

# Investigation of the CIP stabilization parameters and extension of the FenicsR13 solver

*Bachelor Thesis*

*Submitted in partial fulfilment of the requirements of*

*BITS F421T*

*By*

S Aditya Karthik

2017A4TS0216H

*under the supervision of*

Dr. Manuel Torrilhon

(Professor, CCES, RWTH Aachen)

&

Dr. K.R.C. Murthy

(Assistant Professor, BITS Pilani Hyderabad Campus)



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI  
HYDERABAD CAMPUS**

(December, 2020)



**Birla Institute of Technology & Science, Pilani**  
Hyderabad Campus

04/12/2020

### **CERTIFICATE**

This is to certify that the thesis entitled “Investigation of the CIP stabilization parameters and extension of the FenicsR13 solver” submitted by S Aditya Karthik ID No. 2017A4TS0216H in partial fulfilment of the requirement of BITS F421T embodies the original work done by him under my supervision.

Supervisor  
Dr. Manuel Torrilhon  
Professor,  
CCES, RWTH Aachen.  
04/12/2020

Co-supervisor  
Dr. K.R.C. Murthy  
Assistant Professor,  
BITS Pilani, Hyderabad Campus.  
04/12/2020

## Abstract

In the first part of this work, a thorough parametric study investigating the selection of suitable values and scaling for the parameters associated with the CIP stabilization scheme is presented. The recently developed FenicsR13 finite element code, which solves the linearized R13 equations to predict non-equilibrium gas flows, employs a CIP stabilization scheme and has, therefore, been used in this study. A Python code capable of generating analytical solutions of the linearized R13 equations for problems defined over an annular domain was developed. The solutions generated by this script have been used to setup various cases for the parametric study. Qualitative analysis of the convergence curves produced by FenicsR13 for different cases has been used to deduce optimal values and scaling for the CIP stabilization parameters. Finally, observations from the results of the parametric study are presented and discussed to provide a basic intuition for selecting appropriate parameters for different problems.

The focus of the second part of this thesis is on extending the FenicsR13 solver's capability of predicting the physics of flows accurately, by including non-linearity in the conservations equations. A new scaling for the R13 equations is presented in order to improve the convergence of the non-linear iterations carried out by the solver. The non-linear extension for the momentum conservation equation is validated against the benchmark solutions of the lid-driven cavity problem for the cases of  $Re = 100$ ,  $Re = 1000$ ,  $Re = 3200$  and  $Re = 5000$ . Following this, a non-isothermal lid-driven cavity is constructed for the validation of the non-linear extension with respect to the mass and energy conservation equations. Finally, results for this problem generated by the established commercial solvers ANSYS FLUENT and COMSOL are presented.

**Keywords:** R13 Equations, FenicsR13, Finite element method, CIP stabilization, Python, SymPy, Non-equilibrium gas flows, Lid-driven cavity

## ACKNOWLEDGEMENT

Firstly, I would like to thank my supervisor Dr Manuel Torrilhon for providing me with the opportunity to work on this exciting project. The frequent discussions we had regarding the project and his sustained willingness to extend his support across the on-line platform made the tricky situation of a remote thesis, a seamless experience. The computational resources provided by my supervisor were crucial for the studies conducted in this thesis. The topic of the thesis is in-line with my research interests and involves many of the concepts I have learnt over the last three years in my Mechanical Engineering undergraduate program.

I would further like to thank Edilbert Christhuraj, a doctoral candidate under Dr. Torrilhon, with whom I have had many on- and off-topic discussions throughout my thesis. He helped me work myself into the project and provided constant insights based on his work which helped me greatly.

Finally, a special thanks to my family and friends who supported me throughout my journey during this difficult pandemic situation and helped me successfully complete my thesis.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Background	1
1.2	FenicsR13	2
1.3	Thesis objectives	2
1.4	Thesis Outline	2
<b>2</b>	<b>The R13 Equations</b>	<b>4</b>
2.1	Kinetic Gas Theory	4
2.2	Grad's Method of Moments	5
2.3	Regularized 13-Moment Equations	6
2.3.1	R13 Equations	6
2.3.2	R13 Boundary Conditions	7
2.3.3	Linearized R13 Equations	7
2.3.4	Weak Formulation	9
<b>3</b>	<b>Stabilization</b>	<b>10</b>
3.1	Saddle Point Problems	10
3.2	CIP stabilization in FenicsR13	12
<b>4</b>	<b>Analytical Solution Generating Script</b>	<b>13</b>
4.1	Annular Domain	13
4.2	Python Code	15
4.2.1	SymPy Module	15
4.2.2	Code Structure	15
4.2.3	Code Execution	15
4.2.4	Using Exported Solution	17
4.2.4.1	Python Format	17
4.2.4.2	C++ Format	18
4.2.4.3	DOLFIN Format	19
4.2.5	Limitations	20
<b>5</b>	<b>CIP Investigation</b>	<b>21</b>
5.1	Cases and Error Measures	21
5.1.1	Annular Domain	21
5.1.2	Case I : Homogeneous Flow Past Circular Cylinder	22
5.1.3	Case II : Mass and Heat Source Induced Flow in an Extruded Ring	22
5.1.4	Case III : Body Force Induced Flow in an Extruded Ring	22
5.1.5	Case IV : Complex Flow	23
5.1.6	Error Measures	23
5.2	Exponent Value Study	24
5.3	Constant Value Study	26
5.4	Structured Mesh Study	30

---

5.5	Conclusion . . . . .	31
<b>6</b>	<b>Non-Linear Conservation Equations</b>	<b>33</b>
6.1	Scaled R13 Equations . . . . .	33
6.2	Lid-Driven Cavity Validation . . . . .	34
6.2.1	Linear FenicsR13 Solution . . . . .	35
6.2.2	Non-linear FenicsR13 Solution . . . . .	35
6.3	Non-Isothermal Cases for Validation . . . . .	41
6.3.1	Physical Properties of Argon . . . . .	41
6.3.2	Problem Statement . . . . .	41
<b>7</b>	<b>Future Work</b>	<b>45</b>
<b>A</b>	<b>Analytical Expressions of the field variables</b>	<b>46</b>
A.1	Temperature . . . . .	46
A.2	Pressure . . . . .	46
A.3	Velocity . . . . .	47
A.4	Heat Flux . . . . .	47
A.5	Stress . . . . .	48
<b>B</b>	<b>Case Descriptions</b>	<b>49</b>
B.1	Case I . . . . .	50
B.2	Case II . . . . .	51
B.3	Case III . . . . .	52
B.4	Case IV . . . . .	53
	<b>Bibliography</b>	<b>54</b>

# List of Figures

4.1	A schematic of the annular domain. It is important to note the orientation of the normals and tangential velocity vectors at the walls. . . . .	14
4.2	An example of the input YAML file that needs to be supplied to the analytical solution generating script. . . . .	17
4.3	An example of a Python code importing the Exact_Sol_Python.py file . . . . .	18
4.4	Output corresponding to the above Python script . . . . .	18
4.5	An example of a C++ code importing the Exact_Sol_C++.h file . . . . .	18
4.6	Output corresponding to the above C++ script . . . . .	19
4.7	An example of a DOLFIN enabled code importing the Exact_Sol_Dolfin.cpp file . . . . .	19
4.8	Output corresponding to the above DOLFIN code . . . . .	19
5.1	A schematic of the annular domain. It is important to note the orientation of the normals and tangential velocity vectors at the walls. . . . .	22
5.2	Streamlines and velocity contours corresponding to Case 1 and Case 2 . . . . .	23
5.3	Streamlines and velocity contours corresponding to Case 3 and Case 4 . . . . .	24
5.4	Convergence plots obtained after averaging the errors across the 4 cases. Each plot corresponds to the independent variation of one of the three exponents. These graphs illustrate that the selection of $(n_\theta, n_p, n_u) = (3, 1, 3)$ leads to the best convergence. . . . .	25
5.5	Independent Variation - $e_{L^2}$ : These graphs correspond to the relative $e_{L^2}$ errors averaged across the 4 cases and they serve to demonstrate how the stabilization parameters $\delta_\theta$ , $\delta_p$ and $\delta_u$ mostly affect affect the variables $\theta$ , $p$ and $\underline{u}$ respectively. . . . .	27
5.6	Independent Variation - $e_{l^\infty}$ : These graphs correspond to the relative $e_{l^\infty}$ errors averaged across the 4 cases and they serve to demonstrate how the stabilization parameters $\delta_\theta$ , $\delta_p$ and $\delta_u$ mostly affect affect the variables $\theta$ , $p$ and $\underline{u}$ respectively. . . . .	28
5.7	Convergence at extreme values for stabilization parameters. It can be seen how selecting extreme values for the stabilization parameters often results in worsened convergence. Once again here the errors from the 4 cases are combined using the $L^2$ and $L^\infty$ norms. . . . .	29
5.8	Flattening of curves at high values of $\tilde{\delta}_p$ . A total lack of convergence was observed when $\tilde{\delta}_p \simeq 1000$ in all cases for all the variables. Along with the equilibrium variables the non-equilibrium variables $\underline{q}$ and $\underline{\sigma}$ are also shown here. . . . .	30
5.9	Structured mesh convergence comparison. It can be seen here how the use of structured meshes in place of unstructured meshes does not improve the convergence. . . . .	31
5.10	Mesh comparison. . . . .	31
6.1	Schematic of the Lid-Driven Cavity problem. . . . .	35
6.2	Linearized R13 equations solution of the Lid-Driven Cavity problem for the case of $u_0 = 1$ , $\text{Kn} = 0.0001$ and constant temperature boundaries. The streamlines plot and velocity profiles along $x/L = 0.5$ and $y/L = 0.5$ are shown. . . . .	36

6.3	Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for $Re = 100$ . The FenicsR13 solution corresponds to $Ma = 0.01$ and $Kn = 0.0001$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982. . . . .	37
6.4	Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for $Re = 1000$ . The FenicsR13 solution corresponds to $Ma = 0.1$ and $Kn = 0.0001$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982. . . . .	38
6.5	Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for $Re = 3200$ . The FenicsR13 solution corresponds to $Ma = 0.096$ and $Kn = 0.00003$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982. . . . .	39
6.6	Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for $Re = 5000$ . The FenicsR13 solution corresponds to $Ma = 0.1$ and $Kn = 0.00002$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982. . . . .	40
6.7	Schematic of the non-isothermal lid-driven cavity problem . . . . .	42
6.8	Contours produced by FLUENT (left) and COMSOL (right) for the non-isothermal lid-driven cavity case . . . . .	43
6.9	Profiles produced by FLUENT and COMSOL for the non-isothermal lid-driven cavity case . . . . .	44
B.1	Case I . . . . .	50
B.2	Case II . . . . .	51
B.3	Case III . . . . .	52
B.4	Case IV . . . . .	53

# List of Tables

4.1	Examples of Source Function . . . . .	15
5.1	Suggested range for stabilization parameters . . . . .	32

# Chapter 1

## Introduction

The Navier-Stokes equations have been the most popularly used equations to describe gas flows in recent times. Despite its popularity, these equations are based on a continuum assumption and are therefore not valid in all situations. The continuum assumption treats the fluids as continuous, even though in reality they are composed of molecules on a microscopic scale. The Knudsen number,  $\text{Kn}$ , which is defined as the ratio between the mean free path of the molecules and the length scale of the process, determines how valid this continuum assumption is for a given process. In other words, the Navier-Stokes equations are valid at very low Knudsen numbers ( $\text{Kn} \rightarrow 0$ ). Since most phenomena relevant to our daily lives fall in this regime, these equations often provide sufficiently accurate results. However, there are processes for which the Knudsen number is substantial ( $\text{Kn} > 0.1$ ), such as flow past re-entry vehicle and flow related to microtechnology, where the gas dynamics are not accurately predicted by the Navier-Stokes equations. In these situations, we turn to statistical approaches such as the Boltzmann equations, wherein individual particle dynamics are accounted for through microscopic descriptions.

Although the Boltzmann equations are effective in predicting rarefied gas flows accurately, they are often computationally too expensive. Therefore, extended macroscopic models turn out to be an attractive alternative. Similar to the classical models, these models consist of compact sets of partial differential equations involving the important gas quantities. The regularized 13-moment (R13) equations (H.Struchtrup and M.Torrilhon 2003) forms one such example of these macroscopic models. The ability of the R13 equations to predict rarefaction effects at moderate computational costs makes these equations an exciting prospect for the future.

### 1.1 Research Background

The advantages of the R13 equations described above have engendered active research into numerical methods to solve these equations over the past decade. Initially, a finite difference scheme was employed to approximate the steady-state R13 equations in A.Rana, M.Torrilhon and H.Struchtrup 2013. Comparisons with the solutions obtained from the computationally expensive Direct Simulation Monte Carlo (DSMC) validated the accuracy of the solutions to the R13 equations. Following this, an implicit discontinuous Galerkin method was used for hierarchical simulations (M.Torrilhon and N.Sarna 2017) where the solutions to the steady-state linearized R13 equations were compared to the lower order models such as the NSF (Navier-Stokes-Fourier) system. In this report, we focus on the finite element approach to solving the R13 equations.

Recently, the use of finite element methods for the R13 equations was presented in A.Westerkamp 2017; A.Westerkamp and M.Torrilhon 2019 along with methods to tackle the resulting instabilities using stabilization schemes (A.Westerkamp 2012; A.Westerkamp and M.Torrilhon 2014, 2017). Based on these works, FenicsR13, an FEM solver for the linearized R13 equations built on the FEniCS simulation framework, was developed (L.Theisen and M.Torrilhon 2020).

## 1.2 FenicsR13

FenicsR13 (L.Theisen and M.Torrilhon 2020) is a mixed finite element solver designed to solve the linearized R13 equations of non-equilibrium gas dynamics. This solver is built on Python using the FEniCS computing platform. In FenicsR13, the capabilities of the FEniCS's Uniform Form Language (UFL) has been extended to accommodate the tensorial implementation of the variational formulation in FEniCS. FenicsR13 can be used to solve 2D rarefied gas flow problems using a variety of mixed finite elements. A CIP stabilization scheme has been implemented in FenicsR13 in order to remove the restriction on the choice of finite elements for the field variables posed by the Ladyzenskaja-Babuska-Brezzi (LBB) condition (see Section 3.1). The choice of the stabilization parameters involved in the CIP scheme is crucial for the accuracy of the solver, yet, barring a brief look in (A.Westerkamp 2017), there are not many resources that would help make an educated choice here. This is precisely the gap that the first part of this thesis hopes to fill.

## 1.3 Thesis objectives

Following the motivation provided in the previous section, the primary objectives of this thesis are listed below.

- *Development of a script to generate analytical solutions over the annular domain:* We aim to study the stabilization parameters through analysis of convergence curves, and analytical solutions are a pre-requisite for this. Therefore, we seek to develop a script that can generate analytical solutions over an annular domain based on user-specified boundary conditions. To extend the utility of this script, we export the analytical solution in three different formats that can be used by future solvers of the linearized R13 equations.
- *Investigation of the stabilization parameters associated with the CIP scheme:* We aim to gain more insight into the selection of values and scaling for the stabilization parameters associated with the CIP scheme used in FenicsR13. We plan to execute extensive parametric studies to study the effect of the choice of stabilization parameters on the accuracy and convergence of FenicsR13 and make suggestions based on the results.
- *Extension of the FenicsR13 solver to solve the non-linear conservation equations:* Currently, the FenicsR13 solver solves the linearized R13 equations. This linearization restricts the ability of the solver to capture the complex non-linear physics of the flows accurately. We aim to modify the FenicsR13 solver to solve the non-linear conservation laws in the hopes that it will allow for more accurate predictions of the flows.

## 1.4 Thesis Outline

The report has been constructed in a structured manner in order to ensure a comprehensive and cogent thesis. Here we describe the outline of the thesis with a brief introduction to each chapter for the sake of the reader's convenience.

- *Chapter 1:* A brief explanation of the need for a statistical description of gas dynamics in certain scenarios is discussed. The R13 equations are mentioned, and its advantages in comparison to the classical Navier-Stokes equations and DSMC approaches are explained. Following this, the report provides a glimpse into the attempts to solve the R13 equations numerically in recent times with particular emphasis placed on FenicsR13 the finite element solver for solving the linearized R13 equations. Finally, the objectives of the thesis are stated and described.

- *Chapter 2*: As a starting point for the thesis, a theoretical foundation explaining the concepts relevant to this thesis is established in this chapter. Firstly, the Kinetic Gas Theory is introduced to the reader, along with the renowned Boltzmann equation. Following this, extended macroscopic models for the Boltzmann equation are explained, with particular emphasis on Grad's method of moments. With this background a transition to the R13 equations is made explaining clearly how it is an extension of Grad's 13-moment equations. The chapter then concludes with a discussion of the linearized R13 equations and the associated boundary conditions.
- *Chapter 3*: Since one of the main objectives of this thesis revolves around the CIP stabilization scheme, the need for stabilization when employing the finite element approach to solving the R13 equations is thoroughly explained. The concept of saddle point problems is important in understanding this need for stabilization. Therefore, simple examples are used to help the reader understand this concept and see how it relates to the saddle points students are introduced to in multi-variable calculus. Finally, the exact implementation of the CIP stabilization in the FenicsR13 solver is explained and shown.
- *Chapter 4*: This chapter is dedicated to introducing the Python script developed to generate analytical solutions of the linearized R13 equations over an annular domain. Starting with a description of the annular domain set-up, the different boundary conditions and source functions that are allowed to be specified are explained. Then a step-by-step procedure to using the script is provided along with examples. In order to illustrate how the generated solutions can be used in tandem with other solvers, simple examples are provided which can be used as references for future solvers.
- *Chapter 5*: The parametric study investigating the stabilization parameters involved in the CIP scheme, is delineated in this chapter. The four cases used in this study are explained in terms of the boundary conditions and source terms. Following this, the results of the parametric studies are presented along with relevant discussions and observations. The chapter is concluded with a summary of the observations and a suggested optimal range for the stabilization values based on the results of the study.
- *Chapter 6*: The extension of the FenicsR13 solver to solve non-linear conservation equations is discussed in this chapter. A new scaling for the R13 equations is proposed which helps improve the convergence of the solver once the non-linear extension has been implemented. The non-linear extension with respect to the momentum equation is validate against established results for the problem of a lid-driven cavity for various Reynolds numbers. Finally, a non-isothermal lid-driven cavity is constructed in order to validate the complete non-linear conservation equations extension. Solutions for this problem are generated using ANSYS FLUENT and COMSOL, and the results are presented.

# Chapter 2

## The R13 Equations

Since this thesis is fundamentally based on solving the R13 equation set, it is crucial to understand the roots of the R13 system and its derivation. This chapter is dedicated to that very purpose. A solid foundation is laid with an introduction to the Kinetic Gas Theory and the Boltzmann equations. Following this, the derivation of extended macroscopic models using the method of moments is described. Finally, this chapter is concluded with the derivation of the R13 equations (H.Struchtrup and M.Torrilhon 2003) and its corresponding boundary conditions.

### 2.1 Kinetic Gas Theory

In the Kinetic Gas Theory, the state of the gas is described through the phase density, or distribution function,  $f(\underline{x}, t, \underline{c})$ . Physically, this phase density represents the probability of finding particles with velocity  $\underline{c}$  at the location  $\underline{x}$  at the time instant  $t$ . An integration of the phase density over the phase space  $\Delta\underline{c}$  and the spatial domain  $\Delta\underline{x}$  results in the number of particles with velocity in the range  $\Delta\underline{c}$  located in the domain  $\Delta\underline{x}$  at the time instant  $t$  as

$$N_{\Delta\underline{x}, \Delta\underline{c}}(t) = \int_{\Delta\underline{x}} \int_{\Delta\underline{c}} f(\underline{x}, t, \underline{c}) \, d\underline{c} \, d\underline{x} \quad (2.1)$$

Once the phase density  $f$  is known, it can be used to obtain the other quantities describing the gas dynamics such as mass density  $\rho(\underline{x}, t)$ , velocity  $u(\underline{x}, t)$ , internal energy  $\epsilon(\underline{x}, t)$  and so on. A couple of examples are shown below

$$\rho(\underline{x}, t) = m \int_{\mathbb{R}^3} f(\underline{x}, t, \underline{c}) \, d\underline{c} \quad (2.2)$$

$$u_i(\underline{x}, t) = \frac{m}{\rho(\underline{x}, t)} \int_{\mathbb{R}^3} c_i f(\underline{x}, t, \underline{c}) \, d\underline{c} \quad (2.3)$$

$$\epsilon(\underline{x}, t) = \frac{m}{\rho(\underline{x}, t)} \int_{\mathbb{R}^3} \frac{1}{2} (c_i - u_i)^2 f(\underline{x}, t, \underline{c}) \, d\underline{c} \quad (2.4)$$

where  $m$  denotes the mass of each particle.

Based on this, we can say that the knowledge of the phase density  $f$  is sufficient to describe the gas dynamics in its entirety. However, the solution to  $f$  is not quite straightforward as collisions between particles lead to energy and momentum exchange which in turn causes the phase density to change in space and time. The Boltzmann equation shown below describes the evolution of  $f$  through collisions and free flight.

$$\frac{\partial f}{\partial t} + c_i \frac{\partial f}{\partial x_i} + G_k \frac{\partial f}{\partial c_k} = S \quad (2.5)$$

Here  $G_k$  denotes the external body forces on the particles (e.g. gravity) and  $S$ , the collision integral, represents the change in phase density due to interaction between particles.

## 2.2 Grad's Method of Moments

The higher dimensionality of the phase space  $f$  makes its direct computation using the Boltzmann equation computationally expensive. This has consequentially engendered many extended macroscopic models based on the Boltzmann equation, which offer a much more modest computational cost. The popular examples of these would be the Chapman-Enskog (CE) expansion and the method of moments. A detailed explanation of these methods can be found in (H.Struchtrup 2005).

The fundamental assumption in the method of moments is that the state of the gas can be described by an extended set of moments as

$$u_A = \int_{\mathbb{R}^3} \Psi_A(c_k) f(\underline{x}, t, \underline{c}) d\underline{c} \quad (2.6)$$

using a vector of polynomials  $\Psi_A$ . The system of moment equations to be solved are obtained by multiplying the Boltzmann equation with  $\Psi_A$  and integrating over the velocity space. These equations, however, do not form a closed system of partial differential equations as they contain higher-order moments that require explicit closures.

Grad's 13-moment system (G13) uses the vector  $\Psi_A = m \{1, c_i, \frac{1}{2}C^2, C_{\langle i}C_{j \rangle}, \frac{1}{2}C^2C_i\}$  and produces  $u_A = \{\rho, \rho u_i, \rho \epsilon, \sigma_{ij}, s_i\}$  where  $u_i$ ,  $\epsilon$ ,  $\sigma_{ij}$  and  $s_i$  denote the velocity, internal energy, stress tensor and heat flux, respectively. Here  $C$  is the peculiar velocity (a.k.a barycentric velocity) defined as  $C_i = c_i - u_i$  and the notation  $A_{\langle ij \rangle}$  represents the symmetric and deviatoric part of the tensor  $A_{ij}$  as  $A_{\langle ij \rangle} = \frac{1}{2}(A_{ij} + A_{ji}) - \frac{1}{3}A_{kk}\delta_{ij}$  (a detailed explanation of this notation can be found in the Appendix of (H.Struchtrup 2005)). The resulting moment equations, neglecting external body forces, are shown below as

$$\frac{\partial u_A}{\partial t} + \frac{\partial F_{Ak}}{\partial x_k} = P_A \quad (2.7)$$

where  $F_{Ak} = \int_{\mathbb{R}^3} \Psi_A c_k f d\underline{c}$  and  $P_A = \int_{\mathbb{R}^3} \Psi_A S f d\underline{c}$ . This system is not closed as it contains the fluxes of the moments  $F_{Ak}$  and the production terms  $P_A$  which are not a priori related to  $u_A$ . In order to deal with this closure problem, Grad defines  $f_{|G}$  as an expansion of the Maxwellian into Hermite polynomials as

$$f_{|G} = \left( a + a_i \frac{\partial}{\partial C_i} + a_{ij} \frac{\partial^2}{\partial C_i \partial C_j} + \dots \right) f_M \quad (2.8)$$

where  $f_M = \frac{\rho}{m} \left( \frac{m}{2\pi kT} \right)^{3/2} e^{-(m/2kT)C^2}$  is the local Maxwellian. Since the phase density needs to produce  $u_A$  as its moments, the coefficients  $a_{i_1 i_2 \dots i_n}$  can be determined according to

$$u_A = \int_{\mathbb{R}^3} \Psi_A(c_k) \left( a + a_i \frac{\partial}{\partial C_i} + a_{ij} \frac{\partial^2}{\partial C_i \partial C_j} + \dots \right) f_M d\underline{c} \quad (2.9)$$

Upon doing this the phase density assumes the form  $f_{|G} = f_{|G}(u_A(\underline{x}, t), C_i)$  and only depends on space and time through the moments. This allows for the fluxes  $F_{Ak}$  and production terms  $P_A$  to be computed as functions of the moments

$$F_{Ak} = F_{Ak}(u_B) \quad , P_A = P_A(u_B) \quad (2.10)$$

and the system of moment equations is closed.

The final form of Grad's 13-moment equations is given by

$$\frac{D\rho}{Dt} + \rho \frac{\partial u_k}{\partial x_k} = 0 \quad (2.11)$$

$$\rho \frac{Du_i}{Dt} + \theta \frac{\partial \rho}{\partial x_i} + \rho \frac{\partial \theta}{\partial x_i} + \frac{\partial \sigma_{ik}}{\partial x_k} = \rho G_i \quad (2.12)$$

$$\frac{3}{2} \rho \frac{D\theta}{Dt} + \rho \theta \frac{\partial u_k}{\partial x_k} + \frac{\partial s_k}{\partial x_k} + \sigma_{kl} \frac{\partial u_k}{\partial x_l} = 0 \quad (2.13)$$

$$\frac{D\sigma_{ij}}{Dt} + \frac{4}{5} \frac{\partial s_{\langle i}}{\partial x_{j\rangle}} + 2\sigma_{k\langle i} \frac{\partial u_{j\rangle}}{\partial x_k} + \sigma_{ij} \frac{\partial u_k}{\partial x_k} + 2\rho \theta \frac{\partial u_{\langle i}}{\partial x_{j\rangle}} = -\frac{p}{\mu} \sigma_{ij} \quad (2.14)$$

$$\frac{Ds_i}{Dt} + \frac{5}{2} \rho \theta \frac{\partial \theta}{\partial x_i} + \frac{5}{2} \sigma_{ik} \frac{\partial \theta}{\partial x_k} - \sigma_{ik} \theta \frac{\partial \ln \rho}{\partial x_k} - \frac{\sigma_{ik}}{\rho} \frac{\partial \sigma_{kl}}{\partial x_l} + \theta \frac{\partial \sigma_{ik}}{\partial x_k} \quad (2.15)$$

$$+ \frac{7}{5} s_i \frac{\partial u_k}{\partial x_k} + \frac{7}{5} s_k \frac{\partial u_i}{\partial x_k} + \frac{2}{5} s_k \frac{\partial u_k}{\partial x_i} = -\frac{2}{3} \frac{p}{\mu} s_i \quad (2.16)$$

## 2.3 Regularized 13-Moment Equations

While the G13 system is capable of predicting certain rarefaction effects, it fails in predicting shock wave profiles at large enough Mach numbers. In (H.Struchtrup and M.Torrilhon 2003) the R13 equations were introduced as the G13 set with regularization applied to take care of the highest-order terms based on a pseudo-equilibrium approach. This makes the equations non-hyperbolic and yields continuous shock structures across all Mach numbers.

### 2.3.1 R13 Equations

The R13 system in its complete non-linear form is given by

$$\frac{D\rho}{Dt} + \rho \frac{\partial u_k}{\partial x_k} = 0 \quad (2.17)$$

$$\rho \frac{Du_i}{Dt} + \theta \frac{\partial \rho}{\partial x_i} + \rho \frac{\partial \theta}{\partial x_i} + \frac{\partial \sigma_{ik}}{\partial x_k} = 0 \quad (2.18)$$

$$\frac{3}{2} \rho \frac{D\theta}{Dt} + \rho \theta \frac{\partial u_k}{\partial x_k} + \frac{\partial s_k}{\partial x_k} + \sigma_{kl} \frac{\partial u_k}{\partial x_l} = 0 \quad (2.19)$$

$$\frac{D\sigma_{ij}}{Dt} + \sigma_{ij} \frac{\partial u_k}{\partial x_k} + \frac{4}{5} \frac{\partial s_{\langle i}}{\partial x_{j\rangle}} + 2p \frac{\partial u_{\langle i}}{\partial x_{j\rangle}} + 2\sigma_{k\langle i} \frac{\partial u_{j\rangle}}{\partial x_k} + \frac{\partial m_{ijk}}{\partial x_k} = -\frac{p}{\mu} \sigma_{ij} \quad (2.20)$$

$$\frac{Ds_i}{Dt} + \frac{5}{2} p \frac{\partial \theta}{\partial x_i} + \frac{5}{2} \sigma_{ik} \frac{\partial \theta}{\partial x_k} + \theta \frac{\partial \sigma_{ik}}{\partial x_k} - \theta \sigma_{ik} \frac{\partial \ln \rho}{\partial x_k} + \frac{7}{5} s_k \frac{\partial u_i}{\partial x_k} + \frac{2}{5} s_k \frac{\partial u_k}{\partial x_i} \quad (2.21)$$

$$+ \frac{7}{5} s_i \frac{\partial u_k}{\partial x_k} + \frac{1}{2} \frac{\partial R_{ik}}{\partial x_k} + \frac{1}{6} \frac{\partial \Delta}{\partial x_i} + m_{ijk} \frac{\partial u_j}{\partial x_k} - \frac{\sigma_{ij}}{\rho} \frac{\partial \sigma_{jk}}{\partial x_k} = -\frac{2}{3} \frac{p}{\mu} s_i \quad (2.22)$$

with closure for the highest order moments  $\Delta$ ,  $R_{ij}$  and  $m_{ijk}$  given as

$$\Delta = -12 \frac{\mu}{p} \left[ \theta \frac{\partial s_k}{\partial x_k} + \frac{5}{2} s_k \frac{\partial \theta}{\partial x_k} - \theta s_k \frac{\partial \ln \rho}{\partial x_k} - \frac{s_k}{\rho} \frac{\partial \sigma_{kl}}{\partial x_l} + \theta \sigma_{kl} \frac{\partial u_k}{\partial x_i} \right], \quad (2.23)$$

$$R_{ij} = -\frac{24}{5} \frac{\mu}{p} \left[ \theta \frac{\partial s_{\langle i}}{\partial x_{j\rangle}} + s_{\langle i} \frac{\partial \theta}{\partial x_{j\rangle}} - \theta s_{\langle i} \frac{\partial \ln \rho}{\partial x_{j\rangle}} - \frac{s_{\langle i}}{\rho} \frac{\partial \sigma_{j\rangle k}}{\partial x_k} \right] \quad (2.24)$$

$$+ \frac{5}{7} \theta \left[ \sigma_{k\langle i} \frac{\partial u_{j\rangle}}{\partial x_k} + \sigma_{k\langle i} \frac{\partial u_k}{\partial x_{j\rangle}} - \frac{2}{3} \sigma_{ij} \frac{\partial u_k}{\partial x_k} \right] - \frac{5}{6} \frac{\sigma_{ij}}{\rho} \frac{\partial s_k}{\partial x_k} - \frac{5}{6} \frac{\sigma_{ij} \sigma_{kl}}{\rho} \frac{\partial u_k}{\partial x_l} \quad (2.25)$$

$$m_{ijk} = -2 \frac{\mu}{p} \left[ \theta \frac{\partial \sigma_{\langle ij}}{\partial x_{k\rangle}} - \theta \sigma_{\langle ij} \frac{\partial \ln \rho}{\partial x_{k\rangle}} + \frac{4}{5} s_{\langle i} \frac{\partial u_{j\rangle}}{\partial x_k} - \frac{\sigma_{\langle ij}}{\rho} \frac{\partial \sigma_{k\rangle l}}{\partial x_l} \right]. \quad (2.26)$$

### 2.3.2 R13 Boundary Conditions

In addition to the three field variables (temperature  $\theta$ , pressure  $p$  and velocity  $\underline{u}$ ) which are solved for in classical gas dynamics, the R13 equations also solve for the field variables heat flux  $\underline{s}$  and stress tensor  $\underline{\sigma}$ . This necessitates an increased number of boundary conditions. A commonly used model in deriving the wall boundary conditions for the moment equations is the Maxwell accommodation model. This model assumes that a fraction  $\chi$  of the particles hitting the walls are absorbed and re-injected into the gas domain according to a certain distribution function. In our case, we assume this function to be the Maxwellian  $f_M$ . The remaining  $(1 - \chi)$  fraction of the particles are specularly reflected. Accordingly, the distribution function in an infinitesimal neighbourhood of a wall is given by

$$\tilde{f}(\underline{c}) = \begin{cases} \chi f_M(\underline{c}) + (1 - \chi) f_{\text{gas}}^{(*)}(\underline{c}) & \underline{n} \cdot (\underline{c} - \underline{u}_w) > 0 \\ f_{\text{gas}}(\underline{c}) & \underline{n} \cdot (\underline{c} - \underline{u}_w) < 0 \end{cases} \quad (2.27)$$

where  $\underline{n}$  is the normal on the wall pointing into the domain and  $f_{\text{gas}}^{(*)}$  is the distribution function corresponding to the transformed velocities resulting from reflection. Although these equations are non-linear in their fullest extent, we will be considering the linearized version which is shown in Section 2.3.3.

### 2.3.3 Linearized R13 Equations

A linearized steady-state variant of the full non-linear R13 equation set is solved in FenicsR13. This linearization is carried out by considering the unknown field variables as a first-order perturbation of an equilibrium ground state (A.Westerkamp 2017). Physically this makes these equations quite appropriate for cases that involve slow flows (low Mach number). Additionally, the steady-state assumption leads to the exclusion of the temporal derivative terms. The resulting linear equation set consists of three balance laws (mass, energy and momentum), two evolution equations for the non-equilibrium quantities (heat flux  $\underline{s}$  and stress tensor  $\underline{\sigma}$ ) and three closure equations for the highest-order moments  $\underline{\underline{m}}$ ,  $\underline{\underline{R}}$  and  $\Delta$ . The mass source, heat source and body force vector are represented by  $f_m$ ,  $f_h$  and  $\underline{\underline{f}}_b$ , respectively, in the equations below.

$$\nabla \cdot \underline{u} = f_m \quad (2.28)$$

$$\nabla \cdot \underline{s} + \nabla \cdot \underline{u} = f_h \quad (2.29)$$

$$\nabla p + \nabla \cdot \underline{\sigma} = \underline{\underline{f}}_b \quad (2.30)$$

$$\frac{5}{2} \nabla \theta + \nabla \cdot \underline{\sigma} + \frac{1}{2} \nabla \cdot \underline{\underline{R}} + \frac{1}{6} \nabla \Delta = -\frac{1}{\text{Kn}} \frac{2}{3} \underline{s} \quad (2.31)$$

$$\frac{4}{5} (\nabla \underline{s})_{STF} + 2 (\nabla \underline{u})_{STF} + \nabla \cdot \underline{\underline{m}} = -\frac{1}{\text{Kn}} \underline{\sigma} \quad (2.32)$$

Closure for the highest-order moments  $\underline{\underline{m}}$ ,  $\underline{\underline{R}}$  and  $\Delta$  are provided by the following equations.

$$\underline{\underline{m}} = -2\text{Kn} (\nabla \underline{\sigma})_{STF} \quad (2.33)$$

$$\underline{\underline{R}} = -\frac{24}{5} \text{Kn} (\nabla \underline{s})_{STF} \quad (2.34)$$

$$\Delta = -12\text{Kn} (\nabla \cdot \underline{s}) \quad (2.35)$$

where the deviatoric part of the symmetric tensor is denoted by  $(\cdot)_{STF}$ . The above equations are all in the dimensionless form, and the Knudsen number  $\text{Kn}$  describes the rarefaction situation of the gas being considered. A detailed derivation of the analytical solution, including details about the ansatz used, for the cases of slow, rarefied gas flows past a cylinder and a sphere can be found in (A.Westerkamp and M.Torrilhon 2012; M.Torrilhon 2010).

After appropriate linearization and simplification for a two-dimensional space considering a boundary-aligned coordinate system  $(n, t)$ , the boundary condition equations for the R13 system, which are non-linear in their fullest extent, look as shown below.

$$\epsilon_p^w (u_n - u_n^w) = \epsilon_v^w \tilde{\chi} ((p - p^w) + \epsilon_s \sigma_{nn}) \quad (2.36)$$

$$\sigma_{nt} = \tilde{\chi} \left( (u_t - u_t^w) + \frac{1}{5} s_t + m_{nnt} \right) \quad (2.37)$$

$$R_{nt} = \tilde{\chi} \left( -(u_t - u_t^w) + \frac{11}{5} s_t - m_{nnt} \right) \quad (2.38)$$

$$s_n = \tilde{\chi} \left( 2(\theta - \theta^w) + \frac{1}{2} \sigma_{nn} + \frac{2}{5} R_{nn} + \frac{2}{15} \Delta \right) \quad (2.39)$$

$$m_{nnn} = \tilde{\chi} \left( -\frac{2}{5} (\theta - \theta^w) + \frac{7}{5} \sigma_{nn} - \frac{2}{25} R_{nn} - \frac{2}{75} \Delta \right) \quad (2.40)$$

$$\left( \frac{1}{2} m_{nnn} + m_{ntt} \right) = \tilde{\chi} \left( \frac{1}{2} \sigma_{nn} + \sigma_{tt} \right) \quad (2.41)$$

where  $\theta^w, u_n^w, u_t^w$  and  $p^w$  represent the temperature, normal velocity, tangential velocity and pressure at the wall. The parameters  $\epsilon_p^w, \epsilon_v^w$  and  $\epsilon_s$  are used for the inflow modelling and the modified Maxwell accommodation factor is given by  $\tilde{\chi} = \sqrt{\frac{2}{\pi \theta_0}} \frac{\chi}{2 - \chi}$  (A.Westerkamp and M.Torrilhon 2019).

It is important to observe how Eq. 2.36 is structured to understand the inflow model being used here. The parameters involved are the normal velocity prescription  $\epsilon_v^w$ , the pressure prescription  $\epsilon_p^w$  and the stress prescription  $\epsilon_s$ . When  $\epsilon_p^w \gg \epsilon_v^w$  there is greater emphasis on the normal velocity boundary condition  $u_n = u_n^w$  and when  $\epsilon_v^w \gg \epsilon_p^w$  there is greater emphasis on the pressure boundary condition  $p^w = p + \epsilon_s \sigma_{nn}$ . Clearly the stress prescription  $\epsilon_s$  represents the emphasis of the stress term within the pressure boundary condition. For example, a standard zero normal velocity boundary condition can be enabled by setting  $\epsilon_v^w = 0$  and  $u_n^w = 0$ .

### 2.3.4 Weak Formulation

The first step in using the finite element method is to obtain the weak formulation of the equations being solved. To do this, the equation set (Eq 2.28 to 2.32) shown in Section 2.3.3 is multiplied with an appropriate set of test functions  $(\kappa, q, \underline{v}, \underline{r}, \underline{\psi})$  and integrated over the entire computational domain  $\Omega$ . The final discrete weak form of the equations is given by

**Find**  $(\theta, p, \underline{u}, \underline{s}, \underline{\sigma}) \in V_\theta^h \times V_p^h \times V_{\underline{u}}^h \times V_{\underline{s}}^h \times V_{\underline{\sigma}}^h$  **such that**

$$\int_{\Omega} \underline{u} \cdot \nabla q \, d\mathbf{x} - \int_{\Gamma} \left( \frac{\epsilon_v^w}{\epsilon_p^w} \tilde{\chi} ((p - p^w) + \epsilon_s \sigma_{nn}) + u_n^w \right) q \, dl + \int_{\Omega} f_m q \, d\mathbf{x} = 0 \quad (2.42)$$

$$- \int_{\Omega} (\nabla \cdot \underline{s}) \kappa \, d\mathbf{x} + \int_{\Omega} (f_h - f_m) \kappa \, d\mathbf{x} = 0 \quad (2.43)$$

$$\int_{\Omega} (\nabla \cdot \underline{\sigma}) \cdot \underline{v} \, d\mathbf{x} + \int_{\Omega} \nabla p \cdot \underline{v} \, d\mathbf{x} - \int_{\Omega} \underline{f}_b \cdot \underline{v} \, d\mathbf{x} = 0 \quad (2.44)$$

$$\begin{aligned} & \frac{12}{5} \text{Kn} \int_{\Omega} (\nabla \underline{s})_{STF} : \nabla \underline{r} \, d\mathbf{x} + \frac{2}{3} \frac{1}{\text{Kn}} \int_{\Omega} \underline{s} \cdot \underline{r} \, d\mathbf{x} - \frac{5}{2} \int_{\Omega} \theta (\nabla \cdot \underline{r}) \, d\mathbf{x} \\ & \int_{\Omega} (\nabla \cdot \underline{\sigma}) \cdot \underline{r} \, d\mathbf{x} + \int_{\Gamma} \left( \frac{5}{4\tilde{\chi}} s_n - \frac{5}{8} \sigma_{nn} + \frac{5}{2} \theta^w \right) r_n \, dl \\ & + 2\text{Kn} \int_{\Omega} (\nabla \cdot \underline{s}) (\nabla \cdot \underline{r}) \, d\mathbf{x} + \int_{\Gamma} \left( \frac{6\tilde{\chi}}{5} s_t - \frac{1}{2} \sigma_{nt} \right) r_t \, dl = 0 \end{aligned} \quad (2.45)$$

$$\begin{aligned} & 2\text{Kn} \int_{\Omega} (\nabla \underline{\sigma})_{STF} : \nabla \underline{\psi} \, d\mathbf{x} + \frac{1}{\text{Kn}} \int_{\Omega} \underline{\sigma} : \underline{\psi} \, d\mathbf{x} - 2 \int_{\Omega} \underline{u} \cdot (\nabla \cdot \underline{\psi}) \, d\mathbf{x} \\ & + \frac{4}{5} \int_{\Omega} (\nabla \underline{s})_{STF} : \underline{\psi} \, d\mathbf{x} + \int_{\Gamma} \left( \frac{9}{4} \tilde{\chi} \sigma_{nn} - \frac{3}{10} s_n \right) \psi_{nn} \, dl \\ & + \int_{\Gamma} 2\tilde{\chi} \left( \sigma_{tt} + \frac{1}{2} \sigma_{nn} \right) \left( \psi_{tt} + \frac{1}{2} \psi_{nn} \right) \, dl \\ & + \int_{\Gamma} \left( \frac{2}{\tilde{\chi}} \sigma_{nt} + 2u_t^w - \frac{2}{5} s_t \right) \psi_{nt} \, dl \\ & + \int_{\Gamma} \left( 2 \frac{\epsilon_v^w}{\epsilon_p^w} \tilde{\chi} ((p - p^w) + \epsilon_s \sigma_{nn}) + 2u_n^w \right) \psi_{nn} \, dl = 0 \end{aligned} \quad (2.46)$$

**for all**  $(\kappa, q, \underline{v}, \underline{r}, \underline{\psi}) \in V_\theta^h \times V_p^h \times V_{\underline{u}}^h \times V_{\underline{s}}^h \times V_{\underline{\sigma}}^h$ .

where  $V_\theta^h, V_p^h, V_{\underline{u}}^h, V_{\underline{s}}^h, V_{\underline{\sigma}}^h$  represent suitable discrete subspaces of the discrete Sobolev space  $H^1(\Omega)$ . In the above equations integration by parts has been carried out and the boundary conditions (Eq 2.36 to 2.41) have been rearranged and naturally inserted (see L.Theisen and M.Torrilhon 2020). Therefore, there is no need to explicitly enforce essential boundary conditions.

# Chapter 3

## Stabilization

The steady-state linearized R13 equations in two dimensions consist of a complex two-fold saddle point structure which needs to be dealt with when attempting to solve the equations numerically. Familiarity with saddle point problems is essential in understanding why stabilization is a necessary part of the numerical method employed. Therefore, a brief introduction to saddle point problems and methods to deal with them are discussed in this section. Furthermore, the rationale behind choosing to implement stabilization instead of using mixed elements is explained. Finally, the actual implementation of the CIP stabilization in the fenicsR13 solver (L.Theisen and M.Torrilhon 2020) is explicitly depicted and elaborated.

### 3.1 Saddle Point Problems

A general saddle point problem involves a functional of two arguments (say  $L(w, q)$ ) and the objective is to find a pair of arguments (say  $(u, p)$ ) such that the resulting value is minimized with respect to the first variable ( $u$ ) and maximized with respect to the other variable ( $p$ ). Mathematically this can be represented as

$$L(u, q) \leq L(u, p) \leq L(w, p), \quad \forall w, q \quad (3.1)$$

Another way to write this would be

$$L(u, p) = \min_v \max_q L(v, q) = \max_q \min_v L(v, q) \quad (3.2)$$

A finite-dimensional example of such a saddle point problem arises when dealing with quadratic programming. Consider the minimization problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T A x - b^T x \quad (3.3)$$

, where  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^{n \times 1}$  subject to a constraint

$$Bx = g \quad (3.4)$$

where  $B \in \mathbb{R}^{m \times n}$  ( $m < n$ ). Restating the same problem after introducing a Lagrangian multiplier  $y$  as shown below

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} L(x, y) := \frac{1}{2} x^T A x - b^T x + y^T (Bx - g) \quad (3.5)$$

helps us notice that this problem is indeed a saddle point problem. Further, the necessary conditions for a solution pair  $(u, p)$  are

$$\nabla_x L(\mathbf{u}, \mathbf{p}) = \mathbf{A}\mathbf{u} + \mathbf{B}^T \mathbf{p} - \mathbf{b} = 0 \quad (3.6)$$

$$\nabla_y L(\mathbf{u}, \mathbf{p}) = \mathbf{B}\mathbf{u} - \mathbf{g} = 0 \quad (3.7)$$

which can be written in a matrix form as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{g} \end{bmatrix} \quad (3.8)$$

Saddle point problems of this kind tend to arise as a result of Galerkin discretization. A classic example of this would be Stokes problem which describes the flow of a highly viscous fluid. The variables solved for in this case are the pressure  $p$  and velocity vector  $\mathbf{u}$ . This case has been analysed extensively (J.Donea and A.Huerta 2003), and the resulting equations after testing with  $(\mathbf{q}, \mathbf{u})$  look as follows.

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) + \mathbf{b}(\mathbf{v}, \mathbf{p}) = \mathbf{l}(\mathbf{v}), \quad \forall \mathbf{v} \quad (3.9)$$

$$\mathbf{b}(\mathbf{u}, \mathbf{q}) = 0, \quad \forall \mathbf{q} \quad (3.10)$$

where

$$\mathbf{a}(\mathbf{u}, \mathbf{v}) := \sum_{j=1}^3 \int_{\Omega} \nabla \mathbf{u}_j \cdot \nabla \mathbf{v}_j \, dx, \quad \mathbf{b}(\mathbf{v}, \mathbf{p}) := - \int_{\Omega} \operatorname{div} \mathbf{v} \, p \, dx, \quad (3.11)$$

and

$$\mathbf{l}(\mathbf{v}) := \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx \quad (3.12)$$

Again the structure of the final equations (Eq 3.9 and 3.10) is similar to that in the quadratic programming case (Eq 3.8).

The linearized R13 equations considered in this report contains a two-fold saddle-point structure and, a comprehensive illustration of this can be found in (L.Theisen and M.Torrilhon 2020). In the mentioned work a rigorous derivation is presented and it can be clearly seen how the weak form of the linearized R13 equations leads to a discrete algebraic system of the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (3.13)$$

which is again of the same form as Eq 3.8. In this equation  $\mathbf{x} = (\underline{\sigma}, \underline{\varepsilon}, p)$  and  $\mathbf{y} = (\mathbf{u}, \theta)$ . Here  $\mathbf{A}, \mathbf{B}, \mathbf{f}$  and  $\mathbf{g}$  consist of sub-functionals corresponding to the discrete weak form (Eq 2.42 to 2.46)

The Ladyzenskaja-Babuska-Brezzi (LBB) condition is a specific instance of the discrete inf-sup condition (M.Benzi, G.H.Golub and J.Liesen 2005) which is necessary and sufficient for the well-posedness of a discrete saddle point problem arising from Galerkin discretization. The LBB condition for a problem of the form (Eq 3.9-3.10) is stated as

$$\exists \beta > 0, \quad \inf_{\mathbf{q} \in \mathbf{M}} \sup_{\mathbf{v} \in \mathbf{X}} \frac{\mathbf{b}(\mathbf{v}, \mathbf{q})}{\|\mathbf{v}\|_{\mathbf{X}} \|\mathbf{q}\|_{\mathbf{M}}} \geq \beta \quad (3.14)$$

For the Stokes problem discussed above, this LBB condition restricts the selection of bases for the Galerkin approximation. It requires the dimension of the function space used for the  $\mathbf{p}$  variable to be at least equal to that of  $\mathbf{u}$ . A popularly used mixed finite element combination for this problem is the Taylor-Hood  $\mathbb{P}_2\mathbb{P}_1$  element where linear polynomials approximate the pressure variable, and a quadratic approximation is used for the velocity variable.

There are a couple of problems with using such mixed finite elements to deal with the LBB condition. Firstly, the use of mixed elements certainly leads to differences in accuracy among the different field variables being considered. As an example, for the case of the Stokes problem, there is a greater requirement for accuracy in the pressure than the velocity given that the most common goal in fluid problems is to calculate the forces on bodies. Hence it can be clearly seen how the use of mixed finite elements in these scenarios will lead to a drastic increase in computational requirement. The other issue with these mixed elements is that a general straightforward construction scheme for a stable combination of the discrete subspace is unknown in the case of saddle-point problems. This is even more of a problem when considering the two-fold saddle point structure of the linearized R13 equations. For these reasons, the alternative of stabilization is considered.

Stabilization techniques can be used to circumvent the LBB condition and allow for the use of stable equal-order discretization. The general form of such a stabilization scheme involves the addition of a stabilization term  $S$  (scaled by a stabilization parameter  $\delta$ ) to a compound bilinear form  $C$  to get the modified form  $\tilde{C}$ .

$$\tilde{C}((\underline{u}, p), (\underline{v}, q)) = C((\underline{u}, p), (\underline{v}, q)) + \delta S((\underline{u}, p), (\underline{v}, q)) \quad (3.15)$$

On a discrete level, stabilization adds a contribution to the zero block in Eq 3.8 and allows to circumvent the discrete LBB condition. Residual-based stabilization schemes are the most popular when dealing with flow problems. However, the CIP stabilization method is claimed to be better suited for the case of the linearized R13 equations in (A.Westerkamp 2017) and has therefore been implemented in the fenicsR13 code.

### 3.2 CIP stabilization in FenicsR13

This part describes how the CIP stabilization technique has been implemented within the fenicsR13 code. In the following  $\mathcal{A}$  represents the bilinear form corresponding to the weak formulation presented in Section 2.3.4 and  $\tilde{\mathcal{A}}$  represents the modified bilinear form which arises from adding a stabilization term  $S$  to  $\mathcal{A}$  as

$$\mathcal{A}\left((\theta, p, \underline{u}, \underline{s}, \underline{\sigma}), (\kappa, q, \underline{v}, \underline{r}, \underline{\psi})\right) = \tilde{\mathcal{A}}\left((\theta, p, \underline{u}, \underline{s}, \underline{\sigma}), (\kappa, q, \underline{v}, \underline{r}, \underline{\psi})\right) + S((\theta, p, \underline{u}), (\kappa, q, \underline{v})) \quad (3.16)$$

where

$$\begin{aligned} & S((\theta, p, \underline{u}), (\kappa, q, \underline{v})) \\ &= \sum_{E \in \mathcal{E}} \int_E \delta_\theta [\nabla \theta \cdot \underline{n}] [\nabla \kappa \cdot \underline{n}] \, dl + \sum_{E \in \mathcal{E}} \int_E \delta_p [\nabla p \cdot \underline{n}] [\nabla q \cdot \underline{n}] \, dl + \sum_{E \in \mathcal{E}} \int_E \delta_u [\nabla \underline{u} \cdot \underline{n}] [\nabla \underline{v} \cdot \underline{n}] \, dl \end{aligned} \quad (3.17)$$

Note that here  $\mathcal{E}$  denotes the index set of all interior element edges and  $[\underline{f} \cdot \underline{n}] = \underline{f}^+ \cdot \underline{n}^+ + \underline{f}^- \cdot \underline{n}^-$  denotes the jump of quantity  $\underline{f}$  across the edge, weighted with the oppositely oriented edge normals  $\underline{n}^+$  and  $\underline{n}^-$ .

Here the stabilization parameters are of the form

$$\delta_\theta = \tilde{\delta}_\theta h^{n_\theta} \quad (3.18)$$

$$\delta_p = \tilde{\delta}_p h^{n_p} \quad (3.19)$$

$$\delta_u = \tilde{\delta}_u h^{n_u} \quad (3.20)$$

## Chapter 4

# Analytical Solution Generating Script

As mentioned in Chapter 1, one of the main objectives of this thesis is to gain more insight into the optimal choice of stabilization parameters for the CIP scheme employed in FenicsR13. One approach is to perform a parametric study, by repeatedly solving a problem while varying the value and scaling of each of the stabilization parameters described in Section 3.2 and analyzing the effect it has on the solution. If the problems considered in this study possess analytical solutions, we can easily judge the effect each choice of stabilization parameters has on the accuracy of the solver by comparing the obtained solution with the reference analytical solutions. Exact solutions for the linearized R13 have been derived for the problem in an annular domain in A.Westerkamp and M.Torrilhon 2012; M.Torrilhon 2010. In this chapter, we present a Python code that generates exact solutions of the linearized R13 equations for a problem over an annular domain based on user-specified boundary conditions. This script is then later used to generate exact solutions for the cases considered in the CIP investigation presented in Chapter 5. Furthermore, the code exports the generated solution in three different formats; a Python script, a C++ header file and a C++ file compatible with the DOLFIN python module (popularly used to solve PDEs). This allows the script to be used in tandem with a variety of solvers of the linearized R13 equations for validation purposes and for similar parametric investigations that rely on convergence studies.

### 4.1 Annular Domain

The domain considered here is the annular region between two infinitely long coaxial cylinders of radii  $R_0$  and  $R_1$ . An assumption of symmetry and homogeneity along the length of the cylinders allows for the extraction of a 2D model problem while retaining the 3D nature of the field variables such as the stress tensor  $\underline{\underline{\sigma}}$  (refer to A.Westerkamp and M.Torrilhon 2019; L.Theisen 2019 for a more detailed explanation). This model problem is represented by a 2D domain described by the area between two concentric circles of radii  $R_0$  and  $R_1$  as

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2 \mid R_0 \leq \sqrt{x^2 + y^2} \leq R_1 \right\} \quad (4.1)$$

Sharp corners are avoided in this domain, and the fact that the origin of the coaxial circles has been excluded from the domain allows for the prescription of normal fluxes without any issues. A representation of the domain is shown in Fig. 4.1. The values for each parameter on both the boundaries ( $\Gamma_0$  and  $\Gamma_1$  in Fig. 4.1) can be of the form  $a + b \cos(\theta)$  or  $a + b \sin(\theta)$  where  $(a, b) \in \mathbb{R}$ . This allows for a variety of cases to be specified and a few of these are explained below.

- A rotating cylindrical shell could be modelled by setting the tangential velocity at the boundary to a constant value and simultaneously setting the normal velocity to zero. ( $u_t = v_{rot}, u_n = 0$ )

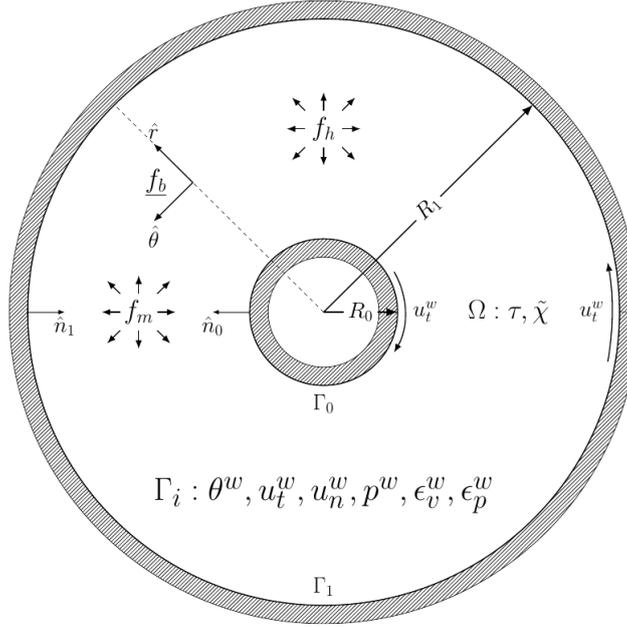


Figure 4.1: A schematic of the annular domain. It is important to note the orientation of the normals and tangential velocity vectors at the walls.

- The classical problem of the flow past a circular cylinder can be modelled by setting the tangential velocity, normal velocity and pressure at the inner boundary to zero ( $u_t = 0, u_n = 0, p = 0$ ) and assigning periodic functions for the the same at the outer boundary ( $u_t = -v_0 \sin(\theta), u_n = v_0 \cos(\theta), p = -p_0 \cos(\theta)$ ).

For the annular domain considered here, analytical solutions of the linearized R13 equations do not exist for any arbitrary source terms  $f_m$ ,  $f_h$  and  $f_b$ . Only specific forms of the source functions allow for non-homogeneous analytical solutions. Therefore, a few general forms of source functions which allow for analytical solutions have been determined empirically. The corresponding expressions for  $f_m$ ,  $f_h$  and  $f_b$  considered in this work are shown in Eq 4.2-4.4

$$f_m = \frac{1}{\text{Kn}} \left( M_o + \frac{M_1 r}{\text{Kn}} \cos(\theta) + \frac{M_2 r^2}{\text{Kn}^2} \right) \quad (4.2)$$

$$f_h = \frac{1}{\text{Kn}} \left( Q_o + \frac{Q_1 r}{\text{Kn}} \cos(\theta) + \frac{Q_2 r^2}{\text{Kn}^2} \right) \quad (4.3)$$

$$f_b = \frac{1}{\text{Kn}} \begin{pmatrix} \left[ F_0 + \left( F_1 \left( \frac{r}{\text{Kn}} + \frac{9\text{Kn}}{5r} \right) + F_{22} \frac{r^2}{\text{Kn}^2} \right) \cos(\theta) + F_2 \frac{r^2}{\text{Kn}^2} \right] \\ \left[ G_0 \left( 1 - \frac{5r^2}{27\text{Kn}^2} \right) + G_1 \frac{r}{\text{Kn}} + \left( G_2 \frac{r^2}{\text{Kn}^2} + G_{11} \frac{\text{Kn}}{r} \right) \sin(\theta) \right] \\ 0 \end{pmatrix} \quad (4.4)$$

where  $M_0, M_1, M_2, Q_0, Q_1, Q_2, F_0, F_1, F_2, F_{22}, G_0, G_1, G_2, G_{11} \in \mathbb{R}$ .

The values of these constants can be adjusted to create a range source functions which can in turn be used to specify a variety of problems. Simple functions for the source terms can be enabled by setting most of these constants to zero. However, it is important that the structure of these functions is unaltered. Examples of some acceptable function for the source terms are listed in Table 4.1

$f_m$	$f_h$	$f_b$
$\frac{2r^2}{Kn^3}$	$\frac{3}{Kn}$	$\begin{pmatrix} \left[ \frac{4}{Kn} + \left( \frac{2r^2}{Kn^3} \right) \cos(\theta) \right] \\ \left[ \frac{1}{Kn} \left( 1 - \frac{5r^2}{27Kn^2} \right) + \frac{3r}{Kn^2} \right] \\ 0 \end{pmatrix}$
$\frac{1}{Kn} \left( 1 + \frac{3r \cos(\theta)}{Kn} \right)$	$\frac{r}{2Kn^2} \cos(\theta) + \frac{r^2}{Kn^3}$	$\frac{1}{Kn} \begin{pmatrix} \left[ 1 + 2 \left( \frac{r}{Kn} + \frac{9Kn}{5r} \right) \cos(\theta) + \frac{3r^2}{Kn^2} \right] \\ \left[ \frac{2r}{Kn} + \left( \frac{3r^2}{Kn^2} + \frac{4Kn}{r} \right) \sin(\theta) \right] \\ 0 \end{pmatrix}$

Table 4.1: Examples of Source Function

## 4.2 Python Code

### 4.2.1 SymPy Module

SymPy is a lightweight open-source Python library used for symbolic mathematics (A.Meurer, C.P.Smith and Paprocki 2017). SymPy is written entirely in Python, which allows for simple installation and use. This library offers a variety of capabilities ranging from basic symbolic arithmetic to calculus, linear algebra, discrete mathematics and quantum physics. Here we use this library to solve the linear system of equations arising from boundary conditions to obtain the values of the integration constants.

### 4.2.2 Code Structure

The boundary condition equations corresponding to the annular domain under consideration along with the generic analytical expressions for the 5 field variables ( $\theta, p, u, q, \underline{\sigma}$ ) (see Appendix A) have been hard-coded into the script presented. The script executes the calculation that needs to be done involving the solving the boundary condition equations according to the boundary values provided by the user, to obtain the required values of the integration constants. Once the integration constant values are determined, these values are plugged into the generic expressions for the field variables to get the final expressions for each field variable corresponding to the specific problem considered. Furthermore, this code exports the final expression of the field variables in three different formats which can then be used in tandem with future solvers for the linearized R13 equations. This could serve the purpose of simply validating the solver or for performing convergence studies.

The script requires a YAML file, to be provided as input, that holds the boundary specifications and formulation parameters for the problem under consideration. The script then reads the YAML file and produces the solution corresponding to the specified problem. As mentioned above, the solution is then exported into the current working directory as a python script (.py), a C++ header file (.h) and a C++ script (.cpp) which conforms to the syntax required by DOLFIN (A.Logg, G.N.Wells and J.Hake 2012) (a python library popularly used to solve PDEs).

### 4.2.3 Code Execution

#### System requirements :

1. A system with Python installed
  - (a) Execute the following commands to check

```
python3 --version
```
2. The SymPy and SciPy libraries need to be installed

- (a) Execute the following commands to check

```
pip3 list
```

- i. If pip3 is not installed then install using (on Ubuntu)

```
sudo apt-get update
sudo apt install python3-pip
```

- (b) If either of the libraries is not listed then install using

```
pip3 install sympy
```

```
pip3 install scipy
```

### Code Execution :

Download the script from:

<https://github.com/aditya314karthik/Analytical-Solution-Generating-Script-for-Lin-R13>

1. Open/Create the YAML file, which will contain the boundary condition values (Fig. 4.2).
2. Adjust/Set the values of the formulation, source and boundary parameters to describe the problem.
  - (a) Comments are provided in the example YAML file (input.yml) describing each parameter.
  - (b) Note that the source terms need to conform to the format specified in Section 4.1. Additionally, the format for each source term is provided in the example YAML file (input.yml) for ease of use.
  - (c) Consider the direction of the normal at both boundaries and assign the sign to the velocity terms at the boundaries appropriately.
3. Ensure that the YAML file is in the same directory as the main script (exact\_sol\_gen\_R13.py).
4. Open a command-line interface.
5. Move into the directory containing the script and the YAML file.
6. Execute

```
python3 exact_sol_gen_R13.py <input_file_name>.yml
```

For example :

```
python3 exact_sol_gen_R13.py input.yml
```

7. Wait until a message saying “Exact Solution Generated in current working directory” appears.
8. Find the exported solutions corresponding to the specified problem in the current working directory.

```

1 # Geometric Parameters
2 # =====
3 # - R0: Inner Radius
4 # - R1: Outer Radius
5 # - fac: For Stress Term in Inflow BC
6 R0: 0.5
7 R1: 2.0
8 fac: 1.0
9
10 # Formulation Parameters
11 # =====
12 # - kn: Knudsen number/Relaxation time
13 # - chi_tilde: Accommodation coefficient in Maxwell model
14 # - heat_source: Heat source function
15 # - mass_source: Mass source function
16 # - body_force_R: Radial Component of Body Force
17 # - body_force_Theta: Tangential Component of Body Force
18
19 # Source Term Formats
20 # =====
21 # - heat_source - (1/kn)*(M0 + M1*(R/kn)*cos(phi) + M2*(R**2/kn**2))
22 # - mass_source - (1/kn)*(Q0 + Q1*(R/kn)*cos(phi) + Q2*(R**2/kn**2))
23 # - body_force_R - (1/kn)*(F0 + (F1*((R/kn) + ((9*kn)/(5*R)))) + F2*(R**2/kn**2))*cos(phi) + F2*(R**2/kn**2))
24 # - body_force_Theta - (1/kn)*(G0*(1-((5*R**2)/(27*kn**2)))) + G1*(R/kn) + (G2*(R**2/kn**2) + G1*(kn/R))*sin(phi))
25 kn: 1.0
26 chi_tilde: 1.0
27 heat_source: (1/kn)*(0.1 + 0.2*(R/kn)*cos(phi) + 0.3*(R**2/kn**2))
28 mass_source: (1/kn)*(0.1 + 0.2*(R/kn)*cos(phi) + 0.3*(R**2/kn**2))
29 body_force_R: (1/kn)*(0.1 + (0.2*((R/kn) + ((9*kn)/(5*R)))) + 0.4*(R**2/kn**2))*cos(phi) + 0.3*(R**2/kn**2))
30 body_force_Theta: (1/kn)*(0.1*(1-((5*R**2)/(27*kn**2)))) + 0.2*(R/kn) + (0.3*(R**2/kn**2) + 0.4*(kn/R))*sin(phi))
31
32 # Boundary Conditions
33 # =====
34 # - bcs:
35 # - bc_id: Inner/Outer Circle
36 # - theta_w: Value for temperature at wall
37 # - u_t_w: Value for tangential velocity at wall
38 # - u_n_w: Value for normal velocity at wall
39 # - p_w: Value for pressure at wall
40 # - epsilon_v: Inflow-model parameter <=> Weight of pressure prescription
41 # - epsilon_p: Inflow-model parameter <=> Weight of velocity prescription
42 bcs:
43   inner:
44     theta_w: 1.0
45     u_t_w: 0
46     u_n_w: 0
47     p_w: 0
48     epsilon_v: pow(10,-3)
49     epsilon_p: 1
50   outer:
51     theta_w: 2.0
52     u_t_w: -1.00 * sin(phi)
53     u_n_w: +1.00 * cos(phi)
54     p_w: -0.27 * cos(phi)
55     epsilon_v: pow(10,+3)
56     epsilon_p: 1

```

Figure 4.2: An example of the input YAML file that needs to be supplied to the analytical solution generating script.

## 4.2.4 Using Exported Solution

The output files generated are :

1. Exact\_Sol\_Python.py
2. Exact\_Sol\_C++.h
3. Exact\_Sol\_Dolfin.cpp

Examples depicting how these files can be imported and used in other programs are elaborated below.

### 4.2.4.1 Python Format

Here a simple python script that imports the Exact\_Sol\_Python.py file and evaluates each field variable at the point (1,1) is shown (Fig. 4.3). The output produced by this python script is shown below (Fig. 4.4)

```

1 import Exact_Sol_Python as sol
2
3 def main():
4     print("theta(1,1)   = {}".format(sol.theta(1,1)))
5     print("p(1,1)      = {}".format(sol.p(1,1)))
6     print("u_x(1,1)    = {}".format(sol.ux(1,1)))
7     print("u_y(1,1)    = {}".format(sol.uy(1,1)))
8     print("s_x(1,1)    = {}".format(sol.sx(1,1)))
9     print("s_y(1,1)    = {}".format(sol.sy(1,1)))
10    print("sigma_xx(1,1) = {}".format(sol.sig_xx(1,1)))
11    print("sigma_xy(1,1) = {}".format(sol.sig_xy(1,1)))
12    print("sigma_yy(1,1) = {}".format(sol.sig_yy(1,1)))
13
14
15 if __name__ == '__main__':
16     main()

```

Figure 4.3: An example of a Python code importing the Exact\_Sol\_Python.py file

```

1
2 theta(1,1)   = 2.007282704443216
3 p(1,1)      = -1.3548211307086665
4 u_x(1,1)    = 1.823765325097661
5 u_y(1,1)    = 0.6946117788614897
6 s_x(1,1)    = -0.07929989062718454
7 s_y(1,1)    = -0.19290970346796807
8 sigma_xx(1,1) = -0.061458934851708946
9 sigma_xy(1,1) = 0.011937308354488918
10 sigma_yy(1,1) = -0.25882432655189946
11
12 Process finished with exit code 0
13

```

Figure 4.4: Output corresponding to the above Python script

#### 4.2.4.2 C++ Format

In a similar fashion to the previous section, the simple C++ code shown here (Fig. 4.5) imports the header file “Exact\_Sol+C++.h” and evaluates the generated solution corresponding to each field variable at the point (1,1). The output generated is also provided (Fig. 4.6). Note that since the evaluation of the modified Bessel functions is required within the header file, it is essential to ensure that the C++ Boost library is installed and available for use.

```

1
2 #include <iostream>
3 #include "Exact_Sol_C++.h"
4
5 int main() {
6     std::cout << "theta(1,1)   = " << theta(1,1) << "\n";
7     std::cout << "p(1,1)      = " << p(1,1) << "\n";
8     std::cout << "u_x(1,1)    = " << u_x(1,1) << "\n";
9     std::cout << "u_y(1,1)    = " << u_y(1,1) << "\n";
10    std::cout << "s_x(1,1)    = " << s_x(1,1) << "\n";
11    std::cout << "s_y(1,1)    = " << s_y(1,1) << "\n";
12    std::cout << "sigma_xx(1,1) = " << sig_xx(1,1) << "\n";
13    std::cout << "sigma_xy(1,1) = " << sig_xy(1,1) << "\n";
14    std::cout << "sigma_yy(1,1) = " << sig_yy(1,1) << "\n";
15    return 0;
16 }

```

Figure 4.5: An example of a C++ code importing the Exact\_Sol\_C++.h file

```

Start
theta(1,1)      = 2.00728
p(1,1)          = -1.35402
u_x(1,1)        = 1.82377
u_y(1,1)        = 0.694612
s_x(1,1)        = -0.0792999
s_y(1,1)        = -0.19291
sigma_xx(1,1)   = -0.0614589
sigma_xy(1,1)   = 0.0119373
sigma_yy(1,1)   = -0.258824
0
Finish

```

Figure 4.6: Output corresponding to the above C++ script

#### 4.2.4.3 DOLFIN Format

Along the same lines as the two previous sections, here this sample python program reads the `Exact_Sol_Dolfin.cpp` file using the DOLFIN python module then interpolates it over a mesh for the annular domain and finally prints the values of each variable at the point (1, 1) (Fig. 4.7). The resulting output (Fig. 4.8) is quite similar to the outputs from the previous two sections.

```

1 from dolfin import *
2 from mshr import *
3
4 def main():
5
6     R0 = 0.5
7     R1 = 2.0
8
9     mesh = generate_mesh(Circle(Point(0, 0), R1) - Circle(Point(0, 0), R0), 50)
10
11     s_field = FunctionSpace(mesh, FiniteElement("Lagrange", mesh.ufl_cell(), 1))
12     v_field = FunctionSpace(mesh, VectorElement("Lagrange", mesh.ufl_cell(), 1))
13     t_field = FunctionSpace(mesh, TensorElement("Lagrange", mesh.ufl_cell(), 1, symmetry=True))
14
15     with open("Exact_Sol_Dolfin.cpp", "r") as file:
16         exact_solution_cpp_code = file.read()
17
18     esol = compile_cpp_code(exact_solution_cpp_code)
19
20     theta = CompiledExpression(esol.Temperature(), degree=2)
21     s = CompiledExpression(esol.Heatflux(), degree=2)
22     p = CompiledExpression(esol.Pressure(), degree=2)
23     u = CompiledExpression(esol.Velocity(), degree=2)
24     sigma = CompiledExpression(esol.Stress(), degree=2)
25
26     theta_exact = interpolate(theta, s_field)
27     p_exact = interpolate(p, s_field)
28     u_exact = interpolate(u, v_field)
29     s_exact = interpolate(s, v_field)
30     sigma_exact = interpolate(sigma, t_field)
31
32     print("theta(1,1) = {}".format(theta_exact(1, 1)))
33     print("p(1,1) = {}".format(p_exact(1, 1)))
34     print("u(1,1) = {}".format(u_exact(1, 1)))
35     print("s(1,1) = {}".format(s_exact(1, 1)))
36     print("sigma(1,1) = {}".format(sigma_exact(1, 1)))
37
38 if __name__ == '__main__':
39     main()

```

Figure 4.7: An example of a DOLFIN enabled code importing the `Exact_Sol_Dolfin.cpp` file

```

theta(1,1) = 2.0072993482085244
p(1,1) = -1.353640707365288
u(1,1) = [1.8233943 0.69473135]
s(1,1) = [-0.07926503 -0.19286674]
sigma(1,1) = [-0.06162601 0.01197821 -0.25863663 0. ]

```

Figure 4.8: Output corresponding to the above DOLFIN code

### 4.2.5 Limitations

Some of the limitations associated with the range of cases that this code can accommodate are mentioned below.

1. As mentioned earlier, although there might be other source functions which allow for non-homogeneous analytical solutions, this code is only able to deal with source functions of the forms specified in Section 4.1.
2. The boundary values have to be of the form  $a + b \cos(\theta)$  or  $a + b \sin(\theta)$  where  $(a, b) \in \mathbb{R}$  as mentioned in Section 4.1.
3. The code is unable to generate solutions when the Knudsen number (Kn) is less than about 0.005. The system becomes ill-conditioned.

# Chapter 5

## CIP Investigation

### 5.1 Cases and Error Measures

To perform a thorough parametric analysis, four different cases with increasingly complex solutions are considered in this report. Analytical solutions for the four cases were generated using the python script presented in Chapter 4 and used to produce convergence curves corresponding to each iteration of this parametric study. This section describes the domain (which is common to the four cases) along with the boundary conditions and formulation parameters associated with each specific case. The Knudsen number  $\text{Kn}$  and modified accommodation coefficient  $\tilde{\chi}$  are both set to unity, and the radii are set to  $R_0 = 0.5$  and  $R_1 = 2.0$  in all the four cases. Note that the input file used for each case is provided in Appendix B for the reader's reference.

#### 5.1.1 Annular Domain

The annular domain consists of the area between two coaxial circles of radii  $R_0$  and  $R_1$  as

$$\Omega = \left\{ \underline{x} = (x, y)^T : R_0 \leq \sqrt{x^2 + y^2} \leq R_1 \right\}, \quad (5.1)$$

Sharp corners are avoided in this domain, and the fact that the origin of the coaxial circles has been excluded from the domain allows for the prescription of normal fluxes without any issues. Note that this 2D domain can be interpreted as the cross-sectional area corresponding to an infinite annular cylinder extending in the third dimension. A representation of the domain is shown below (Fig 5.1)

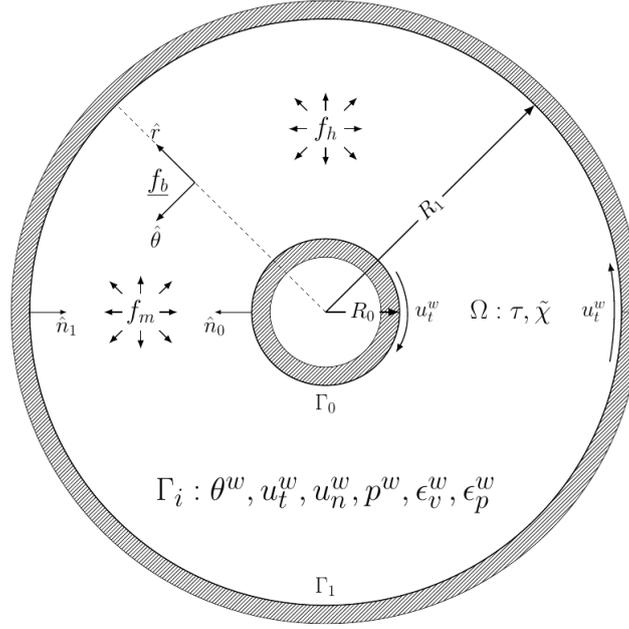


Figure 5.1: A schematic of the annular domain. It is important to note the orientation of the normals and tangential velocity vectors at the walls.

### 5.1.2 Case I : Homogeneous Flow Past Circular Cylinder

In this case, the inner ring is modelled as a solid wall by setting the normal and tangential velocity to zero. The outer wall, on the other hand, is modelled as a cut-off from a larger external homogeneous velocity field directed from left to right. This is achieved by prescribing a periodic velocity and pressure value at this outer ring. No source terms are considered in this case which makes this the most trivial of the four cases considered. A depiction of the streamlines and velocity contours for this case is shown in Fig 5.2

### 5.1.3 Case II : Mass and Heat Source Induced Flow in an Extruded Ring

In this case, both the rings are modelled as solid walls (zero normal velocity); however, the inner wall is prescribed a tangential velocity of -10, whereas the outer wall is kept stationary. Contrary to the previous case, here non-zero heat and mass sources are considered. Both the mass source  $f_m$  and heat source  $f_h$  are described by the function  $\left(1 - \frac{5r^2}{18Kn^2}\right) \cos(\theta)$  and these sources induce the flow in this case. The streamlines and velocity contours for this case is shown in Fig 5.2

### 5.1.4 Case III : Body Force Induced Flow in an Extruded Ring

This case is somewhat similar to the previous case (Case II) except for the fact that this time instead of the mass and heat sources a body force  $f_b$  acting throughout the domain induces the flow (Fig 5.3). The walls are modelled exactly the same as they were in the previous case. The body force vector considered is shown below (in the cylindrical coordinate system)

$$\underline{f}_b = \frac{1}{Kn} \begin{pmatrix} \left[ 0.1 + \left( 0.2 \left( \frac{r}{Kn} + \frac{9Kn}{5r} \right) + 0.4 \frac{r^2}{Kn^2} \right) \cos(\theta) + 0.3 \frac{r^2}{Kn^2} \right] \\ \left[ 0.1 \left( 1 - \frac{5r^2}{27Kn^2} \right) + 0.2 \frac{r}{Kn} + \left( 0.3 \frac{r^2}{Kn^2} + 0.4 \frac{Kn}{r} \right) \sin(\theta) \right] \\ 0 \end{pmatrix} \quad (5.2)$$

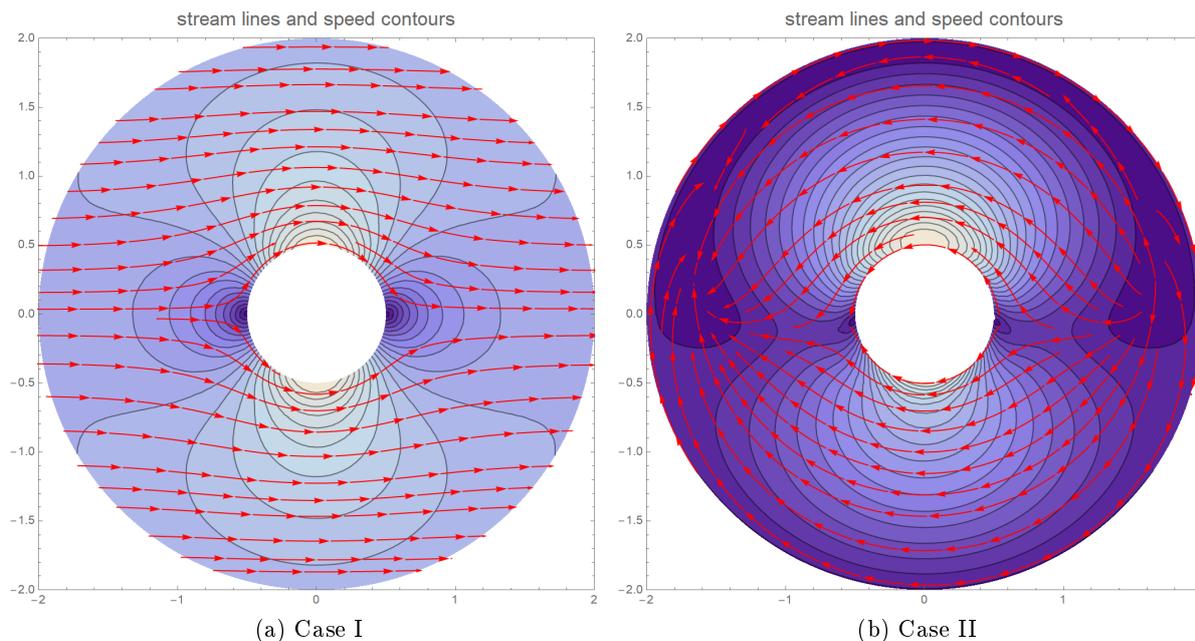


Figure 5.2: Streamlines and velocity contours corresponding to Case 1 and Case 2

Although setting-up the body force vector this way excludes the possibility of any reasonable physical interpretation, this makes the solution more complex and therefore would prove useful when it comes to qualitatively evaluation of the different cases at the end of the study.

### 5.1.5 Case IV : Complex Flow

This final case has been constructed in order to maximize the mathematical complexity of the solution. The mathematical complexity of the is proportional to how many components of the generic solution are active in the solution of the problem considered. It should be noted that analytical solutions are only available for source functions conforming to certain formats. This automatically restricts the range of cases for which convergence studies can be conducted. With that in mind, an attempt to maximize the mathematical complexity was made by selecting an elaborate function for each of the source terms. Additionally, the walls were modelled exactly as they have been in Case II and Case III. The body force vector considered is the same as in Eq 5.2, whereas the mass and heat sources were defined as shown below and the corresponding streamlines and velocity contours can be seen in Fig 5.3

$$f_m = \frac{1}{\text{Kn}} \left( 0.1 + \frac{0.2r}{\text{Kn}} \cos(\theta) + \frac{0.3r^2}{\text{Kn}^2} \right) \quad (5.3)$$

$$f_h = \frac{1}{\text{Kn}} \left( 0.1 + \frac{0.2r}{\text{Kn}} \cos(\theta) + \frac{0.3r^2}{\text{Kn}^2} \right) \quad (5.4)$$

### 5.1.6 Error Measures

The relevant error measures considered in this report are the standard relative  $L^2$ -function error  $e_{L^2}$  and the relative error  $e_{l^\infty}$  which are defined as follows.

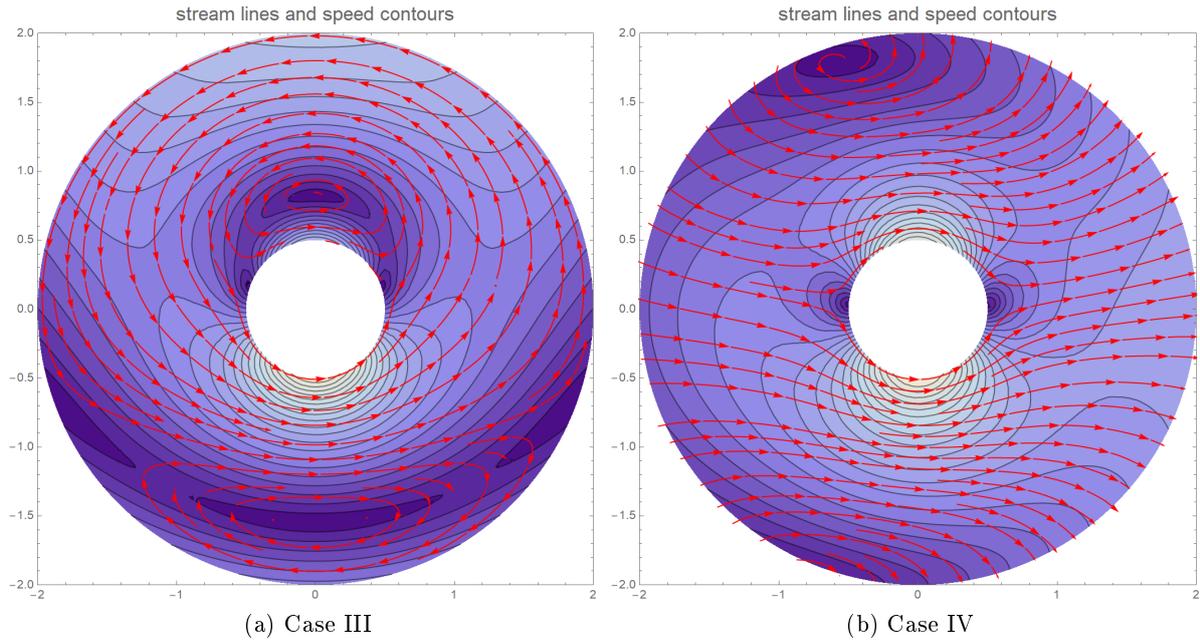


Figure 5.3: Streamlines and velocity contours corresponding to Case 3 and Case 4

$$e_{L^2} = \frac{\|f_{ex} - f\|_{L^2(\Omega)}}{\max\{f_{ex} | n\}_{n \in \eta}} \quad (5.5)$$

$$e_{l^\infty} = \frac{\|\{f_{ex} | n - f | n\}_{n \in \eta}\|_{l^\infty(\eta)}}{\max\{f_{ex} | n\}_{n \in \eta}} \quad (5.6)$$

where  $f_{ex}$  is the exact solution projected onto a function space,  $f$  is the discrete solution, and  $\eta$  represents the set containing all the mesh nodes. Since the resulting solution is in terms of node values, it makes sense to look into the  $e_{l^\infty}$  where point-wise errors are considered based on the nodal values. Throughout the study convergence curves corresponding to these two errors are plotted and used for qualitative comparison purposes.

## 5.2 Exponent Value Study

In A.Westerkamp 2017 a method to select parameter values such that the stabilization behavior is unchanged with refinement is discussed. According to that method, the appropriate exponent values for the case of the linearized R13 equations is  $(n_\theta, n_p, n_u) = (3, 1, 3)$ . In order to verify this selection, a parametric study using fenicsR13 was carried out varying the values for each of  $n_\theta$ ,  $n_p$  and  $n_u$  (from Eq 3.18-3.20) within the set  $(0, 1, 2, 3, 4)$ . The values for the constant part were chosen as  $(\tilde{\delta}_\theta, \tilde{\delta}_p, \tilde{\delta}_u) = (1, 0.01, 1)$  which is in accordance with (L.Theisen 2019). A thorough examination of the convergence curves and error data produced in each case has led to the conclusion that the selection of exponents suggested in A.Westerkamp 2017 is indeed the most appropriate choice. The convergence plots shown in Fig 5.4 are obtained after combining the errors across all 4 cases using the  $L^2$ -norm and  $L^\infty$ -norm of the relative  $e_{L^2}$  and  $e_{l^\infty}$  errors respectively. The definitions of the  $L^2$  and  $L^\infty$  norms used here are given below in Eq 5.7-5.8. In the following graphs, when each of the exponents were varied, the other two exponents were set according to  $(n_\theta, n_p, n_u) = (3, 1, 3)$ .

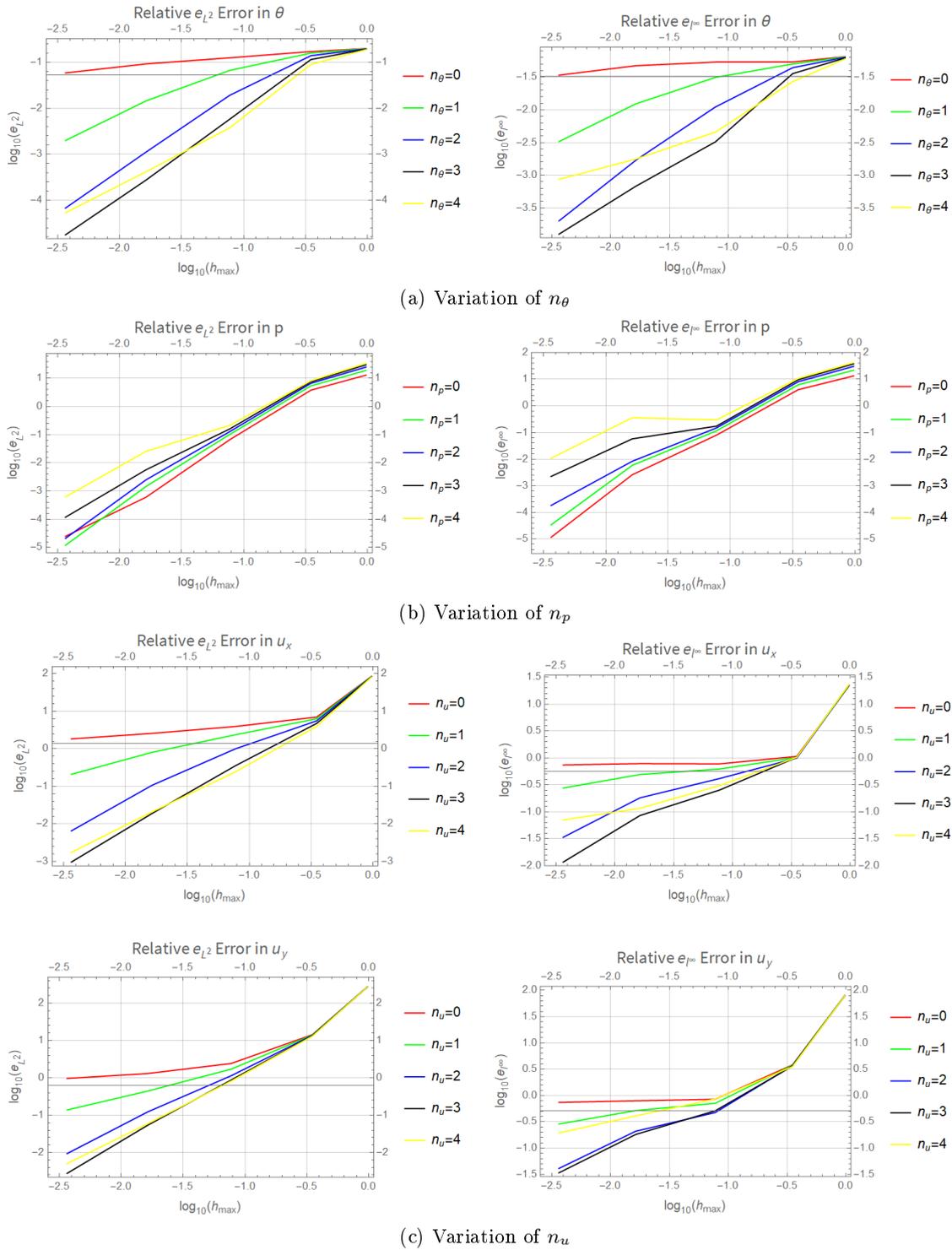


Figure 5.4: Convergence plots obtained after averaging the errors across the 4 cases. Each plot corresponds to the independent variation of one of the three exponents. These graphs illustrate that the selection of  $(n_\theta, n_p, n_u) = (3, 1, 3)$  leads to the best convergence.

$$e_{L^2}^{avg} = \sqrt{\sum_{i=1}^4 (e_{L^2}^i)^2} \quad (5.7)$$

$$e_{l^\infty}^{avg} = \max \{ e_{l^\infty}^i \}, \quad i = \{1, 2, 3, 4\} \quad (5.8)$$

where  $e_{L^2}^i$  and  $e_{L^\infty}^i$  represent the relative  $L^2$  and  $L^\infty$  error in a particular field variable in the  $i^{\text{th}}$  case considered.

### 5.3 Constant Value Study

Once the choice for the exponents  $(n_\theta, n_p, n_u)$  was verified, a parametric study with respect to the constant terms  $(\tilde{\delta}_\theta, \tilde{\delta}_p, \tilde{\delta}_u)$  was carried out setting  $(n_\theta, n_p, n_u) = (3, 1, 3)$ . A wide range of values (0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000) was considered for each constant parameter and a rigorous investigation was carried which resulted in 2197 sets of convergence curves per case. All the convergence curves were then plotted together on Mathematica (n.d.) and a qualitative analysis of the curves was performed. Some of the general observations made are discussed below. Note that in a similar fashion to what was seen in Section 5.2, in each of the graphs below, when one of the three constant terms were varied, the other two were set in accordance with  $(\tilde{\delta}_\theta, \tilde{\delta}_p, \tilde{\delta}_u) = (1, 0.01, 1)$ .

1. As expected, it was noticed that independent variation of each of  $\tilde{\delta}_\theta$ ,  $\tilde{\delta}_p$  and  $\tilde{\delta}_u$  primarily affected the convergence curves corresponding to  $\theta$ ,  $p$  and  $u$  respectively. This is illustrated in Fig 5.5 where the relative  $e_{L^2}$  errors are plotted and in Fig 5.6 where the relative  $e_{L^\infty}$  errors are plotted.
2. In general, the convergence of the non-equilibrium variables  $\underline{s}$  and  $\underline{\sigma}$  are significantly less affected by the choice of  $(\tilde{\delta}_\theta, \tilde{\delta}_p, \tilde{\delta}_u)$  than the equilibrium variables.
3. Curiously, the independent variation of  $\tilde{\delta}_p$  and  $\tilde{\delta}_u$  did have a reduced effect on the other field variables (apart from  $p$  and  $u$  respectively), whereas all the curves but the one corresponding to  $\theta$  remain almost unchanged with the independent variation of  $\tilde{\delta}_\theta$ . Consequentially, the choice of  $\tilde{\delta}_\theta$  has less of an impact on the average error curves than  $\tilde{\delta}_p$  and  $\tilde{\delta}_u$ .
4. In all cases, extreme values (0.001, 0.005, 500, 1000) most often resulted in the drastic worsening on the convergence. This can be seen in Fig 5.7.
5. It was also observed that the errors corresponding to higher values of  $h_{max}$  (coarser meshes) on an average were less affected by the choice of  $(\tilde{\delta}_\theta, \tilde{\delta}_p, \tilde{\delta}_u)$  than those corresponding to lower values of  $h_{max}$  (finer meshes).
6. In all the cases, a complete lack of convergence in all field variables was seen, when the values of  $\tilde{\delta}_p \simeq 1000$ . In other words the errors become independent of the mesh size  $h_{max}$ . Such a phenomenon was not consistently observed with respect to  $\tilde{\delta}_\theta$  and  $\tilde{\delta}_u$ . In Fig 5.8 convergence plots at such high  $\tilde{\delta}_p$  values are shown and it can be seen how this phenomenon is seen across all the four cases and even in the non-equilibrium quantities  $\underline{s}$  and  $\underline{\sigma}$ .

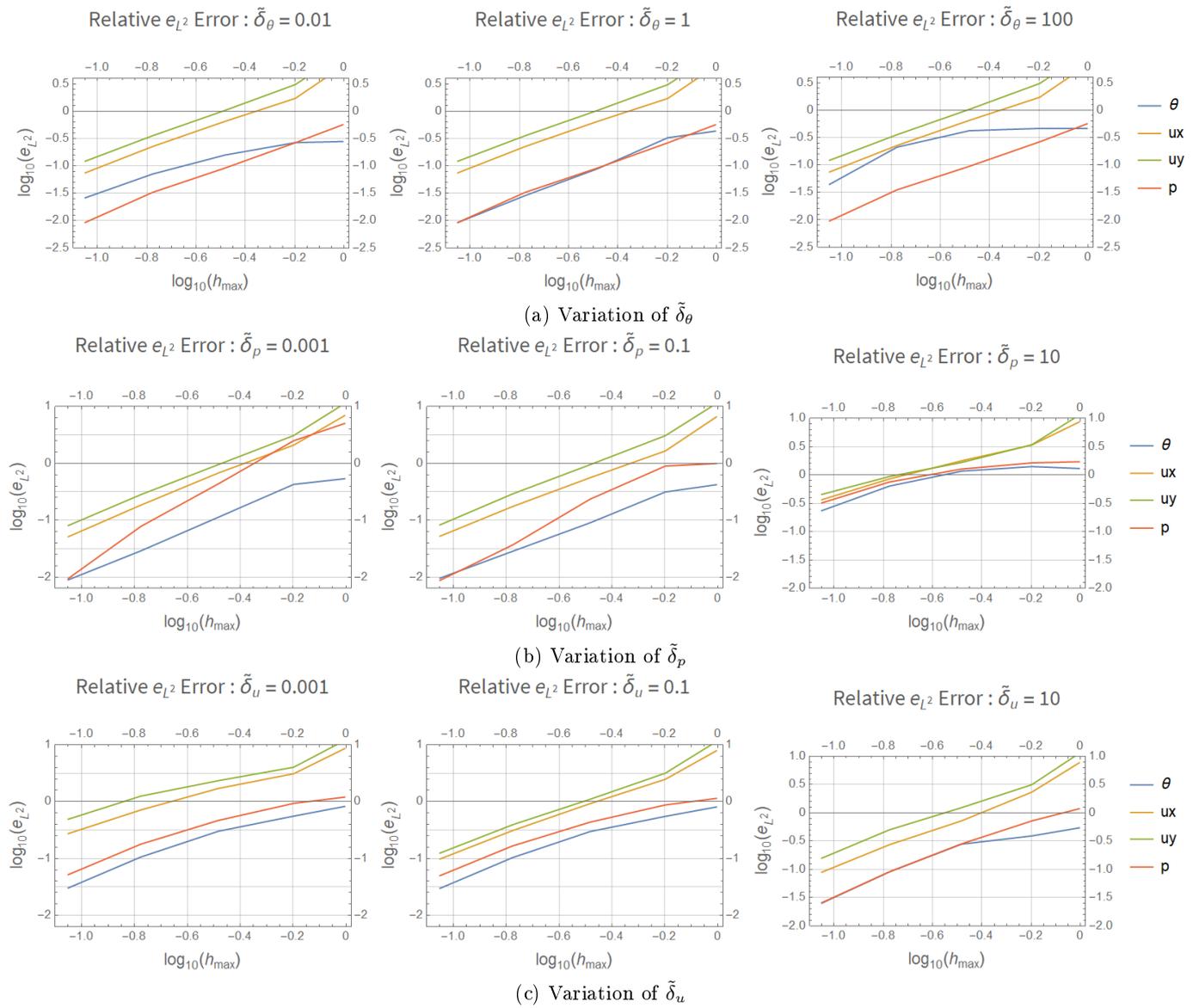


Figure 5.5: Independent Variation -  $e_{L^2}$  : These graphs correspond to the relative  $e_{L^2}$  errors averaged across the 4 cases and they serve to demonstrate how the stabilization parameters  $\delta_\theta$ ,  $\delta_p$  and  $\delta_u$  mostly affect the variables  $\theta$ ,  $p$  and  $\underline{u}$  respectively.

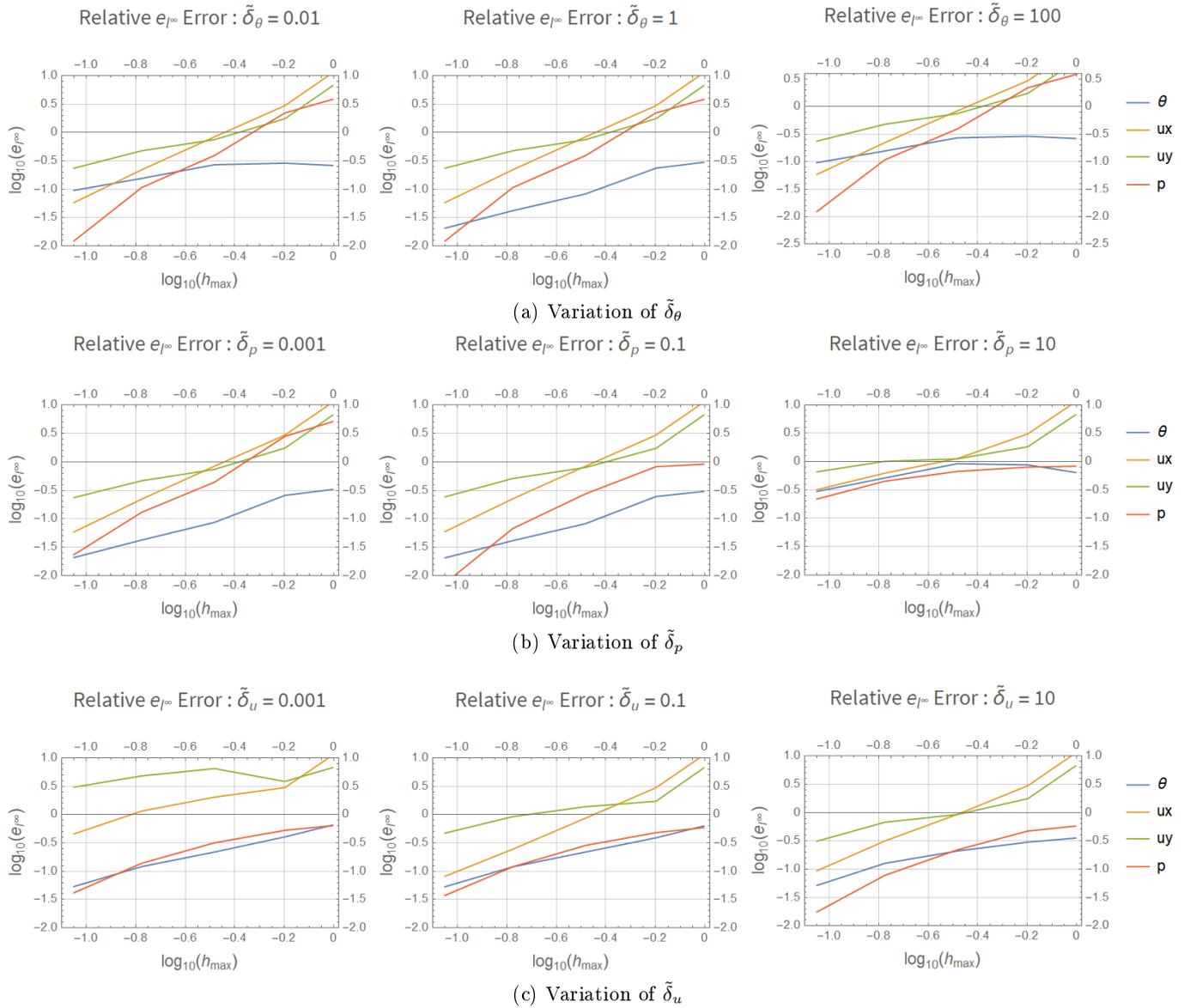


Figure 5.6: Independent Variation -  $e_{l^\infty}$  : These graphs correspond to the relative  $e_{l^\infty}$  errors averaged across the 4 cases and they serve to demonstrate how the stabilization parameters  $\delta_\theta$ ,  $\delta_p$  and  $\delta_u$  mostly affect affect the variables  $\theta$ ,  $p$  and  $\underline{u}$  respectively.

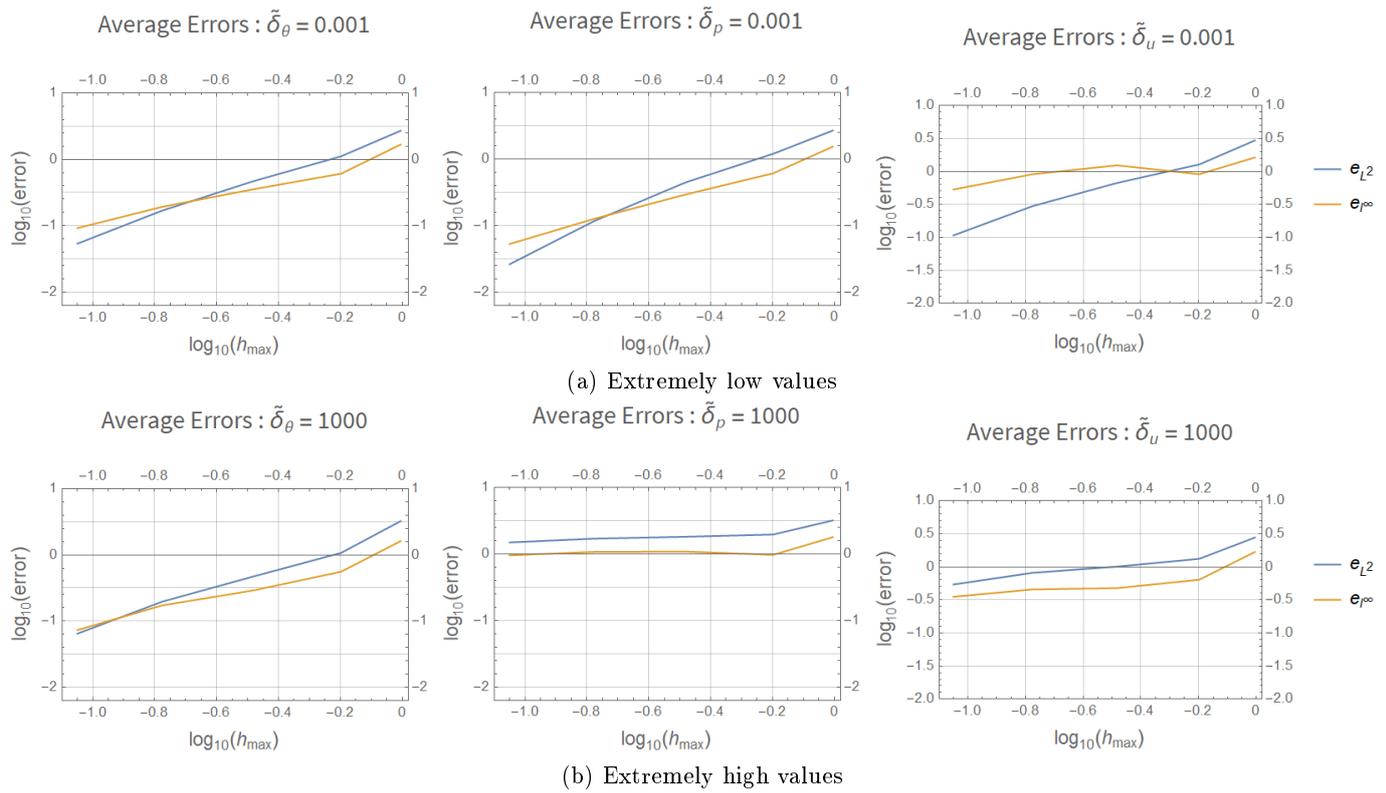


Figure 5.7: Convergence at extreme values for stabilization parameters. It can be seen how selecting extreme values for the stabilization parameters often results in worsened convergence. Once again here the errors from the 4 cases are combined using the  $L^2$  and  $L^\infty$  norms.

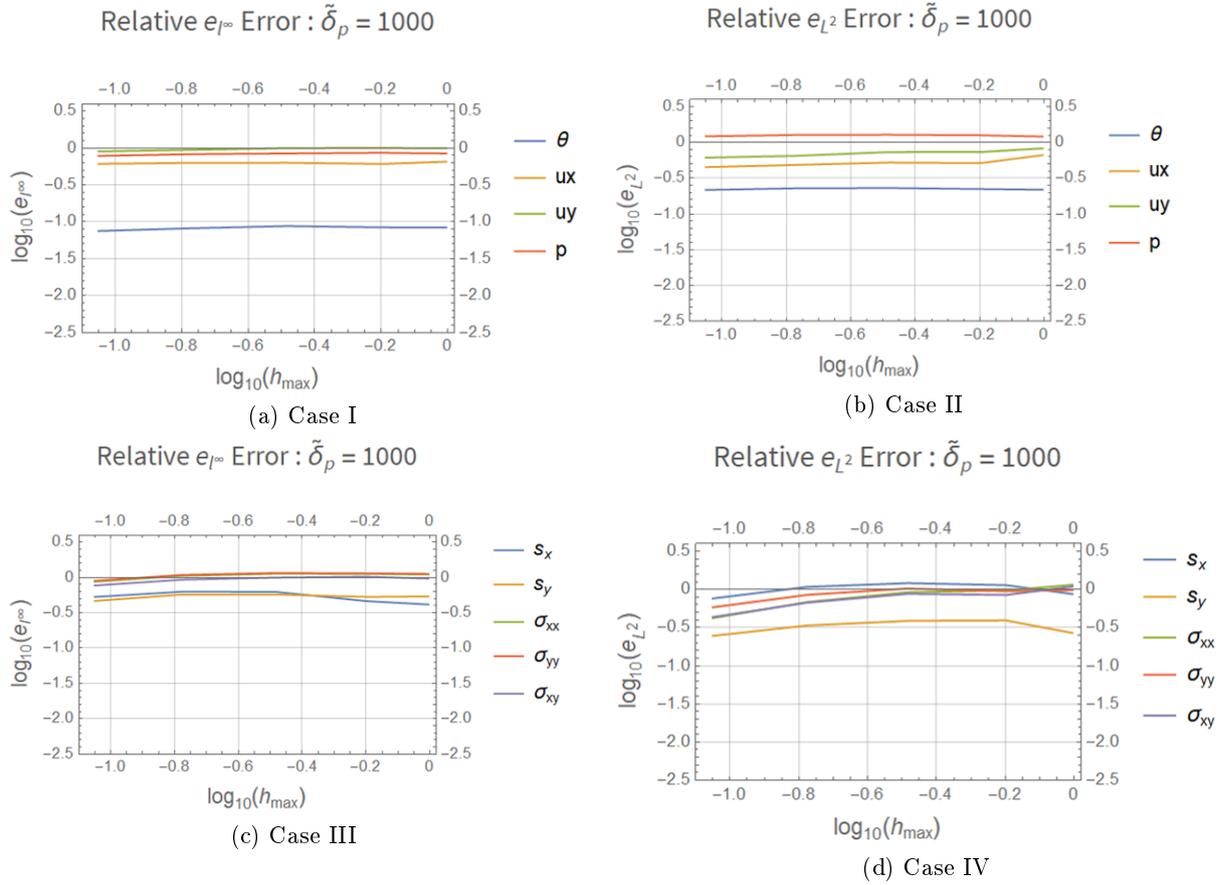


Figure 5.8: Flattening of curves at high values of  $\tilde{\delta}_p$ . A total lack of convergence was observed when  $\tilde{\delta}_p \simeq 1000$  in all cases for all the variables. Along with the equilibrium variables the non-equilibrium variables  $\underline{q}$  and  $\underline{\sigma}$  are also shown here.

## 5.4 Structured Mesh Study

Finally, the entire process from Section 5.3 was repeated using structured meshes instead of the unstructured meshes that were used before. For each case the average errors were calculated by considering the  $L^2$  and  $L^\infty$  norms of the relative  $e_{L^2}$  and  $e_{l^\infty}$  errors across all the field variables within each case (similar to Eq 5.7-5.8). Interestingly, it was noted that change to structured meshes almost did not improve the average error convergence curves at all (Fig 5.9). This is quite surprising because in M.Torrilhon and N.Sarna 2017 it was observed that the use of structured meshes resulted in substantially improved curves. However, in M.Torrilhon and N.Sarna 2017 quad elements were considered, which is in contrast to the triangular elements considered in this report. Unfortunately, FEniCS does not yet allow for the implementation of quad elements, and therefore it was not possible to check whether this difference was the reason behind the contradictory observations. An example of a structured and unstructured mesh used in this study is shown in Fig 5.10.

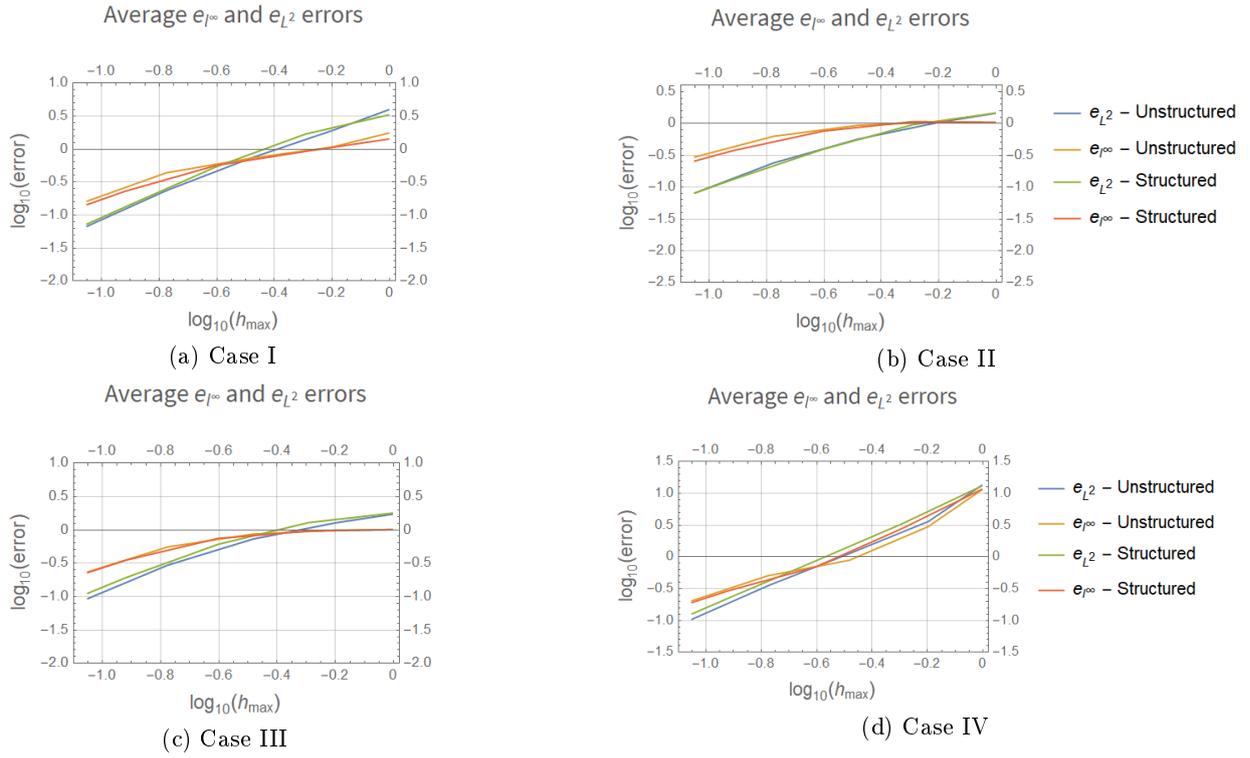


Figure 5.9: Structured mesh convergence comparison. It can be seen here how the use of structured meshes in place of unstructured meshes does not improve the convergence.

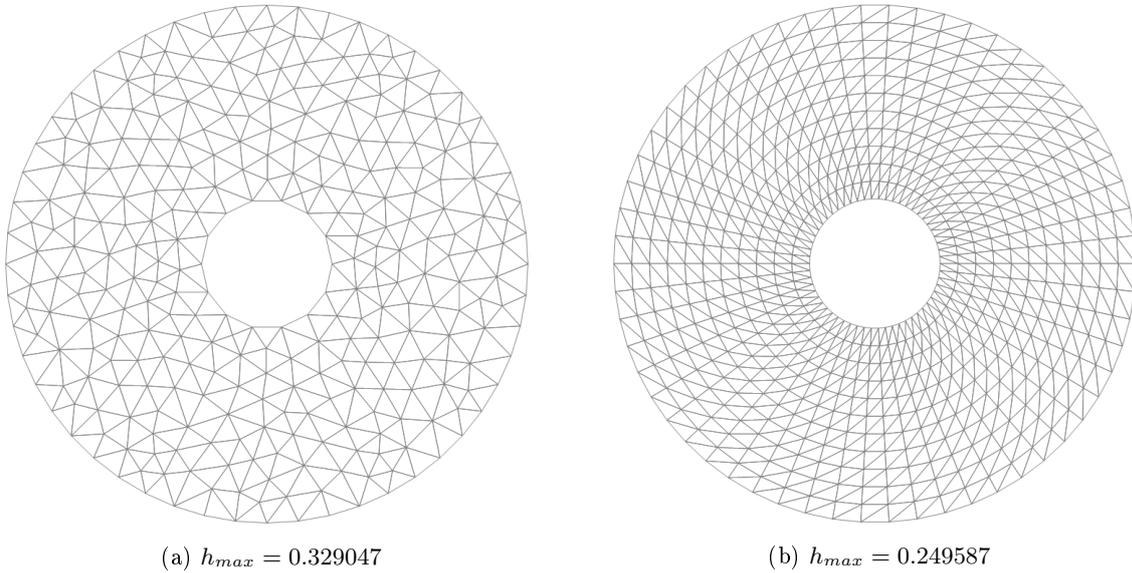


Figure 5.10: Mesh comparison.

## 5.5 Conclusion

It has been verified empirically that the choice of exponents  $n_{\theta} = 3$ ,  $n_p = 1$  and  $n_u = 3$  is indeed the most suitable for the case of the linearized R13 equations. Then a few general observations regarding the effect of the constant terms  $\tilde{\delta}_{\theta}$ ,  $\tilde{\delta}_p$  and  $\tilde{\delta}_u$  on the convergence are discussed and substantiated using various plots. This study should, therefore, provide the reader with a certain level of intuition about choosing the stabilization parameters for the CIP scheme. To help with

this, a range of values for  $\tilde{\delta}_\theta$ ,  $\tilde{\delta}_p$  and  $\tilde{\delta}_u$  which on an average lead to good convergence with respect to the average error is shown in Table 5.1. It is crucial to note that this suggestion is entirely based on these four particular cases and should be used with caution. A final observation is made about the lack of improved convergence when using structured meshes, at least when triangular elements are considered.

<b>Parameter</b>	<b>Range</b>
$\tilde{\delta}_\theta$	0.1 to 5
$\tilde{\delta}_p$	0.01 to 0.1
$\tilde{\delta}_u$	0.05 to 1

Table 5.1: Suggested range for stabilization parameters

## Chapter 6

# Non-Linear Conservation Equations

Although the FenicsR13 solver is capable of solving a variety of rarefied gas flow problems at a modest computational cost, its solution is currently restricted to the linearized version of the R13 equations. This consequentially puts a limitation on the solver's capability to accurately capture the physics of the flow. This latter part of the thesis is focussed on relaxing this restriction by including the non-linear terms in the conservation equations being solved.

The mass, momentum and energy conservation equations in their non-linear form are given as

$$\rho \nabla \cdot \underline{u} + \underline{u} \cdot \nabla \rho = 0 \quad (6.1)$$

$$\rho \underline{u} \cdot \nabla \underline{u} + \nabla p + \nabla \cdot \underline{\underline{\sigma}} = 0 \quad (6.2)$$

$$\frac{3}{2} \rho \underline{u} \cdot \nabla \theta + \nabla \cdot \underline{q} + p \nabla \cdot \underline{u} + \underline{\underline{\sigma}} : (\nabla \underline{u})_{\text{STF}} = 0 \quad (6.3)$$

### 6.1 Scaled R13 Equations

Non-linear systems of equations are traditionally solved using a variant of the classic Newton's iterative method. An inherent problem that comes with this method is the unreliability of the convergence across different problems. If the non-linear terms in the equations are of smaller order than the the linear terms, we expect the convergence to improve significantly. Therefore, we start by introducing a new scaling for the R13 equations that introduces appropriate coefficients in front of the non-linear terms to improve convergence.

We consider a background state at rest, described by mass density  $\rho_0$  and temperature  $T_0$ , upon which thermal and mechanical perturbations are imposed. The gas is considered to be ideal and therefore satisfies the relation

$$p_0 = \rho_0 R_{\text{sp}} T_0 = \rho_0 \theta_0 \quad (6.4)$$

where  $p_0$  is the pressure,  $R_{\text{sp}}$  is the specific gas constant and  $\theta_0$  is the temperature in energy units ( $m^2/s^2$ ). The velocity scale of the background state is given as  $\sqrt{\theta_0}$ .

The magnitude of the mechanical perturbation on the system corresponds to  $u_0$ , the inflow or boundary velocity, whereas the magnitude of the thermal perturbation is represented by  $\Delta T_0$ , the temperature gradient imposed on the system. We then introduce  $c_0$ , the thermal velocity scale as

$$c_0 = R_{\text{sp}} \Delta T_0 \quad (6.5)$$

which makes it convenient to compare the relative magnitudes of the perturbations imposed on the system.

Based on this setting, we define the following new non-dimensional quantities

$$\begin{aligned}\hat{\underline{u}} &= \frac{\underline{u}}{u_0}, & \hat{\theta} &= 1 + \frac{\theta - \theta_0}{c_0^2}, & \hat{p} &= \frac{\sqrt{\theta_0}(p - p_0)}{u_0 p_0}, & \hat{\underline{\sigma}} &= \frac{\underline{\sigma}}{\rho_0 u_0 \sqrt{\theta_0}} \\ \hat{\underline{q}} &= \frac{\underline{q}}{\rho_0 c_0^2 \sqrt{\theta_0}}, & \hat{x} &= \frac{x}{x_0}, & \hat{\underline{m}} &= \frac{\underline{m}}{\rho_0 u_0 \theta_0}, & \hat{R} &= \frac{R}{\rho_0 \theta_0 c_0^2}\end{aligned}$$

where  $x_0$  is the length scale of the process.

The characteristic numbers corresponding to this scaling are the Mach, Knudsen and Reynolds numbers defined as

$$\text{Ma} = \frac{u_0}{\sqrt{\theta_0}} \quad \text{Kn} = \frac{\tau \sqrt{\theta_0}}{x_0} = \frac{\lambda_0}{x_0} \quad \text{Re} = \frac{u_0 x_0}{\tau \theta_0} = \frac{\rho_0 u_0 x_0}{\mu_0} \quad (6.6)$$

where  $\tau$  is the relaxation time,  $\lambda_0$  is the mean free path of the gas particles and  $\mu_0$  is the dynamic viscosity of the gas defined as  $\mu_0 = \rho_0 \theta_0 \tau$ . These definitions can be rearranged to get the relation

$$\text{Ma} = \text{Re Kn} \quad (6.7)$$

which means that the selection of values for two of these quantities fixes the values for the third.

The R13 equations, with non-linear conservation equations and linear evolution equations for the heat flux and stress tensor, are given in their non-dimensional form as

$$\nabla \cdot \hat{\underline{u}} + \frac{\text{Ma}}{(1 + \text{Ma } p)} \hat{\underline{u}} \cdot \nabla p - \frac{\text{Ma}^2 \Gamma^2}{(\text{Ma}^2 \Gamma^2 (\theta - 1) + 1)} \hat{\underline{u}} \cdot \nabla \theta = 0 \quad (6.8)$$

$$\frac{\text{Ma} (1 + \text{Ma } p)}{\text{Ma}^2 \Gamma^2 (\theta - 1) + 1} \hat{\underline{u}} \cdot \nabla \hat{\underline{u}} + \nabla p + \nabla \cdot \hat{\underline{\sigma}} = 0 \quad (6.9)$$

$$\frac{3}{2} \frac{\text{Ma}^2 \Gamma^2 (1 + \text{Ma } p)}{(\text{Ma}^2 \Gamma^2 (\theta - 1) + 1)} \hat{\underline{u}} \cdot \nabla \theta + \text{Ma} \Gamma^2 \nabla \cdot \hat{\underline{q}} + (1 + \text{Ma } p) \nabla \cdot \hat{\underline{u}} + \text{Ma} \hat{\underline{\sigma}} : (\nabla \hat{\underline{u}})_{\text{STF}} = 0 \quad (6.10)$$

$$\frac{5}{2} \nabla \theta + \left( \frac{1}{\text{Ma} \Gamma^2} \right) \nabla \cdot \hat{\underline{\sigma}} + \frac{1}{2} \nabla \cdot \hat{\underline{R}} + \frac{1}{6} \nabla R = - \frac{1}{\text{Kn}} \frac{2}{3} \hat{\underline{q}} \quad (6.11)$$

$$\frac{4}{5} (\text{Ma} \Gamma^2) (\nabla \hat{\underline{q}})_{\text{STF}} + 2 (\nabla \hat{\underline{u}})_{\text{STF}} + \nabla \cdot \hat{\underline{m}} = - \frac{1}{\text{Kn}} \hat{\underline{\sigma}} \quad (6.12)$$

where  $\Gamma = c_0/u_0$  is the ratio between the thermal and mechanical velocity scales and the  $\hat{\star}$  above the non-dimensional parameters have been dropped to enhance readability. When the thermal perturbation dominates, the value of  $\Gamma$  is large and the system reduces to a ‘heat system’, whereas when the mechanical perturbation dominates, the value of  $\Gamma$  is small and the system reduced to a ‘stress system’ according to how these systems are described in A.Westerkamp and M.Torrilhon (2019) and L.Theisen (2019).

## 6.2 Lid-Driven Cavity Validation

As mentioned earlier, the solutions to the Navier-Stokes equations are valid as the Knudsen number and the Mach number approach zero. Therefore, we validate the non-linear extension by solving the benchmark problem of a lid-driven cavity at very low Knudsen and Mach numbers and comparing the solution with established CFD solutions for the Navier-Stokes equations. The results obtained from FenicsR13 are compared with the well-known solutions presented in U.Ghia, et al 1982.

The lid-driven cavity problem is defined in a square-shaped cavity filled with fluid as shown in Fig 6.1. The top boundary moves in the tangential direction with a constant velocity  $u_0$  and the other three boundaries are modelled as impenetrable solid no-slip walls.

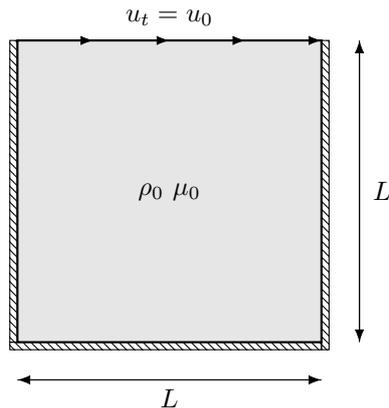


Figure 6.1: Schematic of the Lid-Driven Cavity problem.

### 6.2.1 Linear FenicsR13 Solution

In L.Theisen 2019, the results produced by FenicsR13 for the lid-driven cavity is presented. Since the solution is limited to the linearized R13 equations, these results cannot be directly compared to the results from U.Ghia, et al 1982. The streamlines and velocity profiles along the mid-planes ( $x/L = 0.5$  and  $y/L = 0.5$ ) for the case of  $u_0 = 1$ ,  $\text{Kn} = 0.0001$  and constant temperature for all boundaries is shown in Fig 5.2. We can notice that the clockwise rotation is quite symmetric which is not in accordance with the CFD results. We expect the non-linear conservations equations to be able to capture this asymmetry in streamlines within the cavity.

### 6.2.2 Non-linear FenicsR13 Solution

The solutions presented in U.Ghia, et al 1982 are based on solutions to the stream function and vorticity transport equations and consequentially, do not account for the heat generation due to the flow within the cavity. Therefore, to compare the FenicsR13 solutions to these results we proceed to only add the non-linear terms in the momentum conservation equation shown below

$$\frac{\text{Ma} (1 + \text{Ma} p) \underline{u} \cdot \nabla \underline{u}}{\text{Ma}^2 \Gamma^2 (\theta - 1) + 1} + \nabla p + \nabla \cdot \underline{\underline{\sigma}} = 0 \quad (6.13)$$

while keeping the mass and energy equations in their linear forms.

All the walls are set to a constant temperature such that  $\theta^w = 1$  and the lid is provided a tangential velocity  $u_t^w = u_0 = 1$ . We set  $\Gamma = 0$  as the process is isothermal with the temperature being constant across the domain. We consider four cases in our validation for Reynolds numbers  $\text{Re} = \{100, 1000, 3200, 5000\}$ . An unstructured mesh with 37974 triangular elements has been used for each of the following simulations. We do not consider cases with even higher Reynolds numbers as these flows tend to become quite convection dominant and therefore require stabilization schemes such as SUPG or GLS to attain convergence.

It was observed that when the Knudsen number  $\text{Kn}$  was set to very low values ( $\text{Kn} \leq 0.0001$ ), numerical inaccuracies crept into the solution which needed to be compensated by a finer grid. Therefore, even though, in principle, the FenicsR13 solutions should match with the CFD solutions at any low Knudsen number as long as the Reynolds number is identical, in some cases we have resorted to selecting specific values for Mach number and Knudsen numbers that led to good convergence at reasonable grid sizes. However, it is important to understand that the motive behind these simulations is merely to demonstrate the fact that the solutions

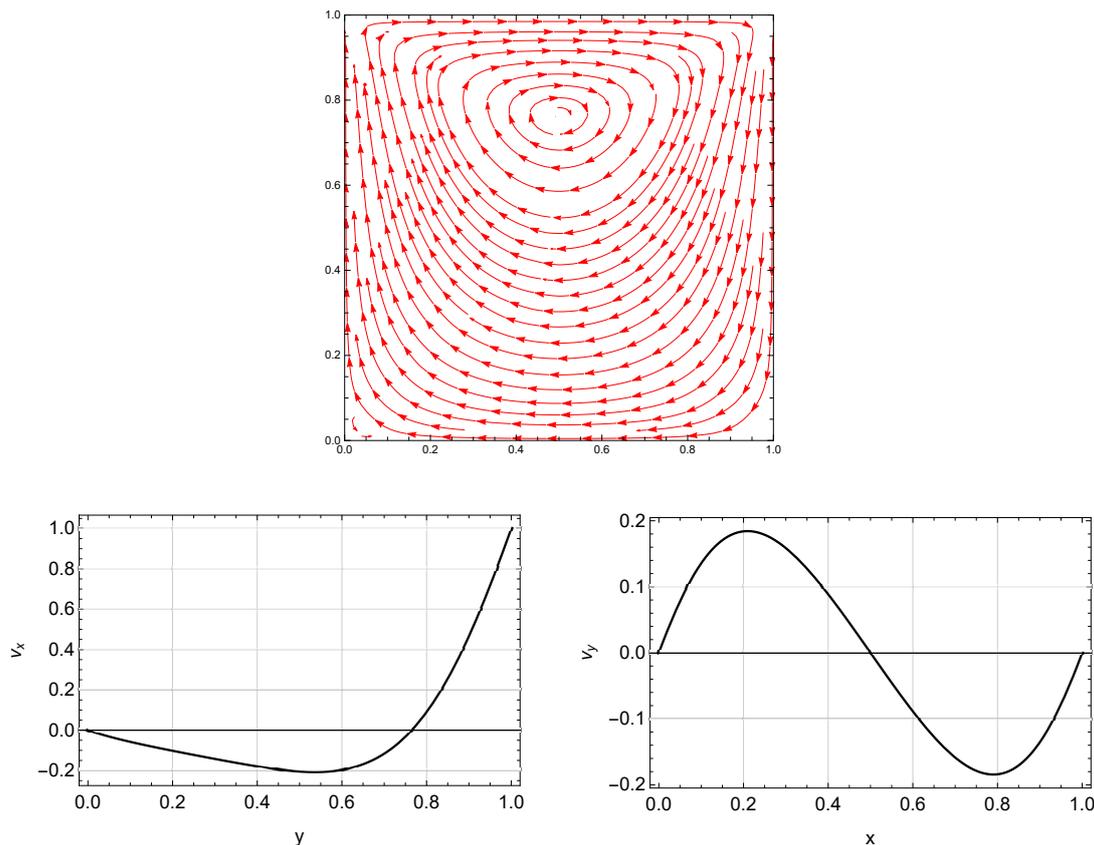


Figure 6.2: Linearized R13 equations solution of the Lid-Driven Cavity problem for the case of  $u_0 = 1$ ,  $\text{Kn} = 0.0001$  and constant temperature boundaries. The streamlines plot and velocity profiles along  $x/L = 0.5$  and  $y/L = 0.5$  are shown.

to the R13 equations do indeed compare well with classical CFD solutions at very low Knudsen and Mach numbers. The typical cases which the FenicsR13 is designed to solve are flows with moderate Knudsen number where the Navier-Stokes-Fourier equations fail. Therefore, this issue with numerical errors due to low Kn values will not be encountered in the typical problems which are intended to be solved using FenicsR13.

For the first case, the Mach and Knudsen numbers have been set to  $\text{Ma} = 0.01$  and  $\text{Kn} = 0.0001$  which corresponds to a Reynolds number of  $\text{Re} = 100$ . In Fig 6.3 the results for the case  $\text{Re} = 100$  are shown and it can be seen that the velocity streamlines produced closely resemble those presented in U.Ghia, et al 1982. Furthermore, the velocity profiles along the mid-planes ( $x/L = 0.5$  and  $y/L = 0.5$ ) are plotted and it can be seen that they show great agreement with the CFD solution. In Fig 6.4, similar comparisons for the case of  $\text{Re} = 1000$  can be seen and the results once again conform well with the CFD solutions. The streamline plots clearly show that the recirculation region in the bottom corners of the cavity have been captured by FenicsR13. Finally, the results for the cases of  $\text{Re} = 3200$  and  $\text{Re} = 5000$  are shown in Fig 6.5-6.6 where great agreement with the CFD results can be seen once again. At these Reynolds numbers the flow develops an additional small recirculation region in the top-left corner of the cavity which has been captured effectively by FenicsR13.

The comparisons made for these four cases, clearly validate the non-linear extension of the FenicsR13 solver with respect to the momentum balance. These results also strongly reinforce the fact that the R13 equations are capable of predicting flows accurately even at low Knudsen and Mach numbers.

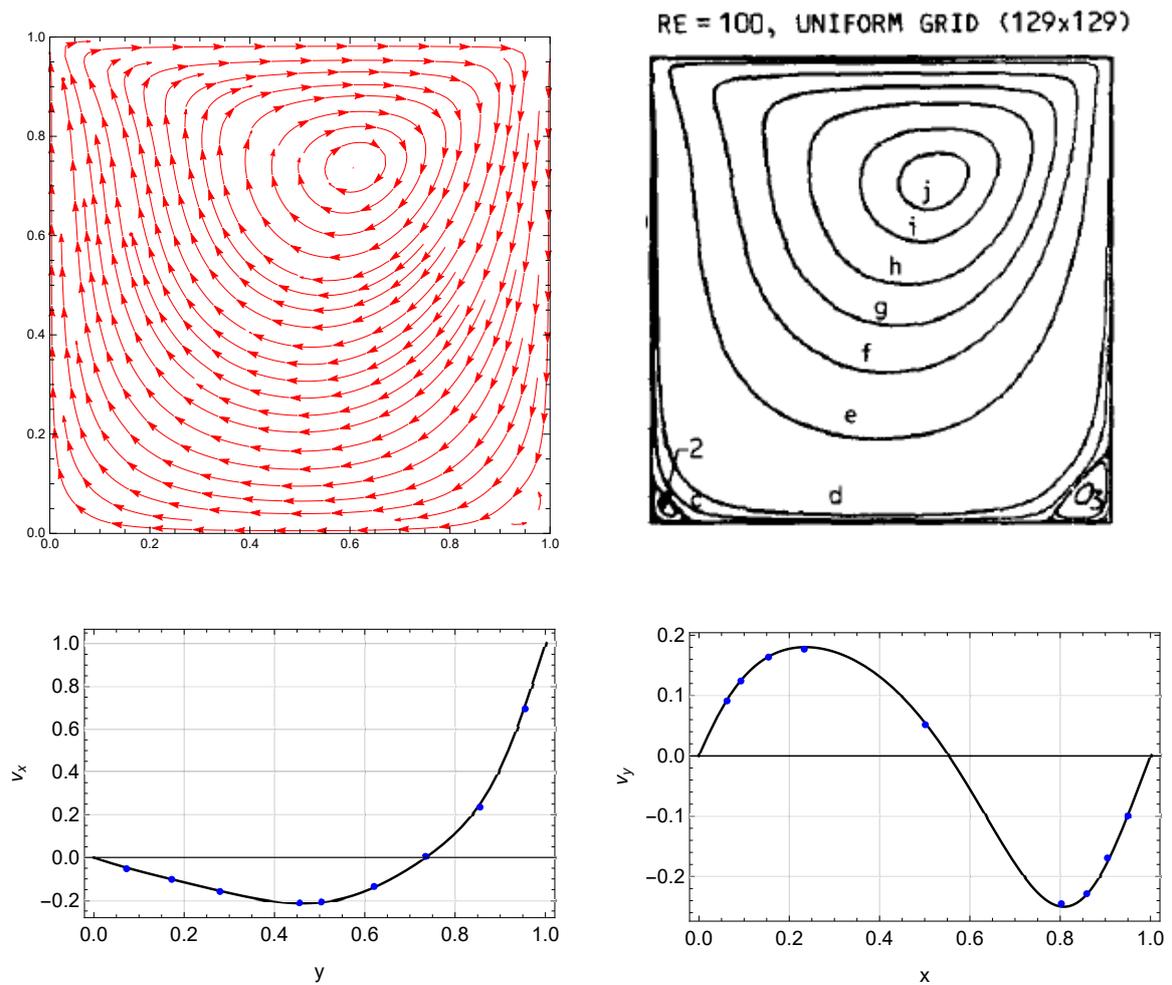


Figure 6.3: Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for  $Re = 100$ . The FenicsR13 solution corresponds to  $Ma = 0.01$  and  $Kn = 0.0001$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982.

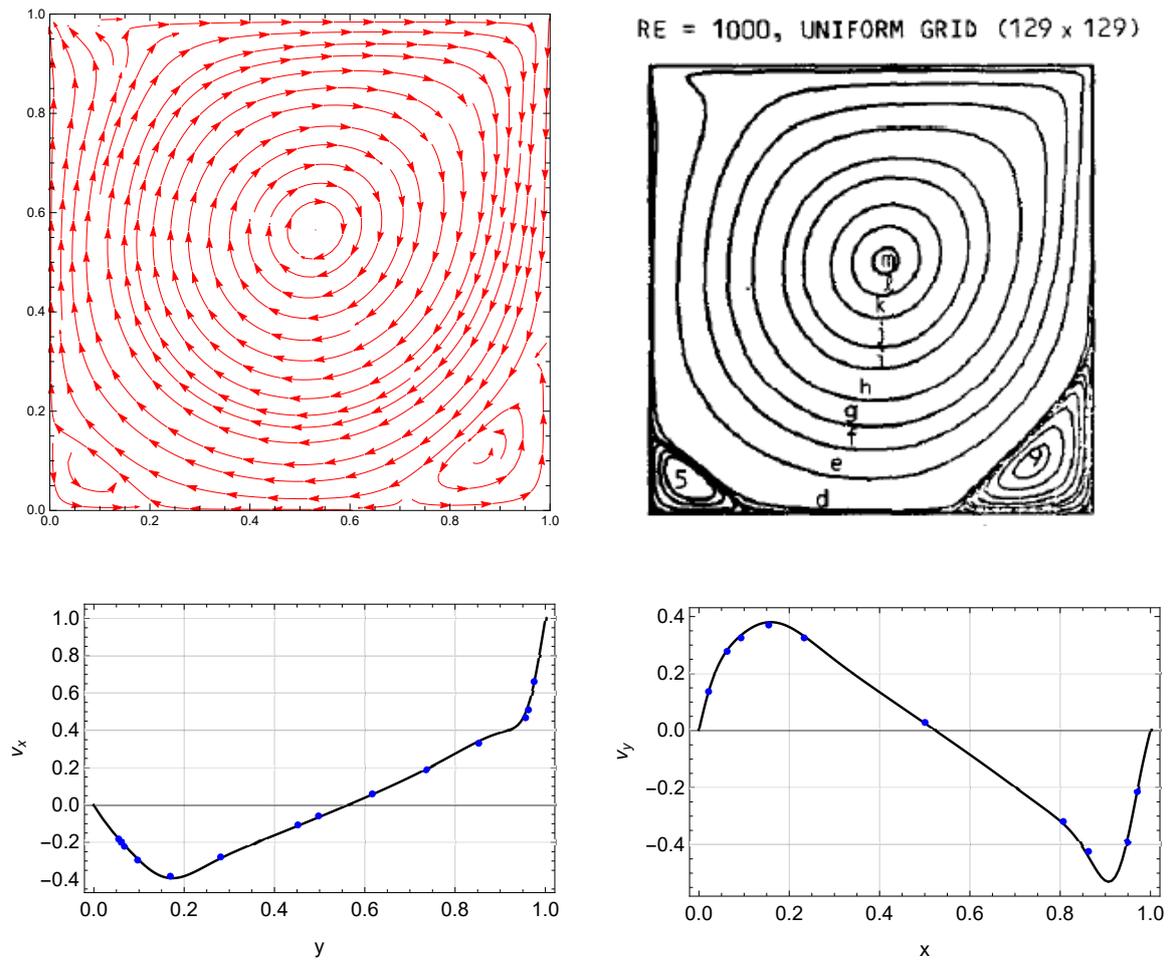


Figure 6.4: Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for  $Re = 1000$ . The FenicsR13 solution corresponds to  $Ma = 0.1$  and  $Kn = 0.0001$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982.

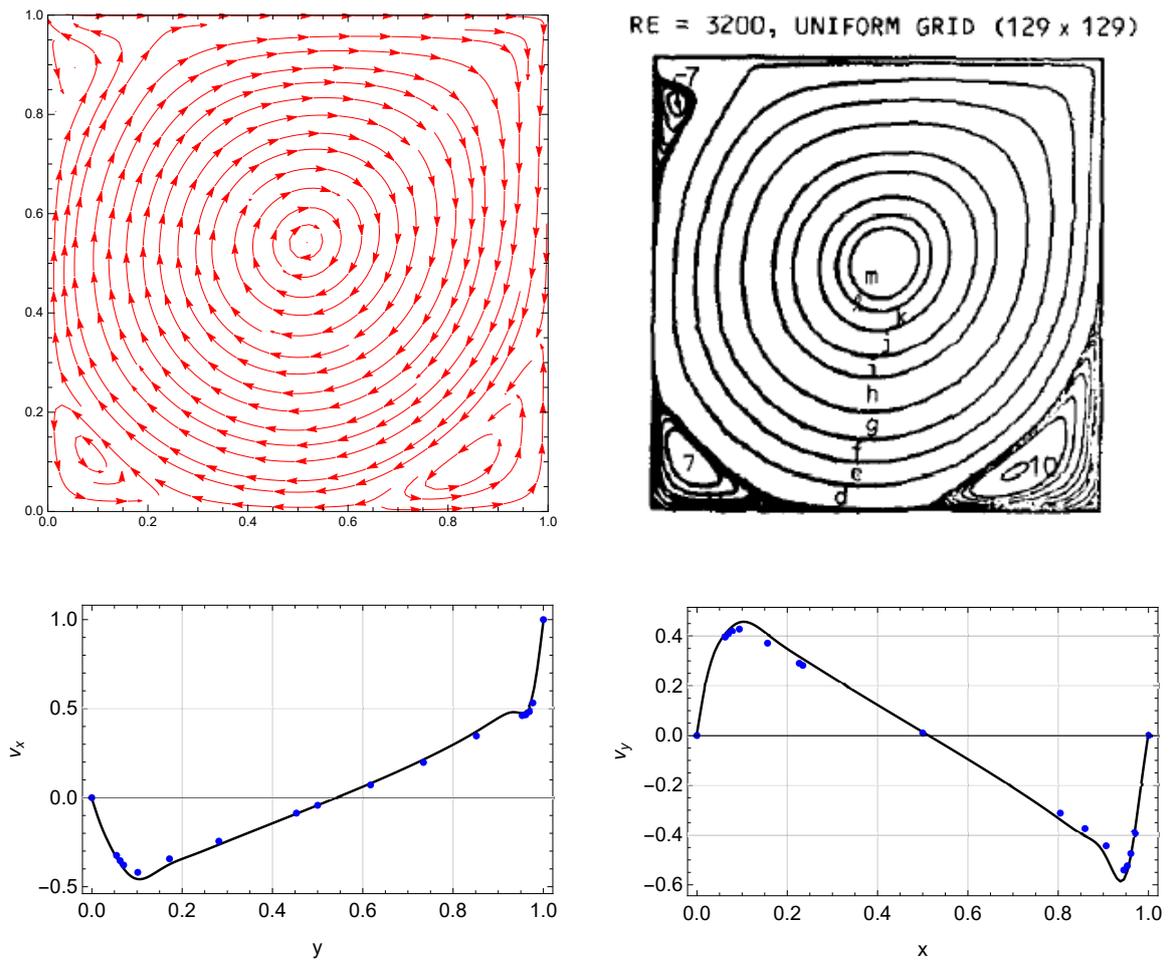


Figure 6.5: Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al [1982](#) for  $Re = 3200$ . The FenicsR13 solution corresponds to  $Ma = 0.096$  and  $Kn = 0.00003$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al [1982](#).

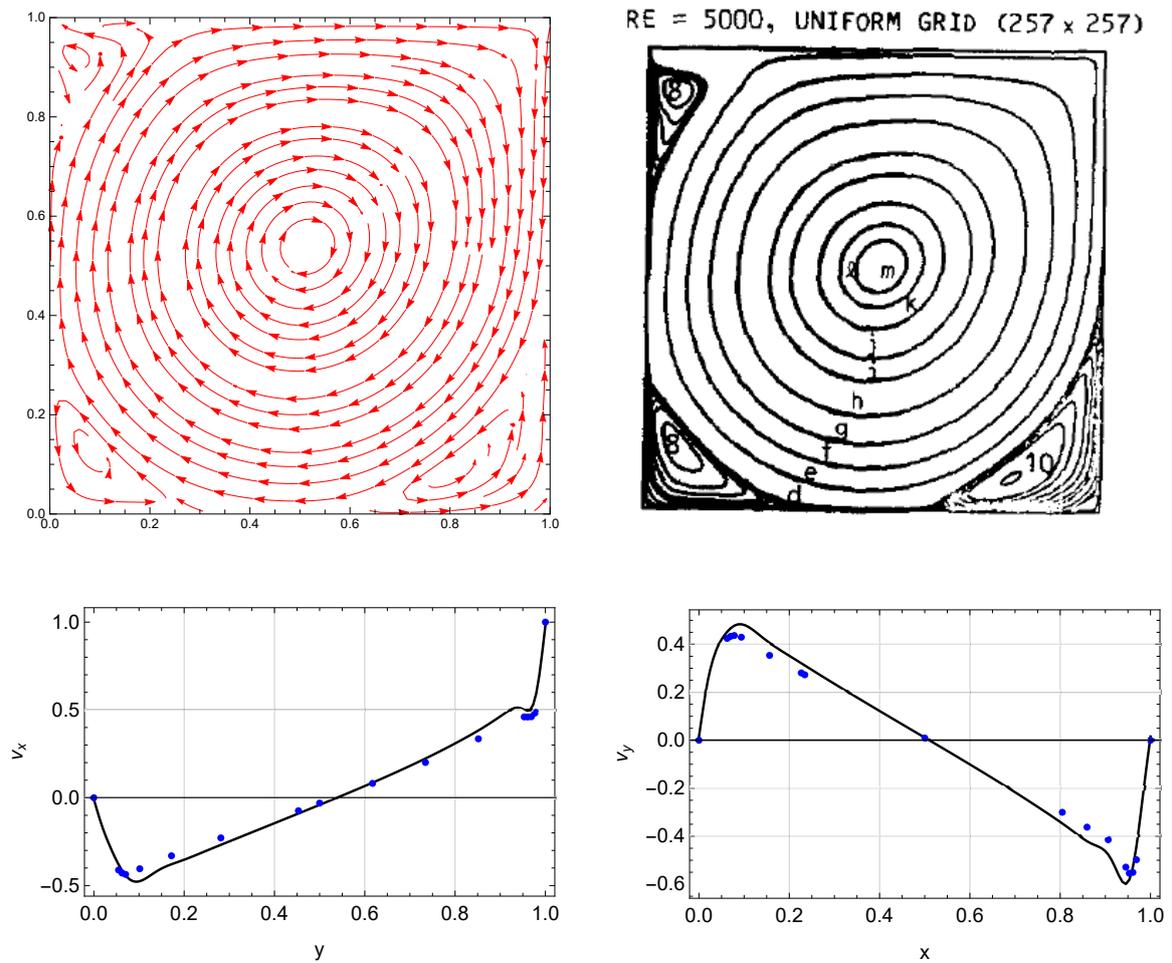


Figure 6.6: Lid-Driven Cavity : Comparison of the solution from the extended FenicsR13 and the results from U.Ghia, et al 1982 for  $Re = 5000$ . The FenicsR13 solution corresponds to  $Ma = 0.1$  and  $Kn = 0.00002$ . The streamlines plot on the top-left and the solid black lines in the velocity profile plots below correspond to FenicsR13 whereas the streamlines plot on the top-right and the blue dots in the velocity profile plots correspond to U.Ghia, et al 1982.

## 6.3 Non-Isothermal Cases for Validation

One way to validate the non-linear extension of the FenicsR13 with respect to the mass and energy conservation equations, is to validate the FenicsR13 solution for a non-isothermal lid-driven cavity problem. The results produced by FenicsR13 can be compared with the solutions produced by established commercial Navier-Stokes-Fourier solvers such as ANSYS FLUENT and COMSOL Multiphysics. In this section we start by defining such a problem with realistic physical properties for the gas and derive the corresponding boundary conditions and characteristic numbers. We then provide results for these cases produced by ANSYS FLUENT and COMSOL in terms of streamline plots and velocity profiles.

### 6.3.1 Physical Properties of Argon

The R13 equations are derived for monatomic gases such as the noble gases; Helium, Neon, Argon, Krypton, Xenon and Radon. We choose Argon in this work as it is a commonly used example for monatomic gases. Argon is the 18th element in the periodic table with a relative atomic mass of 39.948 *amu* or 39.948 *g/mol*. We can calculate the specific gas constant of Argon using its molar mass and the universal gas constant  $R = 8.314 \text{ J/mol K}$  as

$$R_{\text{sp}} = \frac{R}{M} = \frac{8.314}{39.948} = 0.2081 \text{ J/g K} = 280.1 \text{ J/Kg K}$$

We define the background temperature of the system in consideration to be  $T_0 = 20^\circ \text{C}$  and the corresponding physical properties of argon at this temperature are listed below

- Density :  $\rho_0 = 1.6617 \text{ Kg/m}^3$
- Dynamic viscosity :  $\mu_0 = 2.2294 \times 10^{-5} \text{ Kg/m s}$
- Thermal conductivity :  $\kappa_0 = 0.017391 \text{ W/m K}$
- Specific Heat :  $c_p = 521.62 \text{ J/Kg K}$
- Mean free path :  $\lambda_0 \approx 50 \text{ nm}$
- Prandtl number :  $\text{Pr} = 0.66865$

### 6.3.2 Problem Statement

When setting up this non-isothermal lid-driven cavity problem, it is important to choose an appropriate Knudsen number  $\text{Kn}$ . The value of the Knudsen number needs to be low enough such that the rarefied effects are still negligible, but at the same time the value needs to be high enough to ensure that numerical inaccuracies do not creep into the solution. Therefore, considering the mean free path of argon to be around  $\lambda_0 \approx 50 \text{ nm}$ , we select the size of the domain as  $L = x_0 = 100 \mu\text{m}$  which results in a suitable Knudsen number value of  $\text{Kn} \approx 0.0005$ . Following this, we set the temperature of the walls to the background temperature of  $T_{\text{wall}} = 20^\circ \text{C}$  and model them as solid impenetrable no-slip walls ( $u_{\text{wall}} = 0 \text{ m/s}$ ). The temperature and velocity of the lid are set to  $u_{\text{lid}} = 10 \text{ m/s}$  and  $T_{\text{lid}} = 30^\circ \text{C}$  such that  $\Gamma$  works out to be

$$\Gamma = \frac{c_0}{u_0} = \frac{\sqrt{R_{\text{sp}}(T_{\text{lid}} - T_{\text{wall}})}}{u_{\text{lid}}} = 4.562$$

The Reynolds number and Mach number of the flow are calculated as

$$\text{Re} = \frac{\rho_0 u_0 x_0}{\mu_0} = 74.51 \quad \text{Ma} = \frac{u_0}{\sqrt{\theta_0}} = 0.0405$$

An illustration of the problem can be seen in Fig 6.7.

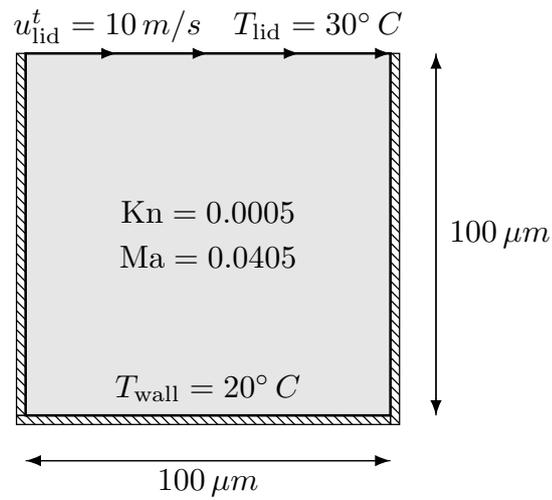
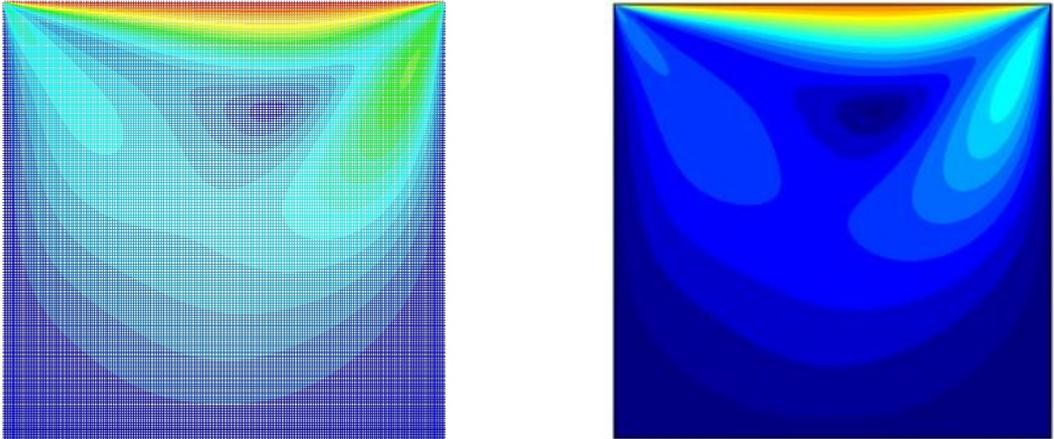
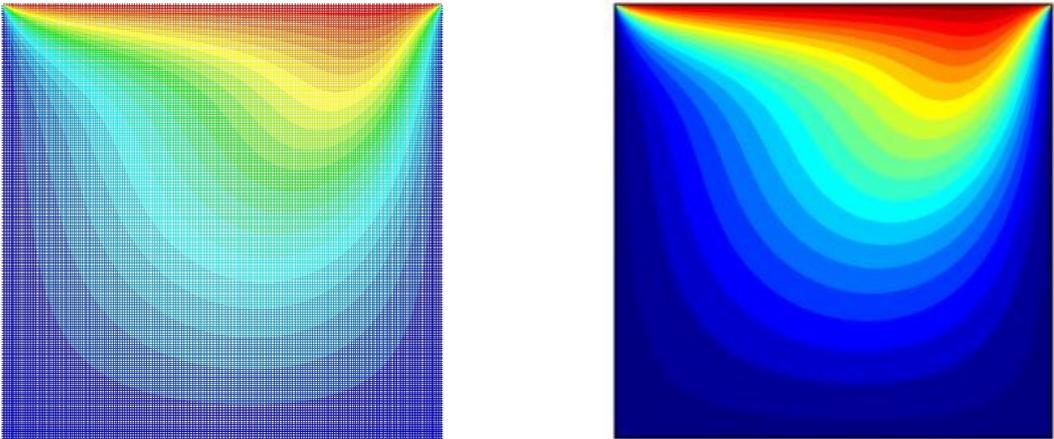


Figure 6.7: Schematic of the non-isothermal lid-driven cavity problem

This problem has been solved on ANSYS FLUENT and COMSOL and the solutions are shown below. The streamlines and profiles along the mid-planes ( $x/L = 0.5$  and  $y/L = 0.5$ ) for velocity and temperature can be seen in Fig 6.8-6.9.



(a) Velocity Contours



(b) Temperature Contours

Figure 6.8: Contours produced by FLUENT (left) and COMSOL (right) for the non-isothermal lid-driven cavity case

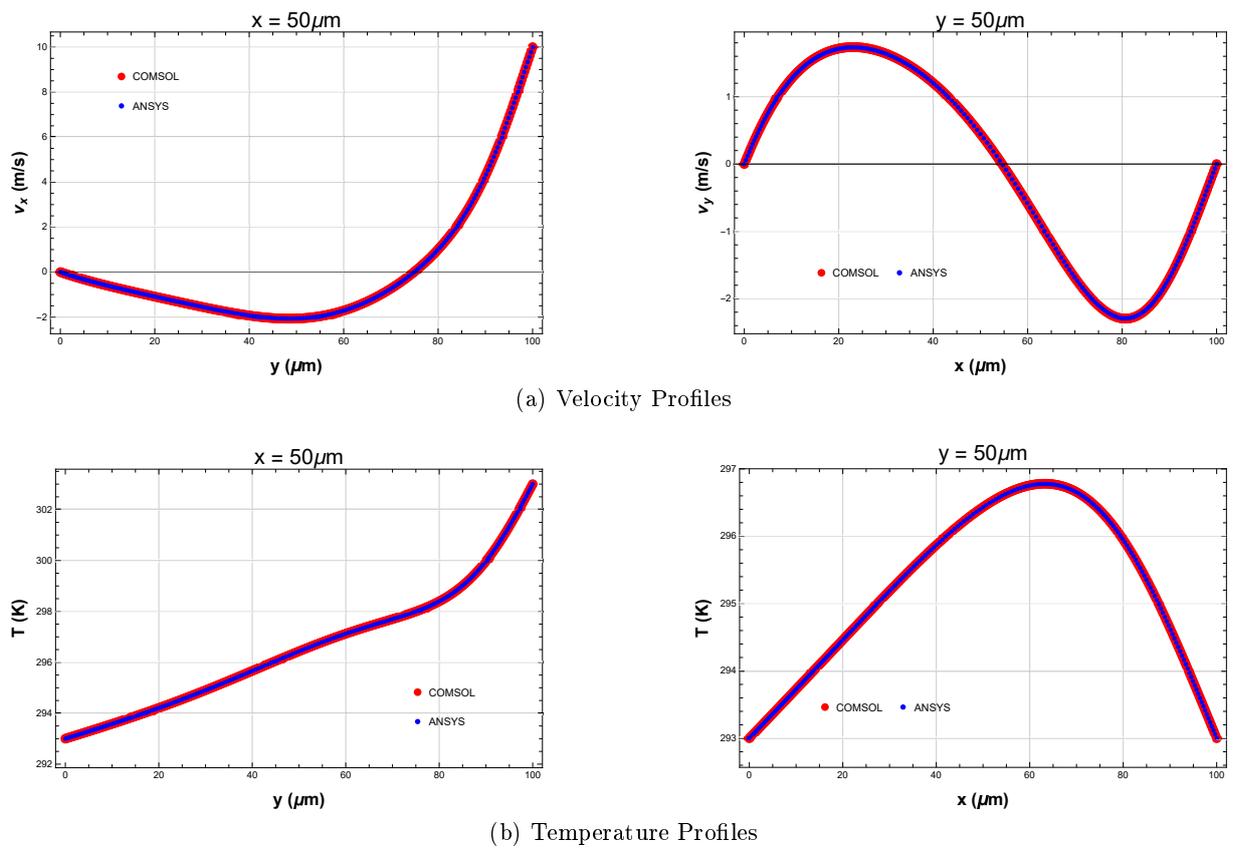


Figure 6.9: Profiles produced by FLUENT and COMSOL for the non-isothermal lid-driven cavity case

# Chapter 7

## Future Work

A summary of the scope for future research based on the work presented in thesis is elucidated in this chapter.

1. The analytical solution generating script presented in Chapter 4 is currently capable of accommodating boundary conditions of the form  $a + b \cos(\theta)$  and  $a + b \sin(\theta)$  where  $(a, b) \in \mathbb{R}$  and source terms of the forms specified in Section 4.1. This script can be further improved to extend the range of acceptable boundary conditions and source terms. However, it is important to note here that although there might be other forms for the source terms that allow for analytical solutions, not all forms of source terms lead to problems that possess exact analytical solutions.
2. In the CIP investigation carried out in Chapter 5, the effect of the Knudsen number  $\text{Kn}$  on the convergence has not been investigated. Therefore, it could be interesting to conduct a study with a wider parametric space to see if the choice of optimal stabilization parameters for the CIP Scheme is influenced by the Knudsen number of the process,
3. The extension of the FenicsR13 for the non-linear terms of the mass and energy conservation equations needs to be validated based on the non-isothermal lid-driven cavity problem that has been defined in Section 6.3. A major obstacle to this validation is the lack of convergence observed once all the non-linear terms have been included. This could be addressed through the implementation of an appropriate stabilization scheme.

## Appendix A

# Analytical Expressions of the field variables

Note that in the expressions shown below  $\log$  denotes the natural logarithm and  $I_n(\cdot)$  and  $K_n(\cdot)$  represent the modified Bessel functions of the first and second kind, respectively. The subscript  $n$  denotes the order of the Bessel Function and  $\tau = \text{Kn}$ . A list of all the integration constants is shown below.

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, C_1^{I,1}, C_1^{K,1}, C_1^{I,2}, C_1^{K,2}, C_2^{I,1}, C_2^{K,1}, C_2^{I,2}, C_2^{K,2}, C_3^{I,1}, C_3^{K,1}, C_3^{I,2}, C_3^{K,2})$$

### A.1 Temperature

$$\begin{aligned} \theta(r, \phi) = \cos(\phi) & \left[ -\frac{16}{15} K_1 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{K,2} - \frac{16}{15} I_1 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{I,2} + \frac{r^3 (M_1 - Q_1)}{30\tau^3} + \frac{c_{11}r}{\tau} + \frac{c_{10}\tau}{r} \right] \\ & - \frac{8}{15} K_0 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{K,1} - \frac{8}{15} I_0 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{I,1} + \frac{r^4 (M_2 - Q_2)}{60\tau^4} - \frac{4}{15} c_3 \log \left( \frac{r}{\tau} \right) \\ & + \frac{r^2 (5M_0 - 84M_2 - 5Q_0 + 124Q_2)}{75\tau^2} + c_5 \end{aligned} \quad (\text{A.1})$$

### A.2 Pressure

$$\begin{aligned} p(r, \phi) = \cos(\phi) & \left[ -\frac{8}{3} I_1 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{I,2} - \frac{8}{3} K_1 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{K,2} + \frac{r^3 (3F_{22} + G_2)}{8\tau^3} + \frac{2F_1 r^2}{3\tau^2} - G_{11} \right. \\ & \left. + \frac{r (3M_1 + 2Q_1)}{5\tau} - \frac{2c_9 r}{\tau} + \frac{c_8 \tau}{r} \right] - \frac{4}{3} I_0 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{I,1} - \frac{4}{3} K_0 \left( \frac{\sqrt{\frac{5}{6}} r}{\tau} \right) C_1^{K,1} \\ & + \frac{F_2 r^3}{3\tau^3} + \frac{F_0 r}{\tau} + \frac{4r^2 (3M_2 + 2Q_2)}{15\tau^2} + c_4 \end{aligned} \quad (\text{A.2})$$

### A.3 Velocity

$$\begin{aligned}
u_r(r, \phi) = \cos(\phi) & \left[ \left( I_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) - I_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) \frac{2C_3^{I,2}}{5} + \left( K_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) - K_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) \frac{2C_3^{K,2}}{5} \right. \\
& + \frac{r^2(-25F_{22} - 75G_2 + 42M_1 - 2Q_1)}{120\tau^2} + \frac{r^4(F_{22} + 3G_2)}{192\tau^4} + \frac{F_1r^3}{45\tau^3} - \frac{52F_1r}{45\tau} + \frac{48F_1\tau}{25r} - \frac{c_9r^2}{4\tau^2} \\
& \left. + \frac{c_7\tau^2}{4r^2} - \frac{3c_{10}\tau^2}{2r^2} + c_8 \left( \frac{5\tau^2}{3r^2} - \frac{1}{2} \log \left( \frac{r}{\tau} \right) + \frac{1}{2} \right) + c_{12} \right] + \frac{M_2r^3}{4\tau^3} + \frac{M_0r}{2\tau} + \frac{c_2\tau}{2r} - \frac{2c_3\tau}{5r}
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
u_\phi(r, \phi) = \sin(\phi) & \left[ \left( I_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) + I_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) \frac{2C_3^{I,2}}{5} + \left( K_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) + K_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) \frac{2C_3^{K,2}}{5} \right. \\
& - \frac{r^2(-25F_{22} - 75G_2 + 2M_1 - 2Q_1)}{40\tau^2} - \frac{5r^4(F_{22} + 3G_2)}{192\tau^4} - \frac{4F_1r^3}{45\tau^3} + \frac{104F_1r}{45\tau} + \frac{3c_9r^2}{4\tau^2} \\
& - \frac{3c_{10}\tau^2}{2r^2} + \frac{c_7\tau^2}{4r^2} - c_8 \left( -\frac{5\tau^2}{3r^2} - \frac{1}{2} \log \left( \frac{r}{\tau} \right) \right) - c_{12} \left. \right] - \frac{2}{5} I_1 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) C_3^{I,1} \\
& - \frac{2}{5} K_1 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) C_3^{K,1} + \frac{G_0r^4}{81\tau^4} - \frac{G_1r^3}{8\tau^3} - \frac{52G_0r^2}{81\tau^2} + \frac{16G_0}{15} + \frac{c_6r}{\tau} + \frac{c_1\tau}{2r}
\end{aligned} \tag{A.4}$$

### A.4 Heat Flux

$$\begin{aligned}
q_r(r, \phi) = \cos(\phi) & \left[ \left( I_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) - I_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) C_3^{I,2} + \left( K_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) - K_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) C_3^{K,2} \right. \\
& + \frac{3r^2(F_{22} + 3G_2 - 2M_1 + 2Q_1)}{16\tau^2} + \frac{3}{10}(9F_{22} + 27G_2 - 15M_1 + 20Q_1) + \frac{F_1r}{2\tau} + \frac{15c_{10}\tau^2}{4r^2} \\
& \left. - \frac{3c_8\tau^2}{2r^2} - 3c_9 - \frac{15c_{11}}{4} \right] + \frac{r^3(Q_2 - M_2)}{4\tau^3} + \frac{r(Q_0 - M_0)}{2\tau} + \frac{c_3\tau}{r}
\end{aligned} \tag{A.5}$$

$$\begin{aligned}
q_\phi(r, \phi) = -\sin(\phi) & \left[ \left( I_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) + I_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) C_3^{I,2} + \left( K_0 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) + K_2 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) \right) C_3^{K,2} \right. \\
& + \frac{r^2(9F_{22} + 27G_2 - 2M_1 + 2Q_1)}{16\tau^2} + \frac{3}{10}(9F_{22} + 27G_2 - 15M_1 + 20Q_1) + \frac{F_1r}{\tau} + \frac{3c_8\tau^2}{2r^2} \\
& \left. - \frac{15c_{10}\tau^2}{4r^2} - 3c_9 - \frac{15c_{11}}{4} \right] + I_1 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) C_3^{I,1} + K_1 \left( \frac{\sqrt{\frac{5}{9}}r}{\tau} \right) C_3^{K,1} + \frac{5G_0r^2}{18\tau^2} - \frac{3G_1r}{2\tau}
\end{aligned} \tag{A.6}$$

## A.5 Stress

$$\begin{aligned}
\sigma_{rr}(r, \phi) = & \cos(\phi) \left[ \left( I_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - I_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{I,2} + \left( \frac{5}{3} I_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + I_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{I,2} \right. \\
& + \left( K_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - K_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{K,2} + \left( \frac{5}{3} K_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + K_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{K,2} \\
& + \left. \frac{r^3(-F_{22} - 3G_2)}{24\tau^3} - \frac{2F_1 r^2}{15\tau^2} + \frac{6F_1}{5} - \frac{2r(3M_1 + 2Q_1)}{15\tau} + \frac{c_7 \tau^3}{r^3} + \frac{c_9 r}{\tau} + \frac{c_8 \tau}{r} \right] \\
& + \left( I_2 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - I_0 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{I,1} + \left( \frac{1}{3} I_0 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + I_2 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{I,1} \\
& + \left( K_2 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - K_0 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{K,1} + \left( \frac{1}{3} K_0 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + K_2 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{K,1} \\
& + \frac{r^2(-3M_2 - 2Q_2)}{6\tau^2} + \frac{1}{75}(-15M_0 - 72M_2 - 10Q_0 - 48Q_2) + \frac{c_2 \tau^2}{r^2}
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
\sigma_{r\phi}(r, \phi) = & \sin(\phi) \left[ \left( I_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - I_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{I,2} + \left( I_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) - I_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{I,2} \right. \\
& + \left( K_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) - K_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{K,2} + \left( K_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) - K_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{K,2} \\
& + \frac{r^3(F_{22} + 3G_2)}{12\tau^3} + \frac{F_1 r^2}{5\tau^2} - \frac{3F_1}{5} + \frac{r(3M_1 + 2Q_1)}{15\tau} + \frac{c_7 \tau^3}{r^3} - \frac{c_9 r}{\tau} \left. - \frac{G_0 r^3}{27\tau^3} \right. \\
& + \left. \frac{G_1 r^2}{4\tau^2} + \frac{G_0 r}{3\tau} + \frac{c_1 \tau^2}{r^2} \right]
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
\sigma_{\phi\phi}(r, \phi) = & -\cos(\phi) \left[ \left( 3I_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) + I_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{I,2} + \left( \frac{1}{3} I_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + I_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{I,2} \right. \\
& + \left( 3K_1 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) + K_3 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{K,2} + \left( \frac{1}{3} K_1 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) + K_3 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{K,2} \\
& + \left. \frac{r^3(-F_{22} - 3G_2)}{24\tau^3} - \frac{2F_1 r^2}{15\tau^2} + \frac{6F_1}{5} + \frac{c_7 \tau^3}{r^3} + \frac{c_9 r}{\tau} + \frac{c_8 \tau}{r} \right] \\
& - \left( I_0 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) + I_2 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{I,1} - \left( I_2 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) - \frac{1}{3} I_0 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{I,1} \\
& - \left( K_0 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) + K_2 \left( \frac{\sqrt{\frac{3}{2}}r}{\tau} \right) \right) C_2^{K,1} - \left( K_2 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) - \frac{1}{3} K_0 \left( \frac{\sqrt{\frac{5}{6}}r}{\tau} \right) \right) C_1^{K,1} \\
& - \frac{r^2(-3M_2 - 2Q_2)}{30\tau^2} + \frac{1}{75}(-15M_0 - 72M_2 - 10Q_0 - 48Q_2) - \frac{c_2 \tau^2}{r^2}
\end{aligned} \tag{A.9}$$



# Appendix B

## Case Descriptions

### B.1 Case I

```

1 # Numerical Parameters
2 # =====
3 # - elements: Must contain the fields: theta, s, p, u, sigma
4 # - fields: List of FEM parameters (shape, degree)
5 # - shape: Element shape, e.g. Lagrange
6 # - degree: Element degree, e.g. 2
7 # - stabilization: Must contain cip
8 # - cip: Collection of Continous Interior Penalty (CIP) parameters
9 # - enable: Enable CIP stabilization
10 # - delta_theta: Stabilization of grad(T)*grad(T_test) over edge
11 # - delta_u: Stabilization of grad(u)*grad(u_test) over edge
12 # - delta_p: Stabilization of grad(p)*grad(p_test) over edge
13 elements:
14   theta:
15     shape: Lagrange
16     degree: 1
17   s:
18     shape: Lagrange
19     degree: 1
20   p:
21     shape: Lagrange
22     degree: 1
23   u:
24     shape: Lagrange
25     degree: 1
26   sigma:
27     shape: Lagrange
28     degree: 1
29 stabilization:
30   cip:
31     enable: True
32     delta_theta: 1.0
33     delta_u: 1.0
34     delta_p: 0.01
35
36 # Formulation Parameters
37 # =====
38 # - nsd: Number of spatial dimensions == 2
39 # - mode: Formulation mode, one of heat, stress, r13
40 # - kn: Knudsen numberkn
41 # - chi_tilde: Refraction coefficient in Maxwell accomodation model
42 # - heat_source: Heat source function for mode==heat||r13
43 # - mass_source: Mass source function for mode==stress||r13
44 # - body_force: Body force for mode==stress||r13
45 nsd: 2
46 mode: r13
47 kn: 1.0
48 chi_tilde: 1.0
49 heat_source: 0
50 mass_source: 0
51 body_force: [0,0]
52
53 # Boundary Conditions
54 # =====
55 # - bcs: Dictionary of all boundary IDs from mesh
56 # - bc_id: must contain theta_w, u_t_w, u_n_w, p_w, epsilon_w
57 # - theta_w: Value for temperature at wall
58 # - u_t_w: Value for tangential velocity at wall
59 # - u_n_w: Value for normal velocity at wall
60 # - p_w: Value for pressure at wall
61 # - epsilon_w: Inflow-model parameter <=> Weight of pressure prescription
62 bcs:
63   3000:
64     theta_w: 1
65     u_t_w: 0
66     u_n_w: 0
67     p_w: 0
68     epsilon_w: pow(10,-3)
69   3100:
70     theta_w: 2
71     u_t_w: -1.00 * sin(phi)
72     u_n_w: +1.00 * cos(phi)
73     p_w: -0.27 * cos(phi)
74     epsilon_w: pow(10,+3)

```

## B.2 Case II

```

1 # Numerical Parameters
2 # =====
3 # - elements: Must contain the fields: theta, s, p, u, sigma
4 # - fields: List of FEM parameters (shape, degree)
5 # - shape: Element shape, e.g. Lagrange
6 # - degree: Element degree, e.g. 2
7 # - stabilization: Must contain cip
8 # - cip: Collection of Continuous Interior Penalty (CIP) parameters
9 # - enable: Enable CIP stabilization
10 # - delta_theta: Stabilization of grad(T)*grad(T_test) over edge
11 # - delta_u: Stabilization of grad(u)*grad(u_test) over edge
12 # - delta_p: Stabilization of grad(p)*grad(p_test) over edge
13 elements:
14   theta:
15     shape: Lagrange
16     degree: 1
17   s:
18     shape: Lagrange
19     degree: 1
20   p:
21     shape: Lagrange
22     degree: 1
23   u:
24     shape: Lagrange
25     degree: 1
26   sigma:
27     shape: Lagrange
28     degree: 1
29 stabilization:
30   cip:
31     enable: True
32     delta_theta: 1.0
33     delta_u: 1.0
34     delta_p: 0.01
35
36 # Formulation Parameters
37 # =====
38 # - nsd: Number of spatial dimensions == 2
39 # - mode: Formulation mode, one of heat, stress, r13
40 # - kn: Knudsen numberkn
41 # - chi_tilde: Refraction coefficient in Maxwell accommodation model
42 # - heat_source: Heat source function for mode==heat||r13
43 # - mass_source: Mass source function for mode==stress||r13
44 # - body_force: Body force for mode==stress||r13
45 nsd: 2
46 mode: r13
47 kn: 1.0
48 chi_tilde: 1.0
49 heat_source: 1.0 * (1.0 - (5.0*pow(R,2))/(18.0*pow(kn,2))) * cos(phi)
50 mass_source: 1.0 * (1.0 - (5.0*pow(R,2))/(18.0*pow(kn,2))) * cos(phi)
51 body_force: [0,0]
52
53 # Boundary Conditions
54 # =====
55 # - bcs: Dictionary of all boundary IDs from mesh
56 # - bc_id: must contain theta_w, u_t_w, u_n_w, p_w, epsilon_w
57 # - theta_w: Value for temperature at wall
58 # - u_t_w: Value for tangential velocity at wall
59 # - u_n_w: Value for normal velocity at wall
60 # - p_w: Value for pressure at wall
61 # - epsilon_w: Inflow-model parameter <=> Weight of pressure prescription
62 bcs:
63   3000:
64     theta_w: 1.0
65     u_t_w: -10
66     u_n_w: 0
67     p_w: 0
68     epsilon_w: 0
69   3100:
70     theta_w: 0.5
71     u_t_w: 0
72     u_n_w: 0
73     p_w: 0
74     epsilon_w: 0

```

Figure B.2: Case II

### B.3 Case III

```

1 # Numerical Parameters
2 # =====
3 # - elements: Must contain the fields: theta, s, p, u, sigma
4 # - fields: List of FEM parameters (shape, degree)
5 # - shape: Element shape, e.g. Lagrange
6 # - degree: Element degree, e.g. 2
7 # - stabilization: Must contain cip
8 # - cip: Collection of Continuous Interior Penalty (CIP) parameters
9 # - enable: Enable CIP stabilization
10 # - delta_theta: Stabilization of grad(T)*grad(T_test) over edge
11 # - delta_u: Stabilization of grad(u)*grad(u_test) over edge
12 # - delta_p: Stabilization of grad(p)*grad(p_test) over edge
13 elements:
14   theta:
15     shape: Lagrange
16     degree: 1
17   s:
18     shape: Lagrange
19     degree: 1
20   p:
21     shape: Lagrange
22     degree: 1
23   u:
24     shape: Lagrange
25     degree: 1
26   sigma:
27     shape: Lagrange
28     degree: 1
29 stabilization:
30   cip:
31     enable: True
32     delta_theta: 1.0
33     delta_u: 1.0
34     delta_p: 0.01
35
36 # Formulation Parameters
37 # =====
38 # - nsd: Number of spatial dimensions == 2
39 # - mode: Formulation mode, one of heat, stress, r13
40 # - kn: Knudsen numberkn
41 # - chi_tilde: Refraction coefficient in Maxwell accommodation model
42 # - heat_source: Heat source function for mode==heat||r13
43 # - mass_source: Mass source function for mode==stress||r13
44 # - body_force: Body force for mode==stress||r13
45 nsd: 2
46 mode: r13
47 kn: 1.0
48 chi_tilde: 1.0
49 heat_source: 0
50 mass_source: 0
51 body_force_R: (1/kn)*(1 + (1*((R/kn) + ((9*kn)/(5*R)))) + 1*(pow(R,2)/pow(kn,2)))*cos(phi) + 1*(pow(R,2)/pow(kn,2))
52 body_force_Theta: (1/kn)*(1*(1-((5*pow(R,2))/(27*pow(kn,2)))) + 1*(R/kn) + (1*(pow(R,2)/pow(kn,2)) + 1*(kn/R))*sin(phi))
53
54 # Boundary Conditions
55 # =====
56 # - bcs: Dictionary of all boundary IDs from mesh
57 # - bc_id: must contain theta_w, u_t_w, u_n_w, p_w, epsilon_w
58 # - theta_w: Value for temperature at wall
59 # - u_t_w: Value for tangential velocity at wall
60 # - u_n_w: Value for normal velocity at wall
61 # - p_w: Value for pressure at wall
62 # - epsilon_w: Inflow-model parameter <=> Weight of pressure prescription
63 bcs:
64   3000:
65     theta_w: 1.0
66     u_t_w: -10
67     u_n_w: 0
68     p_w: 0
69     epsilon_w: pow(10,-5)
70   3100:
71     theta_w: 0.5
72     u_t_w: 0
73     u_n_w: 0
74     p_w: 0
75     epsilon_w: pow(10,-5)

```

Figure B.3: Case III

## B.4 Case IV

```

1 # Numerical Parameters
2 # =====
3 # - elements: Must contain the fields: theta, s, p, u, sigma
4 # - fields: List of FEM parameters (shape, degree)
5 # - shape: Element shape, e.g. Lagrange
6 # - degree: Element degree, e.g. 2
7 # - stabilization: Must contain cip
8 # - cip: Collection of Continuous Interior Penalty (CIP) parameters
9 # - enable: Enable CIP stabilization
10 # - delta_theta: Stabilization of grad(T)*grad(T_test) over edge
11 # - delta_u: Stabilization of grad(u)*grad(u_test) over edge
12 # - delta_p: Stabilization of grad(p)*grad(p_test) over edge
13 elements:
14   theta:
15     shape: Lagrange
16     degree: 1
17   s:
18     shape: Lagrange
19     degree: 1
20   p:
21     shape: Lagrange
22     degree: 1
23   u:
24     shape: Lagrange
25     degree: 1
26   sigma:
27     shape: Lagrange
28     degree: 1
29 stabilization:
30   cip:
31     enable: True
32     delta_theta: 1.0
33     delta_u: 1.0
34     delta_p: 0.01
35
36 # Formulation Parameters
37 # =====
38 # - nsd: Number of spatial dimensions == 2
39 # - mode: Formulation mode, one of heat, stress, r13
40 # - kn: Knudsen numberkn
41 # - chi_tilde: Refraction coefficient in Maxwell accommodation model
42 # - heat_source: Heat source function for mode==heat||r13
43 # - mass_source: Mass source function for mode==stress||r13
44 # - body_force: Body force for mode==stress||r13
45 nsd: 2
46 mode: r13
47 kn: 1.0
48 chi_tilde: 1.0
49 heat_source: (1/kn)*(0.1 + 0.2*(R/kn)*cos(phi) + 0.3*(pow(R,2)/pow(kn,2)))
50 mass_source: (1/kn)*(0.1 + 0.2*(R/kn)*cos(phi) + 0.3*(pow(R,2)/pow(kn,2)))
51 body_force_R: (1/kn)*(0.1 + (0.2*((R/kn) + ((9*kn)/(5*R)))) + 0.4*(pow(R,2)/pow(kn,2)))*cos(phi) + 0.3*(pow(R,2)/pow(kn,2)
52 )
53 body_force_Theta: (1/kn)*(0.1*(1-((5*pow(R,2))/(27*pow(kn,2)))) + 0.2*(R/kn) + (0.3*(pow(R,2)/pow(kn,2)) + 0.4*(kn/R))*
54 sin(phi))
55 # Boundary Conditions
56 # =====
57 # - bcs: Dictionary of all boundary IDs from mesh
58 # - bc_id: must contain theta_w, u_t_w, u_n_w, p_w, epsilon_w
59 # - theta_w: Value for temperature at wall
60 # - u_t_w: Value for tangential velocity at wall
61 # - u_n_w: Value for normal velocity at wall
62 # - p_w: Value for pressure at wall
63 # - epsilon_w: Inflow-model parameter <=> Weight of pressure prescription
64 bcs:
65   3000:
66     theta_w: 1.0
67     u_t_w: -10
68     u_n_w: 0
69     p_w: 0
70     epsilon_w: pow(10,-3)
71   3100:
72     theta_w: 2.0
73     u_t_w: -1.00 * sin(phi)
74     u_n_w: +1.00 * cos(phi)
75     p_w: -0.27 * cos(phi)
76     epsilon_w: pow(10,+3)

```

Figure B.4: Case IV

# Bibliography

- A.Logg, G.N.Wells and J.Hake (2012). “DOLFIN: a C++/Python Finite Element Library”. In: 84, pp. 173–225. URL: [https://doi.org/10.1007/978-3-642-23099-8\\_10](https://doi.org/10.1007/978-3-642-23099-8_10).
- A.Meurer, C.P.Smith and Paprocki (Jan. 2017). “SymPy: symbolic computing in Python”. In: *PeerJ Computer Science* 3, e103. ISSN: 2376-5992. DOI: [10.7717/peerj-cs.103](https://doi.org/10.7717/peerj-cs.103). URL: <https://doi.org/10.7717/peerj-cs.103>.
- A.Rana, M.Torrilhon and H.Struchtrup (2013). “A Robust Numerical Method for the R13 Equations of Rarefied Gas Dynamics: Application to Lid Driven Cavity”. In: *Journal of Computational Physics* 236, pp. 169–186. URL: <https://doi.org/10.1016/j.jcp.2012.11.023>.
- A.Westerkamp (2012). “Finite Element Discretizations for Extended Gas Dynamics”. MA thesis. RWTH Aachen University. URL: [http://www.mathcces.rwth-aachen.de/\\_media/3teaching/00projects/2012\\_masterthesis\\_arminwesterkamp.pdf](http://www.mathcces.rwth-aachen.de/_media/3teaching/00projects/2012_masterthesis_arminwesterkamp.pdf).
- (2017). “A Continuous Interior Penalty Method for the Linear Regularized 13-Moment Equations Describing Rarefied Gas Flows”. PhD thesis. RWTH Aachen. URL: [http://www.mathcces.rwth-aachen.de/\\_media/3teaching/00projects/2017\\_phd\\_arminwesterkamp.pdf](http://www.mathcces.rwth-aachen.de/_media/3teaching/00projects/2017_phd_arminwesterkamp.pdf).
- A.Westerkamp and M.Torrilhon (2012). “Slow Rarefied Gas Flow Past a Cylinder: Analytical Solution in Comparison to the Sphere”. In: *AIP Conference Proceedings* 1501, pp. 207–214. DOI: [10.1063/1.4769505](https://doi.org/10.1063/1.4769505).
- (2014). “Stabilization Techniques in Finite Element Discretizations for Moment Approximations”. In: *AIP Conference Proceedings* 1628, pp. 1016–1023. URL: <https://doi.org/10.1063/1.204902705>.
- (2017). “Curvature-Induced Instability of a Stokes-Like Problem with Non-Standard Boundary Conditions”. In: *Applied Numerical Mathematics* 121, pp. 96–114. URL: <https://doi.org/10.1016/j.apnum.2017.06.012>.
- (2019). “Finite Element Methods for the Linear Regularized 13-Moment Equations Describing Slow Rarefied Gas Flows”. In: *Journal of Computational Physics* 389, pp. 1–21. URL: <https://doi.org/10.1016/j.jcp.2019.03.022>.
- H.Struchtrup (2005). *Macroscopic transport equations for rarefied gas flows*. Springer Berlin Heidelberg. URL: [https://doi.org/10.1007/3-540-32386-4\\_9](https://doi.org/10.1007/3-540-32386-4_9).
- H.Struchtrup and M.Torrilhon (2003). “Regularization of Grad’s 13 moment equations: Derivation and linear analysis”. In: *Physics of Fluids* 15.9, pp. 2668–2680. DOI: [10.1063/1.1597472](https://doi.org/10.1063/1.1597472). eprint: <https://doi.org/10.1063/1.1597472>. URL: <https://doi.org/10.1063/1.1597472>.
- Inc., Wolfram Research (n.d.). *Mathematica, Version 12.1*. Champaign, IL, 2020. URL: <https://www.wolfram.com/mathematica>.
- J.Donea and A.Huerta (2003). *Finite Element Methods for Flow Problems*. John Wiley and Sons. DOI: [10.1002/0470013826](https://doi.org/10.1002/0470013826). URL: <https://doi.org/10.1002/0470013826>.
- L.Theisen (2019). “Simulation of Non-Equilibrium Gas Flows Using the FEniCS Computing Platform”. MA thesis. RWTH Aachen. URL: [http://www.mathcces.rwth-aachen.de/\\_media/3teaching/00projects/2019\\_ma\\_lamberttheisen.pdf](http://www.mathcces.rwth-aachen.de/_media/3teaching/00projects/2019_ma_lamberttheisen.pdf).

- L.Theisen and M.Torrilhon (2020). “fenicsR13: A Tensorial Mixed Finite Element Solver for the Linear R13 Equations Using the FEniCS Computing Platform”. In: *arXiv*. URL: <https://arxiv.org/abs/2007.05944v1>.
- M.Benzi, G.H.Golub and J.Liesen (2005). “Numerical solution of saddle point problems”. In: 14, pp. 1–137. URL: <https://doi.org/10.1017/S0962492904000212>.
- M.Torrilhon (2010). “Slow gas microflow past a sphere: Analytical solution based on moment equations”. In: *Physics of Fluids* 22, p. 072001. URL: <https://doi.org/10.1063/1.3453707>.
- M.Torrilhon and N.Sarna (2017). “Hierarchical Boltzmann simulations and model error estimation”. In: *Journal of Computational Physics* 342, pp. 66–84. URL: <http://dx.doi.org/10.1016/j.jcp.2017.04.041>.
- U.Ghia, et al (1982). “High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method”. In: *Journal of Computational Physics*. DOI: [https://doi.org/10.1016/0021-9991\(82\)90058-4](https://doi.org/10.1016/0021-9991(82)90058-4).