converge. The

computational overhead of mesh reconstruction and subsequent interpolation of all fields is only minor, e.g. around ~ 0.2 and ~ 1.3 seconds for $\sim 10\,000$ and $\sim 150\,000$ degrees of freedom, respectively.

At $r_{\mathrm{cl}} \approx 1.207$, the contact angle becomes zero before the critical Bond number can be reached. Here, the method cannot be continued due to the collapse of the element directly at the contact line. It is hence questionable whether a droplet with a pinned contact line $r_{\mathrm{cl}} \gtrsim 1.207$ can detach at all. In reality, the contact line will depin or form a thin precursor film, which is not accounted for in this simple model.

Since the dynamics is entirely given by the interface, the bulk of the droplet is actually not required to obtain the critical curve. We also solved the Young-Laplace equation directly on a line mesh embedded in a two-dimensional axisymmetric coordinate system. The equation reads

$$-\nabla_S \cdot \hat{\mathbf{n}} + \mathrm{Bo}\, z = p_{\mathrm{ref}} \tag{28}$$

$$\frac{2\pi}{3} \int \hat{\mathbf{n}} \cdot \mathbf{x}\, r \mathrm{d}l + 1 = 0 \tag{29}$$

The second equation enforces a unit volume via the reference pressure p_{ref}. The Young-Laplace equation (28) is solved on moving mesh coordinates in normal direction, whereas for the tangential degrees of freedom, equidistant positioning of the nodes along the arclength is enforced. The radial mesh coordinate at the axis is fixed to zero and the axial mesh coordinate is fixed to zero at the contact line, where also the radial position is prescribed to be r_{cl}. The same fold bifurcation algorithm as before applied on this equation gives the markers in Fig. 3(e), i.e. perfect agreement, but considerably faster due to the lack of bulk equations. However, the method including the bulk domain is flexible to be extended, e.g. to include the effect of thermal Marangoni convection, when a nonuniform temperature profile is induced in the droplet by heating of the top wall. Then, continuation can be performed in the thermal Marangoni number, which will be covered in the next section.

Finally, to validate our results, the critical droplet shape was expressed in new parameters, namely $b^* = r_{\mathrm{cl}}\sqrt{\mathrm{Bo}}$ and $h^* = h\sqrt{\mathrm{Bo}}$ (with the nondimensional height of the droplet h). The comparison with the data extracted from Refs. [38,39] also gives nice agreement, as shown in Fig. 3(f).

5.1.1. Marangoni effects on droplet detachment

The preceding section's analysis can be expanded to encompass a multi-physics framework through the incorporation of thermal Marangoni flow phenomena acting at the interface separating liquid and gas phases. To achieve this, we adopt a physical model akin to that outlined in Ref. [40]. Specifically, a 1 cSt silicone oil pendant droplet of fixed base radius $R = 2.5$ mm (with mass density $\rho^{\mathrm{liq}} = 818\,\mathrm{kg/m^3}$, dynamic viscosity $\mu^{\mathrm{liq}} = 8.18 \times 10^{-4}\,\mathrm{kg/m \cdot s}$, specific heat capacity $c_p^{\mathrm{liq}} = 2000\,\mathrm{J/kg \cdot K}$ and thermal conductivity $k^{\mathrm{liq}} = 0.1\,\mathrm{W/m \cdot K}$), is attached to an isothermal wall at $T_{\mathrm{top}} = 25\,^\circ\mathrm{C}$. The droplet is encircled by air (with $\rho^{\mathrm{liq}} = 1.293\,\mathrm{kg/m^3}$, $\mu^{\mathrm{liq}} = 1.81 \times 10^{-5}\,\mathrm{kg/m \cdot s}$, $c_p^{\mathrm{liq}} = 1005\,\mathrm{J/kg \cdot K}$, $k^{\mathrm{liq}} = 2.587 \times 10^{-2}\,\mathrm{W/m \cdot K}$), which in turn resides within a square domain of side length $L = 3R$. The effects of temperature on the aforementioned properties are disregarded. The surface tension between the phases, however, is considered to change linearly with temperature T, i.e. by $\sigma = (16.87 - 0.0755(T - T_{\mathrm{top}})/1\,\mathrm{K})\mathrm{mN\,m^{-1}}$.

While R remains fixed, the droplet's volume V and the temperature gradient ($\nabla T = (T_{\mathrm{top}} - T_{\mathrm{bottom}})/L$) serve as control parameters in understanding the impact of thermal Marangoni flow on the pendant droplet's pinch-off dynamics. To this end, we employ the Navier-Stokes equations, (30) and (31), supplemented by heat equations (32) in both phases and a kinematic and dynamic boundary conditions, (33) and (34), at the interface:

$$\rho(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot \left[\mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}})\right] + \rho g \mathbf{e}_z \tag{30}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{31}$$

$$\rho c_p(\partial_t T + \mathbf{u} \cdot \nabla T) = k \nabla^2 T, \tag{32}$$

$$[\dot{\mathbf{X}} - \mathbf{u}] \cdot \hat{\mathbf{n}} = 0 \tag{33}$$

$$\left[\!\left[\mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right) - p\mathbf{1}\right]\!\right]_- \cdot \hat{\mathbf{n}} = \kappa\sigma(T)\hat{\mathbf{n}} + \nabla_s \sigma(T). \tag{34}$$

A dimensional variant of the volume constraint (27) is also imposed here. The non-dimensionalization of the equations is automatically handled in the back-end of the code, as detailed in the supplementary information. Fig. 4(a) shows a diagram of the fold bifurcation that dictates the pinch-off as a function of V and ∇T, obtained via bifurcation tracking on V and pseudo-arclength continuation on ∇T. Figs. 4(b) and 4(c) illustrate the contours of temperature (left) and velocity (right) for $\nabla T = -4$ K and $\nabla T = 4$ K, respectively. Remarkably, the critical volume V_{cr} for pinch-off to occur remains nearly unaltered when considering Marangoni flow. Surface tension becomes unable to withhold the mass of the droplet for larger volumes and the existence of flow has a little significance for detachment phenomena. One could expect for the small changes in V_{cr} to be completely due to the modified surface tension induced by temperature changes, which results in a change of the effective $\mathrm{Bo_{eff}} = \rho g V^{2/3}/\sigma_{\mathrm{avg}}$, where σ_{avg} is the surface tension averaged along the interface. However, the non-symmetric slope of the fold bifurcation in the vicinity of $\nabla T = 0$ clearly shows that V_{cr} is altered not only due to the changes in $\mathrm{Bo_{eff}}$. Instead, additional effects, e.g. the shape deformation due to Marangoni flow (Marangoni contraction [41]) can additionally influence the stability.

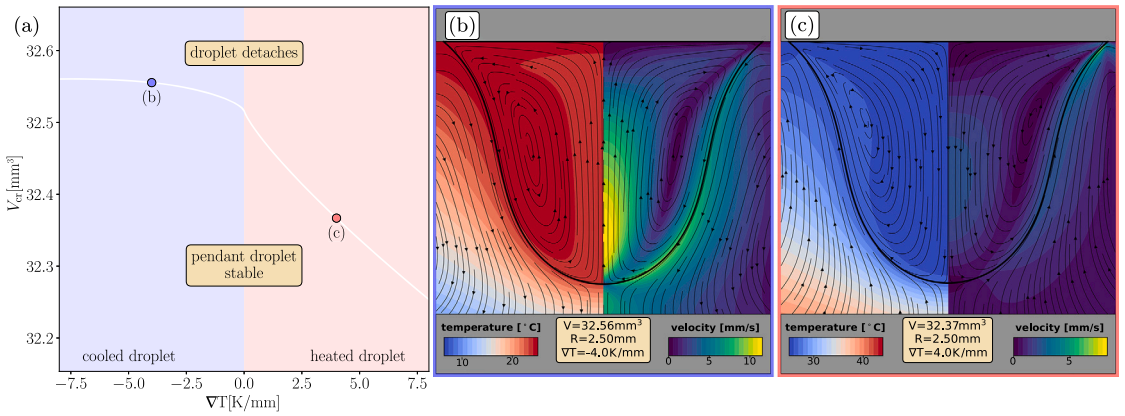

Fig. 4. Critical volume V_{cr} for pinch-off of a droplet as a function of the temperature difference ∇T for a fixed base radius $R = 2.5$ mm, considering both cooling and heating of the droplet (a). Contours of temperature (left) and velocity (right) for heated $\nabla T = -4$ K (b) and cooled $\nabla T = 4$ K (c) droplet.

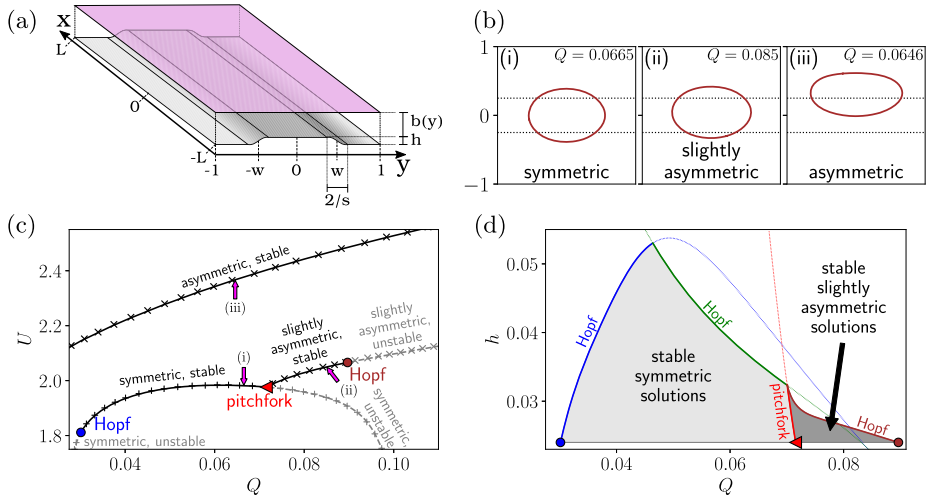

Fig. 5. Validation of the pitchfork and Hopf bifurcation tracking based on a bubble in a Hele-Shaw cell with a centered constriction. (a) Sketch of the problem with the relevant geometric parameters. (b) Representative types of stationary solutions of the bubble. (c) Validation of our results (lines) with the results extracted from Keeler et al. [19] (points) for $h = 0.024$. Arrows indicate the solutions depicted in (b). (d) Bifurcation tracking in the height of the bump h.

5.2. Pitchfork and Hopf bifurcation

For the pitchfork and Hopf bifurcation on moving meshes, an ideal validation case can be found in the articles of Thompson et al. on the solutions of a bubble propagating through a Hele-Shaw cell with a transversal depth perturbation [17–21]. As depicted in Fig. 5(a), a narrow Hele-Shaw cell that has a constriction in the center is considered. When bubbles are advected through this channel in x-direction, they can either move to the left or right side (y-direction), where the channel is higher. However, in some parameter ranges, stable symmetric solutions are also possible, where the bubble propagates along the center line ($y = 0$) of the bump. As shown by Keeler et al. [19], this symmetric centered state can lose the stability in either Hopf or pitchfork bifurcations to an asymmetric state.

Due to the shallow cell, it is sufficient to consider lubrication theory (Darcy's law) to describe the problem, i.e. in the liquid surrounding the bubble, a Laplace equation for the pressure field $p(x, y)$ is solved. Due to the varying nondimensional local height of the channel $b(y)$, the equation actually reads

$$\nabla \cdot \left(b^3 \nabla p\right) = 0\,, \tag{35}$$

and the nondimensional velocity field follows to be $\mathbf{u} = -U\mathbf{e}_x - b^2\nabla p$. Here, U is the bubble velocity in lab frame, i.e. the velocity $\mathbf{u}$ is expressed in the frame co-moving with the bubble. The bubble velocity U is part of the unknowns and follows from the constraint that enforces the averaged x-coordinate of the bubble this co-moving frame to zero. Likewise, the pressure inside bubble p_B is assumed to be homogeneous, i.e. inviscid and inertia-free gas. This pressure p_B corresponds to the volume constraint

$$\int_{\text{bubble}} b\,\mathrm{d}x\mathrm{d}y = V \tag{36}$$

for some given volume V and enters the dynamic boundary condition

$$p_\mathrm{B} - p = \frac{1}{3\alpha Q}\left(\frac{1}{b} + \frac{\kappa}{\alpha}\right), \tag{37}$$

where Q is a non-dimensional flow rate or capillary number with respect to mean through flow velocity of the channel. The aspect ratio, i.e. the channel width in y-direction divided by the plate distance, is denoted by α, whereas κ denotes the x-y-projected curvature of the bubble and $1/b$ accounts for the second curvature [19]. As exterior boundary conditions, $\partial_y p|_{y=\pm 1} = 0$, $\partial_x p|_{x=-L} = -1$ and $p|_{x=L} = 0$ are imposed. The nondimensional constricted height is given by a smoothed double step with a bump height h, sharpness s and half-width w, i.e.

$$b(y) = 1 - \frac{h}{2}\left[\tanh\left(s(y+w)\right) - \tanh\left(s(y-w)\right)\right]. \tag{38}$$

Lastly, the kinematic boundary condition demands $b^2\nabla p \cdot \hat{\mathbf{n}} = -U\hat{n}_x - \dot{\mathbf{X}}\cdot\hat{\mathbf{n}}$.

In accordance with Keeler et al. [19], we set $h = 0.024$, $w = 0.25$, $L = 4$, $\alpha = 40$, $s = 40$ and fix the projected bubble volume to $V = \pi r^2$ with $r = 0.46$. Our implementation yields the same bistable region as described by Keeler et al. [19], where the three types of stationary solutions are represented in Fig. 5(b). The stability diagram also agrees perfectly with the data extracted from their work (cf. Fig. 5(c)), where lines are our results and dots are the results of Keeler et al. The bifurcation points in Fig. 5(c) were obtained by bifurcation tracking. Once a bifurcation point is located, one can obtain entire bifurcation diagrams in minutes by continuation, as e.g. shown exemplary for an increase in the bump height h in Fig. 5(d). With increasing height h, the bistable region shrinks, the pitchfork and the Hopf bifurcation of the slightly asymmetric branch merge and are dominated by another Hopf bifurcation branch. Eventually, at $h \approx 0.053$, symmetric solutions cannot be found anymore for any flow rate Q. A detailed analysis of the entire parameter space is of course not within the scope of this article. However, with our method of dynamic code generation, it is also straightforward to formulate generalizations of this problem, e.g. considering Stokes flow with a Brinkman term instead of Darcy's law. This would allow to impose correct tangential boundary conditions, e.g. no-slip boundaries at the side walls of the channel.

As shown in Table 1, we have tested in total three different methods to locate the pitchfork bifurcation. When using a mesh that complies with the symmetry $y \to -y$, best convergence was obtained. In that case, the symmetry constraint $\mathbf{U}\cdot\mathbf{S} = 0$ can be fulfilled on a discrete level, where the anti-symmetric vector $\mathbf{S}$ can be initialized by the dominant eigenvector close to the bifurcation. As observed for all bifurcations investigated here, a fully symbolical assembly of the augmented pitchfork tracking system (8) outperforms the finite-difference approximation of the second-order derivatives, i.e. the Hessian terms in (8). Likewise, as for the other tested bifurcations, by applying finite differences already for the first-order derivatives in the non-augmented Jacobian $\mathbf{J}$ prevented convergence. When using a mesh that does not comply with the symmetry, the discrete constraint $\mathbf{U}\cdot\mathbf{S} = 0$ in (8) usually cannot be satisfied (cf. discussion in section 2.2.2). Therefore, no convergence can be seen (cf. Table 1), irrespective of whether the derivatives are calculated symbolically or by finite differences. However, we can achieve convergence when replacing the discrete constraint $\mathbf{U}\cdot\mathbf{S} = 0$ by its weakly formulated counterpart. For that, we lift both $\mathbf{U}$ and $\mathbf{S}$ to continuous functions $\mathcal{U}(\mathbf{x})$ and $\mathcal{S}(\mathbf{x})$ of the spatial coordinate $\mathbf{x} = (x, y)$ by expanding the discrete vectors via the shape functions corresponding to each unknown field. Subsequently, the symmetry constraint is evaluated by spatial integration of the product $\mathcal{U}\mathcal{S}$. More details can be found in the supplementary material. With this feature, is not necessary to construct a symmetry-conforming mesh, e.g. when the symmetry is a priori not known.

6. Azimuthal stability analysis of axisymmetric base states

Our method to symbolically evaluate the Hessian has proven to work well in the previous validation section. In principle, it can also be applied to full three-dimensional problems, but the monolithic treatment will result in huge augmented Jacobian matrices which are cumbersome to invert for Newton's method. Direct solvers will run out of memory and finding suitable preconditioners for an iterative solver is complicated for the augmented problem.

If a bifurcation occurs for a base solution with particular symmetry, one can use this symmetry to reduce the problem size, e.g. from full three-dimensional to axisymmetric cylindrical coordinates, but still allow for instabilities that break this symmetry. Here, we focus on axisymmetric problems that lose the azimuthal symmetry in a bifurcation, but the same method can also be applied if the base state is e.g. invariant in the third Cartesian dimension.

6.1. Method outlined on a static mesh

For simplicity, we first develop the method of static meshes and subsequently generalize it for moving meshes. This numerical approach has been e.g. applied by Yim et al. for pancake vortices in stratified liquids [42], but also e.g. to numerically predict the self-propulsion of Leidenfrost droplets [43]. Recently, also the symmetry breaking due to thermal Marangoni flow, both in non-volatile droplets on heated or cooled substrates [44] and in rotating annular pools [45], has been investigated on the basis of the following method.

Axisymmetric problems can best be described in axisymmetric cylindrical coordinates (r, z), i.e. an axisymmetric stationary solution $\mathcal{U}^0$ is just a function of (r, z). However, to also allow for perturbations breaking this symmetry, the residual formulation $\mathcal{R}^*$

must also account for derivatives with respect to the azimuthal coordinate φ, e.g. $\nabla f = (\partial_r f, r^{-1}\partial_\varphi f, \partial_z f)$ for any scalar function f. If this is ensured, the linear evolution of an azimuthally perturbed state

$$\mathcal{U}^{\text{pert}}(r,\varphi,z,t) = \mathcal{U}^0(r,z) + \epsilon e^{\lambda_m t} e^{im\varphi} \mathcal{V}^m(r,z) + \text{c.c.} \tag{39}$$

can be considered. The problem is, however, still formulated on a two-dimensional mesh, i.e. using the reduction by the symmetry, whereas the azimuthal dynamics are entirely given by the mode $e^{im\varphi}$. The goal is to find the complex-valued eigenfunction $\mathcal{V}^m$ and the corresponding eigenvalue λ_m for a given azimuthal mode with integer-valued mode number m. In linear order in the parameter $\epsilon \ll 1$, these modes are independent and upon linerization one obtains

$$\mathcal{R}^*(\mathcal{U}^{\text{pert}}) \approx \epsilon e^{\lambda_m t} \left(\lambda_m \mathcal{M} + \mathcal{J}\right) e^{im\varphi} \mathcal{V}^m(r,z) + \text{c.c.} = 0\,. \tag{40}$$

Here, the operators $\mathcal{M}$ and $\mathcal{J}$ are the continuous analogues of the discretized mass matrix and the Jacobian, i.e. the Gâteaux-derivatives of $\mathcal{R}^*$ with respect to $\partial_t \mathcal{U}$ and $\mathcal{U}$ evaluated at 0 and $\mathcal{U}^0$, respectively. Both operators may contain φ-derivatives, i.e. they are acting on the product $e^{im\varphi}\mathcal{V}^m(r,z)$. Obviously, for (40) to hold after spatial discretization, i.e. $(\mathcal{U}^0, \mathcal{V}^m, \mathcal{M}, \mathcal{J}) \to (\mathbf{U}^0, \mathbf{V}^m, \mathbf{M}^m, \mathbf{J}^m)$, one has to solve the generalized discretized eigenproblem

$$\lambda_m \mathbf{M}^m \mathbf{V}^m = -\mathbf{J}^m \mathbf{V}^m \tag{41}$$

which differs from (4) by having an m-dependent mass and Jacobian matrix. Furthermore, $\mathbf{M}^m$ and $\mathbf{J}^m$ are in general complex-valued now, since odd derivatives with respect to φ induce imaginary contributions.

Since our framework is based on a symbolical treatment of the entered residual formation, it is capable to derive the expressions necessary to assemble $\mathbf{M}^m$ and $\mathbf{J}^m$ automatically from arbitrary weak formulations of the residual functional $\mathcal{R}^*$. Before any derivatives are applied in the entered form of $\mathcal{R}^*$ or any spatial discretization is performed, first all scalar fields f and vector fields $\mathbf{u}$ in $\mathcal{U}$ are expanded according to (40), i.e.

$$f(r,\varphi,z,t) \to f^0(r,z,t) + \epsilon f^m(r,z,t) e^{im\varphi} \tag{42}$$

$$\mathbf{u}(r,\varphi,z,t) \to \mathbf{u}^0(r,z,t) + \epsilon \mathbf{u}^m(r,z,t) e^{im\varphi}\,. \tag{43}$$

The fields f^0 and $\mathbf{u}^0$ are entries of the axisymmetric stationary solution function $\mathcal{U}^0$, whereas f^m and $\mathbf{u}^m$ are part of the eigenmode $\mathcal{V}^m$. Although f^0 and $\mathbf{u}^0$ stationary solutions, i.e. are not explicitly time-dependent, and the time-dependence of f^m and $\mathbf{u}^m$ is later replaced by $e^{\lambda_m t}$, an arbitrary time dependency is considered here in order to automatically generate the mass matrix entries. The corresponding test functions g and $\mathbf{v}$ are replaced by

$$g(r,z) \to g(r,z) e^{-im\varphi} \qquad \mathbf{v}(r,z) \to \mathbf{v}(r,z) e^{-im\varphi} \tag{44}$$

The vectorial fields and test functions are additionally augmented by an additional component φ-direction during this step.

Additional caution must be taken for global unknowns, which are used as Lagrange multipliers to enforce particular constraints. This can be e.g. a volume constraint, which feeds back to the system as a constant pressure offset, as we have used it for e.g. the prescribed volume of the detaching droplet in (27) or the volume of the bubble in (36) (cf. supplementary material for the weak formulations). Also, the nullspace of a pressure field, which arises in case of pure Dirichlet boundary conditions for the normal velocity, can be removed via such a similar Lagrange multiplier. In the azimuthal eigenproblem for $m \neq 0$, such constraints are automatically fulfilled, since the azimuthal integration over $\exp(im\varphi)$ averages out the contribution. Therefore, such global degrees of freedom, are not expanded, i.e. only the base mode corresponding to $m = 0$ is kept.

With the expansions (43) and (44), the original axisymmetric residual formulation can be recovered by setting $\epsilon = m = 0$, whereas the in general complex-valued auxiliary azimuthal residual $\mathcal{R}^{*m}$ is obtained by the first order Taylor coefficient in ϵ.

For the example, an axisymmetric diffusion equation for a scalar field f, i.e. $\partial_t f = \nabla^2 f$ (without Neumann terms and with test function g) yields the following axisymmetric residual $\mathcal{R}^{*,0}$ and auxiliary azimuthal residual $\mathcal{R}^{*m}$ after applying this method:

$$\mathcal{R}^*(\mathcal{U}) = 2\pi \int_\Omega \left(\partial_t f g + \partial_r f \partial_r g + \partial_z f \partial_z g\right) r \,\mathrm{d}r\mathrm{d}z \tag{45}$$

$$\mathcal{R}^{*,m}(\mathcal{U}^0, \mathcal{V}^m) = 2\pi \int_\Omega \left(\partial_t f^m g + \partial_r f^m \partial_r g + \partial_z f^m \partial_z g + \frac{m^2}{r^2} f g\right) r \,\mathrm{d}r\mathrm{d}z\,. \tag{46}$$

Nonlinear terms in the original residual would give additional couplings between the axisymmetric stationary solution $\mathcal{U}^0$ and the azimuthal perturbation $\mathcal{V}^m$, i.e. between f^0 and f^m in this example case.

After this, spatial discretization can be performed, e.g. expansions in shape functions and, by our method, performant C code is generated for both types of residuals and the corresponding mass matrices, Jacobian matrices, potential parameter derivatives and Hessians. The matrices for the azimuthal eigenproblem (41) are then obtained from the discretized auxiliary residual $\mathbf{R}^{*m}$ via

$$\mathbf{M}^m = \frac{\partial \mathbf{R}^{*m}}{\partial \dot{\mathbf{V}}^m} \qquad \text{and} \qquad \mathbf{J}^m = \frac{\partial \mathbf{R}^{*m}}{\partial \mathbf{V}^m} \tag{47}$$

evaluated at $\mathbf{U}^0$ and with $\dot{\mathbf{U}}^0 = \dot{\mathbf{V}}^m = 0$. Since the azimuthal perturbation $\mathcal{V}^m$ enters only linearly in $\mathcal{R}^{*m}$, both of these matrices are independent on $\mathbf{V}^m$, but may depend on the axisymmetric stationary solution $\mathbf{U}^0$ due to nonlinearities.

An axisymmetric stationary solution $\mathbf{U}^0$ is obtained as before, i.e. by Newton's method with the generated code corresponding to $\mathcal{R}^*$. This solution can then be investigated for stability and bifurcations as previously to assess for stability with respect to axisymmetric perturbations ($m = 0$). But the axisymmetric solution $\mathbf{U}^0$ can additionally be checked for axisymmetry-breaking instabilities ($m \neq 0$) by solving the azimuthal eigenproblem (41) for different values of m. For physically reasonable problems, the range of m is limited since, for high values of m, stabilizing terms will dominate and yield only eigenvalues with negative real parts, as e.g. due to the term m^2/r^2 in (46) stemming from azimuthal diffusion. A single line in the driver code entered by the user automatically invokes the code generation for all required matrices, so that arbitrary problems can be investigated for axial symmetry breaking easily.

6.2. Boundary conditions for the eigenvalue problem

Particular care must be taken with the boundary conditions at the axis of symmetry $r = 0$. For the axisymmetric base state, scalar fields have to fulfill $\partial_r f^0 = 0$, likewise the axial component of vector fields fulfills $\partial_r u_z^0 = 0$, whereas the radial and azimuthal component have to vanish, i.e. $u_r^0 = 0$ and $u_\varphi^0 = 0$. For the eigenvector corresponding to the azimuthal perturbation, the boundary conditions at the axis of symmetry depend on m. For $m = 0$, we solve the conventional eigenvalue problem (4), with the same boundary conditions, but for $|m| = 1$, the boundary conditions at $r = 0$ must be changed to $f^m = u_z^m = 0$ and $\partial_r u_r^m = \partial_r u_\varphi^m = 0$, since the basis of the vector components exactly rotates with the mode $m = 1$ and for $|m| \geq 2$, all components have to vanish, i.e. $f^m = u_r^m = u_\varphi^m = u_z^m = 0$ [46,42,43].

Integral constraints associated with a global degree of freedom, e.g. a Lagrange multiplier enforcing some volume or some spatial average of a field, must be deactivated for $m \neq 0$ as well. Due to the rotation by $e^{im\varphi}$, an azimuthal perturbation with $m \neq 0$ always has a vanishing contribution to these constraints when considering the full three-dimensional problem, i.e. the corresponding entry in the eigenvector $\mathbf{V}^m$ has to be removed.

Our framework provides a particular class that must be attached to the axis of symmetry to impose the correct boundary conditions depending on the value of m. This class distinguishes between vectorial and scalar fields (and tensor fields in the future) and modifies the Jacobian and mass matrices between assembly and solving. Whenever necessary for the particular scalar field or vectorial component, a Dirichlet condition of zero value is realized by replacing the corresponding assembled row in the Jacobian and mass matrix to identity and zero, respectively.

6.3. Bifurcation tracking for azimuthal symmetry breaking

We are interested in generalizing the bifurcation tracking approaches discussed in section 2.2 to azimuthal instabilities, i.e. in finding the critical parameter p_c for which the axisymmetric stationary solution breaks its azimuthal symmetry. The following bifurcation tracking method for azimuthal symmetry breaking allows to use two-dimensional discretizations to solve this problem. It consists in solving the eigenproblem (41) with the goal of finding the critical parameter p_c for which the eigenvalue $\lambda^m = \lambda_\mathrm{r}^m + i\lambda_\mathrm{i}^m$ crosses the imaginary axis, i.e. $\lambda_\mathrm{r}^m(p_\mathrm{c}) = 0$. In order to do so, the discretized residual vector from the axisymmetric base state, $\mathbf{R}^*$, is augmented with the eigenproblem (41) and with a constraint to avoid the trivial eigensolution $\mathbf{V}^m = \mathbf{V}_\mathrm{r}^m + i\mathbf{V}_\mathrm{i}^m = \mathbf{0}$. The mass and Jacobian matrices, $\mathbf{M}^m = \mathbf{M}_\mathrm{r}^m + i\mathbf{M}_\mathrm{i}^m$ and $\mathbf{J}^m = \mathbf{J}_\mathrm{r}^m + i\mathbf{J}_\mathrm{i}^m$, corresponding to the m-dependent azimuthal auxiliary residual function $\mathcal{R}^{*,m}$ (cf. (47)) are assembled to the augmented residuals $\mathbf{R}^{\mathrm{aug},m}$ and augmented Jacobian $\mathbf{J}^{\mathrm{aug},m}$:

$$\mathbf{U}^{\mathrm{aug},m} = \begin{pmatrix} \mathbf{U} \\ \mathbf{V}_\mathrm{r}^m \\ \mathbf{V}_\mathrm{i}^m \\ p \\ \lambda_\mathrm{i} \end{pmatrix} \qquad \mathbf{R}^{\mathrm{aug},m} = \begin{pmatrix} \mathbf{R} \\ \mathbf{J}_\mathrm{r}^m\mathbf{V}_\mathrm{r}^m - \mathbf{J}_i^m\mathbf{V}_\mathrm{i}^m - \lambda_\mathrm{i}(\mathbf{M}_\mathrm{r}^m\mathbf{V}_\mathrm{i}^m + \mathbf{M}_\mathrm{i}^m\mathbf{V}_\mathrm{r}^m) \\ \mathbf{J}_\mathrm{i}^m\mathbf{V}_\mathrm{r}^m + \mathbf{J}_\mathrm{r}^m\mathbf{V}_\mathrm{i}^m + \lambda_\mathrm{i}(\mathbf{M}_\mathrm{r}^m\mathbf{V}_\mathrm{r}^m - \mathbf{M}_\mathrm{i}^m\mathbf{V}_\mathrm{i}^m) \\ \mathbf{V}_\mathrm{r}^m \cdot \mathbf{C} - 1 \\ \mathbf{V}_\mathrm{i}^m \cdot \mathbf{C} \end{pmatrix}$$

$$\mathbf{J}^{\mathrm{aug},m} = \begin{pmatrix} \mathbf{J} & \mathbf{0} & \mathbf{0} & \partial_p\mathbf{R} & \mathbf{0} \\ \partial_\mathbf{U}(\mathbf{R}^{\mathrm{aug},m}_2) & \mathbf{J}_\mathrm{r}^m - \lambda_\mathrm{i}\mathbf{M}_\mathrm{i}^m & -\mathbf{J}_\mathrm{i}^m - \lambda_\mathrm{i}\mathbf{M}_\mathrm{r}^m & \partial_p(\mathbf{R}^{\mathrm{aug},m}_2) & -\mathbf{M}_\mathrm{r}^m\mathbf{V}_\mathrm{i}^m - \mathbf{M}_\mathrm{i}^m\mathbf{V}_\mathrm{r}^m \\ \partial_\mathbf{U}(\mathbf{R}^{\mathrm{aug},m}_3) & \mathbf{J}_\mathrm{i}^m + \lambda_\mathrm{i}\mathbf{M}_\mathrm{r}^m & \mathbf{J}_\mathrm{r}^m - \lambda_\mathrm{i}\mathbf{M}_\mathrm{i}^m & \partial_p(\mathbf{R}^{\mathrm{aug},m}_3) & \mathbf{M}_\mathrm{r}^m\mathbf{V}_\mathrm{r}^m - \mathbf{M}_\mathrm{i}^m\mathbf{V}_\mathrm{i}^m \\ 0 & \mathbf{C} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{C} & 0 & 0 \end{pmatrix}. \tag{48}$$

For brevity, the abbreviation $\mathbf{R}^{\mathrm{aug},m}_i$ in the augmented Jacobian $\mathbf{J}^{\mathrm{aug},m}$ corresponds to the i^{th} block row of the augmented residual vector $\mathbf{R}^{\mathrm{aug},m}$. The augmented system is more involved than the one required for e.g. Hopf bifurcations due to the fact that the mass and Jacobian matrices $\mathbf{M}^m$ and $\mathbf{J}^m$ are complex-valued. The constraint vector $\mathbf{C}$ is usually set to the initial guess of the eigenvector $\mathbf{V}^m$. If the initial guess is sufficiently close to the bifurcation, the Newton method (3) can be used to solve the augmented system, yielding the stationary axisymmetric solution $\mathbf{U}_0$, the azimuthal perturbation eigenvector $\mathbf{V}^m$, the imaginary part of the corresponding eigenvalue λ_i and the critical parameter p_c. Since the azimuthal eigenproblem has complex-valued matrices, it is not necessary to distinguish between the different bifurcation types, e.g. fold, pitchfork or Hopf bifurcation.

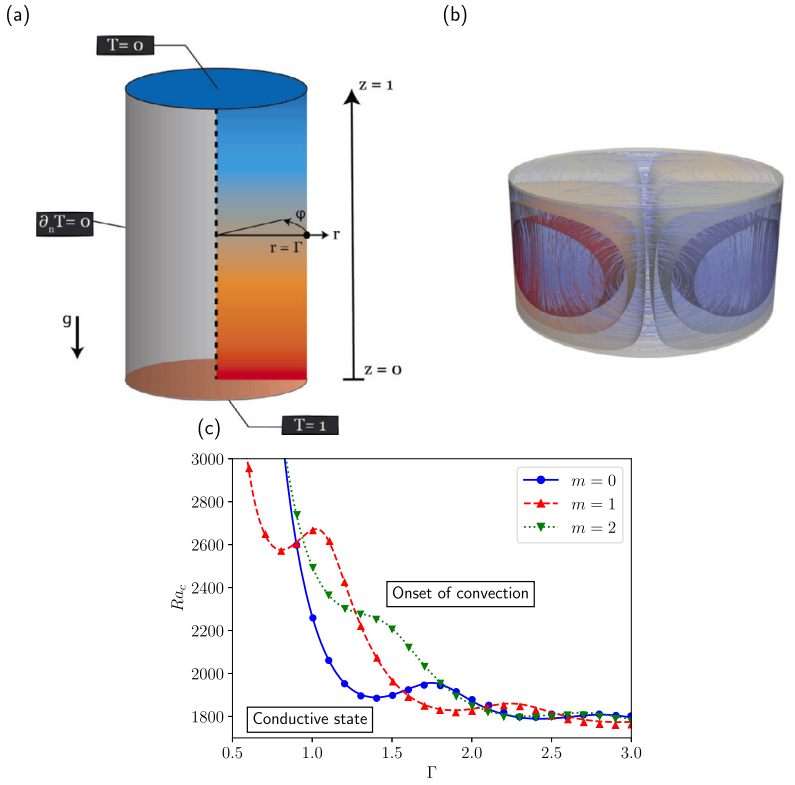

Fig. 6. Rayleigh-Bénard in a cylindrical container. (a) Sketch of the geometry and temperature's boundary conditions. (b) Eigenfunction of temperature field and streamlines of velocity for stationary solution at Ra_c, with aspect ratio $\Gamma = 1$ and azimuthal mode $m = 2$. (c) Comparison of the critical Rayleigh number Ra_c for the onset of convection for different azimuthal modes $m = \{0, 1, 2\}$ between our results (lines) and the ones in [47] (points).

6.4. Validation on a static mesh

As a validation example of the bifurcation tracking for azimuthal symmetry breaking, we consider the Rayleigh-Bénard convection problem in a cylindrical container, as analyzed by Borónska and Tuckerman [47]. A fluid confined within a cylindrical boundary and heated from below, as shown in Fig. 6(a), changes from a motionless conductive state to a flow with convective rolls, due to the action of gravity, when a critical temperature difference ΔT_c between the bottom and top layers is reached. We refer to the Rayleigh number Ra to characterize the temperature difference and to a critical Ra_c for the flow to settle in. The characteristics of these rolls depend on Ra, the Prandtl number Pr and on the cylinder's aspect ratio $\Gamma = r/h$, where r is the radius of the cylinder and h is its height. The rolls can be either axisymmetric or non-axisymmetric, depending on the critical azimuthal mode m of which corresponding eigenvector settles the instability that generates the rolls. For instance, if for a certain Ra there is an instability solely for $m = 2$, then the stationary solution will have four convective rolls in the cylinder, as depicted on the streamlines of the velocity in Fig. 6(b).

The used model equations here are the nondimensionalized Boussinesq equations, which correspond to the Navier-Stokes equations with the Boussinesq approximation for the density, and the advection diffusion equation for temperature T:

$$\nabla \cdot \mathbf{u} = 0 \tag{49}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\mathrm{Pr}\,\mathrm{Ra}\,\nabla p + \mathrm{Pr}\,\nabla^2 \mathbf{u} + \mathrm{Pr}\,\mathrm{Ra}\,T\mathbf{e}_z \tag{50}$$

$$\partial_t T + \mathbf{u} \cdot \nabla T = \nabla^2 T\,. \tag{51}$$

Furthermore, in coherence with the boundary conditions used in [47], we impose no-slip boundary conditions on the velocity field at the cylinder's walls. The temperature is fixed to $T_0 = 1$ at $z = 0$ and to 0 at $z = 1$, while at the sidewalls of the cylinder are adiabatic, i.e. $\nabla T \cdot \hat{\mathbf{n}} = 0$ at $r = \Gamma$, as indicated on Fig. 6(a). In accordance to the analysis of [47], we consider only aspect ratios $\Gamma \sim 1$. The instability of the conductive state is independent of Pr, so $\mathrm{Pr} = 1$ in our simulations. We use a two-dimensional rectangular domain to investigate the Ra_c for which the flow becomes unstable for each $m = \{0, 1, 2\}$. Starting by fixing m to one of the prescribed values, we find a Ra value which is, for an initial Γ, close to the bifurcation, to supply a good initial guess for the critical eigenvector. Then, we augment our system through the method explained on the section 6.3 and we solve it through the Newton method, using the initial guess of the critical eigenvector at the first iteration. Lastly, we use arclength continuation on Γ to obtain the bifurcation curve corresponding to each m. Note that we scale the r-coordinate with the Γ parameter, thus not requiring a constant remeshing in order

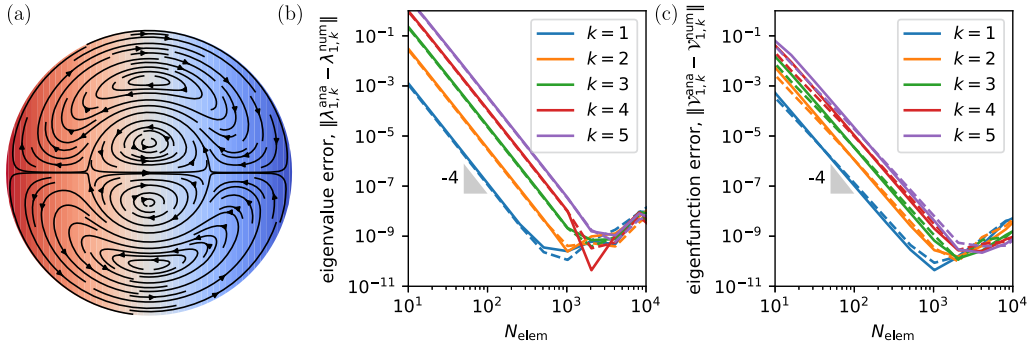

Fig. 7. Discussion of singular terms arising from the viscous stress in the Navier-Stokes equations for $m = 1$, here on a unit disc. (a) Analytical solution of the second ($k = 2$) eigenfunction for $m = 1$ (pressure color-coded, velocity by arrows). (b) Convergence of the eigenvalues as function of the number of elements N_{elem} with singular terms in the Jacobian (solid) and without singular terms after substitution (dashed) to the analytical values. (c) Convergence for the corresponding eigenvectors to the analytical eigenfunctions.

to obtain these curves. The results are depicted in Fig. 6(c), showing a good agreement between our results (lines) and the ones in [47] (points).

6.5. Discussion of singular terms at the axis of symmetry stemming from second-order spatial derivatives

It should be noted that the viscous term in (50) leads to $1/r$-terms in the residual for the azimuthal perturbation, which result in non-integrable entries in the azimuthal Jacobian $\mathbf{J}^m$ (cf. supplementary material for the weak formulation). Such terms can arise from second-order spatial derivatives, as e.g. also seen for the term $2\pi r\, m^2 f g / r^2$ term in (46), which stems from the Laplace operator in the diffusion equation. However, the latter term does not pose any problem, since for $m \neq 0$, the boundary condition at $r = 0$ requires $f, g \to 0$ for $r \to 0$. For vectorial quantities, however, such terms deserve a more careful discussion.

As a minimal example, we consider a Navier-Stokes equation solved on a unit disc, i.e.

$$\begin{aligned} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \nabla \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right) \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{52}$$

with $\mathbf{u} = 0$ at $r = 1$. The axisymmetric base solution is $\mathbf{u}^0 = (u_r^0, u_\varphi^0) = 0$ and $p^0 = \text{const}$ and the corresponding azimuthal residual in weak formulation (with $\mathbf{v}$ and q as test functions) is solved on a line mesh $r \in [0, 1]$ around this trivial base solution. The azimuthal residual reads:

$$\begin{aligned} \mathcal{R}^{*,m} = 2\pi \int_0^1 r \Bigg[& \partial_t \mathbf{u}^m \cdot \mathbf{v} - p^m \left(\frac{1}{r} \partial_r (r v_r) - \frac{im}{r} v_\varphi \right) + \left(\frac{1}{r} \partial_r (r u_r^m) + \frac{im}{r} u_\varphi^m \right) q \\ & + 2 \partial_r u_r^m \partial_r v_r + \frac{im}{r} \left(u_r^m \partial_r v_\varphi - \partial_r u_\varphi^m v_r \right) \\ & + \left(\frac{(m^2+2) u_r^m + 3im u_\varphi^m}{r^2} \right) v_r + \frac{(2m^2+1) u_\varphi^m - 3im u_r^m}{r^2} v_\varphi \Bigg] \, \mathrm{d}r \end{aligned} \tag{53}$$

Due to the factor $2\pi r$ stemming from the integration, all terms in the first two lines of (53) are integrable, whereas the last line contains the in total $1/r$-singularities, which are not integrable if u_r^m, u_φ^m, v_r^m and v_φ^m are arbitrary at $r = 0$. For $m = 0$ and $|m| \geq 2$, the boundary conditions (cf. section 6.2) demand $u_r^m = u_\varphi^m = 0$ (i.e. also $v_r = v_\varphi = 0$ due to the strong boundary condition), so that only the case $|m| = 1$ requires a detailed investigation. Analytically, the case $m = \pm 1$ also does not pose a problem, since $u_\varphi^m = \pm i u_r^m$ must hold for $r \to 0$. Only if this condition is fulfilled, the velocity vector, $\left(u_r^m \mathbf{e}_r + u_\varphi^m \mathbf{e}_\varphi \right) \exp(im\varphi) + \text{c.c.}$, is unique at $r = 0$, i.e. independent of φ, and reads $2\Re(u_r^m)\mathbf{e}_x \mp 2\Im(u_r^m)\mathbf{e}_y$ in Cartesian coordinates. This condition resolves the apparently singular terms in (53) for $m = \pm 1$.

When filling the Jacobian matrix numerically, however, we have to evaluate integrals over the singular terms without being able to apply the necessary condition $u_\varphi^m = \pm i u_r^m$. The two diagonal and two off-diagonal Jacobian entries corresponding to the degrees of freedom of u_r^m and u_φ^m at $r = 0$ are therefore analytically not defined. Numerically, the Gauss-Legendre quadrature will approximate these entries wrongly by rather huge, but finite values. These huge values, however, are sufficient to enforce the necessary condition $u_\varphi^m = \pm i u_r^m$ at $r = 0$, as shown in Fig. 7. Here, we compare the convergence of the numerically calculated eigenvalues and -vectors to the analytical solution (cf. supplementary material) of the m=1-eigenproblem of (52) as function of the number of elements N_{elem}. The solid lines in Fig. 7(b,c) correspond to the numerical error in the eigenvalues and -vectors, respectively, based on the Jacobian assembled by Gauss-Legendre quadrature of the singular terms. Clearly, the error shows a perfect convergence. To compare with numerical data without any singular terms (dashed lines), we applied the proposed substitution of the singular terms by non-singular terms by using the continuity equation as suggested in Refs. [48,44], with virtually the same convergence. Note that for an intense

number of elements, $N_{\text{elem}} > 1000$, the error increases again. This increase can be attributed to round-off errors in the eigensolver, since we observe this behavior for all tested cases, also for axisymmetric cases ($m = 0$) or eigenvalue problems in Cartesian coordinates, e.g. a 1d diffusion equation.

An additional validation with the results of Ref. [44] is part of the supplementary material. One can conclude that, analytically, the discussed terms are only apparently singular and, numerically, they do not require any special treatment, despite of wrongly approximated undefined integrals.

6.6. Method generalized for moving meshes

In the following, we combine the described methods of bifurcation tracking on moving meshes and the investigation of the azimuthal stability of axisymmetric base states. Typical problems that undergo an azimuthal instability by changing the shape are e.g. capillary bridges beyond the Steiner limit [49] or bucking of elastic tubes due to capillary effects, as e.g. analyzed by Hazel & Heil [50].

Of course, these problems can be implemented straight-forward on a three-dimensional Cartesian mesh, but for the azimuthal linear stability, it is sufficient to reduce the dynamics around the axisymmetric stationary solution to a two-dimensional mesh, i.e. evaluated at $\varphi = 0$, and apply the normal mode expansion in terms of $\exp(im\varphi)$ on it, as described in section 6.1. However, for this step, it is crucial to also add perturbations to the mesh coordinates. In a general three-dimensional cylindrical coordinate system, the position vector of the axisymmetric base state is given by

$$\mathbf{x}^0 = r^0 \hat{\mathbf{e}}_r(\varphi) + z^0 \hat{\mathbf{e}}_z \tag{54}$$

and the corresponding perturbed position vector reads $\mathbf{x}^0 + \epsilon \mathbf{x}^m e^{im\varphi}$ with

$$\mathbf{x}^m = r^m \hat{\mathbf{e}}_r + r^0 \varphi^m \hat{\mathbf{e}}_\varphi + z^m \hat{\mathbf{e}}_z \,. \tag{55}$$

If the azimuthal mesh position does not have a physical meaning, i.e. it is just used for parametrization, φ^m can be set to zero, but in general cases, e.g. for the torsion of an elastic body, φ^m must be kept as unknown part of the perturbation, which then must be solved for in the azimuthal eigenvalue problem. As before, during the spatial discretization, both $\mathbf{x}^0$ and $\mathbf{x}^m$ are expanded in terms of the position shape functions $\bar{\psi}^{\bar{l}}$:

$$r^0 = X_r^{0\bar{l}} \bar{\psi}^{\bar{l}}, \qquad z^0 = X_z^{0\bar{l}} \bar{\psi}^{\bar{l}} \tag{56}$$

$$r^m = X_r^{m\bar{l}} \bar{\psi}^{\bar{l}}, \qquad z^m = X_z^{m\bar{l}} \bar{\psi}^{\bar{l}}, \qquad \varphi^m = X_\varphi^{m\bar{l}} \bar{\psi}^{\bar{l}} \tag{57}$$

i.e. in the discrete nodal positions and the complex-valued corresponding azimuthal eigenvector, respectively.

Likewise, the normal $\hat{\mathbf{n}}$ changes with the perturbation. During the perturbation, the normal of the axisymmetric base state

$$\hat{\mathbf{n}}^0 = \hat{n}_r^0 \hat{\mathbf{e}}_r + \hat{n}_z^0 \hat{\mathbf{e}}_z \tag{58}$$

must be replaced by a corresponding linearized perturbed normal, $\hat{\mathbf{n}}^0 + \epsilon \mathbf{n}^m e^{im\varphi}$, where the change of the normal (in general not of unit length) due to the perturbation in first order in ϵ is given by

$$\mathbf{n}^m = \left(\frac{\partial n_r^0}{\partial X_j^{0\bar{l}}} X_j^{m\bar{l}} \right) \hat{\mathbf{e}}_r + \left(\hat{n}_r^0 \varphi^m - \frac{im}{r^0} \hat{\mathbf{n}}^0 \cdot \mathbf{x}^m \right) \hat{\mathbf{e}}_\varphi + \left(\frac{\partial n_z^0}{\partial X_j^{0\bar{l}}} X_j^{m\bar{l}} \right) \hat{\mathbf{e}}_z \tag{59}$$

Additionally, the differential operators must be extended accordingly. Without azimuthally perturbed mesh coordinates, we calculate e.g. the divergence of a vector field $\mathbf{v} = \mathbf{v}^0 + \epsilon e^{im\varphi} \mathbf{v}^m$ by

$$\nabla \cdot \mathbf{v} = g^{\alpha\beta} t_{\alpha,r} \frac{\partial v_r}{\partial s^\beta} + g^{\alpha\beta} t_{\alpha,z} \frac{\partial v_z}{\partial s^\beta} + \frac{v_r}{r} + \frac{1}{r} \partial_\varphi v_\varphi \,, \tag{60}$$

i.e. the metric tensor just accounts for the mapping from the local element coordinates s_β to Cartesian Eulerian coordinates (cf. (19)), while additional terms from the cylindrical coordinate system are added afterwards. Due to the presence of the azimuthal perturbed coordinates, its first order expansion in ϵ reads

$$\begin{aligned} \nabla \cdot \mathbf{v} = {} & g^{\alpha\beta} t_{\alpha,r} \frac{\partial v_r^0}{\partial s^\beta} + g^{\alpha\beta} t_{\alpha,z} \frac{\partial v_z^0}{\partial s^\beta} + \frac{v_r^0}{r^0} \\ & + \epsilon e^{im\varphi} \left(g^{\alpha\beta} t_{\alpha,r} \frac{\partial v_r^m}{\partial s^\beta} + g^{\alpha\beta} t_{\alpha,z} \frac{\partial v_z^m}{\partial s^\beta} + \frac{v_r^m}{r^0} + \frac{1}{r^0} \partial_\varphi v_\varphi^m \right) \\ & + \epsilon e^{im\varphi} \left(\left[D_{rr}^{\bar{l}\beta} X_r^{m\bar{l}} + D_{rz}^{\bar{l}\beta} X_z^{m\bar{l}} \right] \frac{\partial v_r^0}{\partial s^\beta} + \left[D_{zr}^{\bar{l}\beta} X_r^{m\bar{l}} + D_{zz}^{\bar{l}\beta} X_z^{m\bar{l}} \right] \frac{\partial v_z^0}{\partial s^\beta} - \frac{r^m}{(r^0)^2} v_r^0 \right) . \end{aligned} \tag{61}$$

While the first two lines are already present in the azimuthal stability analysis without a moving mesh, the third line considers the effect of the linear azimuthal perturbation of the mesh coordinates. Here, we use the abbreviation

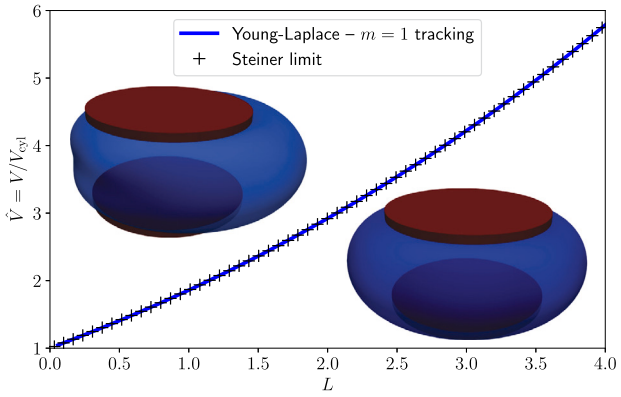

Fig. 8. Azimuthal bifurcation tracking on a moving mesh on the basis of a liquid bridge. If the normalized liquid volume $\hat{V}$ trapped between the two cylindrical plates with distance L exceeds a threshold, an $m = 1$ instability of the shape is observed. Our azimuthal bifurcation tracking method, applied on a moving mesh, perfectly recovers the theoretical Steiner limit. The insets show the axisymmetric base state (lower right) and the base state plus the critical $m = 1$ eigenfunction (upper left) at $L = 1$ and $\hat{V} \approx 1.8603$.

$$D_{ij}^{\bar{I}\beta} = \partial_{X_j^{\bar{I}}} \left(g^{\alpha\beta} t_{\alpha,i} \right) \Big|_{\mathbf{X}^0} , \tag{62}$$

i.e. the derivatives of the transformation terms with respect to the mesh coordinates. While they appear to be cumbersome, they are in fact already calculated in beforehand for the symbolical Jacobian of the moving mesh. Relations for $D_{ij}^{\bar{I}\beta}$ are available in the supplementary material.

All these additional expansions are performed automatically within our framework, if a moving mesh is considered. After the expansion, the m-dependent azimuthal auxiliary residual function $\mathcal{R}^{*,m}$ is again obtained by the first order in ϵ, from which the azimuthal eigenvalue problem matrices are subsequently derived.

6.7. Validation for moving meshes

Due to the complexity, it is hard to find a good validation case for azimuthal shape instabilities in literature. Here, we consider a capillary surface as before in section 5.1, but now in the configuration of a liquid bridge between two cylindrical plates with pinned contact lines and in absence of gravity. Long bridges undergo a Rayleigh-Plateau instability, whereas liquid bridges with high volumes (compared to the volume of a cylinder between the two plates) and small plate distances show asymmetric states. The entire stability dynamics has been already investigated in detail [49,51], so that in particular the transition to asymmetric states provides a suitable validation case of our azimuthal bifurcation tracking method. The bifurcation curve to non-axisymmetric bridges is known [52]. It happens exactly when the liquid volume exceeds the threshold so that the capillary surface becomes tangent to the contact line, which is the limit when the simplifying Steiner symmetrization is not possible anymore [49,52,51].

The axisymmetric base states of a liquid bridge can easily be solved by the Young-Laplace equation, where the capillary surface is represented by a revolved line mesh with coordinates (r, z) in axisymmetric cylindrical coordinates and the corresponding outward pointing normal $\hat{\mathbf{n}}$:

$$\nabla_S \cdot \hat{\mathbf{n}} = p \, . \tag{63}$$

The end points of the surface line are fixed at the nondimensional coordinates $(1, 0)$ and $(1, L)$, where L is the plate distance. The pressure (or curvature) p is constant along the surface. p is either given as a parameter or, alternatively, constitutes a Lagrange multiplier enforcing a prescribed normalized volume $\hat{V}$ by virtue of

$$\frac{2\pi}{3} \int \hat{\mathbf{n}} \cdot \mathbf{x} \, r \mathrm{d}l - \pi L \left(\hat{V} - \frac{1}{3} \right) = 0 \, . \tag{64}$$

The normalized volume $\hat{V}$ is the ratio of the actual volume of the liquid divided by the volume of a cylinder with height L and radius 1.

We can solve this system easily by our moving mesh capabilities and continue the solution e.g. in V and p or $\hat{V}$, respectively. However, this only gives access to axisymmetric $m = 0$ instabilities. By applying the described azimuthal bifurcation tracking, however, the limit of asymmetric states can be found as well without any further changes in the code. The automatic code generation takes care of expanding all quantities and operators, like the normal $\hat{\mathbf{n}}$ and its surface divergence $\nabla_S \cdot \hat{\mathbf{n}}$, to the corresponding azimuthally perturbed variants, from what the azimuthal eigenvalue problem can be assembled. We can jump on the symmetry-breaking $m = 1$ bifurcation by automatically adjusting p correspondingly and subsequently perform pseudo-arclength continuation in the bridge length L to obtain the curve shown in Fig. 8.

7. Conclusion

We have developed and validated a numerical method that allows stability analysis and bifurcation tracking on arbitrary multi-physics problems. Particular complications induced by problems with moving domains, i.e. on moving meshes, are tackled by exact symbolical derivatives of the entered system residual up to the second order, including the derivatives with respect to the moving mesh coordinates. On the basis of the symbolically obtained forms for the residual, the Jacobian and mass matrix, parameter derivatives thereof, as well as the Hessian, efficient C code is automatically generated, which ensures high performance. Due to the numerical exact treatment, our approach does not only outperform the trivially implemented finite-difference approach, but also ensures good convergence of Newton's methods applied to the augmented bifurcation tracking systems. For the latter, we have proven that finite-difference methods generically fail due to the inexact calculation of the derivatives, in particular on moving mesh problems.

Our method has been successfully validated on the basis of versatile literature results. By combining the bifurcation tracking with continuation, entire phase diagrams in the parameter space can be obtained at low computational costs. The definition of the entire equation system, including the geometry, parameters and potentially additional equations, usually takes only ~ 100 lines of easily readable PYTHON code, even for nontrivial multi-physics problems.

For complicated three-dimensional settings, however, the approach can still be rather expensive. As a workaround, our method has been generalized to automatically investigate azimuthal symmetry breaking instabilities of axisymmetric stationary solutions. Thereby, the symmetry of the base state is fully utilized, i.e. again allowing for quick calculations on an axisymmetric mesh in two spatial dimensions only, but yet extracting the full three-dimensional instabilities. Of course, it is always possible to manually code the full symbolical Jacobian, Hessian and azimuthal eigenproblems by hand in frameworks like OOMPH-LIB, but it is cumbersome work and must be adjusted manually for any change in the residuals of the problem.

With this method, it is envisioned to investigate a plethora of bifurcations in fluid dynamics which are hardly accessible by analytical methods. Due to the moving mesh capability, our framework will e.g. easily allow to find the Hopf bifurcation for the onset of bouncing of a droplet in a stratified liquid due to an interplay of Marangoni and Rayleigh forces, as e.g. reported in [53–57]. With the azimuthal symmetry-breaking analysis, also the onset of nonaxisymmetric flow fields in evaporating droplets due to solutal [58] or thermal [59] Marangoni flow can be analyzed. Likewise, the motion of a Leidenfrost droplet due to an m=1-instability can be investigated at a finite capillary number and including the entire gas phase dynamics, and thereby generalizing the analysis of Yim et al. [43]. Due to the moving mesh capability, also the onset of motion of an inverse Leidenfrost droplet levitating on a bath [60] can be obtained. The method furthermore could be applied to the autochemotactic motion of active droplets [61,62], toroidal liquid films [63], and to a plethora of more interesting scenarios.

While the bifurcation tracking can find the location and general type of the bifurcation, weakly nonlinear dynamics is not accessible, i.e. in particular it cannot reveal whether a bifurcation is super- or subcritical. Normally, transient simulations in the vicinity of the bifurcation can bring clarity, but for our azimuthal symmetry approach, only the linear dynamics is available. Here, the automatic code generation of our symbolic framework could easily derive a weakly nonlinear generalization of (39), i.e. including quadratic of cubic order in ϵ, including the nonlinear coupling between different, nonlinearly excited, azimuthal modes. For moving meshes, however, this might be too complicated due to the nonlinear changes of e.g. the normals. Also, the developed bifurcation trackers could be generalized to find codimension-two bifurcations (cf. e.g. [25,27]) or the stability of limit cycles in the future. Finally, it is envisioned to parallelize the entire framework via MPI, which is at the moment only partially implemented.

CRediT authorship contribution statement

Christian Diddens: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Conceptualization. **Duarte Rocha:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation.

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.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the Industrial Partnership Programme *Fundamental Fluid Dynamics Challenges in Inkjet Printing (Phase II): Physicochemical Hydrodynamics of Droplets* (KICH2.V4C.20.001) of the Netherlands Organisation for Scientific Research (NWO) & High Tech Systems and Materials (HTSM), co-financed by Canon Production Printing Netherlands B.V., University of Twente, and Eindhoven University of Technology. The authors thank Dr. Alice Thompson, Dr. Jack Keeler and Dr. Lukas Babor for providing additional information, data and source code.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcp.2024.113306.

References

[1] M.C. Cross, P.C. Hohenberg, Pattern formation outside of equilibrium, Rev. Mod. Phys. 65 (1993) 851–1112, https://doi.org/10.1103/RevModPhys.65.851.
[2] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B 237 (641) (1952) 37–72, https://doi.org/10.1098/rstb.1952.0012.
[3] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophys. J. 1 (6) (1961) 445–466, https://doi.org/10.1016/S0006-3495(61)86902-6.
[4] H. Uecker, D. Wetzel, J.D.M. Rademacher, pde2path - a Matlab package for continuation and bifurcation in 2d elliptic systems, Numer. Math., Theory Methods Appl. 7 (1) (2014) 58–106, https://doi.org/10.1017/S1004897900000295.
[5] E.J. Doedel, B. Oldeman, Auto-07p: continuation and bifurcation software, Concordia University Canada, Montreal, QC, 1998.
[6] M. Heil, A.L. Hazel, oomph-lib - An object-oriented multi-physics finite-element library, Lect. Notes Comput. Sci. Eng. 53 (2006) 19–49, https://doi.org/10.1007/3-540-34596-5{_}2.
[7] A.G. Salinger, E.A. Burroughs, R.P. Pawlowski, E.T. Phipps, L.A. Romero, Bifurcation tracking algorithms and software for large scale applications, Int. J. Bifurc. Chaos Appl. Sci. Eng. 15 (03) (2005) 1015–1032, https://doi.org/10.1142/S0218127405012508.
[8] H.A. Dijkstra, F.W. Wubs, A.K. Cliffe, E. Doedel, I.F. Dragomirescu, B. Eckhardt, A.Y. Gelfgat, A.L. Hazel, V. Lucarini, A.G. Salinger, et al., Numerical bifurcation methods and their application to fluid dynamics: analysis beyond simulation, Commun. Comput. Phys. 15 (1) (2014) 1–45, https://doi.org/10.4208/cicp.240912.180613a.
[9] D.A. Ham, L. Mitchell, A. Paganini, F. Wechsung, Automated shape differentiation in the unified form language, Struct. Multidiscip. Optim. 60 (5) (2019) 1813–1820, https://doi.org/10.1007/s00158-019-02281-z.
[10] P. Gangl, K. Sturm, M. Neunteufel, J. Schöberl, Fully and semi-automated shape differentiation in NGSolve, Struct. Multidiscip. Optim. 63 (3) (2021) 1579–1607, https://doi.org/10.1007/s00158-020-02742-w.
[11] C. Bauer, A. Frink, R. Kreckel, Introduction to the GiNaC framework for symbolic computation within the C++ programming language, J. Symb. Comput. 33 (1) (2002) 1–12, https://doi.org/10.1006/jsco.2001.0494.
[12] C. Diddens, P.J. Dekker, D. Lohse, Non-monotonic surface tension leads to spontaneous symmetry breaking in a binary evaporating drop, preprint, arXiv: 2402.17452, 2024, https://arxiv.org/abs/2402.17452.
[13] S. Hartmann, C. Diddens, M. Jalaal, U. Thiele, Sessile drop evaporation in a gap - crossover between diffusion-limited and phase transition-limited regime, J. Fluid Mech. 960 (2023) A32, https://doi.org/10.1017/jfm.2023.176.
[14] M. Jalaal, B. ten Hagen, H. le The, C. Diddens, D. Lohse, A. Marin, Interfacial aggregation of self-propelled janus colloids in sessile droplets, Phys. Rev. Fluids 7 (11) (2022) 110514, https://doi.org/10.1103/PhysRevFluids.7.110514.
[15] L. Thayyil Raju, C. Diddens, Y. Li, A. Marin, M.N. van der Linden, X. Zhang, D. Lohse, Evaporation of a sessile colloidal water-glycerol droplet: Marangoni ring formation, Langmuir 38 (39) (2022) 12082–12094, https://doi.org/10.1021/acs.langmuir.2c01949.
[16] R.T. van Gaalen, H. Wijshoff, J.G. Kuerten, C. Diddens, Competition between thermal and surfactant-induced Marangoni flow in evaporating sessile droplets, J. Colloid Interface Sci. 622 (2022) 892–903, https://doi.org/10.1016/j.jcis.2022.04.146.
[17] A.B. Thompson, A. Juel, A.L. Hazel, Multiple finger propagation modes in Hele-Shaw channels of variable depth, J. Fluid Mech. 746 (2014) 123–164, https://doi.org/10.1017/jfm.2014.100.
[18] A. Franco-Gómez, A.B. Thompson, A.L. Hazel, A. Juel, Bubble propagation on a rail: a concept for sorting bubbles by size, Soft Matter 13 (2017) 8684–8697, https://doi.org/10.1039/C7SM01478C.
[19] J.S. Keeler, A.B. Thompson, G. Lemoult, A. Juel, A.L. Hazel, The influence of invariant solutions on the transient behaviour of an air bubble in a Hele-Shaw channel, Proc., Math. Phys. Eng. Sci. A 475 (2232) (2019) 20190434, https://doi.org/10.1098/rspa.2019.0434.
[20] A.B. Thompson, Bifurcations of drops and bubbles propagating in variable-depth Hele-Shaw channels, J. Eng. Math. 129 (1) (Jul. 2021), https://doi.org/10.1007/s10665-021-10146-y.
[21] A. Gaillard, J.S. Keeler, G.L. Lay, G. Lemoult, A.B. Thompson, A.L. Hazel, A. Juel, The life and fate of a bubble in a geometrically perturbed Hele-Shaw channel, J. Fluid Mech. 914 (Mar. 2021), https://doi.org/10.1017/jfm.2020.844.
[22] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, Elsevier, 2005.
[23] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, vol. 31, Cambridge University Press, 2002.
[24] M. Heil, A.L. Hazel, J. Boyle, Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches, Comput. Mech. 43 (2008) 91–101, https://doi.org/10.1007/s00466-008-0270-6.
[25] Y.A. Kuznetsov, Elements of Applied Bifurcation Theory, Springer, 1998.
[26] K. Cliffe, S. Spence, A. Tavener, The numerical analysis of bifurcation problems with application to fluid mechanics, Acta Numer. 9 (2000) 39–131, https://doi.org/10.1017/S0962492900000398.
[27] D. Bindel, M. Friedman, W. Govaerts, J. Hughes, Y. Kuznetsov, Numerical computation of bifurcations in large equilibrium systems in Matlab, J. Comput. Appl. Math. 261 (2014) 232–248, https://doi.org/10.1016/j.cam.2013.10.034.
[28] J.S. Umbría, M. Net, Numerical continuation methods for large-scale dissipative dynamical systems, Eur. Phys. J. Spec. Top. 225 (13–14) (2016) 2465–2486.
[29] A.L. Hazel, Spatial and temporal adaptivity in numerical studies of instabilities, with applications to fluid flows, in: A. Gelfgat (Ed.), Computational Modelling of Bifurcations and Instabilities in Fluid Dynamics, Springer International Publishing, Cham, 2019, pp. 75–115.
[30] G. Moore, A. Spence, The calculation of turning points of nonlinear equations, SIAM J. Numer. Anal. 17 (4) (1980) 567–576.
[31] S.H. Strogatz, Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering, CRC Press, 2018.
[32] S.J. Tavener, Symmetric and nonsymmetric equilibria of a rod-and-spring model, IMA J. Appl. Math. 49 (1) (1992) 73–102, https://doi.org/10.1093/imamat/49.1.73.
[33] A. Griewank, G. Reddien, The calculation of Hopf points by a direct method, IMA J. Numer. Anal. 3 (3) (1983) 295–303, https://doi.org/10.1093/imanum/3.3.295.
[34] P.H. Hoffmann, A Hitchhiker's guide to automatic differentiation, Numer. Algorithms 72 (3) (2016) 775–811, https://doi.org/10.1007/s11075-015-0067-6.
[35] A. Logg, K.-A. Mardal, G. Wells, Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, vol. 84, Springer Science & Business Media, 2012.
[36] B. Roman, C. Gay, C. Clanet, Pendulums, drops and rods: a physical analogy, preprint, arXiv:2006.02742, 2020, https://arxiv.org/abs/2006.02742.
[37] J. Eggers, Nonlinear dynamics and breakup of free-surface flows, Rev. Mod. Phys. 69 (1997) 865–930, https://doi.org/10.1103/RevModPhys.69.865.
[38] A. Kumar, M.R. Gunjan, K. Jakhar, A. Thakur, R. Raj, Unified framework for mapping shape and stability of pendant drops including the effect of contact angle hysteresis, Colloids Surf. A, Physicochem. Eng. 597 (2020) 124619, https://doi.org/10.1016/j.colsurfa.2020.124619.
[39] A. Kumar, M. Gunjan, R. Raj, On the validity of force balance models for predicting gravity-induced detachment of pendant drops and bubbles, Phys. Fluids 32 (2020) 101703, https://doi.org/10.1063/5.0025488.

[40] W.-Y. Shi, K.-Y. Tang, J.-N. Ma, Y.-W. Jia, H.-M. Li, L. Feng, Marangoni convection instability in a sessile droplet with low volatility on heated substrate, Int. J. Therm. Sci. 117 (2017) 274–286, https://doi.org/10.1016/j.ijthermalsci.2017.04.007.
[41] S. Karpitschka, F. Liebig, H. Riegler, Marangoni contraction of evaporating sessile droplets of binary mixtures, Langmuir 33 (19) (2017) 4682–4687, https://doi.org/10.1021/acs.langmuir.7b00740.
[42] E. Yim, P. Billant, Analogies and differences between the stability of an isolated pancake vortex and a columnar vortex in stratified fluid, J. Fluid Mech. 796 (2016) 732–766, https://doi.org/10.1017/jfm.2016.248.
[43] E. Yim, A. Bouillant, D. Quéré, F. Gallaire, Leidenfrost flows: instabilities and symmetry breakings, Flow 2 (2022) E18, https://doi.org/10.1017/flo.2022.5.
[44] L. Babor, H.C. Kuhlmann, Linear stability of thermocapillary flow in a droplet attached to a hot or cold substrate, Phys. Rev. Fluids 8 (2023) 114003, https://doi.org/10.1103/PhysRevFluids.8.114003.
[45] H. Liu, J. He, Z. Zeng, Z. Qiu, Instabilities of thermocapillary-buoyancy flow in a rotating annular pool for medium-Prandtl-number fluid, Phys. Rev. E 104 (2021), https://doi.org/10.1103/PhysRevE.104.035101.
[46] G.K. Batchelor, A.E. Gill, Analysis of the stability of axisymmetric jets, J. Fluid Mech. 14 (4) (1962) 529–551, https://doi.org/10.1017/S0022112062001421.
[47] K. Borońska, L.S. Tuckerman, Standing and travelling waves in cylindrical Rayleigh–Bénard convection, J. Fluid Mech. 559 (2006) 279–298, https://doi.org/10.1017/S0022112006000309.
[48] A. Gelfgat, Z. Bar-Yoseph, A. Solan, T. Kowalewski, An axisymmetry-breaking instability of axially symmetric natural convection, Int. J. Transp. Phenom. 1 (1999) 173–190.
[49] B.J. Lowry, P.H. Steen, Capillary surfaces: stability from families of equilibria with application to the liquid bridge, Proc., Math. Phys. Eng. Sci. 449 (1937) (1995) 411–439.
[50] A.L. Hazel, M. Heil, Surface-tension-induced buckling of liquid-lined elastic tubes: a model for pulmonary airway closure, Proc. R. Soc. A, Math. Phys. Eng. Sci. 461 (2058) (2005) 1847–1868, https://doi.org/10.1098/rspa.2005.1453.
[51] J. Bostwick, P. Steen, Stability of constrained capillary surfaces, Annu. Rev. Fluid Mech. 47 (1) (2015) 539–568, https://doi.org/10.1146/annurev-fluid-010814-013626.
[52] L.A. Slobozhanin, J.I.D. Alexander, A.H. Resnick, Bifurcation of the equilibrium states of a weightless liquid bridge, Phys. Fluids 9 (7) (1997) 1893–1905.
[53] Y. Li, C. Diddens, A. Prosperetti, K.L. Chong, X. Zhang, D. Lohse, Bouncing oil droplet in a stratified liquid and its sudden death, Phys. Rev. Lett. 122 (15) (2019) 154502.
[54] Y. Li, C. Diddens, A. Prosperetti, D. Lohse, Marangoni instability of a drop in a stably stratified liquid, Phys. Rev. Lett. 126 (12) (2021) 124502.
[55] Y. Li, J.G. Meijer, D. Lohse, Marangoni instabilities of drops of different viscosities in stratified liquids, J. Fluid Mech. 932 (2022) A11.
[56] J.G. Meijer, Y. Li, C. Diddens, D. Lohse, On the rising and sinking motion of bouncing oil drops in strongly stratified liquids, J. Fluid Mech. 966 (2023) A14.
[57] M.A. Herrada, J.M. Montanero, L. Carrión, Dynamics of a silicone oil drop submerged in a stratified ethanol-water bath, Phys. Rev. E 108 (2023) 065104, https://doi.org/10.1103/PhysRevE.108.065104.
[58] C. Diddens, H. Tan, P. Lv, M. Versluis, J. Kuerten, X. Zhang, D. Lohse, Evaporating pure, binary and ternary droplets: thermal effects and axial symmetry breaking, J. Fluid Mech. 823 (2017) 470–497, https://doi.org/10.1017/jfm.2017.312.
[59] K. Sefiane, J.R. Moffat, O.K. Matar, R.V. Craster, Self-excited hydrothermal waves in evaporating sessile drops, Appl. Phys. Lett. 93 (7) (2008) 074103, https://doi.org/10.1063/1.2969072.
[60] A. Gauthier, C. Diddens, R. Proville, D. Lohse, D. van Der Meer, Self-propulsion of inverse Leidenfrost drops on a cryogenic bath, Proc. Natl. Acad. Sci. 116 (4) (2019) 1174–1179.
[61] C.C. Maass, C. Krüger, S. Herminghaus, C. Bahr, Swimming droplets, Annu. Rev. Condens. Matter Phys. 7 (1) (2016) 171–193, https://doi.org/10.1146/annurev-conmatphys-031115-011517.
[62] S. Michelin, Self-propulsion of chemically active droplets, Annu. Rev. Fluid Mech. 55 (1) (2023) 77–101, https://doi.org/10.1146/annurev-fluid-120720-012204.
[63] A.M.J. Edwards, E. Ruiz-Gutiérrez, M.I. Newton, G. McHale, G.G. Wells, R. Ledesma-Aguilar, C.V. Brown, Controlling the breakup of toroidal liquid films on solid surfaces, Sci. Rep. 11 (1) (2021) 8120, https://doi.org/10.1038/s41598-021-87549-5.