

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MASTER THESIS SPRING 2023

MASTER IN COMPUTATIONAL SCIENCE AND ENGINEERING

Regularized Four-Sided Cavity Flows: A Spectral Bifurcation Benchmark implemented in Julia

Author:

Moritz WALDLEBEN

Supervisor at host university:

Álvaro MESEGUER

Universitat Politècnica Catalunya, Barcelona

Supervisor:

François GALLAIRE

École polytechnique fédérale de Lausanne

EPFL

Abstract

This master project explores a regularized version of the four-sided lid-driven cavity for incompressible fluids to be used as a validator benchmark for Navier-Stokes solvers. The four-sided driven cavity is an extension of the simple one-sided lid-driven case, where all lids move with the same velocity profile and parallel lids move in opposite directions. This study focuses on a regularized version to overcome corner singularities. The considered method recovers exponential convergence with a pseudo-spectral Chebyshev discretization scheme. The flow presents a variety of bifurcation scenarios and could make up for an amenable Navier-Stokes bifurcation benchmark. The algorithms are implemented in Julia, a high-performance language for scientific computing. A developed Julia module provides a reproducible example for the proposed problem.

Acknowledgements

First and foremost, I wish to express my sincere appreciation to my supervisors, Álvaro Meseguer, Arantxa Alonso, and Oriol Batiste, at the Universitat Politècnica Catalunya in Barcelona. I am grateful for their sharing of expertise, patients, and guidance throughout my master's project. A thank you as well to the Nonlinear Fluid Mechanics group.

I would also like to thank my supervisor François Gallaire at EPFL, for supporting me doing my master's project abroad and his valuable feedback.

Finally, I am deeply grateful to my family and friends for their love and support during this process.

Contents

1	Introduction	1
2	2D Driven Cavity Flows	3
2.1	The Single Lid-Driven Cavity Flow	3
2.2	The Four-Sided Lid-Driven Cavity Flow	4
3	Governing Equations	6
4	Regularization	8
4.1	The Regularized Four-Sided Lid-Driven Cavity Flow (R4CF)	8
4.2	A Physical or Mathematical Problem?	9
5	Spatial Discretization	10
5.1	Spectral Methods	10
5.2	Spectral Convergence Rate and Singularities	11
5.3	Pseudo-Spectral Discretization	11
5.4	Chebyshev Differentiation Matrix	11
5.5	2D Chebyshev Discretization	12
5.6	Incorporating the Boundary Conditions	13
6	Temporal Discretization and Solvers	15
6.1	Implicit Euler Method	15
6.2	Streamfunction Time-Stepping Equation	15
6.3	Newton-Raphson Method	16
6.4	Preliminary Results with Time Integration	16
7	Bifurcation Analysis and Dynamical Systems	17
7.1	The Dynamical Systems Perspective	17
7.2	Steady-State Flows	18
7.3	Linear Stability Analysis	18
7.4	Bifurcations	19
7.5	Continuation Algorithms	20
8	Implementation Aspects	21
8.1	The Julia Programming Language	21
8.2	Implementation of the Nonlinear Function	21
8.3	Comparison between MATLAB and Julia	23
9	Results and Discussion	24
9.1	Outline of Bifurcation Diagram	24
9.2	Hopf Bifurcation and Time-Periodic Flows	24
9.3	Pitchfork Bifurcation and New Asymmetric Solutions	27
10	Conclusion and Future Work	30
	Appendix A Implementation of Boundary Conditions in Julia	34
	Appendix B Abstract Conference Presentation at ICTW 2023	35
	Appendix C Abstract Conference Presentation at SFMC 2023	36

1 Introduction

Driven cavity flows have long served as essential benchmarks for validating Navier-Stokes solvers. These problems can test spatial discretization methodologies such as finite elements, finite differences, and spectral methods. They may also be used to assess a variety of boundary condition implementations and time-stepping schemes.

Current theoretical and computational research into Navier-Stokes flows goes beyond the time integration of the equations of motion. Since the publication of the seminal work by Eberhard Hopf 75 years ago (Hopf, 1948), bifurcation and dynamical systems theory have been vindicated as suitable deterministic frameworks to understand hydrodynamic instabilities and the transition to turbulent flow regimes.

Exploring the infinite-dimensional phase space of the Navier-Stokes equations is necessary to anticipate instabilities and comprehend the emergence of new flow states. That implies the seek for invariant sets or manifolds, the simplest cases being equilibria (steady flows) and periodic orbits (time-periodic flows). Although dynamically of high relevance, these manifolds are often linearly unstable and, therefore, systematically overlooked by time integrators of the Navier-Stokes equations, as time-stepping will only approach local attractors.

Accurately computing steady or time-periodic flows, regardless of their stability, requires the implementation of suitable continuation algorithms to monitor them as the Reynolds number is varied (Kuznetsov, 2004). Moreover, the linear stability analysis of these invariant flows is crucial for predicting potential local bifurcations in parameter space. However, this analysis poses challenges due to the high-dimensional generalized eigenvalue problem resulting from the local linearization of the Navier-Stokes operator.

Numerically reliable Navier-Stokes benchmarks require high accuracy and robustness of its solutions. It is well known that some spatial numerical discretizations of the Navier-Stokes equations may lead to spurious solutions that typically appear for coarse grids, eventually vanishing when the spatial resolution is increased accordingly. Although there are many different spatial discretization techniques available for Navier-Stokes flows, spectral methods have long been known for providing unbeatable exponential convergence, provided the domain is simple, and the boundary conditions are mathematically well-posed.

For historical reasons, the one-sided version has received more attention among the wide variety of two-dimensional lid-driven cavity flows (Kuhlmann and Romanò, 2019). This cavity consists of a square with one lid sliding at a uniform speed, and the three other lids are kept stationary. In this case, the flow is steady and stable even for high Reynolds numbers. The first instability found is due to a Hopf bifurcation which may lie somewhere in the interval $[7500, 8100]$ reported by different studies (Kuhlmann and Romanò, 2019). This uncertainty in the reported critical Reynolds number is due to multiple factors. For instance, the large Reynolds number where the instability appears demands very high resolutions in space and time, as well as the different spatial discretization techniques and methodologies used in the works to identify the bifurcation (time integration or linear stability analysis). Due to its simple geometry, the square lid-driven cavity has long been qualified as a good candidate for a canonical benchmark problem. Still, lid-driven cavity flow problems, in their original formulations, typically have singular boundary conditions at the corners, where the velocity profile is discontinuous. In this case, the exponential convergence of spectrally approximated flow solutions is at stake.

The corner-singularity problem has, however, been successfully circumvented using different approaches, such as subtraction methods (Botella and Peyret, 1998), which eliminates the dominant terms of the asymptotic expansion of the solution to the Navier-Stokes equations near the corners, thus recovering spectral accuracy. Another successful strategy to preserve exponential convergence when using spectral methods in lid-driven cavity flows is the regularization of the boundary conditions. By replacing the discontinuous profiles at the walls with polynomial or exponential velocity distributions, the original problem can be mimicked while preserving smoothness at the corners (Shen, 1991; Lopez et al., 2017).

A more recent variant, termed the four-sided cavity flow, has been proposed (Wahba, 2009). Here the four lids are moving at the same speed (top-bottom and right-left lids moving rightwards-leftwards and upwards-downwards, respectively). This cavity has the computational advantage of exhibiting

a variety of bifurcations at low or moderate Reynolds numbers. However, the problem suffers from corner singularities due to the discontinuous boundary conditions. This work addresses this issue by investigating a regularized version of the four-sided cavity flow using a Chebyshev discretization implemented in Julia, a high-performance programming language.

Julia, a free and open-source language for scientific computing, offers performance comparable to compiled C/Fortran codes, making it an attractive platform for implementing scientific computing. A developed Julia module provides a reproducible example of the regularized cavity.

The regularized four-sided lid-driven cavity shows most of the primary bifurcation scenarios. The flow undergoes instabilities, such as pitchfork, saddle-node (fold), and Hopf. Determining the precise location of the bifurcations could present an amenable benchmark when testing and comparing different discretization schemes and implementations.

This master project is structured as follows. First, a review of the driven cavity flow problem is provided. Then the four-sided lid-driven cavity flow is introduced, and the governing equations are presented, followed by an introduction to the regularized version. The necessary theory is developed in the theoretical part regarding spatial (spectral Chebyshev) and time discretization. The numerical tools required for bifurcation analysis of the flow are also explained. The subsequent step focuses on implementation details for the Julia programming language. Finally, the results of the regularized problem are presented and discussed, including stable and unstable solutions and the precise occurrences of bifurcations. The study concludes with a summary of the objectives, limitations, and recommendations for future research.

2 2D Driven Cavity Flows

In two-dimensional (2D) driven cavity flows, a viscous Newtonian incompressible fluid is confined within a square or rectangular domain. In these model problems, the fluid is driven by the motion of the bounding walls. The resulting types of flow configurations have been widely used due to their simplicity and relevance as benchmark problems for validating numerical solvers and investigating various flow phenomena.

A recent and comprehensive review conducted by Kuhlmann and Romanò (2019) provides insights into the extensive research carried out on the many variants of the lid-driven cavity.

2.1 The Single Lid-Driven Cavity Flow

The most basic configuration involves a square cavity where only one wall, referred to as the lid, is in motion while the other walls remain stationary. This simple 2D case has been extensively studied numerically and serves as one of the canonical benchmark problems for numerical Navier-Stokes solvers nowadays. The first thorough computational investigation lies more than half a century back (Burggraf, 1966). Figure 1 shows the basic setup, and figure 2 illustrates typical flow patterns for two different Reynolds numbers. The depicted streamlines show a central vortex driven by the upper lid's motion to the right. Therefore, the primary vortex rotates clockwise. Moreover, the flow shows secondary vortices at the upper and lower corners at higher Reynolds numbers.

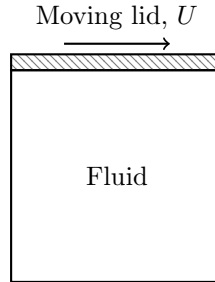


Figure 1: The lid-driven cavity with a tangential velocity U at the top

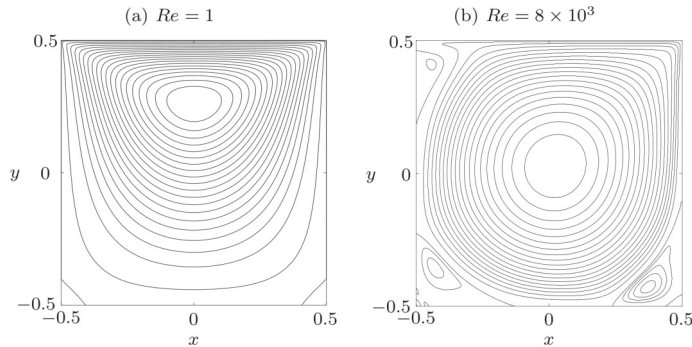


Figure 2: Isolines of the streamfunction of the lid-driven cavity at Reynolds numbers 1 (a) and 8000 (b), figures adapted from Kuhlmann and Romanò (2019)

The flow observed in the upper corners is a result of the discontinuous boundary conditions in the mathematical problem and is caused by the horizontal component being set to a velocity U at the upper corners, whereas the vertical walls are not moving. As a consequence, the pressure and the vorticity diverge at these locations. This phenomenon is a special case of the Taylor's scrapper problem, and there exist closed-form solutions in terms of series expansions (Kuhlmann and Romanò, 2019). The other effect worth mentioning is the formation of two eddies at the lower corners of the

cavity. These vortices rotate in an anti-clockwise way, separated from the central vortex. They are called Moffat eddies, and similarity solutions have been found for them (Moffatt, 1964).

Apart from these studies, calculating the solutions of the cavity while varying the Reynolds number has been done extensively. The single lid-driven cavity becomes unstable at a Hopf bifurcation (see section 7.4) somewhere in the range of the Reynolds number [7500, 8100] as reported by different studies (Kuhlmann and Romanò, 2019). No conclusive results could be drawn due to significant uncertainty. The numerical variations are primarily the result of the large Reynolds number at which the bifurcations appear and the different discretizations used when solving the Navier-Stokes equations numerically. As mentioned before, the other issue is related to the corner singularities caused by the discontinuous boundary conditions.

Until now, we have restricted our attention only to the 2D single lid-driven case. However, there is a whole spectrum of different variants. The analysis can be performed for different aspect ratios (height-width ratio of the box) and one can evaluate the effect on the flow pattern. Shapes other than a rectangular cavity can be considered as well. With more computing resources being available, the 3D variant has also been studied, giving rise to other types of flows and singularities (Lopez et al., 2017). Further, the problem can be combined with heat convection by keeping facing sides at different temperatures (Koseff and Street, 1985).

In this work, we want to consider another type of flow variant by adapting the boundary conditions. Kuhlmann et al. (1997) investigated a two-sided flow problem numerically and experimentally where two facing lids move in opposite directions. It has been found that a primary symmetric flow pattern is stable for low Reynolds numbers. Then the flow transitions to non-unique patterns and becomes unstable for higher Reynolds numbers. Wahba (2009) did further numerical experiments in a modified non-facing version of the two-sided cavity problem. Moreover, an extension was proposed, the four-sided version, where all four walls are moving, having the same tangential speeds but in opposite directions. Figure 3 illustrates the different variants for the boundary conditions.

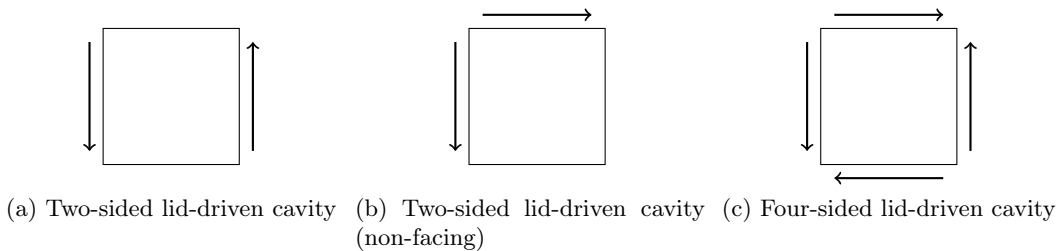


Figure 3: Variations of the boundary conditions for the cavity flow, with the moving lids set to an equal tangential speed U

2.2 The Four-Sided Lid-Driven Cavity Flow

We will focus on the four-sided version of the cavity flow problem. In this configuration, the top and bottom lids move in the rightward and leftward directions, while the right and left lids move in the upward and downward directions, respectively. The dimensional quantities that characterize this cavity flow are depicted in figure 4.

The study by Wahba (2009) reveals the existence of multiple steady solutions and shows an interesting flow bifurcation at a low to moderate Reynolds number. Further, this bifurcation happens at an even lower Reynolds number compared to the two-sided versions. Initially, the global flow pattern exhibits symmetry, and then two stable asymmetric solutions emerge, whereas the symmetric solution becomes unstable. The symmetric state consists of four coexisting vortices, which eventually collapse into two different asymmetric states of central primary rotations and two secondary vortices. Figure 5 provides a visualization of these states and the rotation direction of the resulting vortices. The numerical simulations were carried out on a finite-difference grid using the line successive over-relaxation method (LSOR).

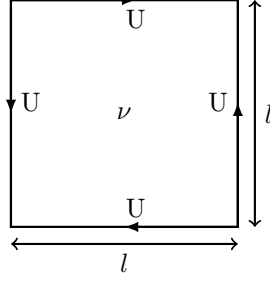


Figure 4: The four-sided cavity ($l \times l$) with tangential velocities U and kinematic viscosity ν

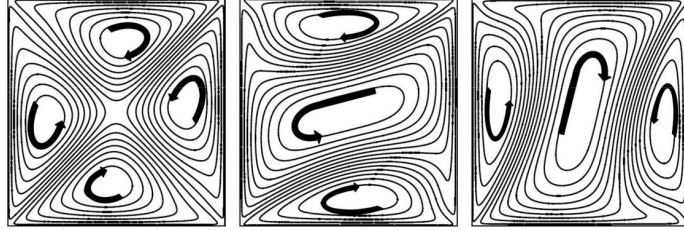


Figure 5: Symmetric and asymmetric solutions at Reynolds number $Re = 250$, computed by Chen et al. (2013)

Later, Arumuga Perumal and Dass (2011) employed a Lattice Boltzmann Method (LBM), recovering the multiple solutions previously obtained by the former finite-difference-based code. Furthermore, other aspect ratios for the cavity have been considered, which exhibit the multiplicity of solutions as well.

Cadou et al. (2012) developed numerical tools for the stability analysis of the two- and four-sided cavities. They identified a secondary bifurcation point of the unstable symmetric solution at a higher Reynolds number for the two-sided non-facing and the four-sided versions. Moreover, a Hopf bifurcation has been discovered for the two-sided non-facing boundary conditions. On the other hand, in the four-sided version, no such bifurcation has been reported in the literature.

A full bifurcation diagram (figure 6) was presented by Chen et al. (2013), which plots the stream-function value at the center of the cavity against an increasing Reynolds number. Through a continuation algorithm, the asymmetric solution branches could be followed (see section 7.5) and are visualized in the diagram. The solid curves represent stable solutions, whereas the dashed curves correspond to unstable ones. One notices that the effect of changing the Reynolds number in the four-sided version reveals two pitchfork bifurcations and saddle-node (fold) points for the asymmetric solutions.

The bifurcation diagram generally illustrates that this could be an ideal bifurcation benchmark. The Reynolds numbers are within a low to moderate range, making it feasible from a computational point of view. The precise detection of the two asymmetric solutions can be used to assess the accuracy of a Navier-Stokes solver under test.

However, it has to be mentioned that the bifurcation diagram was obtained using a second-order finite differences scheme. As pointed out, the goal here is to look at a regularized problem with the aim of applying a spectral discretization and accurately resolving the locations of the bifurcations. To do so, the governing equations will now be introduced formally, and mathematical well-posed boundary conditions will be defined to achieve higher accuracy and an exponential convergence rate.

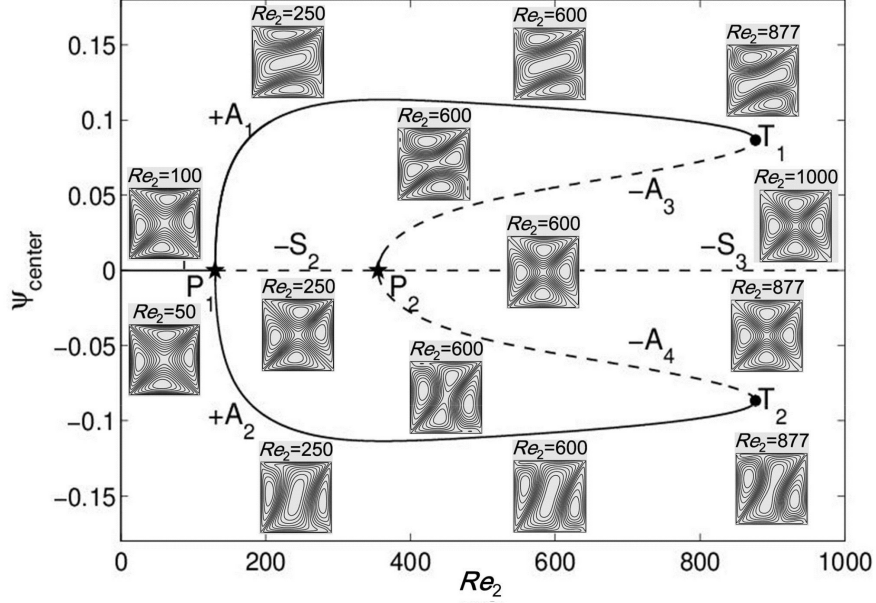


Figure 6: Bifurcation diagram obtained by Chen et al. (2013) for the four-sided cavity flow, Reynolds numbers are double the results obtained in this study

3 Governing Equations

The following presents an overview of the physical and mathematical framework for the 2D cavity flow problem introduced in section 2. The aim is to introduce the theory of the Navier-Stokes equations and the streamfunction formulation.

An incompressible Newtonian fluid in a domain Ω is governed by the Navier-Stokes equations which are

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where \mathbf{u} denotes the velocity vector (2D or 3D). The variable p is the pressure, ρ is the constant fluid density, ν is the kinematic viscosity and \mathbf{g} defines the body acceleration acting on the fluid. The first equation represents the conservation of momentum, while the second is the continuity equation. Furthermore, boundary conditions must be imposed on the domain Ω for these equations.

For cavity flow problems in a plane, we consider the 2D form of the above equations without body acceleration. They can be written explicitly for the two spatial components as

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (3)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \quad (4)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \quad (5)$$

Here, u and v denote the velocities in the x and y directions, respectively. Equations (3) to (5) build the underlying framework for the analysis of the 2D cavity flow.

In the case of a 2D incompressible fluid, it is possible to introduce a scalar function $\Psi(x, y, t)$ called the streamfunction which is defined such that,

$$u = \frac{\partial \Psi}{\partial y}, \quad (6)$$

$$v = -\frac{\partial \Psi}{\partial x}. \quad (7)$$

By its definition, the streamfunction satisfies the continuity equation and therefore the incompressibility condition. Regarding the momentum equations, the expressions (6) and (7) can be used to obtain a formulation of the Navier-Stokes where the pressure can be eliminated and thus only involves the streamfunction (Landau and Lifshitz, 1987):

$$\partial_t \Delta \Psi = \nu \Delta^2 \Psi + (\partial_x \Psi) \partial_y (\Delta \Psi) - (\partial_y \Psi) \partial_x (\Delta \Psi) \quad (8)$$

For a shorter notation, the partial derivatives of the streamfunction are denoted as subscripts in the equation above and from now on. It is important to note that $\Psi = \text{constant}$ represents the family of curves of the streamlines (Landau and Lifshitz, 1987). Hence, if we know the streamfunction, we can visualize the streamlines by setting the function to different constant values.

Viewing a non-dimensional version of the problem is standard for analyzing the cavity flow (figure 4). The relations (9) show the characteristic scales of the 2D cavity. We will use the same length scale l for both sides. Furthermore, as the magnitude for our regularized boundary conditions is set to be the equal for all lids, we can use the same velocity scale U .

$$\begin{aligned} [l] &= \text{length} \\ [U] &= \text{length} * \text{time}^{-1} \\ [\Psi] &= \text{length}^2 * \text{time}^{-1} \\ \left[\frac{l}{U} \right] &= \text{time} \quad (\text{dynamic time}) \end{aligned} \quad (9)$$

All parameters can now be made dimensionless by defining $x = lx^*$, $y = ly^*$, $t = \frac{l}{U} t^*$, $\Psi = lU\Psi^*$. Additionally, the non-dimensional operators are scaled as $\partial_t = \frac{U}{l} \partial_{t^*}$, $\partial_x = \frac{1}{l} \partial_{x^*}$, $\partial_y = \frac{1}{l} \partial_{y^*}$ and $\Delta = \frac{1}{l^2} \Delta_*$. By plugin these definitions into (11), simplifying and dividing by $\frac{U^2}{l^2}$, we get,

$$\partial_{t^*} \Delta_* \Psi^* = \frac{\nu}{U l} \Delta_*^2 \Psi^* + (\partial_{x^*}^* \Psi^*) \partial_{y^*}^* (\Delta_* \Psi^*) - (\partial_{y^*}^* \Psi^*) \partial_{x^*}^* (\Delta_* \Psi^*). \quad (10)$$

We notice that it is possible to recover the Reynolds number, $\text{Re} = \frac{Ul}{\nu}$ that, as a non-dimensional parameter, characterizes the relative importance of inertial forces and viscous forces in a flow. For clarity, we will omit $*$ notation. All quantities will correspond to the dimensionless variables. The final equation reads

$$\partial_t \Delta \Psi = \frac{1}{\text{Re}} \Delta^2 \Psi + (\partial_x \Psi) \partial_y (\Delta \Psi) - (\partial_y \Psi) \partial_x (\Delta \Psi). \quad (11)$$

This equation (11) will be the foundation for the numerical investigation later on. The equation is non-dimensional and only depends on the scalar-valued streamfunction and the Reynolds number. It is a nonlinear ordinary differential equation of order 4, where we can analyze different dynamics and regimes by changing the Reynolds number.

4 Regularization

We have seen that the boundary conditions are discontinuous at the top corners of the single lid-driven cavity. In the four-sided version, there even exist discontinuities at all four corners. The corners are referred to as singular as the pressure and velocity diverge. These diverging values pose a well-known problem for discretization schemes of higher order. In the simple lid-driven cavity Botella and Peyret (1998) used spectral methods (see next section 5) in combination with the subtraction of the leading terms of the singularities. In this way, the numerically solved problem does not involve singularities, and the global discretization of spectral methods can safely be applied and gives highly accurate results.

Another successful strategy is to adapt the boundary condition to study a more well-posed problem. Regularized versions of the boundary conditions are used to replace discontinuous velocity profiles. The velocity is set to zero at the corners and increases to the desired value at the boundary. Polynomials (Shen, 1991) and trigonometric functions have been tested in the literature to construct such regularization functions. Shen (1991) used 4th-order polynomials, which seem too low of an order and give different results than the original problem. A more recent regularization function involves exponentials (Lopez et al., 2017), which was applied to a 3D problem and seems to work well to represent the constant velocity at the lid and still be a smooth function at all the corners.

Resembling the original problem through regularized functions can recover the exponential convergence of spectral discretizations. The resolution of the bifurcation points can thus be calculated more reliably.

4.1 The Regularized Four-Sided Lid-Driven Cavity Flow (R4CF)

The domain we will use for the regularized four-sided cavity flow problem (R4CF) is illustrated in figure 7. To distinguish between different states, it is useful to monitor not only the streamfunction value at the center but also the horizontal and vertical velocities at the left and top midpoints of the domain. The motivation for using a normalized domain of $[-1, 1] \times [-1, 1]$ centered at the origin instead of the classical on $[0, 1] \times [0, 1]$, is that, on the one hand, the reflectional symmetries of the flow are more natural, just implying a change of sign of the spatial variables x or y . On the other hand, the $[-1, 1]$ interval is the canonical domain for orthogonal polynomials.

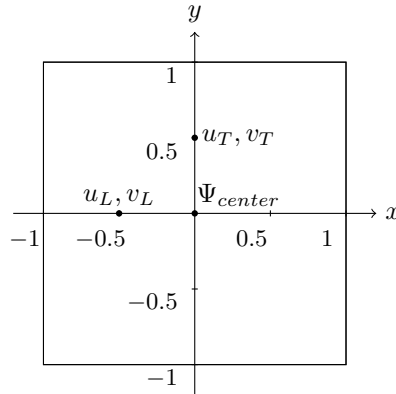


Figure 7: Square domain $(x, y) \in [-1, 1] \times [-1, 1]$ for the four-sided cavity

As shown in figure 4 of the four-sided cavity flow, all lids have the same tangential velocity. The original discontinuous boundary conditions are

$$u(x, \pm 1, t) = \pm 1, \quad (12)$$

$$v(x, \pm 1, t) = 0, \quad (13)$$

$$u(\pm 1, y, t) = 0, \quad (14)$$

$$v(\pm 1, y, t) = \pm 1. \quad (15)$$

To discretize (11) using a spectral method, (12) and (15) are replaced by the regularized boundary conditions, which impose a zero velocity at all four corners of the cavity:

$$u(x, \pm 1, t) = \pm \left[\left(e^{k_0(x-1)} - 1 \right) \left(e^{-k_0(x+1)} - 1 \right) \right]^2 \quad (16)$$

$$v(\pm 1, y, t) = \pm \left[\left(e^{k_0(y-1)} - 1 \right) \left(e^{-k_0(y+1)} - 1 \right) \right]^2 \quad (17)$$

The exponential regularization function is adapted from Lopez et al. (2017). Where k_0 is a suitable parameter controlling the slope of the decay of the velocity profile near the corners. In this study, we set $k_0 = 10$ to get a suitable but not too extreme decay for the velocity at the edges. Figure 8 shows such a profile.

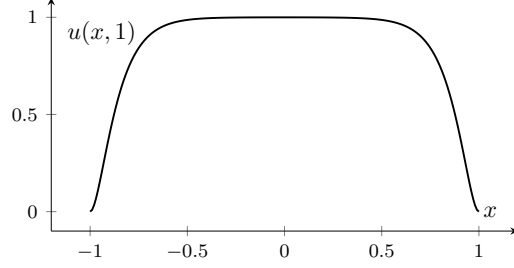


Figure 8: Regularized horizontal velocity profile (16) at the top wall $y = 1$ for $k_0 = 10$

In order to compare the results of this work with the previous literature, the Reynolds number has to be scaled. The $[0, 1] \times [0, 1]$ domain used in Wahba (2009) and the subsequent studies mean that reference length l_{ref} is half the length l used here. Thus the relation of the Reynolds numbers is

$$\text{Re} = \frac{U \cdot l}{\nu} = \frac{s U_{ref} \cdot 2 l_{ref}}{\nu} = 2s \cdot \text{Re}_{ref}. \quad (18)$$

This implies a scaling factor of $s = \frac{1}{2}$, and the Reynolds number of previous studies has to be divided by 2. In all further comparisons of results, the reference Reynolds numbers will be scaled to the presented symmetrical domain.

4.2 A Physical or Mathematical Problem?

The final important point to make is that the 2D problem must be studied cautiously. The paper of Erturk (2009) concludes that the single lid-driven cavity represents a *fictitious* flow at high Reynolds numbers. Fictitious meaning that the problem from a physical point of view would no longer be purely 2D because instabilities in the third dimension would influence the flow. However, numerical solutions exist at high Reynolds numbers for the two-dimensional case.

Furthermore, it is hard to replicate the lid-driven cavity experimentally with regularized boundary conditions. This problem gets even worse considering the four-sided cavity flow and the different velocity directions at all lids.

Using a regularized version, as presented above, will help to obtain numerically accurate results, but it makes it even harder to reproduce in practice. From the point of view of a benchmark for an instability analysis for different solvers, the resulting flow (whether fictional or not) should at least be precisely computable. This study will focus on this main aspect, that is the accurate computation of a well-posed mathematical problem.

5 Spatial Discretization

In order to numerically solve the R4CF problem, we need to approximate the spatial domain of the continuous problem. This is typically done by dividing the domain into grid points forming what is called a mesh. Various discretization techniques and numerical methods exist for solving partial differential equations.

The most established ones are finite elements, finite differences, or finite volume for 3D discretizations. These discretization techniques are referred to as local techniques because the values and derivatives of a grid point only affect other nearby points. They employ local basis functions to approximate the solution and offer flexibility in handling irregular and complex geometries. These techniques are widely adopted in engineering due to their practical implementation and adaptive refinement capabilities. Another class of techniques is called spectral methods. These methods act globally, using basis functions evaluated over the entire domain to approximate individual grid points.

The following discussion will explore why spectral methods are well-suited for studying the R4CF and its simple 2D geometry. Additionally, the significance of regularizing the boundary conditions will become clear. Most of the theory presented in this section is derived from the books Boyd (2001) and Canuto et al. (2006). It is important to note that this section deliberately involves some simplifications and non-rigor to give a broad overview and justify applying a pseudo-spectral Chebyshev discretization for the R4CF problem.

5.1 Spectral Methods

Spectral methods encompass a range of global discretization techniques used to solve differential and integral equations and can be classified into two main categories. To demonstrate some of the underlying theory for spectral methods, we will examine a one-dimensional approximation based on a finite series expansion of $m + 1$ points with basis functions ψ_i and unknown coefficients a_i :

$$u(x) \approx u_m(x) = \sum_{i=0}^m a_i \psi_i \quad (19)$$

This approximation can be plugged into a differential or integral equation of the form $Lu = f(x)$ where L is the operator of the equation. To assess the accuracy of the approximation, we can analyze the residual function, which gives the error between the approximation and the real solution. We note that the residual at a point x can depend on all expansion coefficients, hence the name global. This residual function is formalized within the framework *the methods of mean weighted residuals*. Spectral and other discretization methods can be characterized by how they minimize this residual function.

We can distinguish broadly between two categories of spectral methods, the "interpolating" or "non-interpolating" versions (Boyd, 2001). The so-called *pseudo-spectral* or interpolating methods refer to the strategy of choosing a set of interpolation points. This technique is also referred to as *collocation*. Through the $m+1$ equations that are obtained, the coefficients a_i can be determined. The non-interpolating versions use Galerkin's method or Lanczos' tau-method to solve the equation by multiplying the residual function with a test function ϕ_i and integrating the equation. This results in the integrated so-called weak form of the differential equation and the residual is minimized in terms of the integral. In the case of the widely used Galerkin's method, the test function is chosen to be of the form of the approximating (basis) function. In this sense, the residual is minimized in a weighted manner by multiplying with the basis function. It can be shown that Galerkin's method gives more accurate results than the collocation method. But due to the simplicity of solving the equation at the collocation points exactly, the pseudo-spectral approach is usually preferred and doesn't lose much accuracy. In this study, the spectral method is actually referring to the pseudo-spectral approach.

From the point of view of interpolation theory, using the collocation method is similar to an implicit polynomial interpolation at specified grid points in the physical space. In contrast to the basis function formulation defined in (19), we don't have to compute the coefficients of the basis function but just choose the right collocation points to approximate the solution using these grid points. It turns out that if we use unevenly spaced grids corresponding to the roots of orthogonal

polynomials (Chebyshev, Fourier, or Lagrange), we get a much better interpolation compared to an evenly spaced grid.

5.2 Spectral Convergence Rate and Singularities

Such a well-conditioned set of nodes can achieve a spectral (exponential) convergence rate. This is different from linear or quadratic convergence, where the rate is constant and equal, for example, to 1 or 2. A spectral convergence rate refers to the situation of having a rate that is potentially unbounded. To achieve this, unevenly spaced grids, for example, obtained by Chebyshev polynomials, are preferred as they have a small Lebesgue constant and thus good conditioning of the interpolation (Meseguer, 2020). This mitigates numerical issues.

Another point is that singularities in the function to be approximated broadly affect this convergence rate. If the function has singularities, we can't expect a spectral convergence rate, and due to the global nature of a pseudo-spectral discretization, all the grid points will be affected. Furthermore, the gravest singularity dominates the actual convergence rate and the domain of convergence. In addition, the singularities outside the domain of interest may affect the convergence inside the area of interest.

From a physical perspective, singularities are present in cases where the problems involve discontinuous boundary conditions. Spectral methods are inherently affected by these discontinuities as the global nature of the discretization is "polluted" by the singularity and can render the approximation useless in those regions due to spurious modes and oscillations (Canuto et al., 2006). On the other hand, finite element methods are less affected since the singularity only contributes to a piecewise and low-order polynomial in the vicinity of the corner.

However, the strength of spectral methods lies in their ability to provide exact solutions for well-posed boundary conditions, ensuring exponential convergence and serving as a benchmark for solvers. In real-world scenarios, such extreme mathematical problems are hardly ever encountered, mitigating the impact of singularities to some extent.

5.3 Pseudo-Spectral Discretization

A common choice for the interpolation points is to use the previously mentioned Chebyshev nodes which are given by the roots of the Chebyshev polynomial and minimize the approximation error (Boyd, 2001). Specifically, these discretization points can be expressed as

$$x_i = \cos\left(\frac{i\pi}{m}\right), \quad i = 0, 1, \dots, m. \quad (20)$$

These definition yields $m + 1$ points in the interval $[-1, 1]$. Notably, the grid points cluster at the boundary. Figure 9 illustrates a Chebyshev grid's unevenly spaced grid point distribution.

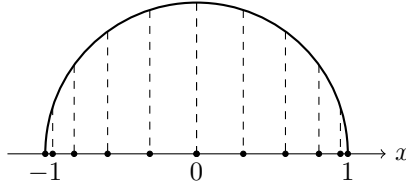


Figure 9: Chebyshev grid visualized as the projection of equally spaced points on a unit circle

5.4 Chebyshev Differentiation Matrix

If we would like to approximate derivatives in a discrete setting, using the concept of differentiation matrices is convenient. A derivative on a given set of nodes $\{x_0, x_1, \dots, x_m\}$ with respect to a function

f , can be written as

$$f(x_i)' = f_i' \approx u_i' = \sum_{j=0}^m \mathbf{D}_{ij} f_j. \quad (21)$$

This formulation is commonly employed for finite difference approximations where the elements of the differentiation matrix \mathbf{D} correspond to the derivatives of the respective cardinal functions. In the context of Chebyshev polynomials, we can utilize the same framework. The Chebyshev grid can be interpreted as an m^{th} -order finite difference approximation at non-equidistant grid points. For the 1D Chebyshev nodes, the differentiation matrix is given by (Meseguer, 2020):

$$\mathbf{D}_{ij} = \begin{cases} (-1)^{i+j} \frac{\delta_j}{\delta_i(x_i - x_j)} & (i \neq j) \\ \frac{(-1)^{i+j}}{\delta_i} \sum_{k=0, k \neq i}^m (-1)^k \frac{\delta_k}{x_i - x_k}, & (i = j) \end{cases} \quad (22)$$

As mentioned before, the grid points cluster near the boundaries of the interval, which can result in small differences in function evaluations. The definition (22) utilizes a numerically more stable formula to address this issue. The coefficients δ_i are defined as $\delta_0 = \delta_m = 1/2$ and $\delta_1 = \delta_2 = \dots = \delta_{m-1}$.

Higher-order derivatives can be simply calculated by setting the matrix to the power of the order. For instance, an approximation of the second derivative of a function f is given by

$$f_i'' \approx \sum_{j=0}^m \mathbf{D}_{ij}^2 f_j. \quad (23)$$

5.5 2D Chebyshev Discretization

To define a two-dimensional Chebyshev grid, we can extend the one-dimensional definition (20) to both spatial directions, (x_i, y_j) :

$$\begin{aligned} x_i &= \cos\left(\frac{i\pi}{m}\right), \quad i = 0, 1, \dots, m \\ y_j &= \cos\left(\frac{j\pi}{n}\right), \quad j = 0, 1, \dots, n \end{aligned} \quad (24)$$

Figure 10 shows such a grid. Analogously to the 1D case, points are clustered in the corner.

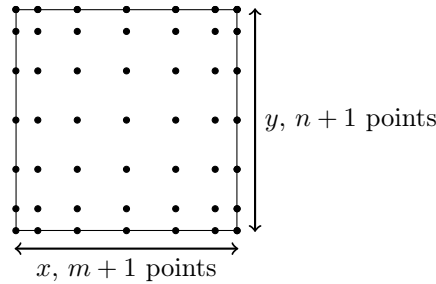


Figure 10: 2D Chebyshev grid points in a square domain ($m = n$)

The next step is to define differentiation in the x and y direction of the grid. One approach is to use the Kronecker delta product ($\mathbf{D} = \mathbf{D}_x \otimes \mathbf{D}_y$) to create the differentiation matrix for a 2D scalar function $f(x, y)$ (Trefethen, 2000). Each unknown value of the function corresponds to an element in a flattened vector. The differentiation matrices in both spatial directions are combined into a single, large matrix $\mathbf{D} \in \mathbb{R}^{(m+1)(n+1) \times (m+1)(n+1)}$. The matrix is multiplied by the flattened vector of the values at the grid points to compute the derivatives. Although this matrix is not completely dense, higher derivatives result in a dense matrix.

Another approach is to visualize the grid points as entries of a matrix. Instead of representing the unknown values as a flattened vector, we can use a matrix $\Psi \in \mathbb{R}^{(m+1) \times (n+1)}$, where the grid points ij correspond to the ji -th entry of the matrix Ψ (shown in figure 11). All derivatives can then simply be obtained by multiplying \mathbf{D}_x from the left with Ψ for the x derivative and $\Psi \mathbf{D}_y^T$ for the y derivative. The spatial derivatives of one grid point are given by the following relations, using Einstein's summation notation,

$$\begin{aligned}\partial_x \Psi_{ij} &= (\mathbf{D}_x)_{ik} \Psi_{kj}, \\ \partial_y \Psi_{ij} &= (\mathbf{D}_y)_{jl} \Psi_{il}.\end{aligned}\tag{25}$$

For the streamfunction equation (11), the discrete Laplace operator $\Delta_{dis}\Psi$ and the biharmonic operator are defined as

$$\Delta_{dis}\Psi = \mathbf{D}_x^2 \Psi + \Psi (\mathbf{D}_y^2)^T, \tag{26}$$

$$\Delta_{dis}^2 \Psi = \mathbf{D}_x^2 (\Delta_{dis}\Psi) + (\Delta_{dis}\Psi) (\mathbf{D}_y^2)^T. \tag{27}$$

Combining all the above, we can state the discrete streamfunction equation we want to solve numerically,

$$\partial_t (\Delta_{dis}\Psi) = \frac{1}{\text{Re}} (\Delta_{dis}^2 \Psi) + \mathbf{D}_x \Psi ((\Delta_{dis}\Psi) \mathbf{D}_y^T) - \Psi \mathbf{D}_y^T (\mathbf{D}_x (\Delta_{dis}\Psi)). \tag{28}$$

Here, Ψ represents the matrix for the finite-dimensional approximation of the streamfunction. The discretized partial derivative with respect to time will be discussed in section 6 on the temporal discretization.

5.6 Incorporating the Boundary Conditions

As illustrated, it is convenient for the R4CF to view the grid points as the transpose of the matrix Ψ , and so we can do derivation in x and y by just applying a matrix multiplication from the left or right with \mathbf{D}_x or the transpose of \mathbf{D}_y respectively.

Now, incorporating the boundary conditions has to be carefully considered and can be done in many different ways. One option is to explicitly substitute the discrete equations into the equations for the boundary conditions (see Meseguer (2020)). Recalling the tangential boundary conditions (16) and (17) and the definition of the streamfunction (6) and (7) we see that the boundary conditions are essential of Neumann-type as the streamfunction is defined through the derivative of the two velocity components:

$$\frac{\partial \Psi(x, \pm 1, t)}{\partial y} = \pm \left[\left(e^{k_0(x-1)} - 1 \right) \left(e^{-k_0(x+1)} - 1 \right) \right]^2 \tag{29}$$

$$\frac{\partial \Psi(\pm 1, y, t)}{\partial x} = \mp \left[\left(e^{k_0(y-1)} - 1 \right) \left(e^{-k_0(y+1)} - 1 \right) \right]^2 \tag{30}$$

Furthermore, the boundary components are only nonzero in the tangential direction of the four lids, meaning this corresponds to the derivative of the streamfunction normal to the boundary. Regarding the normal component of the velocity, we can use the fact that the scalar streamfunction is defined up to a constant. And the streamfunctions values can be set to 0 at all the lids of the cavity (outermost rows and columns of the matrix). As the geometric shape of the cavity is a square or rectangle, this tells us that the derivative of the streamfunction along the lids is 0. Corresponding to the normal component of the velocity field, which ensures no flux outwards from within the cavity.

The next step is to incorporate the boundary conditions (29) and (30) above, by expressing the first inner grid points (outermost are all 0) in terms of the other components. Using the Einstein

summation convention, we can write the derivative of the streamfunction for the vertical walls as:

$$\begin{aligned}\partial_x \Psi_{0j} &= \sum_{k=0}^m (\mathbf{D}_{\mathbf{x}})_{0k} \Psi_{0j} = \underbrace{(\mathbf{D}_{\mathbf{x}})_{00} \Psi_{0j}}_{=0} + (\mathbf{D}_{\mathbf{x}})_{01} \Psi_{1j} + (\mathbf{D}_{\mathbf{x}})_{0m-1} \Psi_{m-1j} + \underbrace{(\mathbf{D}_{\mathbf{x}})_{0m} \Psi_{mj}}_{=0} + (\mathbf{D}_{\mathbf{x}})_{0\bar{k}} \Psi_{\bar{k}j} \\ &= (\mathbf{D}_{\mathbf{x}})_{01} \Psi_{1j} + (\mathbf{D}_{\mathbf{x}})_{0m-1} \Psi_{m-1j} + (\mathbf{D}_{\mathbf{x}})_{0\bar{k}} \Psi_{\bar{k}j}\end{aligned}\quad (31)$$

$$\partial_x \Psi_{mj} = \sum_{k=0}^m (\mathbf{D}_{\mathbf{x}})_{mk} \Psi_{kj} = (\mathbf{D}_{\mathbf{x}})_{m1} \Psi_{1j} + (\mathbf{D}_{\mathbf{x}})_{mm-1} \Psi_{m-1j} + (\mathbf{D}_{\mathbf{x}})_{m\bar{k}} \Psi_{\bar{k}j}\quad (32)$$

The same is the case for the derivatives in the y direction. \bar{k} is the index running from $k = 1$ to $k = n - 1$.

We can recast the above relations into a matrix equation:

$$\underbrace{\begin{bmatrix} (\mathbf{D}_{\mathbf{x}})_{01} & (\mathbf{D}_{\mathbf{x}})_{0m-1} \\ (\mathbf{D}_{\mathbf{x}})_{m1} & (\mathbf{D}_{\mathbf{x}})_{mm-1} \end{bmatrix}}_{A \in \mathbb{R}^{2 \times 2}} \begin{bmatrix} \Psi_{1j} \\ \Psi_{m-1j} \end{bmatrix} = \begin{bmatrix} \partial_x \Psi_{0j} - (\mathbf{D}_{\mathbf{x}})_{0\bar{k}} \Psi_{\bar{k}j} \\ \partial_x \Psi_{mj} - (\mathbf{D}_{\mathbf{x}})_{m\bar{k}} \Psi_{\bar{k}j} \end{bmatrix}$$

$$\begin{bmatrix} \Psi_{1j} \\ \Psi_{m-1j} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{A^{-1}} \begin{bmatrix} \partial_x \Psi_{0j} - (\mathbf{D}_{\mathbf{x}})_{0\bar{k}} \Psi_{\bar{k}j} \\ \partial_x \Psi_{mj} - (\mathbf{D}_{\mathbf{x}})_{m\bar{k}} \Psi_{\bar{k}j} \end{bmatrix}\quad (33)$$

The matrix equation (33) tells us that all elements on the first inner grid points for the horizontal components are explicitly given through the boundary conditions. The 2×2 matrix to the left is known in advance from the Chebyshev differentiation matrix. It follows that the inverse can be pre-computed and reused in succeeding calculations whenever the boundary conditions have to be evaluated. The right part of the equations is given by the partial derivatives and a sum of a matrix-vector product of the inner elements of Ψ .

The same procedure can be applied to the y derivatives of the streamfunction at the horizontal wall. Similarly, we get another matrix where the inverse can be pre-computed:

$$\begin{bmatrix} \Psi_{i1} \\ \Psi_{in-1} \end{bmatrix} = \underbrace{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}}_{B^{-1}} \begin{bmatrix} \partial_y \Psi_{i0} - (\mathbf{D}_{\mathbf{y}})_{0\bar{l}} \Psi_{i\bar{l}} \\ \partial_y \Psi_{in} - (\mathbf{D}_{\mathbf{y}})_{n\bar{l}} \Psi_{i\bar{l}} \end{bmatrix}\quad (34)$$

One subtlety is that the calculations for the boundary conditions for the horizontal and vertical walls have to be done one after the other, as the outermost values in the first rows or columns are dependent on each other. Figure 11 shows in blue and red shading the two inner rows and columns which are computed with equation (33) and (34).

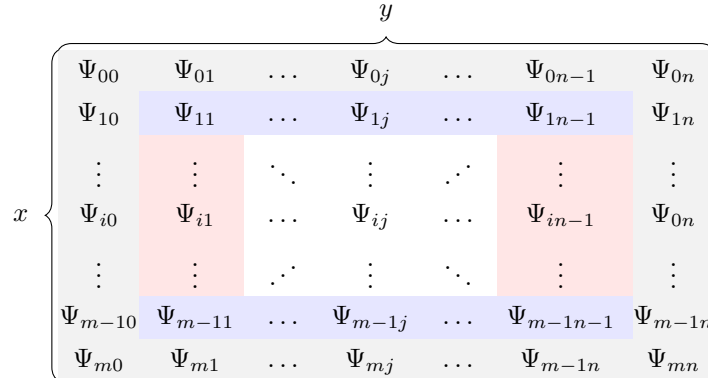


Figure 11: Elements of the streamfunction matrix representation Ψ , blue shading corresponds to the inner rows computed in (33) and red shading to the columns calculated in (34)

6 Temporal Discretization and Solvers

Time-stepping refers to the time integration or time discretization of time-dependent differential equations. In the previous section, we have seen the benefits of using a spectral discretization of the spatial dimension of our problem. Due to its nature, the time component is usually discretized with a local discretization, not a global discretization such as a spectral one. The first thing to note is that time integration is affected by the resolution of the spatial discretization of the problem. However, as mentioned, this will be less of a problem with our pseudo-spectral approach.

The continuous time domain is divided into discrete time intervals or steps. In this case, an equispaced grid $t_j = jh$ (where j goes from 0 to N) is employed in the time domain $t \in [0, T]$, where h corresponds to the time interval for each step.

There exist various discretizations for the time operator. In an explicit time scheme, the values of the dependent variables at a future time step are computed solely based on the values of the dependent variable at the current time step and previous time steps. On the contrary, an implicit time scheme uses the values of the dependent variable of the current and future time steps.

The time-dependent equation (11) is nonlinear and also classified as a stiff equation. Stiff equations generally involve dynamics that change rapidly on one timescale whereas slowly on another. These phenomena are characterized by a large range of magnitude of eigenvalues associated with the discrete problem.

Explicit time schemes become very slow for stiff equations as time step h has to be chosen very small. Otherwise, the numerical scheme will become unstable. A stable time scheme ensures that small errors in the initial conditions or numerical approximations do not amplify and cause the solution to become unbounded or oscillated and will converge to the underlying function. For our Chebyshev discretization already for the 1D case, time-stepping is stable only if h is of $O(1/N^2)$ or smaller (Boyd, 2001).

An implicit time scheme has to be used to ensure stability and means that we have to solve a nonlinear equation at every time step t_j .

6.1 Implicit Euler Method

In this study, the implicit Euler method is used. The Euler method is a stiff, stable method, and it can be characterized as the first-order backward differentiation formula (BDF1) (Meseguer, 2020). It is a straightforward method and, when applicable, is used in many stiff problems.

Given an autonomous ordinary differential equation (ODE) of the form:

$$\frac{dx}{dt} = f(x), \quad (35)$$

where x represents the dependent variable and $f(x)$ is the derivative function with respect to t , the implicit Euler approximates the solution x_{j+1} of the next timestep by

$$x_{j+1} = x_j + h \cdot f(x_{j+1}), \quad (36)$$

where h is the timestep size and t_j and x_j represent the time and solution at the j -th timestep, respectively.

6.2 Streamfunction Time-Stepping Equation

Some adjustments must be made to apply the Euler method to the R4CF. First, the one-dimensional problem naturally generalizes to problems where \mathbf{x} is a finite-dimensional vector. Secondly, the streamfunction equation involves the Laplace operator applied to the streamfunction, which means that the time derivative is applied to the Laplacian of the streamfunction. Overall, the time-stepping equation reads

$$\Delta \Psi_{j+1} = \Delta \Psi_j + h \cdot \left(\frac{1}{\text{Re}} \Delta^2 \Psi_{j+1} + (\partial_x \Psi_{j+1}) \partial_y (\Delta \Psi_{j+1}) - (\partial_y \Psi_{j+1}) \partial_x (\Delta \Psi_{j+1}) \right). \quad (37)$$

The equation can be rewritten as follows:

$$\begin{aligned} F(\Psi_{j+1}) &= \Delta\Psi_j - \Delta\Psi_{j+1} + h\left(\frac{1}{\text{Re}}\Delta^2\Psi_{j+1} + (\partial_x\Psi_{j+1})\partial_y(\Delta\Psi_{j+1}) - (\partial_y\Psi_{j+1})\partial_x(\Delta\Psi_{j+1})\right) \\ &= 0 \end{aligned} \quad (38)$$

This means that in order to calculate the solution for the next timestep t_{j+1} from a given solution at time t_j , we have to solve the nonlinear equation (38). The equation for the cavity flow is usually solved with a Newton-Raphson algorithm (Kuhlmann and Romanò, 2019), which is briefly introduced in the following section.

6.3 Newton-Raphson Method

The Newton-Raphson algorithm is an iterative method used to find the roots of a system of nonlinear equations. Given a system of p nonlinear equations $F(\mathbf{x}) = \mathbf{0}$, the algorithm starts with an initial guess \mathbf{x}_0 and iteratively improves the solution by updating the guess using the formula:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J_F(\mathbf{x}_k)^{-1} \cdot F(\mathbf{x}_k), \quad (39)$$

where \mathbf{x}_{k+1} is the updated guess at the $(k+1)$ -th iteration, $J_F(\mathbf{x}_k)$ is the Jacobian matrix of partial derivatives of F evaluated at \mathbf{x}_k , and $F(\mathbf{x}_k)$ is the vector of function values evaluated at \mathbf{x}_k .

The algorithm continues to iterate until a convergence criterion is satisfied, typically by checking the norm of the update $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ against a tolerance value.

The Newton-Raphson algorithm is known for its fast convergence when the initial guess is sufficiently close to the solution, and the Jacobian matrix is well conditioned.

6.4 Preliminary Results with Time Integration

To motivate the following section on bifurcation analysis, the logical next step would be to use the time integration and the spectral discretization to compute solutions of the streamfunction equation (11). It is then interesting to see if the regularized problem will produce such asymmetric solutions shown in figure 5. Figure 12 shows the time-stepping result for a Chebyshev grid of size 64×64 . The time stepper is launched at different Reynolds numbers with various random initial values of the order of 10^{-3} to avoid converging to the unstable symmetric solution.

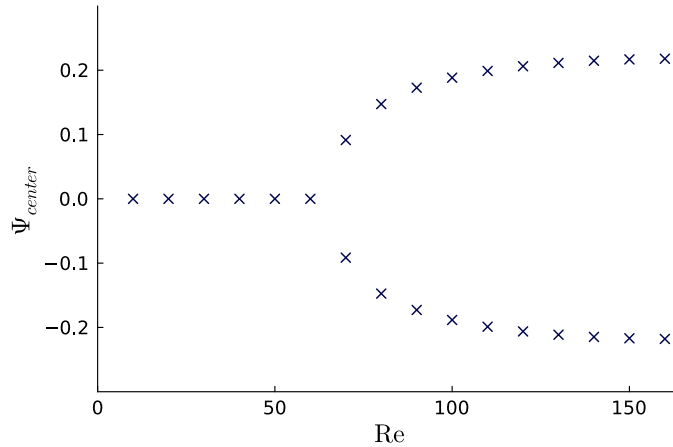


Figure 12: Value of the center grid point of the streamfunction after 150 timesteps with $h = 1$ for different Reynolds numbers up to 160

It becomes clear that at a certain Reynolds number (around 60), two asymmetric branches start to emerge, which correspond to the solutions of the non-regularized case. Furthermore, the base symmetric solutions are not attained anymore by time evolution. To analyze these different solution types and their stability, we will have to introduce tools of bifurcation analysis.

7 Bifurcation Analysis and Dynamical Systems

Hydrodynamic stability theory studies how a fluid flow is affected by small disturbances of an initial state. This analysis involves analytical, experimental, and more and more computational explorations. Flow regimes can be categorized as either linearly unstable or stable. Unstable in the sense that even infinitesimal variations will cause the flow to deviate from its initial state, resulting in a different flow state or the onset of turbulence. Conversely, disturbances do not change the initial system's state in a stable flow. In the context of the R4CF problem, the objective is to investigate the flow arising in the driven cavity for different configurations through numerical computations. We want to mainly understand how the flow is affected by the length of the side walls, the boundary velocity, and the fluid's viscosity, all of these quantities captured by the Reynolds number.

In figure 12, it has been identified that two asymmetric states are possible when the Reynolds number exceeds a critical threshold, and although the symmetric base flow remains a solution to the governing equations, it becomes unstable. The point at which the two states emerge is called a bifurcation point. A bifurcation is the change of a value of qualitative character of the set of possible steady and unsteady flows in dynamical equilibrium (Drazin, 2002). These points are often associated with instabilities, multiple flow patterns, or the emergence of oscillatory behavior. We recall the original bifurcation diagram presented in figure 6. To understand this diagram and to differentiate the points where the flow states are altered, we want to introduce the mathematical underpinnings from a dynamical system point of view. After that, the occurring bifurcations for the R4CF will be explained in detail.

7.1 The Dynamical Systems Perspective

An alternative perspective on our problem at hand is through the theory of dynamical systems:

$$\frac{dx}{dt} = f(x, \mu), \quad (40)$$

where x represents a state variable and μ is a parameter. In this framework x is a multidimensional point within a set of all possible states referred to as the state space or phase space (indicated as X). This point describes the system's current state and includes all the necessary dimensions to determine its future evolution.

The solution $x(t)$ depends on a time variable $t \in T$, where T , in our case, is a set of positive real numbers ($T \in \mathbb{R}^+$). We can define an evolution operator $\varphi_t : X \rightarrow X$, which is a map that describes the system's transformation over time. Formally, a dynamical system is written as a triple $\{T, X, \varphi^t\}$ (Kuznetsov, 2004).

The equation of the streamfunction (11) is an autonomous partial differential equation, which can be reformulated as a dynamical system. We consider an infinite-dimensional point denoted as Ψ in a state space X . Ψ can be thought of as a vector covering all possible points of our cavity and thus is infinite-dimensional. Using the streamfunction at time t , we have not only the locations of all fluid particles but also their evolution (velocities u and v) explicitly given by the definition of the derivatives (6) and (7).

Further, this infinite-dimensional space will be divided into discrete points with the chosen pseudo-spectral discretization. Increasing the number of Chebyshev grid points m and n should result in higher accuracy and the solution is expected to converge exponentially because of the singularity-avoiding regularization. The finite-dimensional space for the numerical computations can be defined as:

$$\frac{d\Psi_j}{dt} = F_j(\Psi, \text{Re}), \quad j = 1, 2, \dots, (m+1)(n+1), \quad (41)$$

where F_j is the j -th component of the nonlinear function at the j -th grid point. Equations (41) is essentially a dynamical system of $(m+1)(n+1)$ equations, and one parameter, namely the Reynolds number Re .

This more abstract framework is helpful in the sense that it provides a geometrical representation of the sought-after solutions. It is useful to look at subsets of the phase space X . Of particular interest

are the so-called invariant sets, subsets S satisfying $\varphi_t(S) \in S$ for all $t \in T$. If S is just one single point, having measure 0, this set is called a fixed point (or equilibrium). One-dimensional invariant manifolds may also be dynamically relevant. An example is an isolated periodic orbit (or limit cycle) which is a closed trajectory such that there exists a time $T > 0$ for which $\varphi_{t+T}(x) = \varphi_t(x)$. A stable periodic orbit corresponds to a closed trajectory, where other nearby trajectories converge toward the orbit. An unstable periodic orbit is characterized by nearby trajectories diverging away from it. Further, periodic orbits can be classified by their period and amplitude and are usually depicted in what is called a phase portrait. Equilibria and periodic orbits interact, leading to homoclinic connections, which are one-dimensional sets of self-connected equilibria. If an orbit reconnects more than one equilibrium, it is called a heteroclinic connection (Kuznetsov, 2004).

These invariant sets help to characterize dynamical systems and understand the fundamental types of solutions when a parameter of interest is varied.

7.2 Steady-State Flows

We have seen that launching a time stepper gives different kinds of states depending on the initial condition. A steady-state solution corresponds to a solution of the differential equation where the time t approaches infinity, satisfying

$$\begin{aligned} F(\Psi, \text{Re}) &= \frac{1}{\text{Re}} \Delta^2 \Psi + (\partial_x \Psi) \partial_y (\Delta \Psi) - (\partial_y \Psi) \partial_x (\Delta \Psi) \\ &= 0, \quad \text{or} \end{aligned} \quad (42)$$

$$F_j(\Psi, \text{Re}) = 0, \quad j = 1, 2, \dots, (m+1)(n+1), \quad (43)$$

in the discretized version. As the outer two grid rows and columns are explicitly known through the boundary conditions (explained in section 5.6), a reduced system of equations $F(\psi, \text{Re}) = 0$, can be formulated, where $\psi \in \mathbb{R}^{(m-1) \times (n-1)}$ corresponds now to the inner grid points. Thus, in practice, the system is $(m-1)(n-1)$ -dimensional and solvable with Newton's method.

7.3 Linear Stability Analysis

To analyze the stability of steady flows converged with Newton's method, we perform a linear stability analysis. Given an equilibrium solution Ψ_0 , satisfying equation (43), we perturb it by adding a small disturbance

$$\Psi = \Psi_0 + \epsilon \tilde{\Psi}, \quad (44)$$

where ϵ is a small amplitude, and $\tilde{\Psi}$ is an arbitrary field satisfying homogeneous boundary conditions. We can insert this expression into the time-dependent streamfunction equation (11). By only identifying first-order terms $\mathcal{O}(\epsilon)$ and using the fact that $F_{\text{steady}}(\Psi_0) = 0$, we get,

$$\partial_t \Delta \tilde{\Psi} = \frac{1}{\text{Re}} \Delta^2 \tilde{\Psi} + (\partial_x \Psi_0) \partial_y (\Delta \tilde{\Psi}) + (\partial_x \tilde{\Psi}) \partial_y (\Delta \Psi_0) - (\partial_y \Psi_0) \partial_x (\Delta \tilde{\Psi}) - (\partial_y \tilde{\Psi}) \partial_x (\Delta \Psi_0). \quad (45)$$

To solve this linear stability analysis problem, an ansatz is used, where we separate time and the spatial variables as follows:

$$\tilde{\Psi} = \tilde{\Psi}(x, y, t) = e^{\lambda t} \Phi(x, y), \quad (46)$$

where Φ is the eigenfunction associated with the eigenvalue λ , whose real part will dictate the stability or instability of Ψ_0 . After formal substitution of the ansatz (46) into (45), the exponential time dependence of the problem disappears, leading to an algebraic generalized boundary eigenvalue problem that reads

$$\lambda \Delta \Phi = \frac{1}{\text{Re}} \Delta^2 \Phi + (\partial_x \Psi_0) \partial_y (\Delta \Phi) + (\partial_x \Phi) \partial_y (\Delta \Psi_0) - (\partial_y \Psi_0) \partial_x (\Delta \Phi) - (\partial_y \Phi) \partial_x (\Delta \Psi_0), \quad (47)$$

where $\Phi = 0$ at the boundaries. In terms of two operators A and B , the problem can be stated as follows:

$$\begin{aligned}
A\Phi &= \lambda B\Phi \\
A &= \frac{1}{Re} \Delta^2 \bullet + (\partial_x \Psi_0) \partial_y (\Delta \bullet) + (\partial_x \bullet) \partial_y (\Delta \Psi_0) \\
&\quad - (\partial_y \Psi_0) \partial_x (\Delta \bullet) - (\partial_y \bullet) \partial_x (\Delta \Psi_0) \\
B &= \Delta
\end{aligned} \tag{48}$$

We recognize that $\Phi(x, y)$ corresponds to an eigenmode, and λ is the associated eigenvalue. If all the eigenvalues of the generalized eigenvalue problem have a positive real part, the equilibrium is stable. On the contrary, if only one of the eigenvalues is found to be positive, then the system is unstable because a small perturbation will grow exponentially in time, which can be directly seen in the definition of the ansatz.

This linear stability problem becomes computationally demanding because the perturbation Φ acts on every grid point, and the numerical eigenvalue problem is solved with matrices of the size $(m+1)(n+1) \times (m+1)(n+1)$. We compute the whole spectrum of eigenvalues using the QZ algorithm (`eigen` function in Julia). The Arnoldi method employed for a partial evaluation of the spectrum has been tried but did not converge in the case of the R4CF.

A final remark is that with linear stability analysis, we only look at linear perturbation, meaning that nonlinear disturbances (second or higher order) are not captured with this technique.

7.4 Bifurcations

We have formally defined the tools needed to characterize all the bifurcations occurring in the RCF4 precisely.

To illustrate the bifurcations we encountered, a parameter of interest μ (Re in the R4CF) will be varied and plotted against a variable that identifies different states. Lets denote this variable by U (Ψ_{center} in the R4CF). The first bifurcation we want to look at is a pitchfork (figure 13), where the symmetry of the problem is broken, and two different flow solutions appear. The base solution is still an equilibrium but linearly unstable for $\mu > \mu_c$.

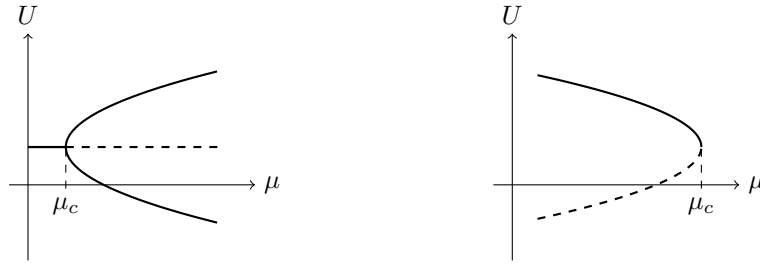


Figure 13: Supercritical pitchfork bifurcation (left) and a saddle-node (right)

Another type of bifurcation is called a saddle-node or fold bifurcation, where a set of stable and unstable fixed points collide at $\mu = \mu_c$. See the bifurcation diagram shown in figure 13 to the right.

Finally, a Hopf bifurcation is characterized by a complex conjugate pair of eigenvalues crossing the imaginary axis at some critical value $\mu = \mu_c$. At a Hopf bifurcation, a stable fixed point becomes unstable, and the system starts to oscillate. A sketch of the amplitudes is depicted in figure 14. One can distinguish between sub- and supercritical Hopf bifurcations. In the supercritical case, the system is changing suddenly from a fixed point to a stable periodic orbit (SPO). On the other hand, the subcritical Hopf is less predictable. Before the critical μ , there are unstable periodic orbits (UPO), and then it is unclear what happens when the fixed point becomes unstable.

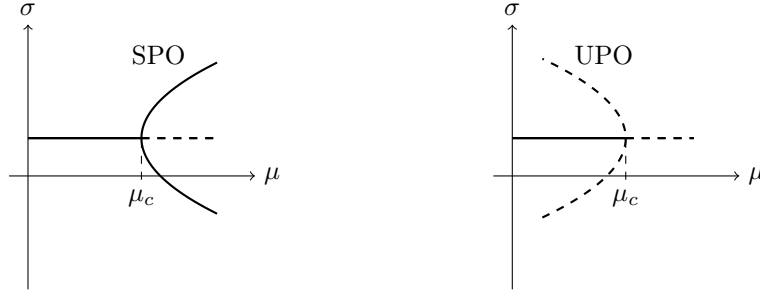


Figure 14: Sketch of amplitudes of stable (SPO) and unstable (UPO) periodic orbits for supercritical (left) and subcritical Hopf (right)

7.5 Continuation Algorithms

The next key idea is how to track branches of steady flows. The simplest numerical algorithm is called natural continuation, where we start with a solution obtained by a time-stepper or the Newton algorithm and then increase the Reynolds number. The initial guess for the next Newton iteration is the solution of the step before. In this way, we can "continue" a branch without using a time integration algorithm for each steady-state solution we want to obtain. The Newton method will converge if the steps in μ are small enough. Furthermore, unstable branches can be followed as well, which can not be done solely by time evolution.

However, there is a limitation. In the fold bifurcation depicted in figure 13, the continuation has to follow a curve when the parameter is decreasing again. The natural continuation cannot properly achieve this, as the Reynolds number is manually set to increase. Another strategy to overcome this issue is called pseudo-arclength continuation (Kuznetsov, 2004). Given two fixed points $x^{(0)}$ and $x^{(1)}$ on a curve with parameter values $\mu^{(0)}$ and $\mu^{(1)}$, we want to find the next $x^{(2)}$ and $\mu^{(2)}$ making a prediction with the approximated tangents from the already given fixed points:

$$\tilde{x}^{(2)} = x^{(1)} + \underbrace{[x^{(1)} - x^{(0)}]}_{\hat{x}} \gamma \quad (49)$$

Here γ (in this study set to 1) is a parameter that determines the prediction step size. $\tilde{x}^{(2)}$ is called the predictor. Now, another equation is imposed, which is referred to as the correction step:

$$(x^{(2)} - x^{(1)}) \cdot \hat{x} = 0 \quad (50)$$

This equation tells us that we want to find the next point on the curve, that is orthogonal to the tangent drawn from our previous points. Figure 15 illustrates this idea geometrically. Consequently, $\mu^{(2)}$ is determined explicitly by satisfying the equation (50). To find the next point, the initial system of nonlinear equations (41) is augmented by

$$F(x^{(2)}, \mu^{(0)}) = \begin{bmatrix} F(x^{(2)}, \mu) \\ (x^{(2)} - x^{(1)}) \cdot \hat{x} \end{bmatrix}, \quad (51)$$

which means the parameter μ is now part of the equation. The augmented Jacobian can be formulated as follows:

$$J = \begin{bmatrix} J_x & F_\mu \\ (x^{(2)} - x^{(1)})^T & \end{bmatrix} \quad (52)$$

First of all, we calculate the original Jacobian J_x . The components of the last row of the augmented Jacobian are given by the transpose of the distance in phase space between the two already computed states. The last column (excluding the last row) corresponds to the finite difference derivative of the nonlinear function with respect to μ . It is necessary to scale the parameter μ to obtain changes with similar magnitudes as for the x components. This ensures a well-conditioned Jacobian.

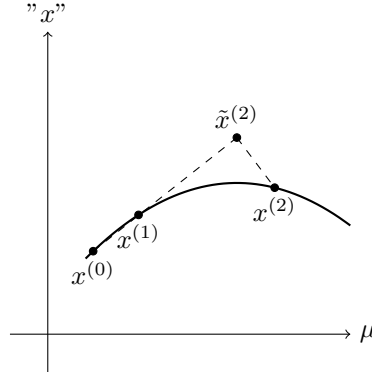


Figure 15: Pseudo-arclength continuation, a simplified illustration by projecting the vector x on a plane

8 Implementation Aspects

8.1 The Julia Programming Language

Another important aspect of this work is the implementation using the open-source programming language Julia (Bezanson et al., 2017). This programming language is designed for high-performance computing. It is a dynamically typed language but compiles efficient code via LLVM. The LLVM compiler infrastructure gives a code language-intermediated representation that is platform-independent. Julia offers performance comparable to compiled code of statically typed languages such as C or Fortran.

Another essential feature is the use of the multiple dispatch paradigm. This paradigm allows a function to be dynamically dispatched based on the run-time information of the function's arguments. Once a function has been compiled for a set of input types, the successive call of the function will not have to be recompiled and will give the expected performance.

For the regularized four-sided cavity flow, the Group of Nonlinear Fluid Dynamics at UPC has developed a MATLAB code implementing the necessary algorithms for solving this problem. This project has dealt with the translation and extension of the already existing code. The developed Julia model aims to be a simple open-source example of the proposed problem. This section will skip many details but highlight some optimization and design choices for the implementation.

8.2 Implementation of the Nonlinear Function

Julia provides a way to develop the code similar to scripting languages such as Python, but then one can add optimizations where needed without changing the language. This coding style will be demonstrated by showing the implementation of the nonlinear function. The implementation was carried out for the square cavity where the grid is of size $(m + 1) \times (m + 1)$.

We want to recall the dynamical system defined in equation (41). The code of this nonlinear function $F(\Psi, Re)$ has to be highly optimized, as it is evaluated in every Newton-Raphson iteration multiple times. The most costly part consists of calculating the Jacobian, where the function is called $\mathcal{O}(m^2)$ times.

The following code snippet shows a straightforward implementation of the nonlinear function.

Code snippet 1: Simple implementation of the nonlinear function F in Julia

```
function f(u, p::CavityStruct)
    @unpack Re, m, D1, D2, D4 = p

    Ψ = constructBC(u, p)

    biharmΨ = D4*Ψ + Ψ*D4' + 2*(D2*Ψ)*D2'
    laplΨ = D2*Ψ + Ψ*D2'
    nonlinterm = (D1*Ψ).*(laplΨ*D1') - (D1*laplΨ).*(Ψ*D1')

    fΨ = 1/Re*biharmΨ - nonlinterm
    fu = fΨ[3:(m-1), 3:(m-1)][:]

    return fu
end
```

In the above, the input vector u corresponds to the inner elements of the streamfunction. The second input p is a **struct** containing the necessary parameters and the differentiation matrices.

The coding style of defining a nonlinear function of a differential equation as $f(u, p)$ is widely used in the Julia ecosystem. This standard helps to easily pass the function to other libraries, for example, to compute the Jacobian. Moreover, one can use Unicode input such as the Ψ symbol to facilitate the notations for mathematical expressions. The `constructBC(u, p)` is a function (see appendix) which returns the streamfunction matrix by constructing the two outer rows and columns from the given boundary conditions by applying them to the vector u .

The nonlinear function could be more efficient. If we want to use it repeatedly, every function call will allocate new variables for the matrices. A useful approach is to pre-allocate memory which we pass to the function and reuse at every function call. The optimized, allocation-free version is shown in the second code snippet.

Code snippet 2: Optimized implementation of the nonlinear function F in Julia

```
function f!(fu, u, p::CavityStruct)
    @unpack Re, m, D1, D2, D4 = p.params
    @unpack fΨ, Ψ, D2Ψ, ΨD2, D4Ψ, ΨD4, laplΨ, biharmΨ = p.cache

    @inbounds @views Ψ[3:(m-1), 3:(m-1)][:] .= u

    constructBC!(Ψ, p)

    mul!(D4Ψ, D4, Ψ)
    mul!(ΨD4, Ψ, D4')

    mul!(D2Ψ, D2, Ψ)
    mul!(ΨD2, Ψ, D2')

    mul!(laplΨ, D2Ψ, D2') # using as intermediate memory
    @inbounds @. biharmΨ = D4Ψ + ΨD4 + 2 * laplΨ
    @inbounds @. laplΨ = D2Ψ + ΨD2

    mul!(D2Ψ, D1, Ψ)
    mul!(ΨD2, Ψ, D1')

    mul!(ΨD4, laplΨ, D1')
    mul!(D4Ψ, D1, laplΨ)

    @inbounds @. laplΨ = D2Ψ * ΨD4 - D4Ψ * ΨD2
    @inbounds @. fΨ = (1 / Re) * biharmΨ + laplΨ

    @inbounds @views fu .= fΨ[3:(m-1), 3:(m-1)][:]

    return nothing
end
```

The exclamation mark in front of the f in the function definition is standard to inform the programmer that the function is changing its input variable (here fu). $f!(fu, u, p)$ is then referred to as an in-place function. In general, this cache-like programming style is important when writing efficient Julia code. In the optimized version, for clarity, the **struct** p is now split into two secondary **structs**, containing the parameters and cache variables. The cache variables are reused to store the

intermediate results. The `mul!(C, A, B)` function does a matrix multiplication AB and stores the values in C without allocating a new variable. Secondly, a dot `(.)` before an operation applies an element-wise operation. The macro, denoted by an `@.`, will apply it to all the operations of an expression. The `@inbounds` macro turns off bound checking (time-consuming) and should be used carefully. Moreover, the macro `@views` avoids creating a copy of the accessed subarray.

A bottleneck in this implementation is the conversion between \mathbf{u} and Ψ representation in every function call. The vector \mathbf{u} accesses the inner elements of the streamfunction matrix Ψ . This access is not aligned in memory, making it a slower operation.

Below is a list of the algorithms implemented in Julia for square domains with $(m+1) \times (m+1)$ grid points:

- Newton-Raphson (Jacobian construction through *FinitDiff.jl*), absolute tolerance set to 10^{-10}
- Time-stepping with an implicit Euler scheme
- Pseudo-arclength continuation, and saving results to a CSV file
- Linear stability analysis, by use of the `eigen`-function of the Julia base library *LinearAlgebra.jl*

The functions are implemented in a Julia module named *FourSidedCavityFlow.jl*, which will be available on GitHub. Finally, it has to be mentioned that there exists a package for Julia called *BifurcationKit.jl* (Veltz, 2020). This module provides parameter continuation for a given nonlinear problem, a Newton-Krylov (matrix-free) solver and forward mode automatic differentiation. This is indispensable in large problems. However, the Newton-Raphson and an explicitly constructed Jacobian work best for the four-sided cavity and through regularization, the successful convergence of the spectral discretizations does not need too many grid points.

8.3 Comparison between MATLAB and Julia

Table 1 compares some basic operations for the R4CF implemented in Julia and MATLAB. The benchmark was performed on an Intel Core i5-5200U processor.

The table shows a slight decrease in time using the Julia module. One might expect that the Julia code should be much faster, but the problem is that most of the time is spent doing matrix multiplication, where MATLAB is highly optimized, and essentially both implementations will run the same lower level *Blas* (Basic Linear Algebra Subprograms) routines. Still, as the Newton’s method will be called repeatedly in the continuation algorithm, this performance gain can add up quickly. Nonetheless, this cache-like programming style in Julia is indispensable to achieve a similar efficiency as in MATLAB, but one has to do these kinds of optimizations only for specific functions where efficiency matters.

Table 1: MATLAB vs. Julia comparison of median time to solve for a steady-state solution at Reynolds number 100 (MATLAB: `timeit`, Julia: `@benchmark`)

Test	m	MATLAB time (ms)	Julia time (ms)
Function evaluation	32	0.0949	0.0526
	64	0.3267	0.2232
Creating the Jacobian	32	83.6	49.5
	64	1152.6	858.3
Newton (3 iterations)	32	196.7	126.7
	64	5861	4878

9 Results and Discussion

9.1 Outline of Bifurcation Diagram

The primary objective of this project is to reproduce and complete the bifurcation diagram presented in figure 6 of Chen et al. (2013). The resulting bifurcation diagram of the regularized version is depicted in figure 16, and table 2 lists the bifurcations that occur along with their corresponding critical Reynolds numbers. Table 3 shows the critical Reynolds numbers obtained for different grid sizes. As mentioned, only a square cavity is considered for this study, and no aspect ratios have been varied.

The critical Reynolds numbers reported here are determined by interpolating the leading eigenvalues between states of the pseudo-arclength continuation algorithm where the leading eigenvalue crosses the imaginary axis in the complex plane. The reported results are rounded to three significant digits.

Table 2: List of bifurcations encountered and the critical Re at which they occur

Bifurcation	Re
P_1 , first supercritical pitchfork of the base flow	66.197
P_2 , second pitchfork of the base flow	172.708
P_3 , pitchfork of the asymmetric steady solution	353.358
SN , saddle-node bifurcation of the asymmetric steady solution	353.654
H , Hopf bifurcation of the asymmetric steady solution	348.321

Table 3: Critical Reynolds numbers for different grid sizes m of the two pitchfork bifurcations of the symmetric base flow, $Re_c^{P_1}$ and $Re_c^{P_2}$, and of the saddle-node, the third pitchfork, and Hopf bifurcations corresponding to the asymmetric steady solution, Re_c^{SN} , $Re_c^{P_3}$ and Re_c^H respectively. For the Hopf bifurcation, the imaginary part ω_c^H corresponding to the crossing eigenvalue has also been included. In comparison, the scaled critical values for the un-regularized version are shown as well.

m	$Re_c^{P_1}$	$Re_c^{P_2}$	Re_c^H	ω_c^H	$Re_c^{P_3}$	Re_c^{SN}
32	66.197	172.731	348.210	0.0582	352.160	352.527
48	66.197	172.708	348.315	0.0600	353.365	353.663
64	66.197	172.708	348.321	0.0599	353.356	353.656
96	66.197	172.708	348.321	0.0599	353.358	353.654
Chen et al. (2013)	65.154	177.723	-	-	-	438.285

Analyzing the new bifurcation diagram in figure 16 shows that the regularized version resembles the non-regularized one regarding bifurcations and the shape of the asymmetric branches. The reported critical Reynolds numbers for the pitchforks P_1 and P_2 are close to those of the original problem (a factor of 2, with respect to Chen’s scaling, see section 4). We can also see how the regularized version converges rapidly, with only changes within the third decimal digit (or smaller) for a grid of size 48×48 or larger for the two pitchforks and the saddle-node bifurcation.

On the other hand, it has been observed that the critical value for the saddle-node in the regularized version deviates significantly from the original value by approximately 80 units of Reynolds. This variation is probably caused by the regularization. However, measuring the discrepancy between the regularized and original version is out of the scope of this work because the focus lies on the development of a robust benchmark.

Moreover, the Group of Nonlinear Fluid Dynamics at UPC has recently discovered a Hopf bifurcation (H) at the asymmetric branch. A pitchfork bifurcation (P_3) has also been found with a critical Reynolds number close to the saddle-node. Another branch is emerging at that location, displaying a different asymmetric flow pattern. The following two subsections will discuss these two bifurcation scenarios in more detail.

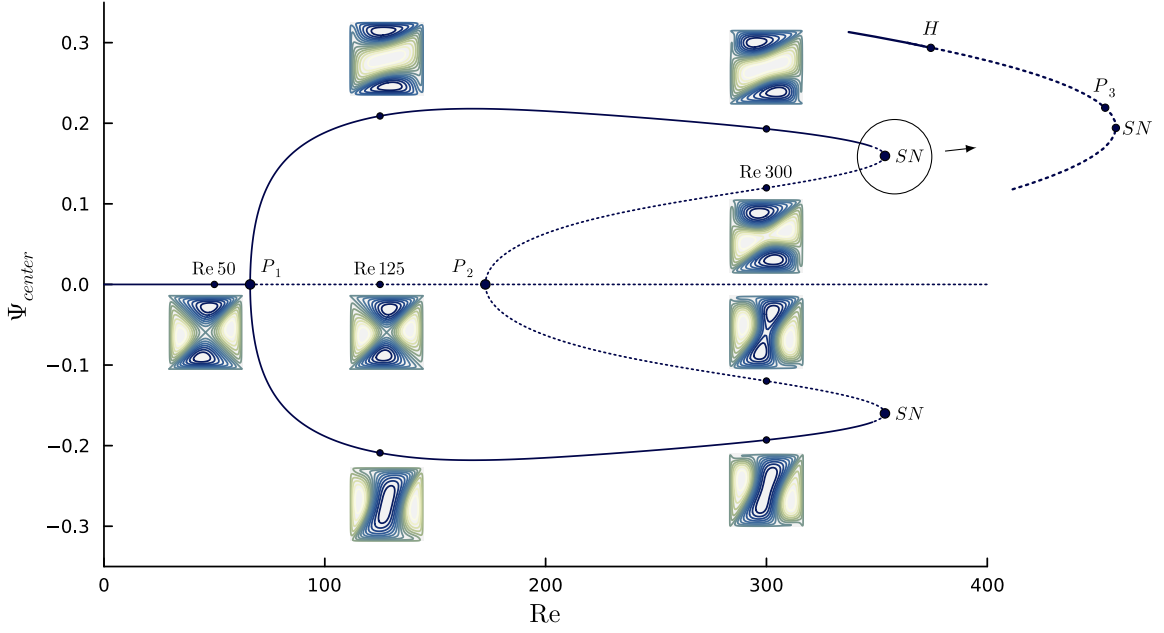


Figure 16: Bifurcation diagram for the regularized version of the four-sided cavity flow, calculated by using the custom-developed Julia module

9.2 Hopf Bifurcation and Time-Periodic Flows

The Hopf bifurcation is found to be located at a critical Reynolds of 348.321 and has not been reported in previous studies to our knowledge. The question that arises is whether the Hopf bifurcation is subcritical or supercritical.

Figure 14 in section 7.4 on bifurcation scenarios shows how these two types differ. For a supercritical Hopf, we would expect stable periodic orbits for a Reynolds number above the critical value of the Hopf bifurcation. On the contrary, for the subcritical case, unstable periodic orbits should be located below the critical value. The amplitudes of the periodic orbits, for both cases, should scale following a square root envelope of the form $A \propto |R - R_c|^{\frac{1}{2}}$ (Kuznetsov, 2004).

To investigate this, the time-stepper is launched from the perturbed symmetric unstable solution at Reynolds number larger than the critical value of the Hopf. These time evolutions reveal trajectories that converge to stable periodic orbits (see figure 17), but the amplitudes of these oscillations are larger than the expected square root growth for a supercritical Hopf. Further, the Group of Nonlinear Fluid Dynamics has computed unstable periodic orbits that converge subcritically from the Hopf bifurcation at $Re = Re_c^H$. At smaller Reynolds numbers, the unstable periodic orbits collapse in a homoclinic connection with the lower asymmetric branch. These results confirm that the Hopf is subcritical.

The follow-up questions revolve around the detected stable periodic orbits with large amplitudes. They are depicted in figure 17 for three distinct Reynolds numbers. The orbits are visualized in a projection of the phase space using the average of the vertical and horizontal velocities at the top and left mid-points of the cavity (as defined in figure 7). The amplitudes of these oscillations are also shown by extracting the center value of the streamfunction. They are compared to the steady-state lower (LB) and upper (UB) branches in the bifurcation diagram.

Moreover, as shown in figure 17, these stable periodic orbits with large amplitude exist below the Hopf bifurcation's critical value and were obtained by extracting the solutions of the matrix Ψ with the maximum amplitude from an orbit at a higher Reynolds number. This solution is then used as an initial condition for the time-stepper. With this approach, one can "continue" stable periodic orbits backward in Reynolds, similar to a natural continuation algorithm. These periodic orbits collapse at a Reynolds number of around 335, and time evolutions at lower values will converge to one of the stable asymmetric branches.

Figure 18 shows a zoom of the phase portrait of figure 17 on the trajectories of the stable periodic

orbits. The blue dots refer to the values of the stationary solutions. One can recognize that the stable periodic orbits transiently visit the locally attracting lower saddle steady-state (LB), and the periodic orbit with a Reynolds number of 336 is approaching this fixed point even more (see figure 18). This suggests that the collapse at around 335 in Reynolds is through a heteroclinic connection with the unstable lower branches.

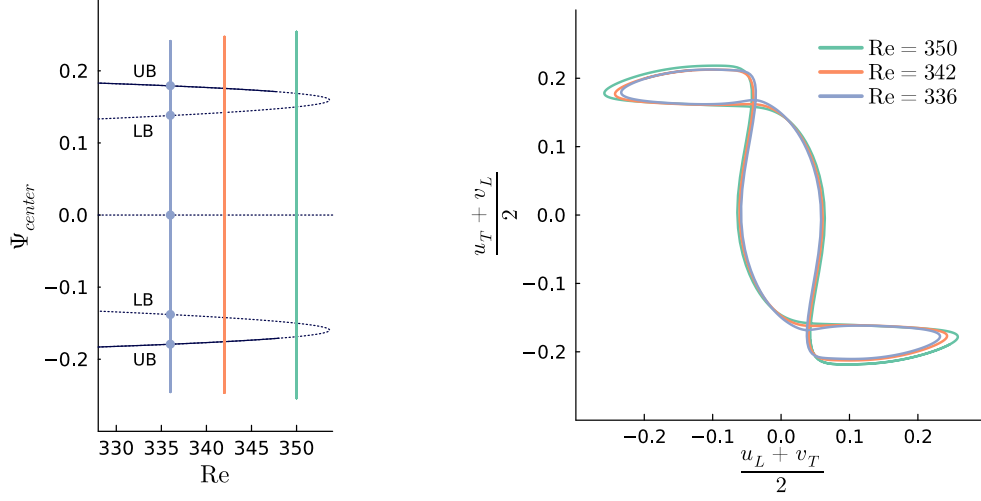


Figure 17: Periodic orbits at different Reynolds numbers, amplitudes depicted in the bifurcation diagram and in a phase portrait using the velocities at the top and left mid-points of the cavity

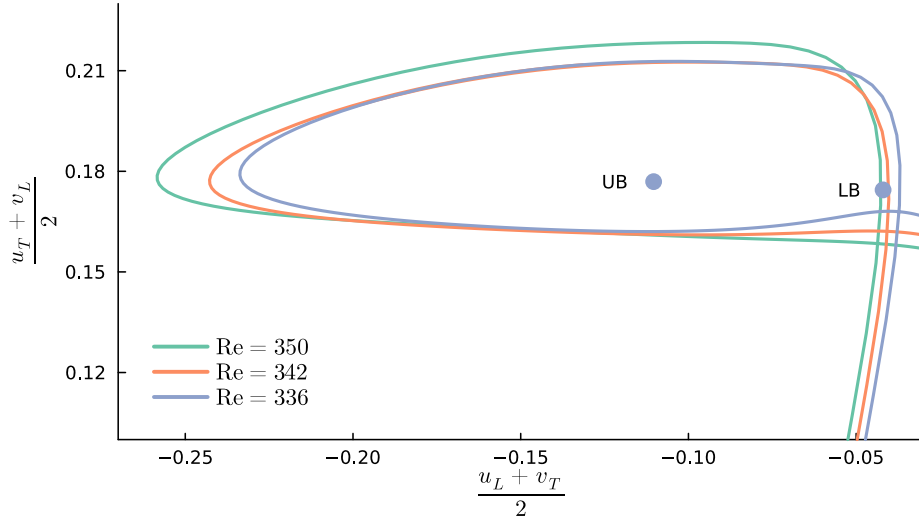


Figure 18: Zoom of periodic orbits at different Reynolds numbers, blue dots denote the locations of the upper stable (UB) and unstable (LB) steady solutions for $Re = 336$

It is unclear if the stable periodic orbits with a large amplitude below the critical Reynolds number are related to the unstable periodic orbits of the subcritical Hopf. Figure 19 illustrates what seems to be happening. One can see that there is an interval where two stable solutions coexist, namely the stable asymmetric branch and the stable periodic orbits. From a dynamical systems point of view, these stable solutions must be separated by an unstable branch.

An explanation, therefore, could be that the two periodic orbits are connected, forming what is called a saddle-node of orbits. However, this does not seem to be the case because the homoclinic connection and the collapse of the subcritical periodic orbits take place at distinct Reynolds numbers. What the separating branch is, remains unclear, but the Hopf bifurcation itself is likely of subcritical type.

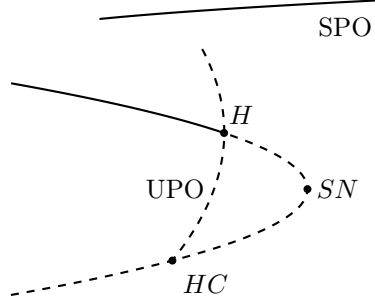


Figure 19: Illustration of amplitudes in the bifurcation diagram for the stable periodic (SPO) and the unstable periodic orbits (UPO) with its homoclinic connection (HC)

9.3 Pitchfork Bifurcation and New Asymmetric Solutions

A linear stability analysis performed along the upper asymmetric branch has also revealed that another real eigenvalue crosses the imaginary axis slightly after the Hopf bifurcation and before reaching the saddle-node point. The details of these crossings are presented in figure 20 and in table 4. Figure 21 shows the three rightmost eigenvalues in a plot where the Reynolds number increases and then decreases again from the saddle-node onwards. The crossing labeled *A* corresponds to the complex conjugate pair associated with the Hopf bifurcation, which is denoted by λ_1 and λ_2 in green. However, another eigenvalue crossing is found just before the saddle-node, denoted as *B*. This second eigenvalue crossing (λ_3 in orange) lies within a range of 0.3 in Reynolds.

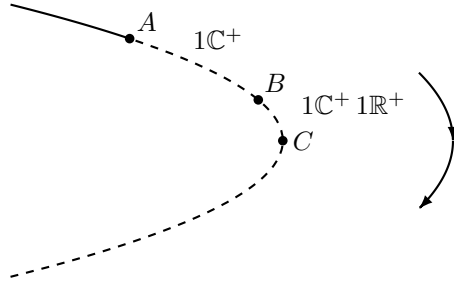


Figure 20: A schematic of the encountered eigenvalue crossing and the direction of the linear stability analysis in figure 21, $1C^+$: conjugate pair of unstable eigenvalues, $1R^+$: real unstable eigenvalue

n	$A (H)$	$B (P_3)$	$C (SN)$
64	348.321	353.356	353.656
96	348.321	353.358	353.654

Table 4: Critical Reynolds numbers where crossings appear

Additionally, close to the saddle-node, another branch emerges, which was detected by starting the natural continuation algorithm near the critical Reynolds of the fold. The branch consists of unstable steady solutions because a time-stepper starting from this branch will diverge. The branch is depicted in figure 22. The figure shows the projections onto the Ψ and u_T planes to distinguish the branches. One can see that for the new branch, the center value of the streamfunction is not changing significantly, whereas, for both the asymmetric and the second branch, there is a slight change in the horizontal velocity u_T . Further, this branch is connected to point *B*, corresponding to another pitchfork, denoted by P_3 . Linear stability analysis around the pitchfork of this new branch (figure 24) reveals how the eigenvalues can be related to point *B* (i.e. P_3).

The emergence of the second branch provides an alternative perspective on the problem. Since we are dealing with symmetric boundary conditions, we can observe that the streamlines of the symmetric base solution in the four-sided lid-driven cavity flow remain unchanged under a π -rotation and a reflection on the diagonal combined with a change of sign (in terms of streamfunction) of the cavity. The second symmetry is broken for the two asymmetric branches, which only conserve the π -rotation symmetry. Figure 23 illustrates the two solutions of the additional pitchfork emerging at P_3 , and it can be seen that the rotational symmetry is no longer valid for these branches. However, both branches exhibit the rotational symmetry to one another.

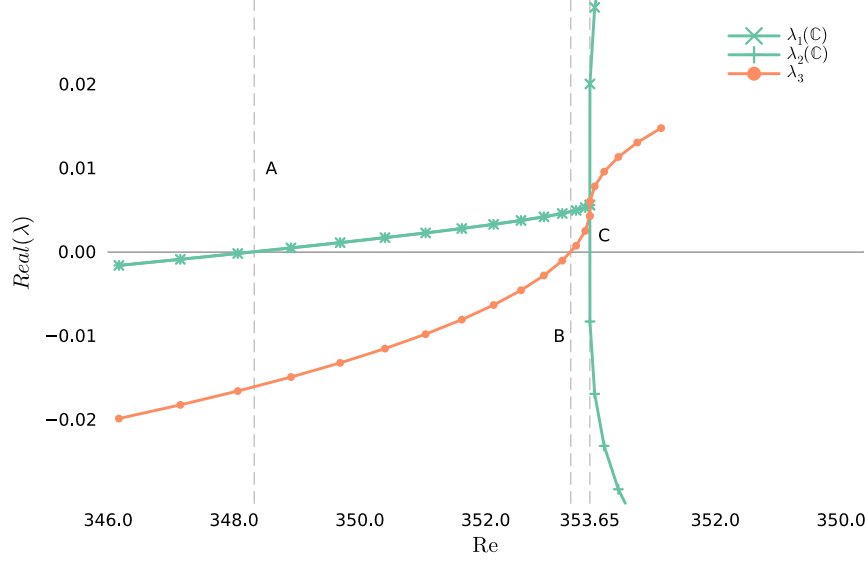


Figure 21: Linear stability analysis around saddle-node (increasing and decreasing Reynolds number), the real part of the three largest eigenvalues are shown, grid 96×96

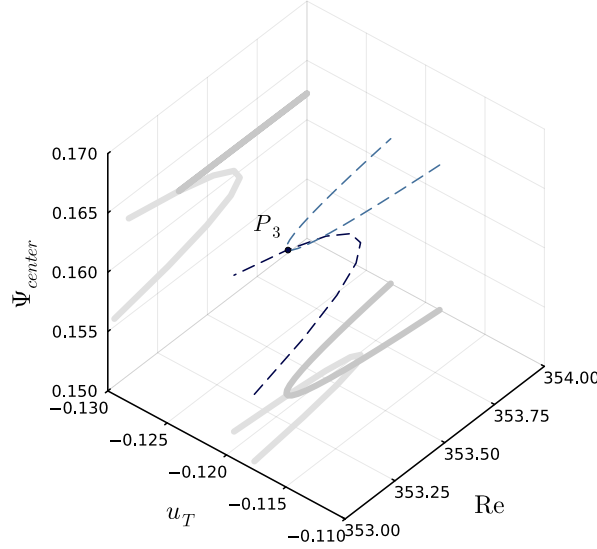


Figure 22: 3D plot of the asymmetric branch (blue) and second branch (light blue) and the connection of the two branches at pitchfork P_3)

The question that remains is what happens at the saddle-node at point C . The schematic in figure 20 shows the evolution of the eigenvalues, and figure 25 shows a detail on the real part of the eigenvalues for figure 21. After the Hopf bifurcation (A), we have a complex conjugate pair with nonzero imaginary components and a positive real part. From pitchfork P_3 (B) onwards, the additional eigenvalue λ_3 has a real part that also becomes positive. What occurs at the saddle-node C is, first of all, that the two eigenvalues lose their imaginary part and become purely real. Moreover, this takes place at a nonzero real part. What seems to be happening is that one of the complex pairs (now real) "jumps" and becomes negative. The eigenvalue configuration is depicted in the complex plane in figure 26 to illustrate the behavior at the saddle-node. The preceding explanation is speculative because we cannot be sure that it is the eigenvalue corresponding to the complex pair that becomes stable. It could be that two positive eigenvalues meet at the saddle-node.

Another reason for being cautious with our interpretation is that the saddle-node is a highly degenerated point where the Jacobian is ill-conditioned. Therefore the accuracy of the steady flows,

along the linear stability analysis, may be at stake. Due to the degeneracy, it may even be that the saddle-node corresponds to a double or triple zero, and multiple eigenvalue crossings coincide. Nonetheless, these results were obtained on a large grid for a pseudo-spectral code of size 96×96 .

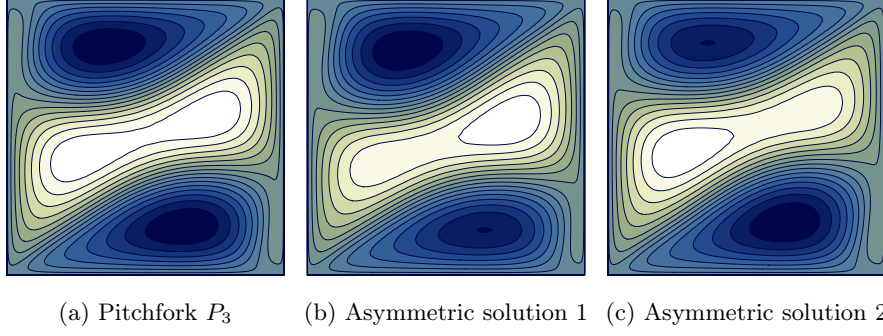


Figure 23: Streamlines of new branch, (a) π -rotation invariant, (b) and (c) are π -invariant with respect to each other

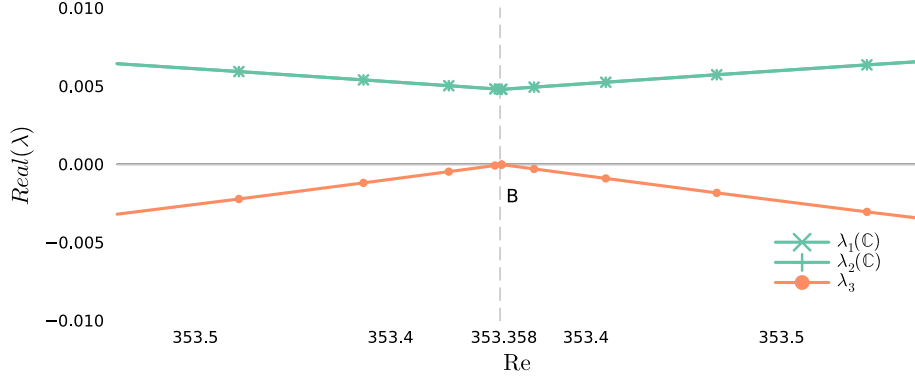


Figure 24: Linear stability analysis around connection B of second branch (decreasing and then increasing Reynolds number), grid 96×96

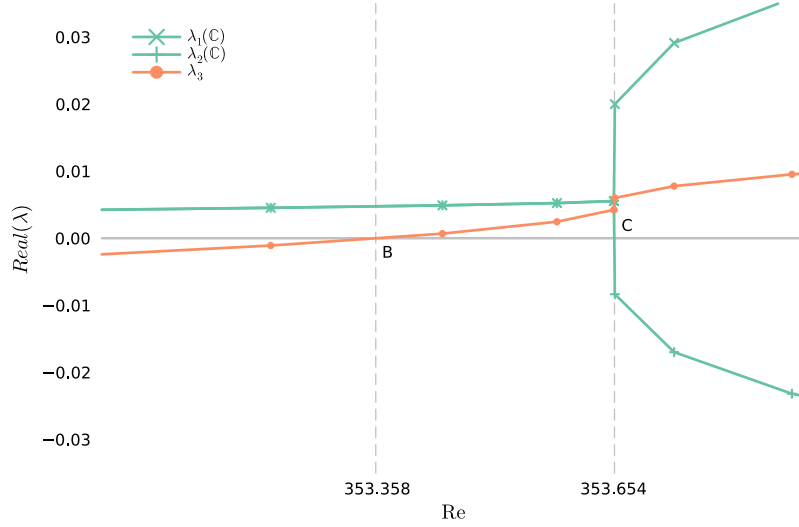


Figure 25: Zoom of figure 21 on saddle-node bifurcation, grid 96×96

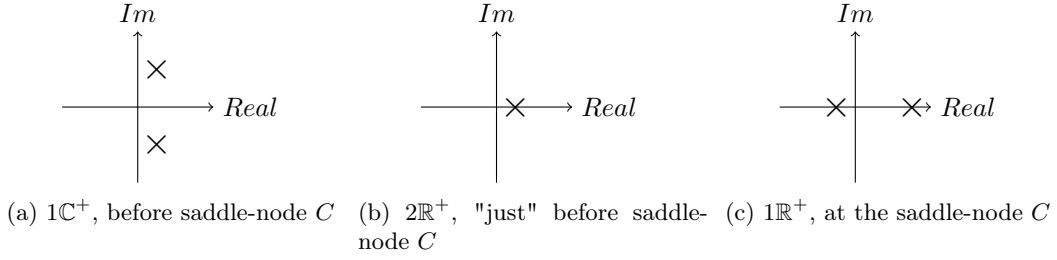


Figure 26: Visualization of the real and imaginary part of the complex conjugate pair λ_1 and λ_2 of figure 21 and 25.

10 Conclusion and Future Work

We have seen that the problem involves more bifurcation scenarios than anticipated. We want to finish with the discussion on further improvements and future work. The abovementioned more complicated crossing at the saddle-node needs further refinement of the states of the continuation algorithm. An exact comparison to the MATLAB code at a higher resolution of the points on the continuation curve would be required to clarify the issue. Nevertheless, it seems that the Hopf bifurcation and pitchfork P_3 can be seen independent of the problem arising at the saddle-node.

Another possible future investigation is related to the symmetries of the cavity. Using a square cavity and having the four lid's velocities changing at the same time means that we are varying 4 Reynolds numbers simultaneously. From a geometrical point of view, this indicates the movement along a line in a four-dimensional space, where the four Reynolds numbers could be varied. To get a better picture of what is happening, one could break the symmetries of the problem and impose different velocities at the four lids and, thus, different Reynolds numbers. This could separate the dense scenario of eigenvalues near the saddle-node.

Also, in future research, it has to be clarified how the stable upper branch is separated from the stable orbits below the critical Reynolds number of the Hopf. Moreover, we have always worked with the regularized version to be sure of the convergence rate for our pseudo-spectral discretization. It could be interesting to see if one can reproduce the periodic orbits and the secondary branch and compare the critical Reynolds numbers in the discontinuous case. Conversely, Chen et al. (2013) used a finite difference discretization. With the pseudo-spectral approach, it is certainly not guaranteed to get a useful approximation of the original boundary conditions. Another strategy is trying the spectral approach on the non-regularized version and seeing how the convergence rates and results are affected.

Lastly, it has to be mentioned that there is another eigenvalue crossing happening close to the pitchfork P_2 of the unstable base solution. This phenomenon was not investigated in this study but could suggest a more complex picture of the bifurcation scenario at P_2 .

In terms of implementation, the Julia module could be further extended. As a first step, the code can be adapted for rectangular grids. Also, the continuation algorithm and the linear stability analysis results are saved using a CSV format, and the values of the streamfunctions at the grid points are stored in plain text files. A further optimization could be to save results as binaries in an HDF5 format directly. This could easily be done with a package called *JLD.jl* and would probably reduce the code size and simplify the results handling. Another improvement worth mentioning is that the code is divided into the core functions explained in section 8 and into the code necessary to run the simulations and generate the plots. The Julia module only includes the core functions mentioned in the implementation section for readability. It could be beneficial to have both parts reconciled, and the binary savings of results would greatly help.

In conclusion, this master project focused on the four-sided lid-driven cavity flow using regularized boundary conditions. The study's spectral Chebyshev discretization technique combined with the regularization yields highly accurate results having an exponential convergence rate as the grid size increases. We have seen that the R4CF exhibits a wide variety of bifurcation scenarios at much more moderate Reynolds numbers so that different discretizations of the Navier-Stokes equations can be

easily tested.

The regularized version recovers the asymmetric branches of the problem with discontinuous boundary conditions. The original bifurcations have been detected, namely the two pitchforks of the base flow and the saddle-node of the non-regularized version. Further, in the regularized version, a Hopf bifurcation of the asymmetric solution branch was found at a Reynolds number of 348.321. Additionally, a third pitchfork occurs close to the saddle-node at 353.358 in Reynolds. The connected unstable branch breaks the π -rotation symmetry of the asymmetric solutions. On the other hand, the saddle-node bifurcation needs further investigation due to the unclear eigenvalue crossing, possibly associated with a complex conjugate pair. The presence of three pitchfork bifurcations and the Hopf bifurcation in this problem offers robust scenarios to benchmark a Navier-Stokes solver. The regularized four-sided lid-driven cavity only requires slight adaptations from the classical one-sided version, making it easily applicable to existing codes. This modified setup allows for testing the multiplicity of states and periodic orbits at low to moderate Reynolds numbers providing the means to assess a solver's capability in identifying these solution types and the oscillatory behavior. In general, it can make up for an interesting bifurcation benchmark.

References

- E. Hopf. A mathematical example displaying features of turbulence. *Commun. Pure Appl. Math.*, 1: 59–62, 1948.
- Y. A. Kuznetsov. *Elements of Applied Bifurcation Theory, 3rd Ed.* Springer-Verlag, 2004.
- Hendrik C Kuhlmann and Francesco Romanò. The lid-driven cavity. *Computational Modelling of Bifurcations and Instabilities in Fluid Dynamics*, pages 233–309, 2019.
- O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers and Fluids*, 27:421–433, 1998. doi: 10.1016/S0045-7930(98)00002-4.
- J. Shen. Hopf bifurcation of the unsteady regularized driven cavity flow. *J. Comp. Phys.*, 95:228–245, 1991. doi: 10.1016/0021-9991(91)90261-i.
- J. M. Lopez, B. D. Welfert, K. Wu, and J. Yalim. Transition to complex dynamics in the cubic lid-driven cavity. *Physical Review Fluids*, 2:074401, 2017. doi: 10.1103/PhysRevFluids.22.074401.
- E.M. Wahba. Multiplicity of states for two-sided and four-sided lid driven cavity flows. *Computers & Fluids*, 38(2):247–253, 2009. doi: <https://doi.org/10.1016/j.compfluid.2008.02.001>.
- Odus R. Burggraf. Analytical and numerical studies of the structure of steady separated flows. *Journal of Fluid Mechanics*, 24(1):113–151, 1966. doi: 10.1017/S0022112066000545.
- H. K. Moffatt. Viscous and resistive eddies near a sharp corner. *Journal of Fluid Mechanics*, 18(1): 1–18, 1964. doi: 10.1017/S0022112064000015.
- J. R. Koseff and R. L. Street. Circulation structure in a stratified cavity flow. *Journal of Hydraulic Engineering*, 111(2):334–354, 1985. doi: 10.1061/(ASCE)0733-9429(1985)111:2(334).
- H. C. Kuhlmann, M. Wanschura, and H. J. Rath. Flow in two-sided lid-driven cavities: non-uniqueness, instabilities, and cellular structures. *Journal of Fluid Mechanics*, 336:267–299, 1997. doi: 10.1017/S0022112096004727.
- Kuen-Tsan Chen, Chih-Ching Tsai, and Win-jet Luo. Multiplicity flow solutions in a four-sided lid-driven cavity. *Applied Mechanics and Materials*, VOL.368:pp. 615–618, 11 2013. doi: 10.4028/www.scientific.net/AMM.368-370.838.
- D. Arumuga Perumal and Anoop K. Dass. Multiplicity of steady solutions in two-dimensional lid-driven cavity flows by lattice boltzmann method. *Computers & Mathematics with Applications*, 61(12):3711–3721, 2011. doi: <https://doi.org/10.1016/j.camwa.2010.03.053>.
- J.M. Cadou, Yann Guevel, and Gregory Girault. Numerical tools for the stability analysis of 2d flows: application to the two and four-sided lid-driven cavity. *Fluid Dynamics Research*, 44, 06 2012. doi: 10.1088/0169-5983/44/3/031403.
- L. D. Landau and E. M. Lifshitz. *Fluid Mechanics, Second Edition: Volume 6 (Course of Theoretical Physics)*. Butterworth-Heinemann, 2 edition, 1987. ISBN 0750627670.
- Ercan Erturk. Discussions on driven cavity flow. *International Journal for Numerical Methods in Fluids*, 60:275 – 294, 05 2009. doi: 10.1002/fld.1887.
- John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, second edition, 2001. ISBN 0486411834 9780486411835.
- Claudio Canuto, M. Youssuff Hussaini, Quarteroni Alfio, and Zang Thomas A. *Spectral Methods: Fundamentals in Single Domains*. Springer-Verlag, 2006. doi: <https://doi.org/10.1007/978-3-540-30726-6>.
- Álvaro Meseguer. *Fundamentals of Numerical Mathematics for Physicists and Engineers*. John Wiley & Sons, Inc., 2020. doi: 10.1002/9781119425762.

- Lloyd Nick Trefethen. *Spectral Methods in MATLAB*. Society for Industrial and Applied Mathematics, 2000. doi: 10.1137/1.9780898719598.
- P. G. Drazin. *Introduction to Hydrodynamic Stability*. Cambridge University Press, 2002.
- Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. URL <https://doi.org/10.1137/141000671>.
- Romain Veltz. BifurcationKit.jl, Jul 2020. URL <https://hal.archives-ouvertes.fr/hal-02902346>.

Appendix A Implementation of Boundary Conditions in Julia

Code snippet 3: Boundary construction function for the matrix Ψ , below the inplace and out of place versions of this function

```
function constructBC!(Ψ, p::CavityStruct)
    @unpack m, D1, a11, a12, a21, a22 = p.params
    @unpack bcleft, bcright, bctop, bcbottom = p.params
    @unpack h1, h2, k1, k2 = p.cache

    @inbounds @views Ψ1 = Ψ[3:(m - 1), 3:(m - 1)]
    @inbounds @views Ψ2 = Ψ[3:(m - 1), 2:m]'
    @inbounds @views Dcs = D1[1, 3:(m - 1)]
    @inbounds @views Dce = D1[m + 1, 3:(m - 1)]

    mul!(h1, Ψ1, Dcs)
    mul!(h2, Ψ1, Dce)

    @inbounds @views @. Ψ[3:(m - 1), 2] = a11 * (bctop[3:(m - 1)] - h1) +
                                           a12 * (bcbottom[3:(m - 1)] - h2)
    @inbounds @views @. Ψ[3:(m - 1), m] = a21 * (bctop[3:(m - 1)] - h1) +
                                           a22 * (bcbottom[3:(m - 1)] - h2)

    mul!(k1, Ψ2, Dcs)
    mul!(k2, Ψ2, Dce)

    @inbounds @views @. Ψ[2, 2:m] = a11 * (bcleft[2:m] - k1) +
                                     a12 * (bcright[2:m] - k2)
    @inbounds @views @. Ψ[m, 2:m] = a21 * (bcleft[2:m] - k1) +
                                     a22 * (bcright[2:m] - k2)

    return nothing
end

function constructBC(u, p::CavityStruct)
    @unpack m = p.params

    Ψ = zeros(m + 1, m + 1)
    @inbounds @views Ψ[3:(m - 1), 3:(m - 1)][:] .= u

    constructBC!(Ψ, p)

    return Ψ
end
```

The code snippet shows how the outer rows and columns are computed to incorporate the boundary conditions. a_{11} , a_{12} , a_{21} and a_{22} are the pre-computed elements of the inverse of the matrix A in equation (33). As the domain is square $((m + 1) \times (m + 1))$, this matrix is the same for both spatial directions. The variables with a `bc` in front correspond the prescribed values for the grid points at the boundary.

Appendix B Abstract Conference Presentation at ICTW 2023



Regularized four-sided cavity flows: A spectral bifurcation benchmark implemented in Julia

Moritz Waldleben¹, Álvaro Meseguer², Oriol Batiste², and Arantxa Alonso²

moritz.waldleben@epfl.ch

¹Section de Mathématiques, École Polytechnique Fédérale de Lausanne, EPFL (Switzerland)

²Departamento de Física, Universitat Politècnica de Catalunya, Barcelona, UPC (Spain)

Abstract

Driven cavity flows are commonly used as benchmarks to validate Navier-Stokes solvers. These problems can test spatial discretization methodologies such as finite elements, finite differences, and spectral methods. They also assess a variety of boundary condition implementations and time-stepping schemes. The simple lid-driven cavity flow has received considerable attention. This flow is steady for high Reynolds numbers, with the first instability occurring within a large uncertainty interval (7500, 8100) [Kuhlmann (2018)].

More recent variants, such as the four-sided version of the cavity flow, have been proposed [Wahba (2009)]. In this problem, lids move with the same velocity profile and parallel lids move in opposite directions. Later works tested different numerical techniques and studied the bifurcations with linear stability analysis, arc-length continuation, and time-stepping [Perumal (2011), Cadou (2012), Chen (2013)]. This cavity has the computational advantage of exhibiting a variety of bifurcations at low or moderate Reynolds numbers.

Still, the problem suffers from corner singularities due to the discontinuous boundary conditions and affects the exponential convergence of spectral methods. This work presents a regularized version of the four-sided cavity flow to address the issue. A spectral Chebyshev discretization of the flow problem is implemented in Julia, an open-source, high-performance language for scientific computing. A developed Julia module provides a reproducible example of the proposed cavity.

The regularized four-sided lid-driven cavity shows most of the primary bifurcation scenarios. The flow undergoes instabilities, such as pitchfork, saddle-node (fold), and Hopf. Predicting the precise location of the bifurcations could present an amenable Navier-Stokes bifurcation benchmark when testing and comparing different schemes and implementations.

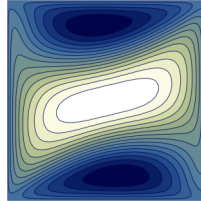


Figure 1: Asymmetric solution on a 64×64 grid at Reynolds 100

Acknowledgement

This research was supported by the Spanish Ministerio de Ciencia e Innovación, under grant No. PID2020-114043GB-I00.

References

- Wahba E. M. (2009) “Multiplicity of states for two-sided and four-sided lid driven cavity flows”, *Comput. Fluids*, **38**, 247–253
- Arumuga Perumal D., Dass A. K. (2011) “Multiplicity of steady solutions in two-dimensional lid-driven cavity flows by Lattice Boltzmann Method”, *Comput. Math. with Appl.*, **61**, 3711–3721
- Cadou J. M., Guevel Y. and Girault G. (2012) “Numerical tools for the stability analysis of 2D flows: application to the two- and four-sided lid- driven cavity”, *Fluid Dyn. Res.*, **44** (3), 031403
- Chen, K. T., Tsai C. C. and Luo W. J. (2013) “Multiplicity Flow Solutions in A Four-sided Lid-driven Cavity”, *Appl. Mech. Mater.*, **368-370**, 838–843
- Kuhlmann H., Romanò F. (2018) “The Lid-Driven Cavity”, *Comput. Methods Appl. Sci.*

2nd Spanish Fluid Mechanics Conference
Barcelona (2 - 5 July 2023)

Regularized four-sided cavity flows: A spectral bifurcation benchmark implemented in Julia

Moritz Waldleben¹, Álvaro Meseguer², Oriol Batiste², and Arantxa Alonso²

moritz.waldleben@epfl.ch

¹Section de Mathématiques, École Polytechnique Fédérale de Lausanne, EPFL (Switzerland)

²Departamento de Física, Universitat Politècnica de Catalunya, Barcelona, UPC (Spain)

Abstract

This work presents a regularized version of the four-sided lid-driven cavity for incompressible fluids. The four-sided driven cavity is an extension of the simple lid-driven case where all lids move with the same velocity profile and parallel lids move in opposite directions. Lid-driven cavity flows are used to validate Navier-Stokes solvers. This study focuses on the regularized version to overcome corner singularities. The proposed method recovers near exponential convergence in spectral discretization schemes. The flow presents a variety of bifurcation scenarios and can make up for an amenable Navier-Stokes bifurcation benchmark. The algorithms are implemented in Julia, a high-performance language for scientific computing. A developed Julia module provides a reproducible example of spectral discretization for the proposed benchmark.

1 Introduction

Driven cavity flows are commonly used as benchmarks to validate Navier-Stokes solvers. These problems can test spatial discretization methodologies such as finite elements, finite differences, and spectral methods. They also assess a variety of boundary condition implementations and time-stepping schemes. The simple lid-driven cavity flow has received considerable attention. This flow is steady for high Reynolds numbers, with the first instability due to a supercritical Hopf bifurcation occurring within a large uncertainty interval (7500, 8100) [Kuhlmann (2018)].

More recent variants, such as the four-sided version of the cavity flow, have been proposed [Wahba (2009)]. Here the four lids are moving at the same speed (top-bottom and right-left lids moving rightwards-leftwards and upwards-downwards, respectively). Later works tested different numerical techniques and studied the bifurcations with linear stability analysis, arc-length continuation, and time-stepping [Perumal (2011), Cadou (2012), Chen (2013)]. This cavity has the computational advantage of exhibiting a variety of bifurcations at low or moderate Reynolds numbers.

Still, the problem suffers from corner singularities due to the discontinuous boundary conditions and affects the exponential convergence of spectral methods. This work presents a regularized version of the four-sided cavity flow to address the issue. A spectral Chebyshev discretization of the flow problem is implemented in Julia, a high-performance language for scientific computing. Julia is free, open-source, and provides good performance comparable to compiled C/Fortran codes, making it an attractive platform for scientific computing. A developed Julia

module provides a reproducible example of the proposed cavity.

The regularized four-sided lid-driven cavity shows most of the primary bifurcation scenarios. The flow undergoes instabilities, such as pitchfork, saddle-node (fold), and Hopf. Predicting the precise location of the bifurcations could present an amenable benchmark when testing and comparing different schemes and implementations.

2 Mathematical formulation

We consider the two-dimensional and time-dependent Navier-Stokes equations in streamfunction formulation $\Psi(x, y, t)$ for incompressible fluids,

$$\partial_t \Delta \Psi = \frac{1}{\text{Re}} \Delta^2 \Psi + (\partial_x \Psi) \partial_y (\Delta \Psi) - (\partial_y \Psi) \partial_x (\Delta \Psi), \quad (1)$$

within the square domain $(x, y) \in [-1, 1] \times [-1, 1]$.

To discretize (1) using a spectral method, we define regularized boundary conditions in terms of exponential functions

$$u(\pm 1, y, t) = 0, \quad (2)$$

$$v(x, \pm 1, t) = 0, \quad (3)$$

$$u(x, \pm 1, t) = \pm (e^{k_0(x-1)} - 1)(e^{-k_0(x+1)} - 1)^2, \quad (4)$$

$$v(\pm 1, y, t) = \pm (e^{k_0(y-1)} - 1)(e^{-k_0(y+1)} - 1)^2. \quad (5)$$

The regularized boundary conditions impose a zero velocity at all four corners of the cavity. Figure 1 shows such a profile for the regularization parameter $k_0 = 10$.

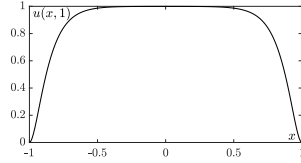


Figure 1: Regularized horizontal velocity profile (4) at the top wall $y = 1$ for $k_0 = 10$.

3 Computational methods and results

The cavity is discretized using a two-dimensional Chebyshev grid $[\cos(\frac{i\pi}{m}), \cos(\frac{j\pi}{n})]$ where $i = 0, 1, \dots, m$ and $j = 0, 1, \dots, n$.

To incorporate the boundary condition, the streamfunction is set to be zero at the lids. The first inner grid points are defined through the derivative of the streamfunction (2-5).

For the steady-state flows, the zeros of equation (1) have to be computed. As the outer grid points are explicitly known through the boundary conditions, a reduced system of equations $F(\psi, \text{Re}) = 0$ can be formulated. $\psi \in \mathbb{R}^{(m-1) \times (n-1)}$ corresponds now to the inner grid points.

We apply a Newton-Raphson algorithm to solve this system of non-linear equations, and discretize the associated Jacobian using finite differences. This Newton method can then be used in a pseudo-arclength continuation algorithm to track the solutions as a function of the Reynolds number.

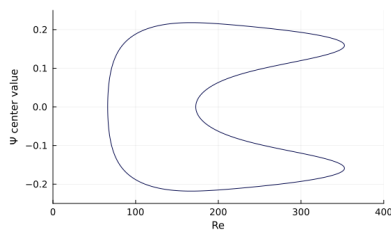


Figure 2: Branch of asymmetric solutions on a 32×32 grid obtained with pseudo-arclength continuation

Figure 2 shows the asymmetric branches obtained by the continuation algorithm and figure 3 illustrates one of these solutions. The pitchfork bifurcations seem to agree with the ones found in [Wahba (2009), Chen (2013)] whereas the locations of the saddle nodes are different as a result of the regularization process. Furthermore, it has

been vindicated that there is a Hopf bifurcation occurring at a Reynolds number of around 350.

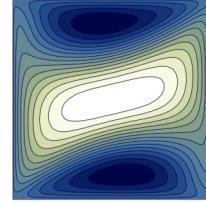


Figure 3: Asymmetric solution for 64×64 at Reynolds 100

4 Conclusions

We have shown some primary results of the regularized four-sided cavity flow. The problem has been discretized with a Chebyshev spectral method and provides a way to accurately determine a rich number of bifurcation scenarios. The computed pitchfork bifurcations seem to coincide with the un-regularized version. Additionally, it could be used as a benchmark to test other discretization schemes.

Acknowledgement

This research was supported by the Spanish Ministerio de Ciencia e Innovación, under grant No. PID2020-114043GB-I00.

References

- Wahba E. M. (2009) "Multiplicity of states for two-sided and four-sided lid driven cavity flows", *Comput. Fluids*, **38**, 247–253
- Arumuga Perumal D., Dass A. K. (2011) "Multiplicity of steady solutions in two-dimensional lid-driven cavity flows by Lattice Boltzmann Method", *Comput. Math. with Appl.*, **61**, 3711–3721
- Cadou J. M., Guevel Y. and Girault G. (2012) "Numerical tools for the stability analysis of 2D flows: application to the two- and four-sided lid-driven cavity", *Fluid Dyn. Res.*, **44** (3), 031403
- Chen, K. T., Tsai C. C. and Luo W. J. (2013) "Multiplicity Flow Solutions in A Four-sided Lid-driven Cavity", *Appl. Mech. Mater.*, **368-370**, 838–843
- Kuhlmann H., Romanò F. (2018) "The Lid-Driven Cavity", *Comput. Methods Appl. Sci.*