

NUMERICAL SIMULATION OF
COMPRESSIBLE GAS FLOW
COUPLED TO HEAT CONDUCTION
IN TWO SPACE DIMENSIONS

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Mathematics and Statistics
University of Saskatchewan
Saskatoon

By
Daria Korneeva

©Daria Korneeva, May 2011. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mathematics & Statistics,
142 McLean Hall,
106 Wiggins Road,
University of Saskatchewan,
Saskatoon, Saskatchewan
Canada
S7N 5E6

ABSTRACT

The current thesis studies a model of two dimensional convection of an ideal gas in a rectangular domain having walls of finite thickness. The temperature outside of walls is considered constant ($T_{left} > T_{in} > T_{right}$). Heat exchange between walls and outside/inside air is computed using Newton's law of cooling. Heat transfer inside walls is modelled with the heat equation. The mathematical model inside enclosure involves Navier-Stokes equations coupled with equation of state for gas. The model is numerically studied using the method of large particles. One of the main goals of the current thesis was to develop a software in C# language for numerical solution of the above-described model. Physically meaningful results, including stream lines and distribution of parameters of gas and temperature inside solid walls were obtained.

ACKNOWLEDGEMENTS

First of all I am grateful to my supervisor Prof. Alexei Cheviakov for his guidance and invaluable advice. His comments and corrections greatly contributed to the writing of this work, and it could not be finished without his constant support. I would like to thank the members of my committee, Prof. Mikelis G. Bickis, Prof. Chris Soteris, Prof. George Patrick, and my external examiner, Prof. Sam Butler for reading my thesis. Last but not least, I want to thank my family and friends, for their support and encouragement.

Dedicated to my family.

CONTENTS

Permission to Use	i
Abstract	ii
Acknowledgements	iii
Contents	v
List of Tables	vii
List of Figures	viii
1 Introduction	1
1.1 Basic equation of gas and fluid dynamics	1
1.1.1 Continuity equation (Conservation of mass)	1
1.1.2 Equation of motion (Conservation of momentum)	2
1.1.3 Navier-Stokes equations	2
1.1.4 Equations of state	3
1.1.5 Laminar and turbulent flows. Reynolds numbers.	6
1.2 Basic computational method for fluid/gas dynamics	7
1.2.1 Finite difference method	7
1.2.2 Lax-Friedrichs method	8
1.2.3 Godunov's method	9
1.2.4 Finite element method	10
1.2.5 Finite volume method	13
1.2.6 Eulerian and Lagrangian flow description	14
1.2.7 Coupled Eulerian-Lagrangian (CEL) methods	14
1.2.8 Particle-In-Cell (PIC) method	15
1.2.9 Monte Carlo methods (MCM) methods	16
1.2.10 Spectral methods	16
1.3 Discussion	16
2 The method of large particles	19
2.1 Overview	19
2.2 Eulerian stage	20
2.3 Lagrangian stage	21
2.4 Final stage	23
2.5 Stability condition and error estimation	24
2.6 Artificial and numerical viscosity	24
2.7 Discussion	25

3	The mathematical model and numerical computations	26
3.1	Test model 1	26
3.1.1	Effects of step sizes	28
3.1.2	Effects of artificial viscosity	29
3.2	Test model 2	30
3.3	Main model	31
4	Conclusions and future work	34
4.1	Conclusion	34
	References	36
A	Programming code	38
A.1	Code for solving heat equation	38
A.2	Euler stage code	39
A.3	Lagrange stage code	40
A.4	Final stage code	42
A.5	Boundary conditions for final stage	42
B	Pictures of results	44

LIST OF TABLES

LIST OF FIGURES

1.1	Stencil for Lax-Friedrichs method	8
3.1	Test model 1	26
3.2	Test model 1, Input form	27
3.3	Test model 2	30
3.4	Main model	31
3.5	Types of heat transfer	32
B.1	Test model 1, graphs for density (left) ρ , temperature (middle) T and pressure (right) p at time 2.52 s for $N = 50$, $h_0 = 0.04$ m, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.	44
B.2	Test model 1, graphs for velocity field at time 2.52 s	45
B.3	Test model 1, graphs for density (left) ρ , temperature (middle) T and pressure (right) p at time 12 s	45
B.4	Test model 1, graphs for average temperature (left) and energy (right)	46
B.5	Test model 1, graphs for x and y components of velocity field and velocity field for $N = 50$, $h_0 = 0.04$ m, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.	46
B.6	Test model 1, velocity fields for $N = 10$, $h = 0.2$ m $> h_0$ (left) and $N = 30$, $h = 0.067$ m $> h_0$ (right), $\tau_0 = 0.008$, $\mu = 1$ m ² /s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s.	47
B.7	Test model 1, velocity fields for $N = 70$, $h = 0.028$ m $< h_0$ (left) and $N = 100$, $h = 0.02$ m $< h_0$ (right), $\tau_0 = 0.008$, $\mu = 1$ m ² /s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.	47
B.8	Test model 1, velocity fields for $\tau = 0.005 < \tau_0$ (left) and $\tau = 0.01 > \tau_0$ (left), $N = 50$, $h = h_0 = 0.04$ m, $\mu = 1$ m ² /s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.	48
B.9	Test model 1, graphs of differences $ u - u_0 $ and $ v - v_0 $ of x - and y - components of velocity field for $N = 50$, $h = h_0 = 0.04$ m, $\tau = 0.01$ and τ_0 . Here $\mu = 1$ m ² /s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.	48
B.10	Test model 1, graphs for x - and y - components of the velocity field and streamlines for computations without artificial viscosity: $\mu = 0$, $N = 50$, $h = 0.4$ m, $\tau = 0.008$, $L_x = L_y = 20$ m, $V_{in} = V_{out} = 10$ m/s, $x_1 = 8$ m, $x_2 = 12$ m, $x_3 = 8$ m, $x_4 = 12$ m.	49
B.11	Test model 1, graphs for x - and y - components of the velocity field and streamlines for computations without artificial viscosity: $\mu = 1$ m ² /s, $N = 50$, $h = 0.4$ m, $\tau = 0.008$, $L_x = L_y = 20$ m, $V_{in} = V_{out} = 10$ m/s, $x_1 = 8$ m, $x_2 = 12$ m, $x_3 = 8$ m, $x_4 = 12$ m.	49
B.12	Test model 2, graphs for velocity at time 10 s.	50
B.13	Test model 2,graphs for velocity at time 18.9 s.	50

B.14	Test model 2,graphs for velocity at time 31 s.	51
B.15	Main model, initial temperature distribution at time 0.3 s.	51
B.16	Main model, initial velocity field at time 0.3 s.	52
B.17	Main model, final temperature distribution at 171 s.	52
B.18	Main model, final form of the velocity field inside gas chamber at 171 s.	53
B.19	Main model, comparison of results for different glass thermal conductivity at time 41 s. The conductivity is $K_0 = 3.4E - 7 W/m \cdot K$ (left) and $K_0 = 3.4E - 4 W/m \cdot K$ (right).	53
B.20	Main model, comparison of velocity fields for different viscosity values at time 41 s, $\mu = 1 m^2/s$ (left) and $\mu = 2 m^2/s$ (right).	54
B.21	Main model, comparison of temperature distribution inside glasses result for different outside temperatures at time 41s. $T_{left} = 300 K$ (left) and $T_{left} = 320 K$ (right).	54
B.22	Main model, comparison of velocity fields for different mesh sizes at time 25s (left: $N = 50, h = 0.04 m$, right: $N = 100, h = 0.02 m$).	55
B.23	Main model, a typical instability result for gas velocity components when the artificial viscosity is insufficient.	55

CHAPTER 1

INTRODUCTION

The current thesis is devoted to a mathematical model involving a two-dimensional flow of a compressible gas in a rectangular chamber, coupled through the Newton's law of cooling to the process of heat conduction in walls of the chamber.

The research is motivated by the problem of modeling high-performance sealed multi-pane windows. In cold climates, sophisticated window designs are used in order to provide sufficient thermal insulation. A reliable model for such windows is required to take into account both gas dynamics and heat transfer effects. Models considered within the current thesis were created as initial steps towards a complete and reliable model of window units, which would be capable of accurate prediction of heat transfer properties of actual windows that satisfy a given set of specifications.

1.1 Basic equation of gas and fluid dynamics

In this chapter discussed necessary theory for describing model of this thesis, also represented different basic methods in area of fluid dynamics.

1.1.1 Continuity equation (Conservation of mass)

The continuity equation represents the law of conservation of mass. The quantity of fluid entering a certain volume in space must be balanced by the quantity leaving, unless compression occurs [17]. One can obtain this by using equality between net rate of mass outflow and net rate of mass inflow at arbitrary control volume. The continuity equation is given by

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{V}) = 0 \quad (1.1)$$

where $\mathbf{V} = (u, v, w)$ is velocity vector field and ρ is density.

1.1.2 Equation of motion (Conservation of momentum)

Another important quantity in the description of continuum gas/fluid dynamics is the linear momentum density. This equates the rate of change of momentum density of a body of the fluid with the sum of all forces acting on that body of fluid. Thus using the second Newton's law of classical mechanics, one can obtain a vector equation(three scalar equations):

$$\rho \frac{D\mathbf{V}}{Dt} \equiv \rho \left(\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} \right) = \rho \mathbf{F} - \text{grad } p + \mu \Delta \mathbf{V}, \quad (1.2)$$

where \mathbf{F} is the body force per unit mass acting on the fluid element, \mathbf{V} is velocity vector, μ is viscosity coefficient and p is pressure.

1.1.3 Navier-Stokes equations

The Navier-Stokes equations describe the motion of a viscous compressible fluid in a given domain. They are given by

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{V}) = 0, \\ \rho \left(\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} \right) = \rho F - \text{grad } p + \mu \Delta \mathbf{V}. \end{array} \right. \quad (1.3)$$

The inviscid fluid case for the Navier-Stokes equations is known as Euler's equations and are given by

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{V}) = 0, \\ \rho \left(\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} \right) = \rho F - \text{grad } p. \end{array} \right. \quad (1.4)$$

There are four independent variables in the equations: the spatial coordinates x, y, z , and the time t , and five dependent variables: the pressure p , the density ρ , and three components of the velocity vector \mathbf{V} [16]. It follows that both Euler and Navier-Stokes systems need an extra equation in order to be closed. There are different possibilities in different cases, which are described below.

1.1.4 Equations of state

Ideal gas and its generalization

Gases are highly compressible in comparison to liquids, with changes in gas density directly related to changes in pressure and temperature through the classical equation of state for ideal gas:

$$p = \rho RT \quad (1.5)$$

where R is the gas constant, p is pressure, ρ is density and T is the absolute temperature. It is used to closely approximate the behavior of real gases under normal conditions when the gases are not approaching liquefaction.

The kind of effect which intermolecular forces have on the equations of state of a ideal gas can be illustrated by examining modified equations of state.

In the hard sphere model of a gas, each molecule has a definite volume so that if V is volume of the vessel containing one mole of the gas, then the volume which is not occupied by molecules is $V - b$, where b is approximately N_0 times the volume of one molecule. The pressure is thereby increased, so that instead of classical equation of state one should write

$$p = \frac{RT}{V - b} \quad (1.6)$$

Equation (1.6) is known as the Clausius equation of state. To include attractive forces, one must consider the reduced velocity with which the molecules on the average strike the walls of the container. The reduction in pressure will be proportional both to the number of molecules striking the wall and to the number of molecules in the interior of the container. Both are proportion to the number density of molecules, n . Then p will be decreased proportionally to n^2 , or $\frac{1}{V^2}$. Thus

$$p = \frac{RT}{V - b} - \frac{a}{V^2} \quad (1.7)$$

where a is constant. Equation (1.7) is called van der Waals equation of state.

Conservation of energy

To close system (1.3) or (1.4), one can add equation of state such as the ideal gas law, but it will add new unknown T , and consequently system will be needed one more equation such

as conservation of energy

$$\frac{\partial \rho E}{\partial t} + \operatorname{div}(\rho E \mathbf{V}) + \operatorname{div}(p \mathbf{V}) = 0 \quad (1.8)$$

where energy is assumed to be

$$E = \rho e + \frac{1}{2} \rho (u^2 + v^2 + w^2) \quad (1.9)$$

is the total energy per unit volume. In (1.9), e being the internal energy per unit mass for the fluid. The internal energy at constant volume of an ideal gas is given by:

$$e = C_V n R T \quad (1.10)$$

where C_V is a constant dependent on temperature, and n is the amount of substance of the gas.

Incompressibility condition

All liquids and gases are compressible to some extent that is changes in pressure or temperature will result in changes in density. However, in many situations the changes in pressure and temperature are sufficiently small that the changes in density are negligible. In this case the flow can be modeled as an incompressible flow. Mathematically, incompressibility is expressed by saying that the density ρ of a fluid parcel does not change as it moves in the flow field, i.e., ρ , is a constant throughout the flow field so that (1.1) becomes

$$\operatorname{div} \mathbf{V} = 0 \quad (1.11)$$

This additional constraint simplifies the governing equations, especially in the case when the fluid has a uniform density. Otherwise the more general compressible flow equations must be used. Incompressibility condition plays the role of the equation of state. It is used in many applications, in particular, for distribution of water, plasma flow and etc. For flow of gases, to determine whether to use compressible or incompressible fluid dynamics, the Mach number is a commonly used dimensionless parameter. The Mach number Ma at any point in a compressible flow field is given by ratio of the local stream velocity \mathbf{V} and the speed of sound c :

$$Ma = \frac{|\mathbf{V}|}{c} \quad (1.12)$$

The speed of sound is computed from the local thermodynamic conditions of the stream. For an ideal gas,

$$c = c_{ideal} = \sqrt{\gamma \frac{p}{\rho}} = \sqrt{\frac{\gamma RT}{M}} = \sqrt{\frac{\gamma kT}{m}} \quad (1.13)$$

where R is the molar gas constant, k is the Boltzmann constant, γ is the adiabatic index, T is the absolute temperature, M is the molar mass, m is the mass of a single molecule.

When the Mach number is relatively small ($\lesssim 0.2 - 0.3$), the inertial forces induced by the fluid motion are not sufficiently large to cause a significant change in the fluid density, and in this case the compressibility of the fluid can be neglected. Thus to close system (1.3) or (1.4), one can use condition (1.11) as extra equation.

Adiabatic process

Processes in which no heat is transferred to, or removed from, the system and which the work is done reversibly are known as adiabatic, isentropic or reversible, processes. So using the definition of an adiabatic process is that heat transfer to the system is zero and the first law of thermodynamic, one can obtain the mathematical equation for an ideal fluid undergoing a reversible adiabatic process

$$pV^{\frac{\alpha+1}{\alpha}} = pV^\gamma = \text{const}, \quad (1.14)$$

where p is pressure, V is specific or molar volume, and

$$\gamma = \frac{C_p}{C_V} = \frac{\alpha + 1}{\alpha}, \quad (1.15)$$

C_p being the specific heat for constant pressure, C_V being the specific heat for constant volume, γ is the adiabatic index, and α is the number of degrees of freedom of molecule divided by 2 ($\frac{3}{2}$ for monatomic gas, $\frac{5}{2}$ for diatomic gas). Using the ideal gas law and assuming a constant molar quantity, one also can obtain

$$T_2 = T_1 \left(\frac{p_2}{p_1} \right)^{\frac{\gamma-1}{\gamma}} \quad (1.16)$$

Incompressible flows there are two additional unknowns: density and temperature. the continuity and momentum equation must now be augmented by the equation of state and the energy equation to render the system determinable. The second law of thermodynamics

asserts that all the thermodynamic systems possess a state variable called entropy, s , defined by

$$s_1 - s_2 = \Delta s = \int_r ev \frac{dQ}{T}. \quad (1.17)$$

this implies that the change of entropy between two states 1 and 2 is the sum of all $\frac{dQ}{T}$ produced by any reversible change which joins two states. For a perfect gas,

$$s = \int C_V \frac{dT}{T} + R \ln v + \text{const} \quad (1.18)$$

So, for compressible gas to close system (1.3) or (1.4), one can combine the ideal gas equation of state (1.5) with adiabatic condition:

$$\frac{\partial S}{\partial t} + \mathbf{V} \cdot \text{grad } S = 0, \quad P = \rho^\gamma e^{\frac{S}{C_V}}, \quad (1.19)$$

where S is the entropy density.

1.1.5 Laminar and turbulent flows. Reynolds numbers.

Laminar flow is characterized by low velocity and calm flow without distribution, swirls of fluid and mixing between layers, whereas turbulent flow is characterized by chaotic, stochastic property changes. The Reynolds number

$$Re = \frac{\rho |\mathbf{V}| l}{\mu} \quad (1.20)$$

was created as a criterion to distinguish laminar and turbulent flow. In (1.20), l is characteristic length, \mathbf{V} is characteristic velocity, ρ is characteristic density, and μ is viscosity.

The Reynolds number is a measure of the ratio of the inertia force on an element of fluid to the viscous force on an element. Flows with high Reynolds numbers usually become turbulent, while those with low Reynolds numbers usually remain laminar. For example, for pipe flows, Reynolds numbers above about 4000 will most likely correspond to turbulent flow, while a Reynold's number below 2100 indicates laminar flow. The region in between ($2100 < Re < 4000$) is called the transition region.

1.2 Basic computational method for fluid/gas dynamics

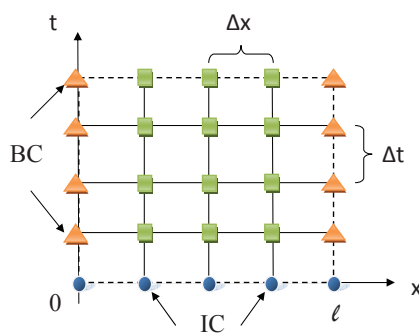
1.2.1 Finite difference method

In order to illustrate method, we apply it to the one-dimensional diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad (1.21)$$

For using finite difference method first of all one should establish the domain where the equation is studied, i.e., area in spatial and time coordinates. If, for example, we model the diffusion of a gas into a tube of length l the spatial domain is the interval of the Ox axis associated to this length, i.e., $(0, l)$. The time domain begins at $t = 0$ and ends at final time $t = T$, i.e., $(0, T)$. Concluding the equation domain is $\Omega = (0, l) \times (0, T) \subset \mathbb{R}^2$.

Second step is to choose the grid. The grid can be chosen by determining steps for all axis and grid will be formed by intersection of hyper surface. For this one should choose constant step sizes $\Delta x = x_{j+1} - x_j$ and $\Delta t = t_{k+1} - t_k$. Then the nodes will be the intersection of these straight lines.



After these procedures one is able to discretize the equation by the derivatives by finite differences. If $u_j^k = u(x_j, t_k)$, then for the node (x_j, t_k) equation is

$$\frac{u_j^{k+1} - u_j^k}{\Delta t} = \frac{u_{j-1}^k - 2u_j^k + u_{j+1}^k}{(\Delta x)^2} \quad (1.22)$$

which could be reset in the form

$$u_j^{k+1} = \frac{\Delta t}{(\Delta x)^2} u_{j-1}^k + \left(1 - 2 \frac{\Delta t}{(\Delta x)^2}\right) u_j^k + \frac{\Delta t}{(\Delta x)^2} u_{j+1}^k \quad (1.23)$$

Applying these formulas for any $j = 1, \dots, n-1$, one can see that from known values for $k = 0$ (the initial conditions) we can calculate those for $k = 1$, then from these values we calculate those for $k=2$ and so on. At each step, we must know the values u_0^k and u_n^k (from the boundary conditions) in order to complete the time level values.

1.2.2 Lax-Friedrichs method

This method is first order finite difference method, which is build using stencil on figure 1.1.

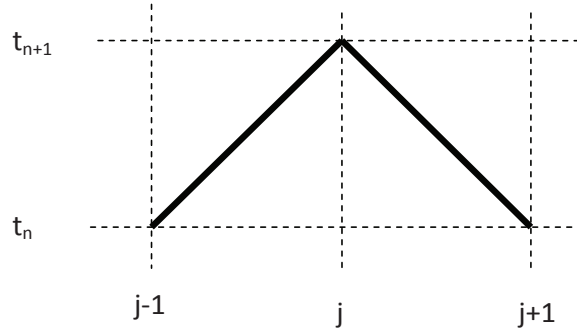


Figure 1.1: Stencil for Lax-Friedrichs method

First of all, let's approximate this linear equation with this method

$$\frac{\partial u}{\partial t} + A \frac{\partial u}{\partial x} = 0, \quad A = \text{const.} \quad (1.24)$$

The value of u_j^n in the time derivative $\frac{\partial u}{\partial t}$ is replaced by average at neighbor nodes:

$$\frac{\partial u}{\partial t} = \frac{u_j^{n+1} - \frac{1}{2}(u_{j-1}^n + u_{j+1}^n)}{\Delta t} \quad (1.25)$$

and

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^n - u_{j-1}^n}{2 \Delta x}. \quad (1.26)$$

Using (1.25) and (1.26) for approximation of (1.24) one gets the expression for u_j^{n+1} :

$$u_j^{n+1} = \frac{1}{2}(u_{j-1}^n + u_{j+1}^n) - \frac{\Delta t}{2 \Delta x} A(u_{j+1}^n - u_{j-1}^n). \quad (1.27)$$

The method is stable for the linear hyperbolic equation (1.24) if $|\frac{A\Delta t}{\Delta x}| \leq 1$. [13]

The method can be extended to solve, for example, Riemann problems for nonlinear systems (conservation laws) given by

$$\frac{\partial}{\partial t} \mathbf{V} + \frac{\partial}{\partial x} f(\mathbf{V}) = 0 \quad (1.28)$$

with initial conditions

$$\mathbf{V}_0(x) = \mathbf{V}(x, 0) = \begin{cases} \mathbf{V}_l & x < 0 \\ \mathbf{V}_r & x > 0 \end{cases}. \quad (1.29)$$

For any finite difference model of (1.28) a robust first order numerical flux function $F_{j+\frac{1}{2}}^n = F(\mathbf{V}_j^n, \mathbf{V}_{j+1}^n)$ usually is used in the conservation form

$$\mathbf{V}_j^{n+1} = \mathbf{V}_j^n - \frac{\Delta t}{\Delta x} (F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n). \quad (1.30)$$

In conservation form Lax-Friedrichs scheme defined through the numerical flux [22]

$$F_{j+\frac{1}{2}}^n = \frac{\Delta t}{2 \Delta x} (\mathbf{V}_{j+1}^n - \mathbf{V}_j^n) + \frac{1}{2} \{f(\mathbf{V}_{j+1}^n) + f(\mathbf{V}_j^n)\}. \quad (1.31)$$

For the one-dimensional Euler equations, \mathbf{V} and $f(\mathbf{V})$ have form

$$\mathbf{V} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad f(\mathbf{V}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix}, \quad E = \rho \left(\frac{u^2}{2} + p \right) \quad (1.32)$$

Lax-Friedrichs method has first order precision in time and space, and is stable when $|\frac{\Delta t}{\Delta x} f'(\mathbf{V}_j^n)| \leq 1$ for all \mathbf{V}_j^n . However this method has very strong dissipation, which is often not acceptable for computing discontinuous solutions.

1.2.3 Godunov's method

Godunov's method is built by using information about characteristic solution of Riemann problems (1.28) forward in time.

First step is to define initial conditions $\tilde{\mathbf{V}}^n(x, t_n)$ as piecewise constant function with the value v_j^n on the grid cell $x_{j-\frac{1}{2}} \leq x \leq x_{j+\frac{1}{2}}$, but $\tilde{\mathbf{V}}^n$ is not a constant over the time $t_n \leq t \leq t_{n+1}$ and $\tilde{\mathbf{V}}^n(x, t_n)$ is used as initial data for the conservation law. Then conservation law is solved exactly to get $\tilde{\mathbf{V}}^n(x, t)$ for $t_n \leq t \leq t_{n+1}$. After obtaining the solution over $t_n \leq t \leq t_{n+1}$, one needs to define the piecewise constant approximate solution at t_{n+1} using the formula

$$\tilde{\mathbf{V}}^{n+1} = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tilde{\mathbf{V}}^n(x, t_{n+1}) dx. \quad (1.33)$$

And then process can be repeated using $\tilde{\mathbf{V}}^{n+1}(x, t_{n+1})$ as initial data for time t_{n+2} . Also this process can be written in conservative form

$$v_j^{n+1} = v_j^n - \frac{\Delta t}{\Delta x} (F(v_{j+1}^n) - F(v_{j-1}^n)), \quad (1.34)$$

with numerical flux

$$F(v_{j+1}^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(\tilde{\mathbf{V}}^n(x_{j+\frac{1}{2}}, t)) dt. \quad (1.35)$$

The method is stable when $|\lambda_p(v_j^n) \frac{\Delta t}{\Delta x}| \leq 1$ for all eigenvalues λ_p at each v_j^n . Also the method has numerical viscosity $s \frac{\Delta t}{\Delta x}$, where s is Rankine-Hugoniot shock speed. Godunov's method, similarly to Lax-Friedrichs method, is accurate to the first order in time and space. In practice to solve Riemann problem at every cell boundary in each time step is very expensive and the exact solution is average over the cell. Therefore modern methods, where Godunov's scheme is a building block, use approximate Riemann solution obtained by less expensive algorithms.

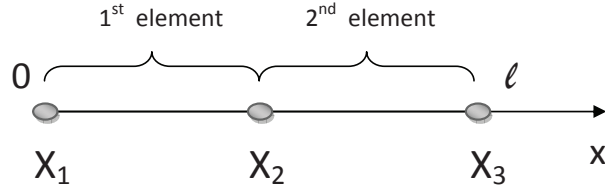
1.2.4 Finite element method

The solution of a continuum problem by the finite element method is approximated by the following step-by-step process.

1. *Discretize the continuum*

Divide the solution region into non-overlapping elements or sub-regions. The finite element discretization allows a variety of element shapes, for example, triangles, quadrilaterals. Each element is formed by the connection of a certain number of nodes

As example to illustrate method we would apply it to the same problem as in section 1.2.1. So to discretize the solution region, one can divide the spatial domain $(0, l)$ into elements, for example $(x_1, x_2) \cup (x_2, x_3)$.



2. *Select interpolation or shape functions*

The next step is to choose the type of interpolation function that represents the variation of the field variable over an element. The number of nodes form an element; the nature and number of unknowns at each node decide the variation of a field variable within the element.

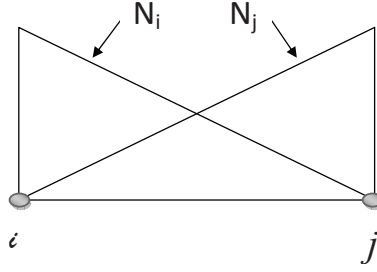
By dividing the solution region into a number of small regions, called elements, and approximating the solution over these regions by a suitable known function, a relation between the differential equations and the elements is established. The functions employed to represent the nature of the solution within each element are called shape functions, or interpolating functions, or basis functions. Polynomial type functions have been most widely used as they can be integrated, or differentiated, easily and the accuracy of the results can be improved by increasing the order of the polynomial. On each element we seek the unknown function under the form $u = \sum_{j=1}^2 N_j u_j$ where N_j are the shape functions and u_j are those corresponding to that elements nodal values.

Main properties of the Shape Functions

(a) *Shape functions are linearly-independent.*

(b)

$$N_i(x_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (1.36)$$



where δ_{ij} is the delta function. The delta function property implies that the shape function N_i should be unit at its home node i , and vanishes at the remote nodes $j \neq i$ of the element.

(c) Shape functions are partitions of unity:

$$\sum_i^n N_i(x) = 1 \quad (1.37)$$

if the constant is included in the basis.

Considering all of above, for an element with n nodes, the shape function could be $(n-1)$ order Lagrange interpolation polynomial

$$N_k(x) = \prod_{i=1}^n \frac{x - x_i}{x_k - x_i}. \quad (1.38)$$

For a one-dimensional linear element it can be rewritten, as $(n = 2)$

$$N_1 = \frac{x - x_2}{x_1 - x_2}, \quad N_2 = \frac{x - x_1}{x_2 - x_1} \quad (1.39)$$

3. Form element equations

Next, we have to determine the matrix equations that express the properties of the individual elements by forming matrix form, these equations are, for each element

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u_1^{k+1} \\ u_2^{k+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

Assembling these elements, the local numbering 12 becomes a global numbering 123 and the above systems become

$$\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1^{k+1} \\ u_2^{k+1} \\ u_3^{k+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ 0 \end{pmatrix} \quad (1.40)$$

for the first element and

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & b_{11} & b_{12} \\ 0 & b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} u_1^{k+1} \\ u_2^{k+1} \\ u_3^{k+1} \end{pmatrix} = \begin{pmatrix} 0 \\ g_1 \\ g_2 \end{pmatrix}. \quad (1.41)$$

for the second.

4. Assemble the element equations to obtain a system of simultaneous equations

Combining (1.40) and (1.41) local systems into a global system , one get

$$\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} + b_{11} & b_{12} \\ 0 & b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} u_1^{k+1} \\ u_2^{k+1} \\ u_3^{k+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 + g_1 \\ g_2 \end{pmatrix} \quad (1.42)$$

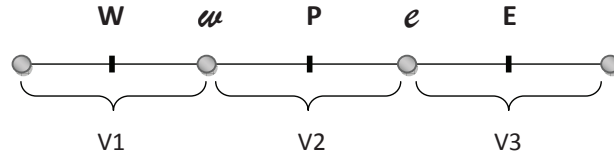
Here one introduce the boundary conditions and then, by solving the system, one get the nodal values of the solution at the instant t_{k+1} from the values at the instant t_k (which appear on the right-hand side).

1.2.5 Finite volume method

At the first step we discretize in time the equation,

$$\frac{u^{k+1} - u^k}{\Delta t} = \frac{\partial^2 u}{\partial x^2}.$$

Then, at the time instant t_k we divide the spatial domain $(0, l)$ into finite volumes (in 1D case they are intervals too) but having the reference point P at the center. Considering three such neighboring finite volumes, with centers at the points W and E (to West respectively to East of P), these volumes have their interior boundaries placed at the points w between W and P , respectively e between P and E .



The discretization of the spatial derivative is now performed by the formula

$$\frac{\partial^2 u}{\partial x^2} \Big|_P = \frac{\frac{\partial u}{\partial x} \Big|_e - \frac{\partial u}{\partial x} \Big|_w}{x_e - x_w} \quad (1.43)$$

and then

$$\frac{\partial u}{\partial x} \Big|_e = \frac{u_E - u_P}{x_E - x_P}, \quad \frac{\partial u}{\partial x} \Big|_w = \frac{u_P - u_W}{x_P - x_W}. \quad (1.44)$$

Replacing into the above equation for every reference point E, P, W we obtain another system from which we can calculate u_E, u_P, u_W at the next time instant t_k . This step is performed as for the finite differences method, using the initial and boundary conditions. What is different in these two methods is the discretization procedure.

1.2.6 Eulerian and Lagrangian flow description

There are two approaches in analyzing fluid problems. The first method, called Eulerian method, in this method the fluid motion is given by completely prescribing the necessary properties (pressure, density, velocity, etc.) as functions of space and time. From this method one can obtain information about the flow in terms of what happens at fixed points in space as fluid flows past those points.

The second method, called the Lagrangian method, involves following individual fluid particles as they move about and determining how the fluid properties associated with these particles change as a function of time.

1.2.7 Coupled Eulerian-Lagrangian (CEL) methods

It should be pointed out that all methods introduced in the previous chapters are based on the Eulerian coordinates in which computational nodes are fixed in space and all variables

are calculated at these fixed nodes. In some instances in reality, however, it is of interest to compute variables in the Lagrangian coordinates where the mesh points are allowed to move along with the fluid particles. Furthermore, it is often convenient to have both Eulerian and Lagrangian coordinates coupled, known as the coupled Eulerian - Lagrangian methods, useful in highly distorted flows or multiphase flows. The basic idea is that the boundary Γ of the region Ω given by

$$\Omega = \bigcup_{i=1}^n \Omega_i \quad (1.45)$$

and the curves D_i which separate the subregions Ω_i are to be approximated by time dependent Lagrangian lines $L_i(t)$. A subregion R_i which is approximated by the time independent Eulerian mesh E will consequently have its boundary Γ_i , prescribed by the Lagrangian calculations. Thus, the Eulerian calculation reduces to a calculation on a fixed mesh having a prescribed moving boundary and therefore contributes one of the central calculations in the CEL methods. The calculations that are made at each time step are divided into three main parts: Lagrange calculations, Eulerian calculations, and a calculation that couples the Eulerian and Lagrangian regions by defining that part of the Eulerian mesh which is active and by determining the pressures from the Eulerian region which act on the Lagrangian boundaries.

1.2.8 Particle-In-Cell (PIC) method

In this method, Eulerian mesh is used and the cell is filled with particles of the same kind or a mixture of different kinds. The calculation of changes in the fluid configuration proceeds through a series of time steps or cycles. Each cell is characterized by a set of variables describing the mean components of velocity, the internal energy, the density, and the pressure in the cell. In the Eulerian part of the calculations, only the cellwise quantities are changed and the fluid is assumed to be momentarily completely at rest. In order to accomplish the particle motion, it is convenient to prepare as a first step for the possibility of particles moving across cell boundaries. For this purpose, the specific quantities in each of the cells are transformed to cellwise totals.

1.2.9 Monte Carlo methods (MCM) methods

Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to compute their results. Monte Carlo methods are often used in simulating physical and mathematical systems. Because of their reliance on repeated computation of random or pseudo-random numbers. MCM have been successfully used in many problems in physics and engineering where stochastic or statistical approaches can describe the physical phenomena more realistically. They have been extensively applied to electron distributions, neutron diffusion, radiative heat transfer, probability density functions for turbulent microscale eddies, etc.

1.2.10 Spectral methods

Spectral methods are a class of techniques used in applied mathematics and scientific computing to numerically solve certain PDEs, often involving the use of the Fast Fourier Transform. Where applicable, spectral methods have excellent error properties, with the so called "exponential convergence" being the fastest possible. The spectral method and the finite element method are closely related and built on the same ideas; the main difference between them is that the spectral method approximates the solution as linear combination of continuous functions that are generally nonzero over the domain of solution (usually sinusoids or Chebyshev polynomials), while the finite element method approximates the solution as a linear combination of piecewise functions that are nonzero on small subdomains. Because of this, the spectral method takes on a global approach while the finite element method is a local approach. This is part of why the spectral method works best when the solution is smooth.

1.3 Discussion

A wide variety of models exists in field of computational fluid and gas dynamics. For numerical solution of the corresponding problem, the above methods, their modifications, as well as other methods are used in practice. To name few, such models include laminar and tur-

bulent gas and fluid flows, flows of plasma and flows with chemical, thermal and mechanical interactions. A few examples are discussed below.

In [21], iterative finite volume method is developed for calculation of compressible, viscous, heat conductive gas flows at all speeds. The method is derived from a general form of system of equations describing the motion of compressible, viscous is given in a general form of the Navier-Stokes-Fourier equations. In work concluded that model captures the basic flow effects of the motion of compressible viscous heat-conducting gas in continuum.

Work [5] studied the effect of surface and gas radiation on the turbulent natural convection in a tall differentially heated cavity. The turbulent flow is described by Navier-Stokes equations with added the subgrid-scale viscosity. An explicit finite volume fractional step based algorithm is used.

In [1], direct numerical simulations of an anisothermal reacting turbulent channel flow with and without radiative source terms have been performed to study the influence of the radiative heat transfer on the optically non-homogeneous boundary layer structure. In the studied case, representing a low optical thickness and low temperature gradient configuration. It is shown that it is not necessary to resolve the coupled Navier-Stokes/radiation equations in the turbulent boundary layer to capture the correct physics of the energy transfer to the wall. The contribution of the radiative heat fluxes at the boundaries can be independently calculated using a radiation solver and added to the flux prediction of the law-of-the-wall. In work the adding radiative source influenced the heat flux prediction of 7%.

Paper [20] presented a comparative study of the physical differences between numerical simulations obtained with both the conservation and incompressible forms of the Navier-Stokes equations for natural convection flows in simple geometries with the goal of investigating the possible consequences of assuming incompressible flow in Next Generation Nuclear Plant simulations, specifically for helium-cooled reactor concepts. Results show that there was only a 1.0% variation in local density from the incompressible reference density, for this low-temperature difference and the comparison results diverged significantly for the high-temperature difference. In general, the incompressible flow assumption is valid in the case of isothermal, or nearly isothermal, single phase flows under relatively small pressure gradients.

Work [2] represents model of gas flow in pipeline networks ranging from partial differential

equation models (the isothermal Euler equations) to algebraic models from an analytical and a numerical point of view. Work shows that models of different types can be coupled and this coupling yields well-posed mathematical problems.

Turbulent natural convection in a rectangular enclosure having finite thickness heat-conducting walls at local heating at the bottom of the cavity provided that convective-radiative heat exchange with an environment on one of the external borders has been numerically studied in [11]. Mathematical simulation has been carried out in terms of the dimensionless Reynolds averaged Navier-Stokes equations in stream function-vorticity formulations. Equations have been solved numerically using finite difference method with the implicit two-layer difference scheme. The enclosure is filled with uniform rectangular grid. The convective terms are discretised applying the scheme of the second order allowing considering a sign of velocity and the diffusive terms with the central difference scheme.

CHAPTER 2

THE METHOD OF LARGE PARTICLES

2.1 Overview

Firstly for modeling compressible ideal gas, in inviscid fluid case one should describe it with Euler's equations plus continuity equation.

$$\begin{cases} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{V}) = 0, \\ \rho \left(\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} \right) = \rho F - \operatorname{grad} p. \end{cases} \quad (2.1)$$

Main idea of the method is to split initial Euler or Navier-Stokes system, represent as conservation laws system, on every time step onto separate calculation stages, which are divided by physical processes. Fluid is considered continuous complex of N large particles, which match with Eulerian mesh in initial moment of time. Particles determine parameters of fluid such as mass, energy and velocity, and Eulerian mesh are used to detect parameters of field: pressure, density and temperature.

For each time steps there is three calculation stages:

1. Eulerian stage

At this stage particles are considered static and only changes in inner state of cell are studied, i.e. there is no mass flow through boundaries of cell at this moment. So to determine transient values of flow parameters \tilde{u}, \tilde{v} and \tilde{E} one can only consider effects of pressure in cell, so one should take look at differences of momentum and energy of large particle for Δt time.

2. Lagrangian stage

After first stage, next step is to take into consideration displacements of cells proportionally their velocity and Δt time, without changes in inner state. So at this stage flow of mass through the boundaries of cell should be computed to consider interaction between surrounding cells.

3. Final stage

Computing of redistribution mass, energy and momentum in space accordingly with conservation laws for each cell and whole system. This helps to determine new spreading of velocity, energy and density in domain.

2.2 Eulerian stage

As discussed earlier at this stage density is distribution considered static, so $\frac{\partial \rho}{\partial t} = 0$ and from conservation of mass one can get that $\text{div } \rho \mathbf{V} = 0$. Then in (2.1) ρ can be taken out of time derivation and parts of type $\text{div } \phi \rho \mathbf{V}$ can be neglected, then this modified system can be group by time derivatives. Below Navier-Stokes system described using this assumptions.

$$\left\{ \begin{array}{l} \rho \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = \mu \Delta u, \\ \rho \frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = -\rho g + \mu \Delta v, \\ \rho \frac{\partial E}{\partial t} + \text{div } p \mathbf{V} = \mathbf{E}(\mathbf{g}). \end{array} \right. \quad (2.2)$$

Using finite difference approximation for (2.2) at time t^n , one can get following equations for i, j cell:

$$\begin{aligned}\rho_{ij}^n \left(\frac{\partial u}{\partial t} \right)_{i,j}^n &= - \frac{[p_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n]}{\Delta x} + \mu \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta x)^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2} \right) \\ \rho_{ij}^n \left(\frac{\partial v}{\partial t} \right)_{i,j}^n &= - \frac{[p_{i,j+\frac{1}{2}}^n - p_{i,j-\frac{1}{2}}^n]}{\Delta y} - \rho_{ij}^n g + \mu \left(\frac{v_{i+1,j}^n - 2v_{i,j}^n + v_{i-1,j}^n}{(\Delta x)^2} + \frac{v_{i,j+1}^n - 2v_{i,j}^n + v_{i,j-1}^n}{(\Delta y)^2} \right) \\ \rho_{ij}^n \left(\frac{\partial E}{\partial t} \right)_{i,j}^n &= - \left(\frac{[p_{i+\frac{1}{2},j}^n u_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n u_{i-\frac{1}{2},j}^n]}{\Delta x} + \frac{[p_{i,j+\frac{1}{2}}^n v_{i,j+\frac{1}{2}}^n - p_{i,j-\frac{1}{2}}^n v_{i,j-\frac{1}{2}}^n]}{\Delta y} \right) - g \rho_{ij}^n v_{ij}\end{aligned}\quad (2.3)$$

where derivatives in times are defined as:

$$\left(\frac{\partial f}{\partial t} \right)_{ij}^n = \frac{\tilde{f}_{ij}^n - f_{ij}^n}{\Delta t}, \quad f = (u, v, E).$$

After these steps one can get equations for velocity and energy at Euler's stage:

$$\begin{aligned}\tilde{u}_{ij}^n &= u_{ij}^n - \frac{p_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n}{\Delta x} \frac{\Delta t}{\rho_{ij}^n} + \frac{\mu \Delta t}{\rho_{ij}^n} \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta x)^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2} \right), \\ \tilde{v}_{ij}^n &= v_{ij}^n - \frac{p_{i,j+\frac{1}{2}}^n - p_{i,j-\frac{1}{2}}^n}{\Delta y} \frac{\Delta t}{\rho_{ij}^n} - g \Delta t + \frac{\mu \Delta t}{\rho_{ij}^n} \left(\frac{v_{i+1,j}^n - 2v_{i,j}^n + v_{i-1,j}^n}{(\Delta x)^2} + \frac{v_{i,j+1}^n - 2v_{i,j}^n + v_{i,j-1}^n}{(\Delta y)^2} \right), \\ \tilde{E}_{ij}^n &= E_{ij}^n - \left[\frac{p_{i+\frac{1}{2},j}^n u_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n u_{i-\frac{1}{2},j}^n}{\Delta x} + \frac{p_{i,j+\frac{1}{2}}^n v_{i,j+\frac{1}{2}}^n - p_{i,j-\frac{1}{2}}^n v_{i,j-\frac{1}{2}}^n}{\Delta y} \right] \frac{\Delta t}{\rho_{ij}^n} - g v_{ij} \Delta t,\end{aligned}\quad (2.4)$$

where $\tilde{u}, \tilde{v}, \tilde{E}$ are transitional values of velocity and energy at time $t^n + \Delta t$. And variable with fractional indexes should be computed as:

$$u_{i+\frac{1}{2},j}^n = \frac{u_{ij}^n + u_{i+1,j}^n}{2}.$$

2.3 Lagrangian stage

At this stage flow of masses ΔM^n through boundaries of cell are computed at time Δt with consideration that only normal to boundary component of velocity move all masses. For

example, continuity equation (1.1) can be written in difference form:

$$\rho_{ij}^{n+1} \Delta x \Delta y = \rho_{ij}^n \Delta x \Delta y - \Delta M_{i+\frac{1}{2},j}^n + \Delta M_{i-\frac{1}{2},j}^n - \Delta M_{i,j+\frac{1}{2}}^n + \Delta M_{i,j-\frac{1}{2}}^n,$$

where flow of mass is

$$\Delta M_{i+\frac{1}{2},j}^n = \langle \rho_{i+\frac{1}{2},j}^n \rangle \langle \tilde{u}_{i+\frac{1}{2},j}^n \rangle \Delta y \Delta t. \quad (2.5)$$

Here symbol $\langle \rangle$ means values of ρ and \tilde{u} on boundary of cell. ΔM^n could be computed using formulas of second order. Then $\Delta M_{i+\frac{1}{2},j}^n$ is computed as:

$$\text{If } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n > 0 \text{ and } \tilde{u}_{i+\frac{1}{2},j}^n = \tilde{u}_{ij}^n + \left(\frac{\partial \tilde{u}}{\partial x} \right)_{ij}^n \frac{\Delta x}{2} = \tilde{u}_{ij}^n + \frac{\tilde{u}_{i+1,j}^n - \tilde{u}_{i-1,j}^n}{4} > 0,$$

$$\text{then } \Delta M_{i+\frac{1}{2},j}^n = \left(\tilde{u}_{ij}^n + \frac{\tilde{u}_{i+1,j}^n - \tilde{u}_{i-1,j}^n}{4} \right) \left(\rho_{ij}^n + \frac{\rho_{i+1,j}^n - \rho_{i-1,j}^n}{4} \right) \Delta y \Delta t.$$

$$\text{If } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n < 0 \text{ and } \tilde{u}_{i+\frac{1}{2},j}^n = \tilde{u}_{i+1,j}^n - \left(\frac{\partial \tilde{u}}{\partial x} \right)_{i+1,j}^n \frac{\Delta x}{2} = \tilde{u}_{i+1,j}^n - \frac{\tilde{u}_{i+2,j}^n - \tilde{u}_{i,j}^n}{4} < 0,$$

$$\text{then } \Delta M_{i+\frac{1}{2},j}^n = \left(\tilde{u}_{i+1,j}^n - \frac{\tilde{u}_{i+2,j}^n - \tilde{u}_{i,j}^n}{4} \right) \left(\rho_{i+1,j}^n - \frac{\rho_{i+2,j}^n - \rho_{i,j}^n}{4} \right) \Delta y \Delta t. \quad (2.6)$$

$$\text{In other case, if } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n > 0 \text{ and } \tilde{u}_{ij}^n + \frac{\tilde{u}_{i+1,j}^n - \tilde{u}_{i-1,j}^n}{4} < 0,$$

$$\text{or } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n < 0 \text{ and } \tilde{u}_{i+1,j}^n - \frac{\tilde{u}_{i+2,j}^n - \tilde{u}_{i,j}^n}{4} > 0,$$

$$\text{then } \Delta M_{i+\frac{1}{2},j}^n = 0.$$

Such order of computation allows to have stable result if only first stage was stable, so for this purpose in (2.4) thermodynamic pressure p should be replaced by $p + q$, where q is artificial viscosity.

Also flow of mass ΔM^n could be computed using formulas of first order. In this case it

can be done without adding artificial viscosity.

$$\Delta M_{i+\frac{1}{2},j}^n = \begin{cases} \rho_{ij}^n \frac{\tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n}{2} \Delta y \Delta t, & \text{if } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n > 0, \\ \rho_{i+1,j}^n \frac{\tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n}{2} \Delta y \Delta t, & \text{if } \tilde{u}_{ij}^n + \tilde{u}_{i+1,j}^n < 0. \end{cases} \quad (2.7)$$

2.4 Final stage

When Euler and Lagrangian stages completed, one can finish iteration and compute new distribution parameters at time $t^{n+1} = t^n + \Delta t$. Equation of this stage are conservation laws of mass M , momentum P and full energy E , which are written for cell in difference form:

$$M^{n+1} = M^n + \sum \Delta M_{boundary}^n,$$

$$P^{n+1} = P^n + \sum \Delta P_{boundary}^n, \quad (2.8)$$

$$E^{n+1} = E^n + \sum \Delta E_{boundary}^n.$$

Here $\Delta M_{boundary}^n$ is mass of gas, which move through one of boundaries for time Δt . So final values of ρ , $X = (u, v, E)$ at new time $t^{n+1} = t^n + \Delta t$ are computed as:

$$\rho_{ij}^{n+1} = \rho_{ij}^n + \frac{\Delta M_{i-\frac{1}{2},j}^n + \Delta M_{i,j-\frac{1}{2}}^n - \Delta M_{i,j+\frac{1}{2}}^n - \Delta M_{i+\frac{1}{2},j}^n}{\Delta x \Delta y},$$

$$\begin{aligned} X_{ij}^{n+1} = & \left\{ D_{ij}^n(1) \tilde{X}_{i-1,j}^n \Delta M_{i-\frac{1}{2},j}^n + D_{ij}^n(2) \tilde{X}_{i,j-1}^n \Delta M_{i,j-\frac{1}{2}}^n + D_{ij}^n(3) \tilde{X}_{i+1,j}^n \Delta M_{i+\frac{1}{2},j}^n + \right. \\ & + D_{ij}^n(4) \tilde{X}_{i,j+1}^n \Delta M_{i,j+\frac{1}{2}}^n + \tilde{X}_{ij}^n \left(\rho_{ij}^n \Delta x \Delta y - [1 - D_{ij}^n(1)] \Delta M_{i-\frac{1}{2},j}^n - \right. \\ & \left. \left. - [1 - D_{ij}^n(2)] \Delta M_{i,j-\frac{1}{2}}^n - [1 - D_{ij}^n(3)] \Delta M_{i+\frac{1}{2},j}^n - [1 - D_{ij}^n(4)] \Delta M_{i,j+\frac{1}{2}}^n \right) \right\} / \{ \rho_{ij}^{n+1} \Delta x \Delta y \}, \end{aligned} \quad (2.9)$$

where $D_{ij}^n(k)$ is using to determine direction of flow at a cell boundary, k is index for each of four boundaries: 1 is left, 2 is bottom, 3 is right and 4 is top boundary of cell.

$$D_{ij}^n(k) = \begin{cases} 1, & \text{if fluid flow in } (ij) \text{ cell through } k\text{-boundary;} \\ 0, & \text{otherwise.} \end{cases} \quad (2.10)$$

2.5 Stability condition and error estimation

Thorough analysis can be found in [3]. The method of large particles is an explicit first order differences schemes which lead to stability condition in the form $C \frac{\Delta t}{\Delta s} < 1$, where C is speed of sound, s is coordinate x or y . The method is first order. The stability properties can be improved by introducing artificial viscosity as outlined below.

2.6 Artificial and numerical viscosity

Numerical viscosity appears in methods that use discretization of domain and approximation of values at nodes or cell volumes. This type of viscosity follows from approximation structure of difference scheme and mesh itself. For example, the value of numerical viscosity for Lax-Friedrichs method (discussed above) is $A \frac{\Delta x}{\Delta t}$, where A is constant from equation (1.24). For method of large particle numerical viscosity is $\varepsilon_x = \frac{1}{2}|u| \Delta x$, $\varepsilon_y = \frac{1}{2}|v| \Delta y$.

On the other hand, artificial viscosity can be manually added to method to increase numerical stability. Method of large particles in areas with small velocities tends to accumulate instability. So in order to correct this effect one needs to change pressure from p to $p + q$ in Euler equations, adding necessary viscous pressure q to the method. For example, in [3], the following form of artificial viscosity q is used:

$$q = \nu \frac{\Delta u}{\Delta x} = BC\rho \Delta u, \quad (2.11)$$

where $C = \sqrt{\gamma p/\rho}$ is local speed of sound, B is some empirical constant and $\nu = BC\rho \Delta x$. Then at the Euler stage the finite difference approximation yields

$$\tilde{u}_{ij}^n = u_{ij}^n - \frac{p_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n}{\Delta x} \frac{\Delta t}{\rho_{ij}^n} + \frac{\nu \Delta t}{\rho_{ij}^n} \frac{(u_{i+1} - u_{i-1})}{\Delta x^2}. \quad (2.12)$$

On the other hand, one may choose to discretize Navier-Stokes equations with physical viscosity μ instead of Euler equations with artificial viscous pressure q , i.e. follow the physical model for viscous fluid. Then the formulas for Euler stage take the form

$$\tilde{u}_{ij}^n = u_{ij}^n - \frac{p_{i+\frac{1}{2},j}^n - p_{i-\frac{1}{2},j}^n}{\Delta x} \frac{\Delta t}{\rho_{ij}^n} + \frac{\mu \Delta t}{\rho_{ij}^n} (u_{xx} + u_{yy}) \quad (2.13)$$

with a similar form for the y -component \tilde{v}_{ij}^n of the velocity. Computations for most models considered in the current Thesis show that results are consistent for added viscosity $\mu \geq 0.9$; results for smaller viscosities involve development of instability (see, e.g., Fig. B.23).

2.7 Discussion

Method of large particles has been used to perform computations for different models. Method of large particles allows to make computations using one algorithm for different complex models without preliminary determination of singularities, such models as streamlines of different shapes, transition through sound speed, inner shock waves and others. Some examples are given below.

Work [4] obtains results for three dimensional viscous flows with shock waves in channels and engine nozzles using modified method of large particles. In this model existence of mixed subsonic and transonic flows makes it impossible to use many classical methods, however as shown in [4], method of large particles is the most appropriate for this type of problems.

For example, in [12], the flow of gas and dust in atmosphere after volcanic eruption was described by system of equations of gas-suspension mechanics. Parameters of volcanic eruption were computed, using method of large particles in cylindrically symmetric formulation.

In [10] studied model of probe was placed in a flow of slowly moving collisional plasma in the wall region in the neighborhood of a cylindrical. The problem was solved by the time relaxation method using the algorithm of the method of large particles. The mathematical model of the problem included equations of continuity for ions, electrons, and neutral particles, equations of motion for all components of plasma, energy equation for neutral component, and Poisson equation for self-consistent electric field. The computations were compared with experiment and showed adequate agreement.

CHAPTER 3

THE MATHEMATICAL MODEL AND NUMERICAL COMPUTATIONS

Firstly two test models are discussed, in order to examine properties of method of bulk particles applied to different physical situations. After that main model is formulated, and computations are performed.

3.1 Test model 1

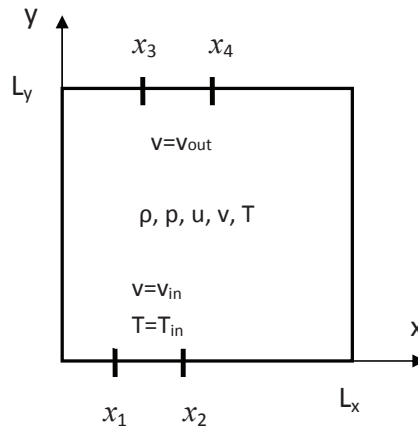


Figure 3.1: Test model 1

First test model considers a rectangular area with two “no-leak” boundaries. At the bottom, there is a part of the boundary ($x_1 \leq x \leq x_2$) with a gas stream of higher temperature flowing in vertically with velocity v_{in} ; in the top wall, there is a part of the boundary ($x_3 \leq x \leq x_4$) has corresponding to a “fan” blowing the gas out vertically with velocity v_{out} .

Thus the model represent enclosure with some current with higher temperature passing through. Parameters $x_1, x_2, x_3, x_4, v_{in}, v_{out}$ can be change manually in program form 3.2.

The screenshot shows a software window titled "Current Temperature&Velocity" with various input fields and a results display area.

Parameter	Value	Parameter	Value
N	50	g	0
M	50	visc	1
K	1000		
tau	0.01		
Lx	2	hx	0.04
Ly	2	hy	0.04

Buttons: Run, Print

Boundary Conditions:

Variable	Value	Unit
x1	0.2	m
x2	0.4	m
x3	0.4	m
x4	0.6	m
V in	1	m/s
V out	1	m/s
T in	290	K
T BC	300	K

Results done
 Stability condition=True tau/hx= 0.000790569330084199
 T=0.031622773203368

--Mass--
 M_IC = 3.99999999999982
 M_Comp = 4.04610364988323
 dM = 0.046103649883408

--Energy--
 E_IC = 5202.06153739033
 E_Comp = 5288.53689459238
 dE = 86.4753572020518

Figure 3.2: Test model 1, Input form

Boundary conditions are:

$$\left\{ \begin{array}{l} u(0, y, t) = u(L_x, y, t) = 0, 0 \leq y \leq L_y, \\ v(x, L_y, t) = \begin{cases} 0, 0 \leq x \leq x_3, x_4 \leq x \leq L_x \\ v_{out}, x_3 \leq x \leq x_4, \end{cases} \\ v(x, 0, t) = \begin{cases} 0, 0 \leq x \leq x_1, x_2 \leq x \leq L_x \\ v_{in}, x_1 \leq x \leq x_2, \end{cases} \\ T(x, 0, t) = T_{in}, \\ \rho(x, 0, t) = \rho_0, \\ p(x, 0, t) = p_0, \end{array} \right. \quad (3.1)$$

Initial conditions are:

$$\left\{ \begin{array}{l} \mathbf{V} = 0, \\ T = T_0, \\ \rho = \rho_0, \\ p = \frac{R}{M} \rho T, \end{array} \right. \quad (3.2)$$

where p is pressure, ρ is density, R is the gas constant, M is the molar mass and T is temperature, T_{in} is temperature of flow in and $T_0 < T_{in}$ is initial inside temperature. Gravity coefficients equal 0. Results contain velocity field u, v ; density ρ ; pressure p and temperature distribution T . Results for two different times 2.52 s and 12 s for model with initial temperature inside $T_0 = 290$ K and temperature of influent flow $T_{in} = 300$ K can be seen in Figures B.1, B.2 and B.3; they show physically meaningful development of a gas stream between inflow and outflow locations.

3.1.1 Effects of step sizes

For analyzing the computation results, let us look at the effects of changing different parameters in Test model 1. As a benchmark, we take results corresponding to the following parameters: $L_x = L_y = 2$ m, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $v_{in} = 1$ m/s, $x_3 = 0.4$ m, $x_4 = 0.6$ m, $v_{out} = 1$ m/s, dimensionless time step $\tau = 0.008$, control time $t = 0.316$ s, artificial viscosity

coefficient $\mu = 1 \text{ m}^2/\text{s}$. For such parameters, the flow current passes through the left side of the domain, and a swirl is formed in the right side (Figure B.5).

The first series of tests was done to compare benchmark results with results for the same physical model, but with different numerical space steps (bigger and smaller than for chosen model). The time step is considered fixed at $\tau = \tau_0$. Results in Figures B.6 shows result for bigger cell approximation in particular $h = 0.2 \text{ m}$ and $h = 0.067 \text{ m}$. For $N = 30$ results very close to benchmark, but further increase in space steps ($N \leq 10$) leads to the loss of accuracy. Even for large particle sizes, the numerical method is stable and gives acceptable results which can be used for fast first computation to estimate the behavior of the flow. Comparatively small number of cell for domain allows to reduce computational time. Results in Figures B.7 shows result for smaller cell approximation in particular $h = 0.02\text{m}$ and $h = 0.028 \text{ m}$ shows no significant change in results. From this series of tests, it can be concluded that one may start by choosing a large step size h , without loss of stability, and and continue until desired accuracy is obtained.

The next comparisons are made with fixed space step $h = h_0$ and for different time steps (bigger and smaller than the chosen benchmark). Results in Figure B.8 are close to the benchmark results. Results of comparison of velocity components (absolute differences) in the case when $\tau = 0.01$ with the benchmark case $\tau_0 = 0.008$ are given in Figure B.9 and demonstrate rather small differences ($\sim 10^{-3}$). This test shows that for a faster computation, one might choose the largest possible time step from the stability condition.

3.1.2 Effects of artificial viscosity

Model 1 is also helpful to demonstrate the work of the method of large particles for higher velocities, where there is no need in artificial viscosity. Results in Figure B.10 correspond to velocities $\sim 10 \text{ m/s}$ and no artificial viscosity (Reynolds number $Re \gg 1$); Figure B.11 contains comparison results for the same problem with viscosity $\mu = 1 \text{ m}^2/\text{s}$ ($Re = 200$).

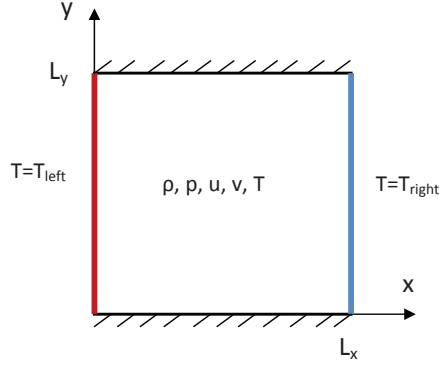


Figure 3.3: Test model 2

3.2 Test model 2

The second test model considers a rectangular domain, with initial conditions corresponding to hydrostatic equilibrium of gas at a fixed temperature, with gravity taken into account:

$$\left\{ \begin{array}{l} \mathbf{V} = 0, \\ T = T_{in} = T_0, \\ \rho = \rho_0 e^{-\frac{Mg}{RT}y}, \\ p = \frac{R}{M}\rho T_0. \end{array} \right. \quad (3.3)$$

In this model, artificial viscosity is introduced for stable computations, and Navier-Stokes equations (1.3) are solved instead of Euler's equations (1.4). Boundary conditions are set as follows: “no-leak” boundaries on top and bottom of rectangle and with left boundary temperature $T_{left} > T_{in}$ and right boundary temperature $T_{right} < T_{in}$.

$$\left\{ \begin{array}{l} u(0, y, t) = u(L_x, y, t) = 0, 0 \leq y \leq L_y, \\ v(x, 0, t) = v(x, L_y, t) = 0, \\ T(0, y, t) = T_{left}, \\ T(L_x, y, t) = T_{right}. \end{array} \right. \quad (3.4)$$

As a result, thermal convection is expected to be observed in the gas domain.

Results for progressing times are presented in Figure B.12 (10 s, flow goes straight from left to right due to compressibility assumptions), Figure B.13 (18.9 s), and finally , Figure B.14 (31 s, stabilized convection roll). The results show development of convection, with gas warming and rising near the hot boundary, and cooling and going down near the cold boundary.

3.3 Main model

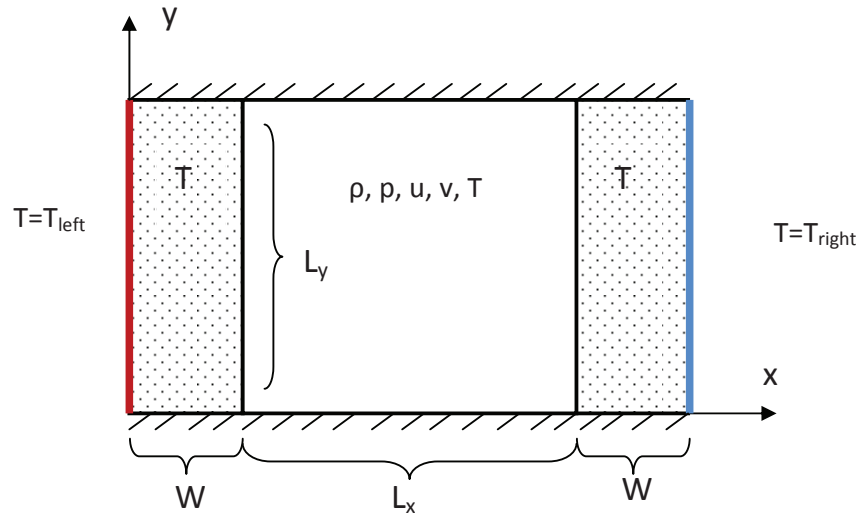


Figure 3.4: Main model

The main model is based on the Test model 2, with modifications at boundaries. For this model, two rectangular domains representing glass walls are added on the left and on the right. At the left boundary of the left glass, the temperature of the environment assumed constant and equal to T_{left} ; similarly at the right boundary of the right glass, the temperature of the environment is $T_{right} < T_{left}$.

In this model, the linear heat equation is used to describe temperature distributions $s(x, y, t)$ inside glass domains:

$$s_t = k(s_{xx} + s_{yy}), \quad (3.5)$$

where k is thermal diffusivity of material. Using explicit finite differences to approximate

the heat equation, one obtains

$$s_{ij}^{n+1} = s_{ij}^n + k \Delta t \left\{ \frac{s_{i-1,j}^n - 2s_{ij}^n + s_{i+1,j}^n}{(\Delta x)^2} + \frac{s_{i,j-1}^n - 2s_{ij}^n + s_{i,j+1}^n}{(\Delta y)^2} \right\} \quad (3.6)$$

In places of contact between glass and gas, one needs to set proper boundary conditions. There are various kinds of heat transfer. Conduction is heat transfer which occur inside medium due to molecular activity of material. Convection is heat transfer come due to a superposition of energy transport by the random motion of the molecules and by the bulk motion of the fluid. In this model particularly, convection heat transfer occurs between glass surface and a fluid in motion when the two are at different temperatures. Thermal radiation is energy emitted by matter that is at finite temperature [8].

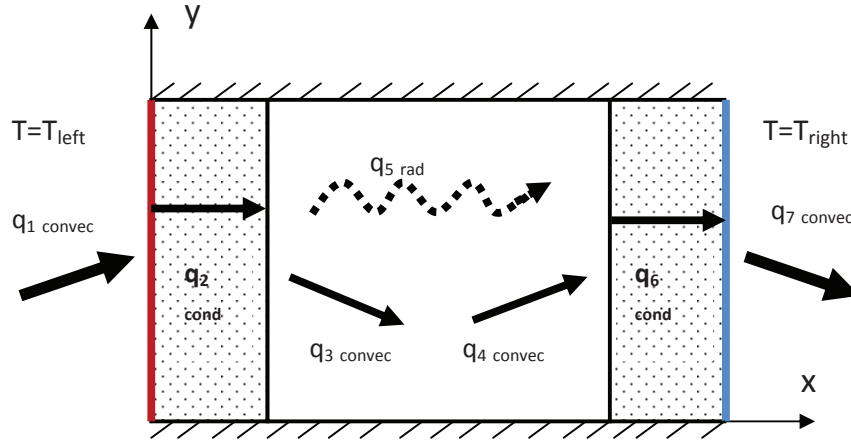


Figure 3.5: Types of heat transfer

Figure 3.5 shows different types of heat transfer discussed above: q_1 is the heat flux corresponding to convection between gas and the left glass, q_2 corresponds to heat conduction through the left glass, q_3 and q_4 correspond to convection between glasses, q_5 is net radiation exchange heat flux between the inner surface of the left glasses, q_6 is the conduction heat flux through the right glass, q_7 is the heat flux corresponding to convection between gas and the right glass. The main model takes into account all types of heat transfer except for the radiation flux q_5 .

Initial conditions for the gas domain is the same as in previous test model: (3.3). For glass areas initial temperature distributions are $s_{ij}^0 = T_0$. Top and bottom boundaries of glass

domains are assumed perfectly insulated:

$$\frac{\partial s}{\partial y}(x, 0, t) = 0, \quad \frac{\partial s}{\partial y}(x, L_y, t) = 0. \quad (3.7)$$

Inner and outer surface boundary conditions are computed using the mixed boundary condition (Newton's law of cooling)

$$-K_0 \frac{\partial s}{\partial x}(0, y, t) = h(T_{air} - s(0, y, t)), \quad (3.8)$$

where $s(0, y, t)$ is $T_{surface}$ temperature of surface of a glass, K_0 ($W/m \cdot K$) is thermal conductivity of material, and h ($W/m^2 \cdot K$) is convection heat transfer coefficient. Other parameters use the same boundary conditions as for Test model 2.

Results for time $t = 0.3$ s are given in Figures B.15 (temperature distribution in glass and gas domains) and B.16 (velocity field inside the gas domain). We note that time needed to establish convection is longer than in Test model 2, due to time necessary for glasses to be warmed/cooled, in order to create differences of temperature at air domain boundaries.

Final results at time 171 s where convection is established are shown in Figure B.18; the corresponding temperature distribution is given in Figure B.17. Computations were conducted in the presence of artificial viscosity $\mu = 1$ m²/s. It was shown numerically that the viscosity cannot be decreased lower than 0.92 m²/s without loss of numerical stability (see Figure B.23).

Comparison of results for different computation parameters are presented in Figure B.19 (a different thermal conductivity), Figure B.20 (a different viscosity coefficient), Figure B.21 (different temperatures to the left of the left glass) and Figure B.22 (different mesh size). All comparison results are well within physical expectations.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

4.1 Conclusion

The current thesis is devoted to numerical solution of gas dynamics problem, in particular, a system of two dimensional Navier-Stokes equations. Fundamentals of fluid and gas dynamics, conservation principles and various equations of state for fluid and gas dynamics were discussed in Chapter 1. In particular, the conservation of mass follows from assumption that mass leaving the given volume must be balanced by the same quantity of mass entering the volume. The same principle holds for conservation of momentum and energy. These equations together form a system of equations describing motion known as Navier-Stokes equations for viscous fluids, or Euler's equations for inviscid fluids/gases. When modeling gas or fluid dynamics, equation of state is used to complete the system of equations. and relate parameters of the system such as pressure p , density ρ and temperature T . Different assumptions can be applied to problem in order to make computations simpler, such as incompressibility, using the adiabatic process, etc. Depending on the type and complexity of model, the gas/fluid dynamics equations can be coupled with other equations, describing particular process. For example, this is applicable for flows with chemical reactions, plasma flows, flame propagation, etc. A final nonlinear system usually contains several unknown functions which makes the exact solution solution of the system unlikely; in practice, such problems are often solved numerically.

In the last decades, many numerical methods have been developed for particular problems in fluid dynamics. Benefits of methods and views was combined to obtain universal methods. Even nowadays this quest is not finished. Some of fundamental methods in this area discussed

in second part of Chapter 1. Most of the classical methods aren't used directly, but the ideas were used to develop new methods in the modern computational fluid dynamics.

In the current thesis, we performed computations using the method of large particles. This is a finite volume method developed based on the idea of combining the benefits of Eulerian and Lagrangian frameworks. The process of computation is split in different stages by physical processes. The domain is subdivided using Eulerian mesh which is considered static, the medium in it considered to be a set of "particles" with fixed mass that move through the Eulerian mesh. Thus the mesh is used to determine parameters of the field (pressure, density and temperature), whereas particles are used to determine parameters of fluid itself (mass, energy, velocity). Stability issues and their possible solutions are discussed in Chapter 2.

The obtained results correspond to three two-dimensional models, to which the method of large particles was applied in the current work. For the method of large particles, a computational software package was developed using *C#* programming language. The software provides results in text files, which then visualize the data using *Matlab*. The first considered model describes a rectangular cavity with open segments on top and bottom boundaries, through which a gas flow of slightly higher temperature moves in and out of the gas cavity. The second model describes rectangular enclosure with fixed temperatures at left and right boundaries ($T_{left} > T_{in} > T_{right}$). Initially gas inside is in the state of hydrostatic equilibrium; due to temperature differences, an eventually stabilized convection roll is observed.

The third model is the main model for this work. There, the method of large particles is applied for two dimensional convection problem in rectangular domain with finite thickness walls. This test model was developed as a step towards a model of processes taking place in a multi-pane window. Within the model, glass walls participated in heat exchange through Newton's law of cooling used as a boundary condition. Obtained results show the development of convection and have a direct physical interpretation. Future work in the direction of extension of numerical model may include extension of the model into three dimensions. One also could include radiative heat exchange component in the model. For comparison purposes, the model can also be studied using alternative numerical methods.

REFERENCES

- [1] AMAYA J., CABRIT O., POITOU D., CUENOT B., ELHAFI M. Unsteady coupling of Navier-Stokes and radiative heat transfer solvers applied to an anisothermal multi-component turbulent channel flow. *Journal of Quantitative Spectroscopy and Radiative Transfer* 111 (2010), 295-301
- [2] BANDA M.K., HERTY M., KLAR A. Multiscale modeling for gas flow in pipe networks. *Math. Meth. Appl. Sci.* 31 (2008), 915-936
- [3] BELOTSEKOVSKII O.M., DAVIDOV Y.M. *Method of Bulk Particles in Gas Dynamics*. Nauka, Moscow (in Russian), 1982.
- [4] BORISOV D.M., VASYUTICHEV A.S., LAPTEV I.V., RUDENKO A.M. Computational modeling for three-dimensional viscous sub/supersonic flows with shock waves. *Mathematical Modeling* Vol.19, No. 11 (2007), 112-120
- [5] CAPDEVILA R., PREZ-SEGARRA C.D., LEHMKUHL O., COLOMER G. Numerical simulation of turbulent natural convection and gas radiation in differentially heated cavities using FVM, DOM and LES. *6th International Symposium on Radiative Transfer* (2010).
- [6] CHERNII G.G. *Gas Dynamics*. Nauka, Moscow (in Russian), 1988.
- [7] CHUNG T.J. *Computational Fluid Dynamics*. Cambridge University Press, 2002.
- [8] INCROPERA F.P., DEWITT D.P. *Fundamentals of Heat and Mass Transfer*. John Wiley&Sons Inc., 1990.
- [9] KAVIANY M. *Principles of Heat transfer*. John Wiley&Sons Inc., 2002.
- [10] KOTEL'NIKOV V.A., KOTEL'NIKOV M.V. A cylindrical probe in a flow of slowly moving collisional plasma. *High Temperature* Vol. 46, No. 3 (2008), 306-310.
- [11] KUZNETSOV G.V., SHEREMET M.A. Numerical simulation of turbulent natural convection in a rectangular enclosure having finite thickness walls. *International Journal of Heat and Mass Transfer* 53 (2010), 163-177.
- [12] KUDRYASHOV N.A., ROMANOV G.S., SHEVYAKOV A.F. Numerical modeling of the quasistationary regime of eruption of a volcano at the initial stage. *Journal of Engineering Physics and Thermophysics*, Vol. 76, No. 4 (2003) 795-803.

- [13] LEVEQUE R.J. *Numerical Methods for Conservation Laws*. Basel;Boston;Berlin: Birkhauser, 1992.
- [14] LEVEQUE R.J. *Finite-Volume Methods for Hyperbolic problems*. Cambridge, 2004.
- [15] LEWIS R.W., NITHIARASU P., SEETHARAMU K.N. *Fundamentals of the Finite Element Method for Heat and Fluid Flow*. John Wiley&Son Inc., 2004.
- [16] MCCORMACK P.D., LAWRENCE C. *Physical Fluid Dynamics*. New York and London, Academic press, 1973.
- [17] MUNSON B.R., YOUNG D.F., OKIISHI T.H. *Fundamentals of Fluid Mechanics*., John Wiley&Sons Inc., 1992.
- [18] ORAN E.S., BORIS J.P. *Numerical Simulation of Reactive Flow*. Elsevier, 1987.
- [19] PETRILA T., TRIF D. *Basics of Fluid Mechanics and Introduction to Computational Fluid Dynamics*. Springer, 2005.
- [20] RICHARD C.M., RAY A.B., ESTEVE A., HAMMAN K.D., KNOLL D.A., PARK H., TAITANO W. Comparison of natural convection flows under VHTR type conditions modeled by both the conservation and incompressible forms of the Navier-Stokes equations. *Nuclear Engineering and Design* 240 (2010), 1371-1385.
- [21] SHTEREV K.S., STEFANOV S.K. Pressure based finite volume method for calculation of compressible viscous gas flows. *Journal of Computational Physics* 229 (2010), 461-480.
- [22] SONAR T., ANSGORE R. *Mathematical Models of Fluid Dynamics*. Wiley-wch,verlag GmbH & Co.KGaA,Weinheim, 2009.
- [23] TRITON D.J. *Physical Fluid Dynamics*. Oxford University Press, 1988.
- [24] WARSI Z.U.A. *Fluid Dynamics: Theoretical and Computational Approaches*. Taylor&Francis Group, 2006.
- [25] ZUCKER R.D., BIBLARZ O. *Fundamentals of Gas Dynamics*. John Wiley&Sons Inc., 2002.

APPENDIX A

PROGRAMMING CODE

This chapter contains source code written in C#.

A.1 Code for solving heat equation

This section contains code for solving equation (3.5) using scheme (3.6), where $T_{glassRight}$ is a matrix for values of function $u(x, y, t)$ at nodes of mesh. The same procedure for left glass, where $T_{glassLeft}$ is a matrix for function $u(x, y, t)$.

```

public static void HeatEqRight()
{
    double externalTempr;

    //insulated BC for top& bottom
    for (int i = 0; i < W; i++)
    {
        TglassRight[i, 0, newl] = TglassRight[i, 1, old];
        TglassRight[i, M + 1, newl] = TglassRight[i, M, old];
    }

    for (int j = 1; j < M + 1; j++)
    {
        ///BC Newton's law of cooling for inner and outer side of glass
        externalTempr = p[N, j, old] * P * mu / (ro[N, j, old] * R * Rgas);
        TglassRight[0, j, newl] = (TglassRight[1, j, old] + coefNewt *
            externalTempr) / coefNewtPlusOne;
        TglassRight[W - 1, j, newl] = (TglassRight[W - 2, j, old] + coefNewt *
            TemprRight) / coefNewtPlusOne;
    }
    //Computing of main explicit scheme
    for (int j = 1; j < M+1; j++)
    {
        for (int i = 1; i < W - 1; i++)
        {
            TglassRight[i, j, newl] = TglassRight[i, j, old]
                + Ktau * (coefX * (TglassRight[i - 1, j, old] - 2 *
                    TglassRight[i, j, old] + TglassRight[i + 1, j, old])
                + coefY * (TglassRight[i, j - 1, old] - 2 * TglassRight[i, j,
                    old] + TglassRight[i, j + 1, old]));
        }
    }
}

```


A.2 Euler stage code

This section represent code for eulerian stage scheme (2.4) of method of bulk particles form chapter 2. In code $p\{p/m\}xHalf$ stands for $p_{i\{+/-\}\frac{1}{2},j}^n$ and $p\{p/m\}yHalf$ for $p_{i,j\{+/-\}\frac{1}{2}}^n$; uXX, uYY are for u_{xx}, u_{yy} ; $u\{p/m\}Half$ is for $u_{i\{+/-\}\frac{1}{2},j}^n$; v_eu, u_eu and E_eu are for $\tilde{u}_{ij}^n, \tilde{v}_{ij}^n$ and \tilde{E}_{ij}^n .

```

for (int i = 1; i < N + 1; i++)
{
    for (int j = 1; j < M + 1; j++)
    {
        ppxHalf = (p[i, j, old] + p[i + 1, j, old]) * 0.5;
        pmxHalf = (p[i, j, old] + p[i - 1, j, old]) * 0.5;
        ppyHalf = (p[i, j, old] + p[i, j + 1, old]) * 0.5;
        pmyHalf = (p[i, j, old] + p[i, j - 1, old]) * 0.5;

        uXX = (u[i + 1, j, old] - 2.0 * u[i, j, old] + u[i - 1, j, old]) /
            hx2;
        uYY = (u[i, j + 1, old] - 2.0 * u[i, j, old] + u[i, j - 1, old]) /
            hy2;

        vXX = (v[i + 1, j, old] - 2.0 * v[i, j, old] + v[i - 1, j, old]) /
            hx2;
        vYY = (v[i, j + 1, old] - 2.0 * v[i, j, old] + v[i, j - 1, old]) /
            hy2;

        u_eu[i, j] = u[i, j, old] + (-tau_hx * (ppxHalf - pmxHalf) +
            coeff_mu * (uXX + uYY)) / ro[i, j, old]; //velocity for x
        v_eu[i, j] = v[i, j, old] + (-tau_hy * (ppyHalf - pmyHalf) +
            coeff_mu * (vXX + vYY)) / ro[i, j, old] - cf_g; //velocity for
            y

        upHalf = (u[i, j, old] + u[i + 1, j, old]) * 0.5;
        umHalf = (u[i, j, old] + u[i - 1, j, old]) * 0.5;
        vpHalf = (v[i, j, old] + v[i, j + 1, old]) * 0.5;
        vmHalf = (v[i, j, old] + v[i, j - 1, old]) * 0.5;
        E_eu[i, j] = E[i, j, old] - (tau_hx * (ppxHalf * upHalf - pmxHalf
            * umHalf) + tau_hy * (ppyHalf * vpHalf - pmyHalf * vmHalf)) -
            cf_g * ro[i, j, old] * v[i, j, old];
        if (i == 1)
        {
            leftT = TglassLeft[W - 1, j, newl] + inv_coefNewt * (
                TglassLeft[W - 1, j, newl] - TglassLeft[W - 2, j, newl]);
                //Newton's law of cooling
            E_eu[i, j] = (R * ro[i, j, old]) * (0.5 * C * C * (v_eu[i, j]
                * v_eu[i, j] + u_eu[i, j] * u_eu[i, j]) + kappa * leftT /
                gammaMinus1) / Edim;
        }
        if (i == N)
        {
            rightT = TglassRight[0, j, newl] + inv_coefNewt * (TglassRight
                [0, j, newl] - TglassRight[1, j, newl]); //Newton's law of

```

```

        cooling
        E_eu[i, j] = (R * ro[i, j, old]) * (0.5 * C * C * (v_eu[i, j]
        * v_eu[i, j] + u_eu[i, j] * u_eu[i, j]) + kappa * rightT /
        gammaMinus1) / Edim;
    }
}
}
//-----
//BC for Euler matrix
//-----
for (int i = 1; i < N + 1; i++)
{
    //normal component is v=> v = -v
    v_eu[i, 0] = -v_eu[i, 1];
    u_eu[i, 0] = u_eu[i, 1];
    E_eu[i, 0] = E_eu[i, 1];

    //normal component is v=> v = -v
    v_eu[i, M + 1] = -v_eu[i, M];
    u_eu[i, M + 1] = u_eu[i, M];
    E_eu[i, M + 1] = E_eu[i, M];
}
for (int j = 1; j < M + 1; j++)
{
    //normal component is u=> u = -u
    v_eu[0, j] = v_eu[1, j];
    u_eu[0, j] = -u_eu[1, j];
    E_eu[0, j] = E_eu[1, j];

    //normal component is u=> u = -u
    v_eu[N + 1, j] = v_eu[N, j];
    u_eu[N + 1, j] = -u_eu[N, j];
    E_eu[N + 1, j] = E_eu[N, j];
}
}

```

A.3 Lagrange stage code

In this section lagrange stage code uses equations (2.7) for mass flux, where $M\{p/m\}xHalf$ stands for $M_{i\{+/-\}^1_{5,j}}^n$ in formula.

```

upHalf = (u_eu[i, j] + u_eu[i + 1, j]) * 0.5;
//compute M(p/m)xHalf (border 1 and 3)
if (upHalf > eps)//right
{
    MpxHalf = ro[i, j, old] * upHalf * htau_y;
    d3 = 0.0;
}
else if (upHalf < -eps)
{
    MpxHalf = ro[i + 1, j, old] * upHalf * htau_y;
}

```

```

    d3 = 1.0;
}
else
{
    MpxHalf = 0.0;
    d3 = 0.0;
}
umHalf = (u_eu[i, j] + u_eu[i - 1, j]) / 2;
if (umHalf > eps)//left
{
    MmxHalf = ro[i - 1, j, old] * umHalf * htau_y;
    d1 = 1.0;
}
else if (umHalf < -eps)
{
    MmxHalf = ro[i, j, old] * umHalf * htau_y;
    d1 = 0.0;
}
else
{
    MmxHalf = 0.0;
    d1 = 0.0;
}
//compute M(p/m)yHalf (border 2 and 4)
vpHalf = (v_eu[i, j] + v_eu[i, j + 1]) / 2;
if (vpHalf > eps)
{
    MpyHalf = ro[i, j, old] * vpHalf * htau_x;
    d4 = 0.0;
}
else if (vpHalf < -eps)
{
    MpyHalf = ro[i, j + 1, old] * vpHalf * htau_x;
    d4 = 1.0;
}
else
{
    MpyHalf = 0.0;
    d4 = 0.0;
}
vmHalf = (v_eu[i, j] + v_eu[i, j - 1]) / 2;
if (vmHalf > eps)//bottom
{
    MmyHalf = ro[i, j - 1, old] * vmHalf * htau_x;
    d2 = 1.0;
}
else if (vmHalf < -eps)
{
    MmyHalf = ro[i, j, old] * vmHalf * htau_x;
    d2 = 0.0;
}
else
{

```

```

    MmyHalf = 0.0;
    d2 = 0.0;
}

```

A.4 Final stage code

Formulas (2.9) represented in code in this section.

```

d1m = d1 * MmxHalf;
d2m = d2 * MmyHalf;
d3m = d3 * MpxHalf;
d4m = d4 * MpyHalf;
ro[i, j, newl] = ro[i, j, old] + (MmxHalf + MmyHalf - MpyHalf - MpxHalf) /
    (hyx);
ro_hyx = ro[i, j, newl] * hyx;
inv_ro_hyx = 1.0 / ro_hyx;

calctmp = (ro_hyx + (1.0 - d1) * MmxHalf + (1.0 - d2) * MmyHalf - (1.0 -
    d3) * MpxHalf - (1.0 - d4) * MpyHalf);
v[i, j, newl] = (d1m * v_eu[i - 1, j] + d2m * v_eu[i, j - 1] - d3m * v_eu[
    i + 1, j] - d4m * v_eu[i, j + 1] + v_eu[i, j] * calctmp) * (inv_ro_hyx
);
u[i, j, newl] = (d1m * u_eu[i - 1, j] + d2m * u_eu[i, j - 1] - d3m * u_eu[
    i + 1, j] - d4m * u_eu[i, j + 1] + u_eu[i, j] * calctmp) * (inv_ro_hyx
);
E[i, j, newl] = E_eu[i, j]
    + (1 / hyx) * (
        d1m * E_eu[i - 1, j] + MmxHalf * (1.0 - d1) * E_eu[i, j]
        + d2m * E_eu[i, j - 1] + MmyHalf * (1.0 - d2) * E_eu[i, j]
        - d3m * E_eu[i + 1, j] - MpxHalf * (1.0 - d3) * E_eu[i, j]
        - d4m * E_eu[i, j + 1] - MpyHalf * (1.0 - d4) * E_eu[i, j]
    );

```

A.5 Boundary conditions for final stage

```

//BC for new t for unleak wall
for (int i = 1; i < N + 1; i++)
{
    //normal component is v=> v = -v
    v[i, 0, newl] = -v[i, 1, newl];
    u[i, 0, newl] = u[i, 1, newl];
    //other components bring without change
    ro[i, 0, newl] = ro[i, 1, newl];
    p[i, 0, newl] = 2 * p[i, 1, newl] - p[i, 2, newl];
    E[i, 0, newl] = E[i, 1, newl];

    //normal component is v=> v = -v
    v[i, M + 1, newl] = -v[i, M, newl];
    u[i, M + 1, newl] = u[i, M, newl];
}

```

```

//other components bring without change
ro[i, M + 1, newl] = ro[i, M, newl];
p[i, M + 1, newl] = 2 * p[i, M, newl] - p[i, M - 1, newl];
E[i, M + 1, newl] = E[i, M, newl];
}
for (int j = 1; j < M + 1; j++)
{
//normal component is  $u \Rightarrow u = -u$ 
v[0, j, newl] = v[1, j, newl];
u[0, j, newl] = -u[1, j, newl];
//other components bring without change
ro[0, j, newl] = ro[1, j, newl];
p[0, j, newl] = 2 * p[1, j, newl] - p[2, j, newl];
E[0, j, newl] = E[1, j, newl];

//normal component is  $u \Rightarrow u = -u$ 
v[N + 1, j, newl] = v[N, j, newl];
u[N + 1, j, newl] = -u[N, j, newl];
//other components bring without change
ro[N + 1, j, newl] = ro[N, j, newl];
p[N + 1, j, newl] = 2 * p[N, j, newl] - p[N - 1, j, newl];
E[N + 1, j, newl] = E[N, j, newl];
}

```

APPENDIX B

PICTURES OF RESULTS

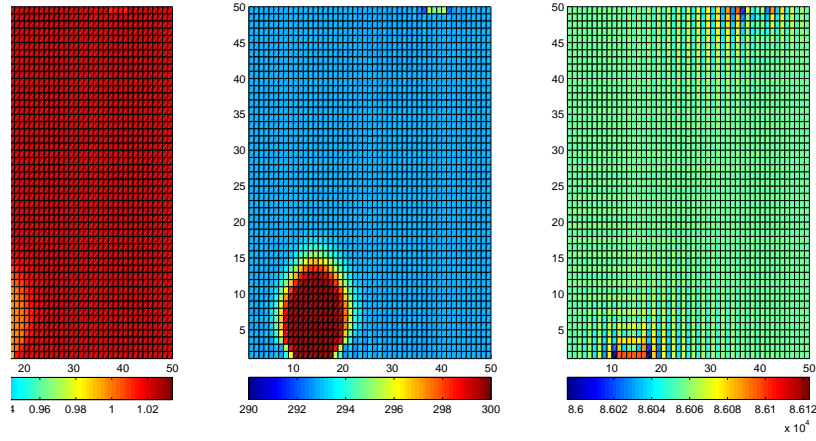


Figure B.1: Test model 1, graphs for density (left) ρ , temperature (middle) T and pressure (right) p at time 2.52 s for $N = 50$, $h_0 = 0.04$ m, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.

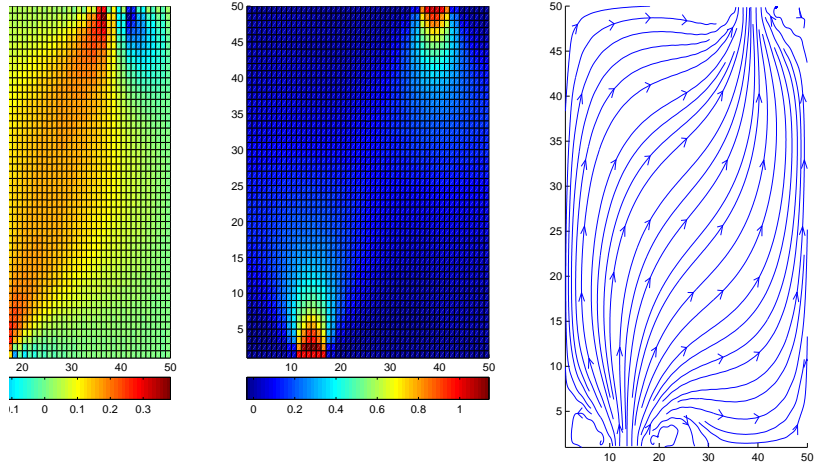


Figure B.2: Test model 1, graphs for velocity field at time 2.52 s

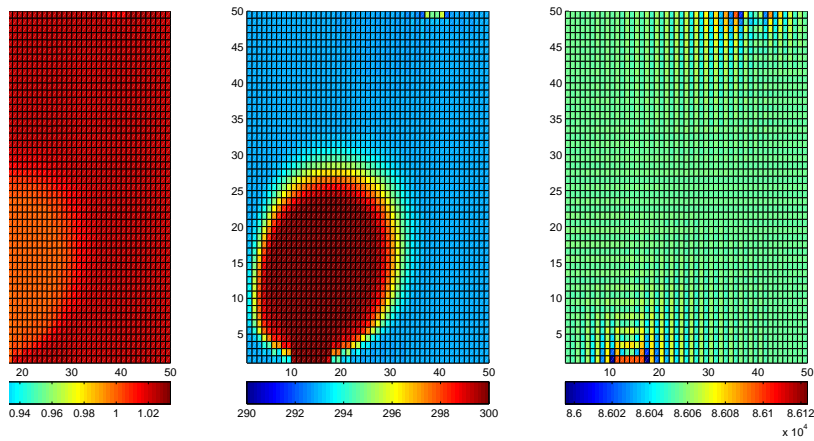


Figure B.3: Test model 1, graphs for density (left) ρ , temperature (middle) T and pressure (right) p at time 12 s

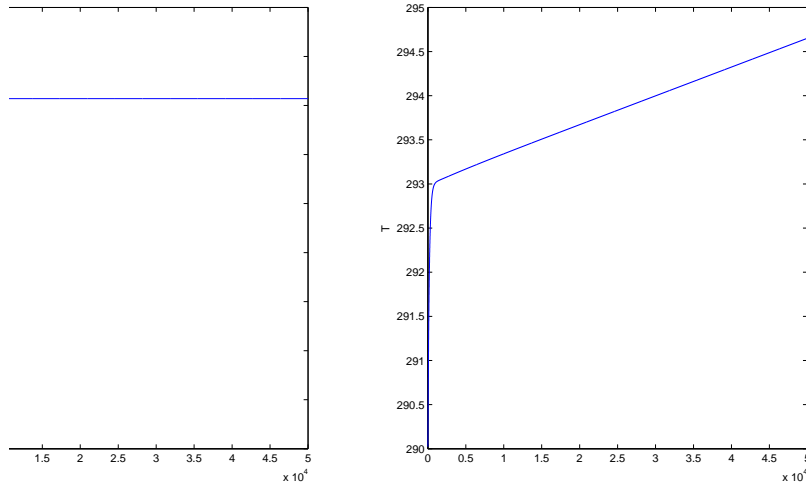


Figure B.4: Test model 1, graphs for average temperature (left) and energy (right)

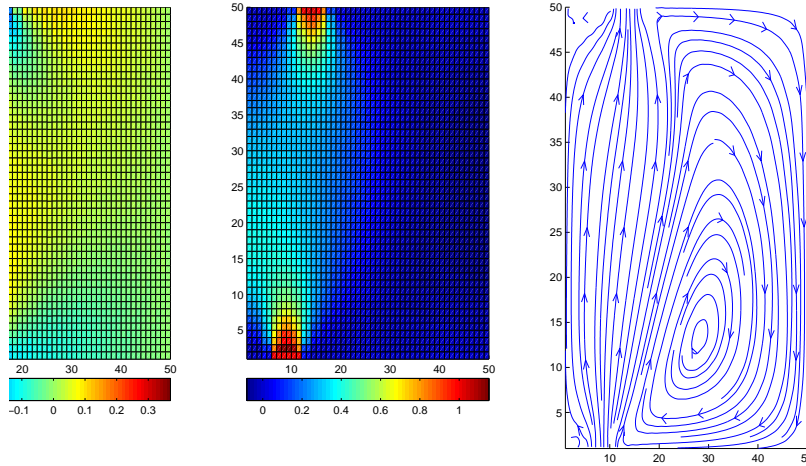


Figure B.5: Test model 1, graphs for x and y components of velocity field and velocity field for $N = 50$, $h_0 = 0.04\text{m}$, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2\text{m}$, $V_{in} = V_{out} = 1\text{m/s}$, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.

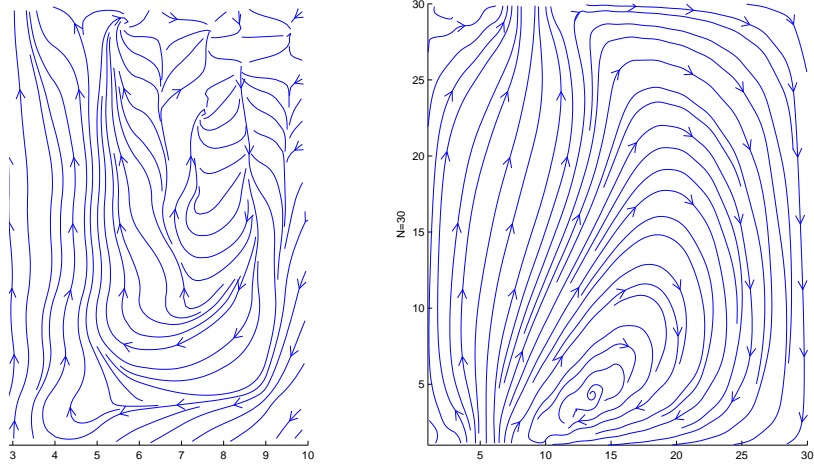


Figure B.6: Test model 1, velocity fields for $N = 10$, $h = 0.2 \text{ m} > h_0$ (left) and $N = 30$, $h = 0.067 \text{ m} > h_0$ (right), $\tau_0 = 0.008$, $\mu = 1 \text{ m}^2/\text{s}$, $L_x = L_y = 2 \text{ m}$, $V_{in} = V_{out} = 1 \text{ m/s}$.

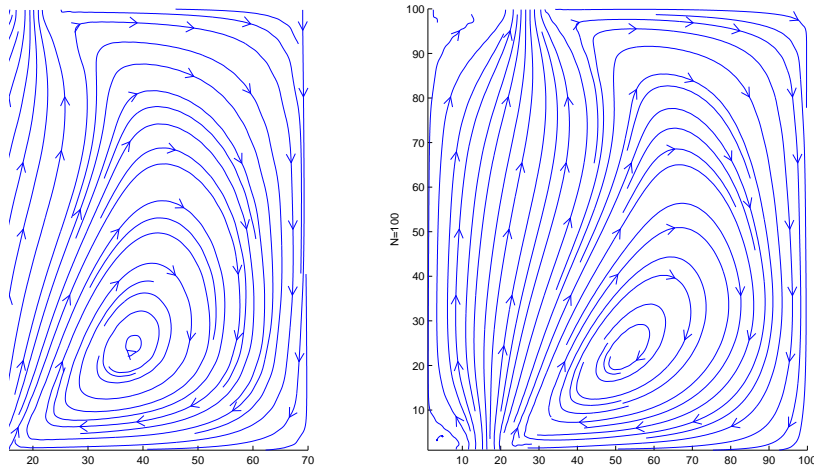


Figure B.7: Test model 1, velocity fields for $N = 70$, $h = 0.028 \text{ m} < h_0$ (left) and $N = 100$, $h = 0.02 \text{ m} < h_0$ (right), $\tau_0 = 0.008$, $\mu = 1 \text{ m}^2/\text{s}$, $L_x = L_y = 2 \text{ m}$, $V_{in} = V_{out} = 1 \text{ m/s}$, $x_1 = 0.2 \text{ m}$, $x_2 = 0.4 \text{ m}$, $x_3 = 0.4 \text{ m}$, $x_4 = 0.6 \text{ m}$.

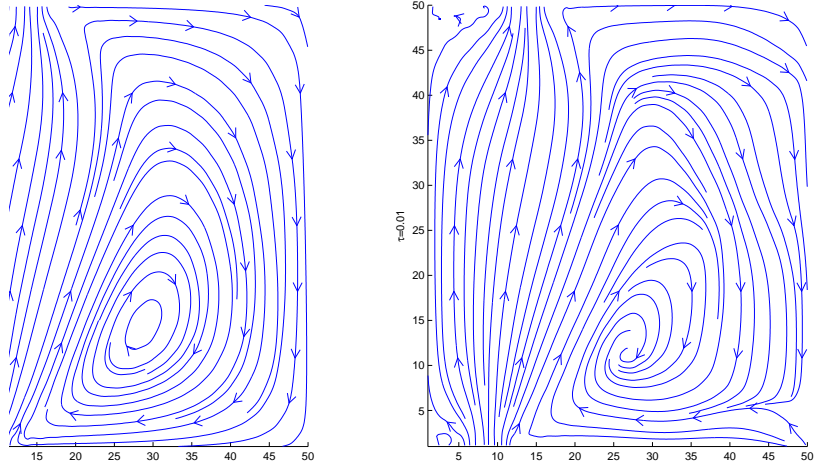


Figure B.8: Test model 1, velocity fields for $\tau = 0.005 < \tau_0$ (left) and $\tau = 0.01 > \tau_0$ (right), $N = 50$, $h = h_0 = 0.04$ m, $\mu = 1$ m²/s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.

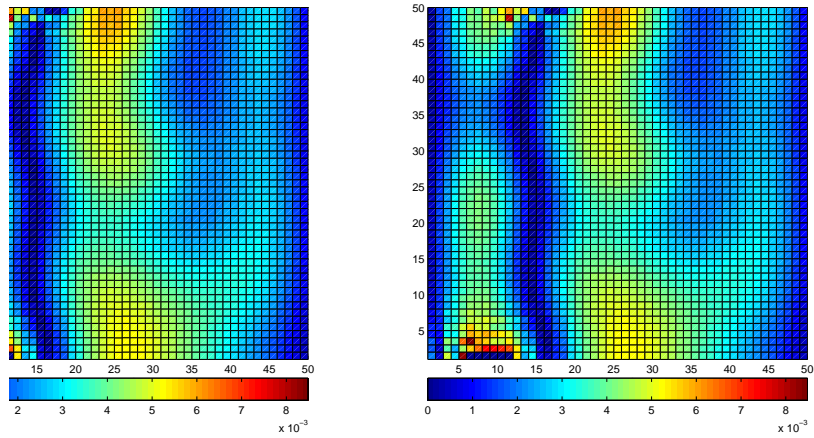


Figure B.9: Test model 1, graphs of differences $|u - u_0|$ and $|v - v_0|$ of x - and y -components of velocity field for $N = 50$, $h = h_0 = 0.04$ m, $\tau = 0.01$ and τ_0 . Here $\mu = 1$ m²/s, $L_x = L_y = 2$ m, $V_{in} = V_{out} = 1$ m/s, $x_1 = 0.2$ m, $x_2 = 0.4$ m, $x_3 = 0.4$ m, $x_4 = 0.6$ m.

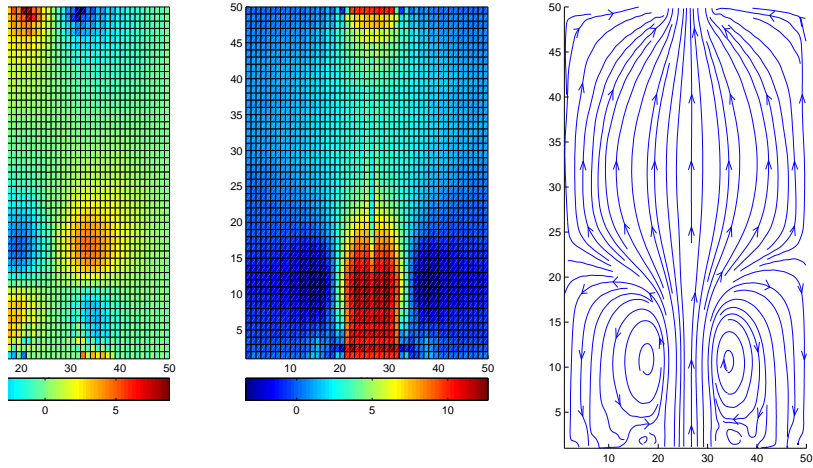


Figure B.10: Test model 1, graphs for x - and y - components of the velocity field and streamlines for computations without artificial viscosity: $\mu = 0$, $N = 50$, $h = 0.4$ m, $\tau = 0.008$, $L_x = L_y = 20$ m, $V_{in} = V_{out} = 10$ m/s, $x_1 = 8$ m, $x_2 = 12$ m, $x_3 = 8$ m, $x_4 = 12$ m.

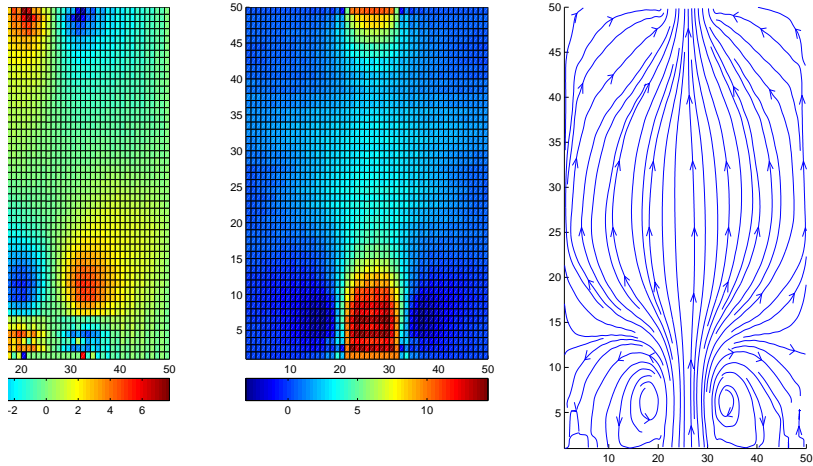


Figure B.11: Test model 1, graphs for x - and y - components of the velocity field and streamlines for computations without artificial viscosity: $\mu = 1$ m²/s, $N = 50$, $h = 0.4$ m, $\tau = 0.008$, $L_x = L_y = 20$ m, $V_{in} = V_{out} = 10$ m/s, $x_1 = 8$ m, $x_2 = 12$ m, $x_3 = 8$ m, $x_4 = 12$ m.

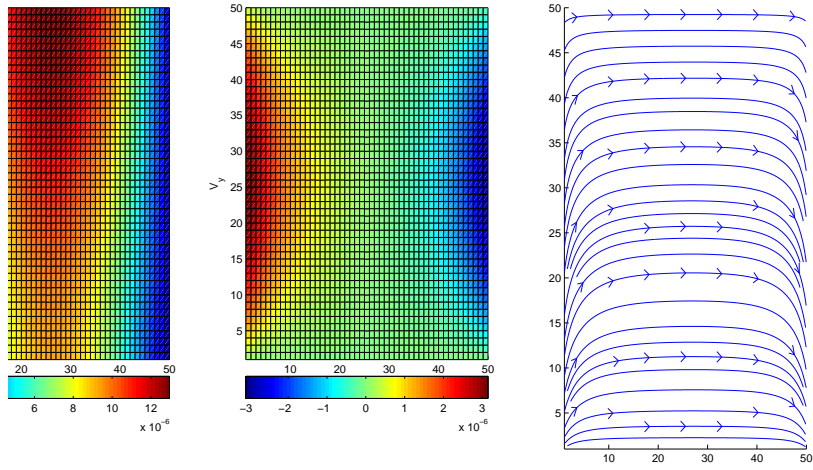


Figure B.12: Test model 2, graphs for velocity at time 10 s.

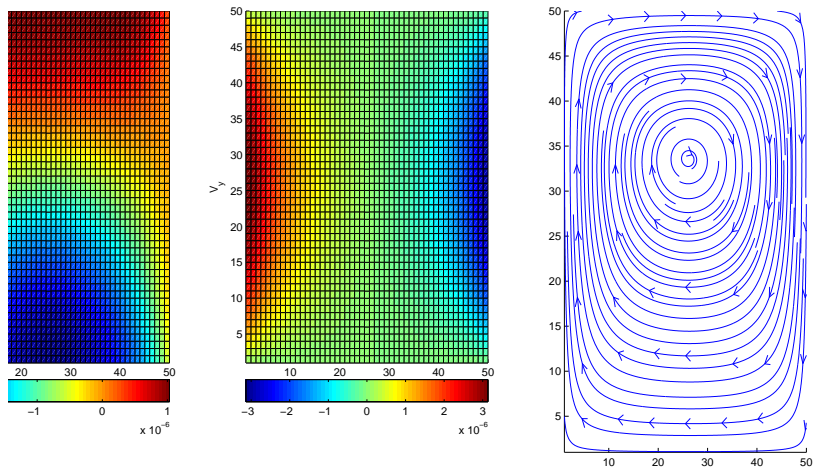


Figure B.13: Test model 2, graphs for velocity at time 18.9 s.

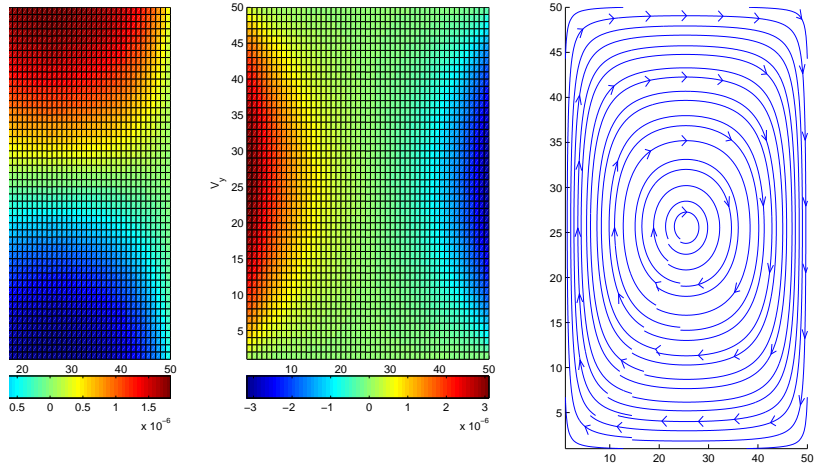


Figure B.14: Test model 2, graphs for velocity at time 31 s.

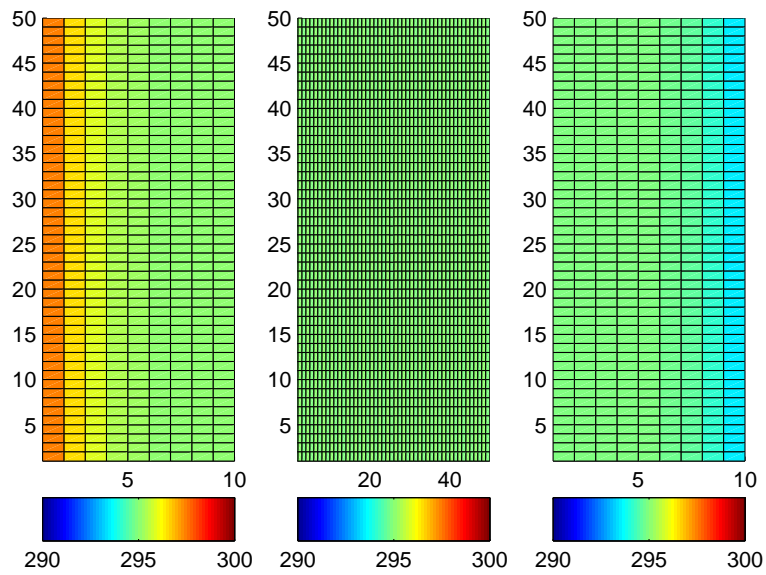


Figure B.15: Main model, initial temperature distribution at time 0.3 s.

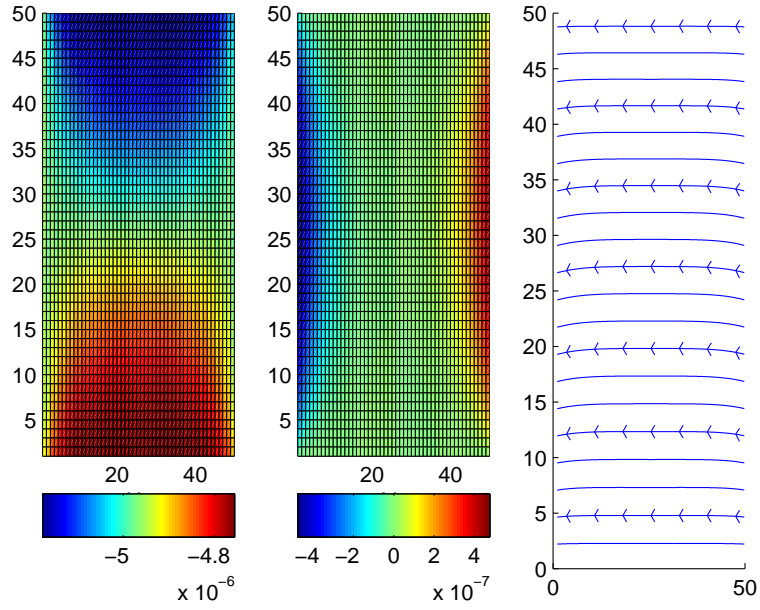


Figure B.16: Main model, initial velocity field at time 0.3 s.

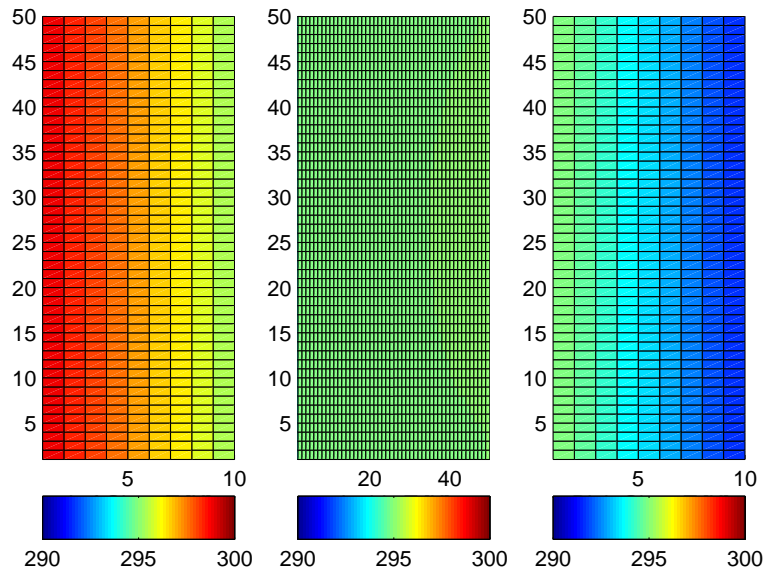


Figure B.17: Main model, final temperature distribution at 171 s.

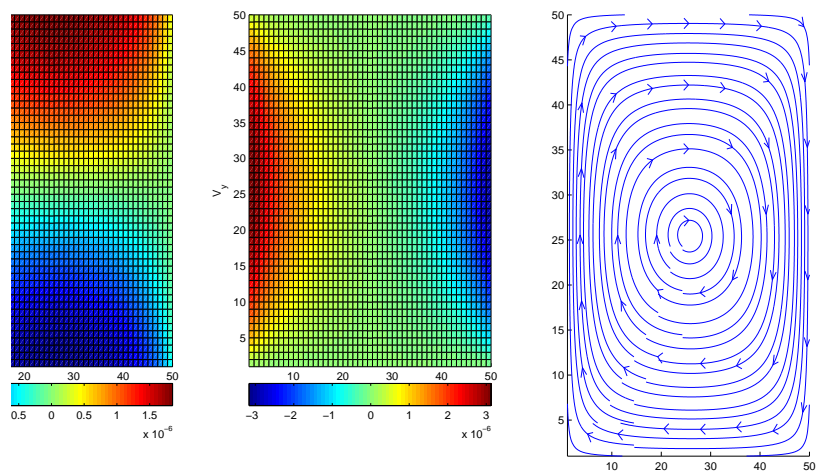


Figure B.18: Main model, final form of the velocity field inside gas chamber at 171 s.

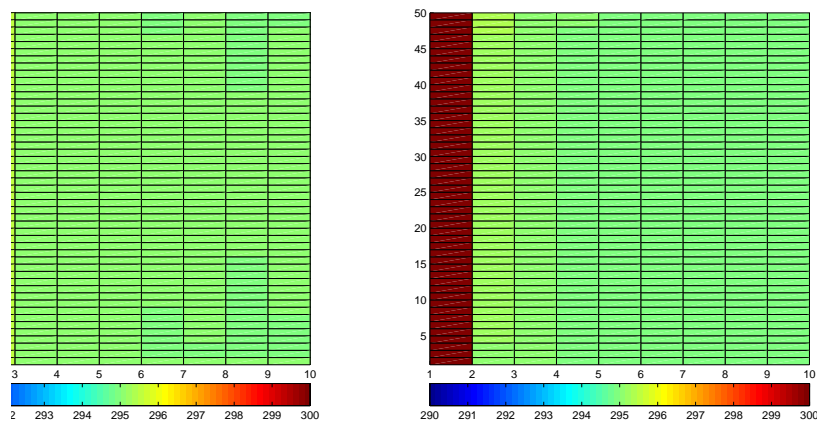


Figure B.19: Main model, comparison of results for different glass thermal conductivity at time 41 s. The conductivity is $K_0 = 3.4E - 7 W/m \cdot K$ (left) and $K_0 = 3.4E - 4 W/m \cdot K$ (right).

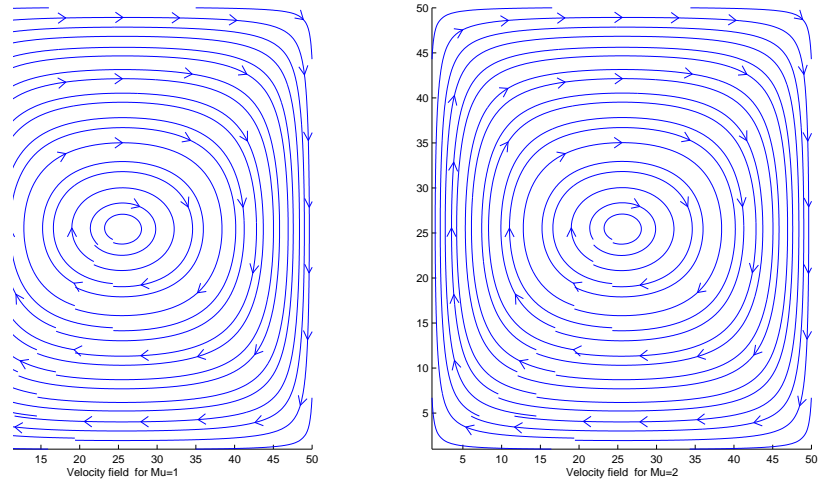


Figure B.20: Main model, comparison of velocity fields for different viscosity values at time 41 s, $\mu = 1 \text{ m}^2/\text{s}$ (left) and $\mu = 2 \text{ m}^2/\text{s}$ (right).

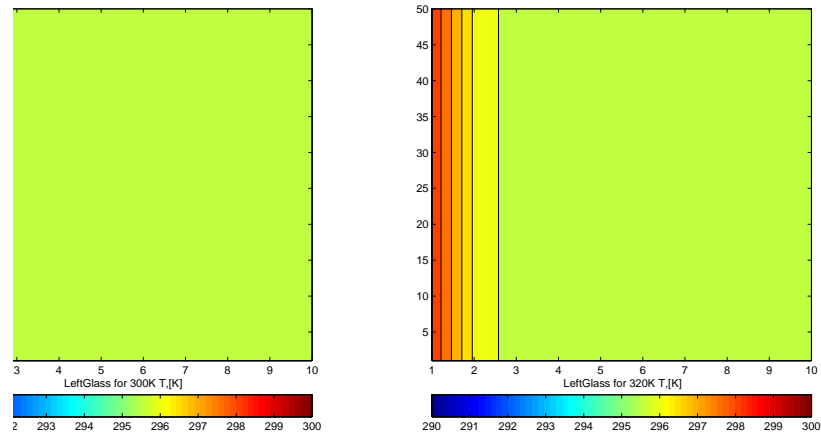


Figure B.21: Main model, comparison of temperature distribution inside glasses result for different outside temperatures at time 41s. $T_{left} = 300 \text{ K}$ (left) and $T_{left} = 320 \text{ K}$ (right).

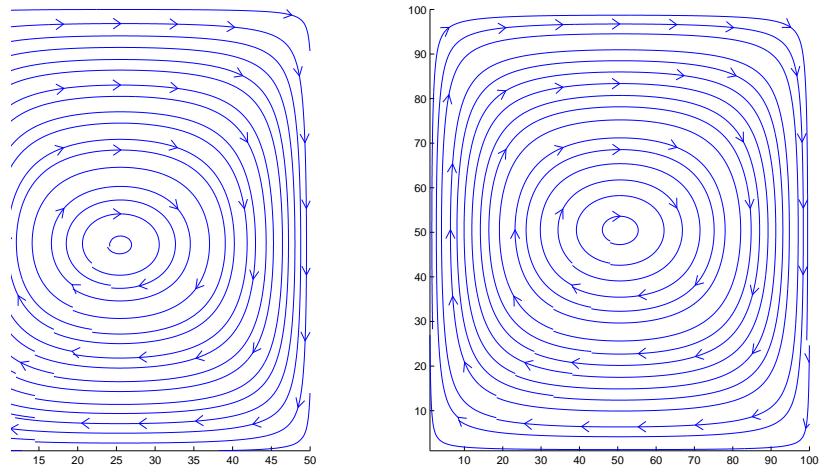


Figure B.22: Main model, comparison of velocity fields for different mesh sizes at time 25s (left: $N = 50$, $h = 0.04$ m, right: $N = 100$, $h = 0.02$ m).

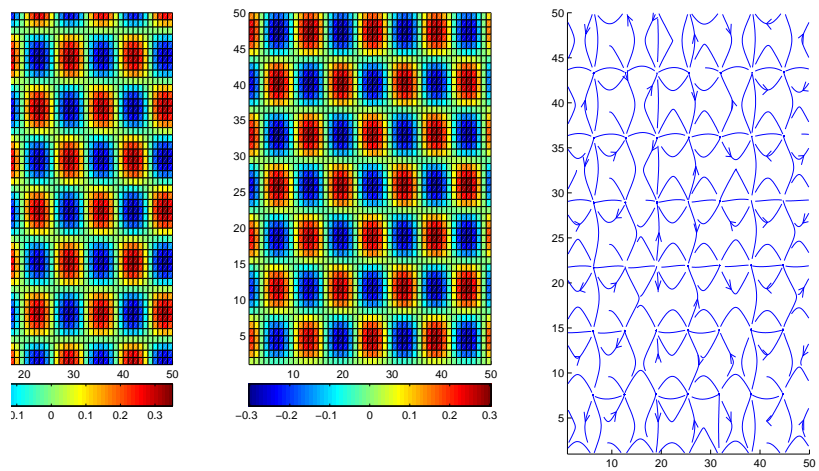


Figure B.23: Main model, a typical instability result for gas velocity components when the artificial viscosity is insufficient.