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

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 studied a model of two dimensional convection of an ideal gas in a rectangular domain having finite thickness walls. Temperature outside of walls considered constant ($T_{left} > T_{in} > T_{right}$). Heat exchange between walls and outside/inside air is computed using Newton's law of cooling. Inside walls is using mechanism of conductive heat transfer through heat equation. Mathematical model inside enclosure includes Navier-Stokes equations coupled with equation of state for gas. Model numerically studied using method of large particles. One of the main goal of the work was to develop a software in $C\#$ language for above described model. Results including stream lines and temperature distribution have been obtained and showed good correlation with physical processes.

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 Soteros, 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

List of Tables

List of Figures

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

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

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):

$$\rho \frac{D\mathbf{V}}{Dt} \equiv \rho \left(\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \triangledown) \mathbf{V} \right) = \rho \mathbf{F} - \operatorname{grad} p + \mu \bigtriangleup \mathbf{V}, \tag{1.2}$$

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

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{V}) = 0, \\ \\ \rho \left(\dfrac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \triangledown) \mathbf{V} \right) = \rho F - \operatorname{grad} p + \mu \bigtriangleup \mathbf{V}. \end{cases} \tag{1.3}$$

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

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{V}) = 0, \\ \\ \rho \left(\dfrac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \triangledown) \mathbf{V} \right) = \rho F - \operatorname{grad} p. \end{cases} \tag{1.4}$$

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:

$$p = \rho RT \tag{1.5}$$

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

$$p = \frac{RT}{V - b} \tag{1.6}$$

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

$$p = \frac{RT}{V - b} - \frac{a}{V^2} \tag{1.7}$$

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

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

where energy is assumed to be

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

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:

$$e = C_V nRT \tag{1.10}$$

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

$$\operatorname{div}\mathbf{V} = 0 \tag{1.11}$$

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:

$$Ma = \frac{|\mathbf{V}|}{c} \tag{1.12}$$

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

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

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

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

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

$$\gamma = \frac{C_p}{C_V} = \frac{\alpha + 1}{\alpha}, \tag{1.15}$$

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

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

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

$$s_1 - s_2 = \triangle s = \int_r ev \frac{dQ}{T}. \tag{1.17}$$

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,

$$s = \int C_V \frac{dT}{T} + R \ln v + \mathrm{const} \tag{1.18}$$

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:

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

where S is entropy density.

1.1.5 Laminar and turbulent flows. Reynolds numbers.

Laminar flow is characterized with 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 was created as a criterion to distinguish laminar and turbulent flow. 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.

$$Re = \frac{\rho \mathbf{V} l}{\mu} \tag{1.20}$$

where length l,velocity $\mathbf{V}$, density ρ and viscosity μ. For pipe flow, a Reynolds number 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

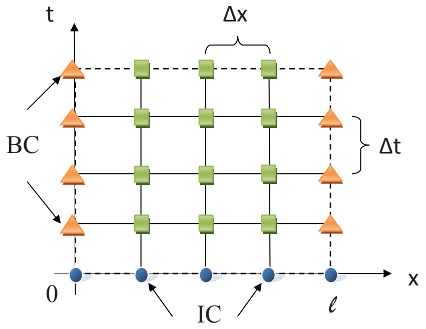
1.2.1 Finite difference method

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

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \tag{1.21}$$

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 $\Omega = (0, l) \times (0, T) \subset R^2$.

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 $\triangle x = x_{j+1} - x_j$ and $\triangle t = t_{k+1} - t_k$. 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

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

which could be reset in the form

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

Applying these formulas for any $j = 1, ..., n - 1$, 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.

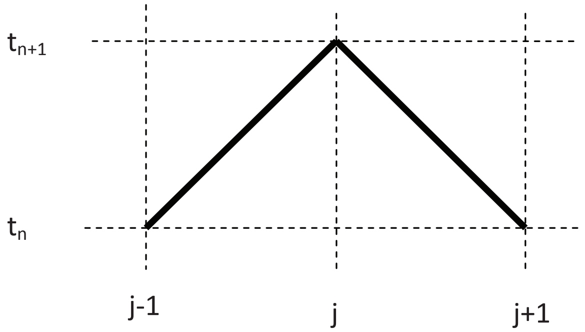


Figure 1.1: Stencil for Lax-Friedrichs method

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

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

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

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

and

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^n - u_{j-1}^n}{2 \triangle x}. \tag{1.26}$$

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

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

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

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

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

with initial conditions

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

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

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

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

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

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

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

Lax-Friedrichs method has first order precision in time and space, and is stable when $|\frac{\triangle t}{\triangle x} f'(\mathbf{V}_j^n)| \leq 1$ 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

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

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

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

with numerical flux

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

The method is stable when $\left|\lambda_p(v_j^n)\frac{\triangle t}{\triangle x}\right| \leq 1$ for all eigenvalues λ_p at each v_j^n. Also the method has numerical viscosity $s\frac{\triangle t}{\triangle x}$, 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.

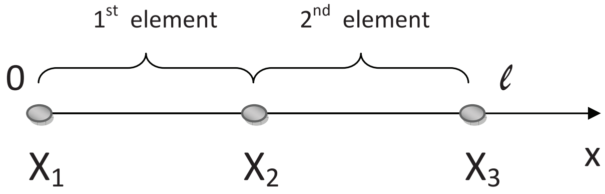
1.2.4 Finite element method

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 $(x_1, x_2) \bigcup (x_2, x_3)$.



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.

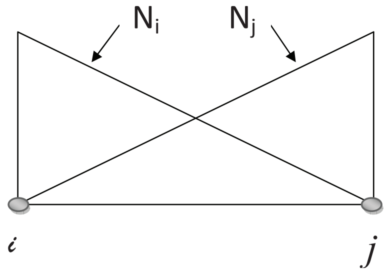
By dividing the solution region into a number of small regions, called elements, and approximating the solution over these regions by a suitable known function, a relation between the differential equations and the elements is established. The functions employed to represent the nature of the solution 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)

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



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:*

$$\sum_{i}^{n} N_i(x) = 1 \tag{1.37}$$

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

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

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

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

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*

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

for the first element and

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

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*

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

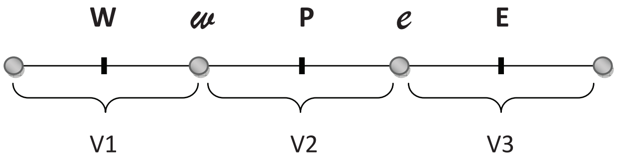
   Here one introduce the boundary conditions and then, by solving the system, one get the nodal values of the solution at the instant t_{k+1} from the values at the instant t_k (which appear on the right-hand side).

1.2.5 Finite volume method

At the first step we discretize in time the equation,

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

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

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_P = \frac{\left.\frac{\partial u}{\partial x}\right|_e - \left.\frac{\partial u}{\partial x}\right|_w}{x_e - x_w} \tag{1.43}$$

and then

$$\left.\frac{\partial u}{\partial x}\right|_e = \frac{u_E - u_P}{x_E - x_P}, \left.\frac{\partial u}{\partial x}\right|_w = \frac{u_P - u_W}{x_P - x_W}. \tag{1.44}$$

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 arc 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

$$\Omega = \bigcup_{i=1}^{n} \Omega_i \tag{1.45}$$

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 turbulent 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.

[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.

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

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 $\triangle t$ time.

2. Lagrangian stage

After first stage, next step is to take into consideration displacements of cells proportionally their velocity and $\triangle t$ 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 $\dfrac{\partial \rho}{\partial t} = 0$ and from conservation of mass one can get that $\operatorname{div} \rho \mathbf{V} = 0$. Then in (2.1) ρ can be taken out of time derivation and parts of type $\operatorname{div} \phi \rho \mathbf{V}$ can be neglected, then this modified system can be group by time derivatives. Below Navier-Stokes system described using this assumptions.

$$\begin{cases} \rho \dfrac{\partial u}{\partial t} + \dfrac{\partial p}{\partial x} = \mu \Delta u, \\ \\ \rho \dfrac{\partial v}{\partial t} + \dfrac{\partial p}{\partial y} = -\rho g + \mu \Delta v, \\ \\ \rho \dfrac{\partial E}{\partial t} + \operatorname{div} p \mathbf{V} = \mathbf{E}(\mathbf{g}). \end{cases} \tag{2.2}$$

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

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

where derivatives in times are defined as:

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

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

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

Where $\tilde{u}, \tilde{v}, \tilde{E}$ are transitional values of velocity and energy at time $t^n + \triangle t$. 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 $\triangle M^n$ through boundaries of cell are computed at time $\triangle t$ 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:

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

where flow of mass is

$$\triangle M_{i+\frac{1}{2},j}^{n} =< \rho_{i+\frac{1}{2},j}^{n} >< \tilde{u}_{i+\frac{1}{2},j}^{n} > \triangle y \triangle t. \tag{2.5}$$

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

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

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

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

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

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

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

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

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 $\triangle M^n$ could be computed using formulas of first order. In this case it can be done without adding artificial viscosity.

$$\triangle M^n_{i+\frac{1}{2},j} = \begin{cases} \rho^n_{ij}\dfrac{\tilde{u}^n_{ij}+\tilde{u}^n_{i+1,j}}{2} \triangle y \triangle t, \text{ if } \tilde{u}^n_{ij}+\tilde{u}^n_{i+1,j} > 0, \\ \\ \rho^n_{i+1,j}\dfrac{\tilde{u}^n_{ij}+\tilde{u}^n_{i+1,j}}{2} \triangle y \triangle t, \text{ if } \tilde{u}^n_{ij}+\tilde{u}^n_{i+1,j} < 0. \end{cases} \tag{2.7}$$

2.4 Final stage

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

$$M^{n+1} = M^n + \sum \triangle M^n_{boundary},$$

$$P^{n+1} = P^n + \sum \triangle P^n_{boundary}, \tag{2.8}$$

$$E^{n+1} = E^n + \sum \triangle E^n_{boundary}.$$

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

$$\rho^{n+1}_{ij} = \rho^n_{ij} + \frac{\triangle M^n_{i-\frac{1}{2},j} + \triangle M^n_{i,j-\frac{1}{2}} - \triangle M^n_{i,j+\frac{1}{2}} - \triangle M^n_{i+\frac{1}{2}}, j}{\triangle x \triangle y},$$

$$X^{n+1}_{ij} = \Big\{ D^n_{ij}(1)\tilde{X}^n_{i-1,j} \triangle M^n_{i-\frac{1}{2},j} + D^n_{ij}(2)\tilde{X}^n_{i,j-1} \triangle M^n_{i,j-\frac{1}{2}} + D^n_{ij}(3)\tilde{X}^n_{i+1,j} \triangle M^n_{i+\frac{1}{2},j} +$$

$$+ D^n_{ij}(4)\tilde{X}^n_{i,j+1} \triangle M^n_{i,j+\frac{1}{2}} + \tilde{X}^n_{i,j}\Big(\rho^n_{ij} \triangle x \triangle y - [1 - D^n_{ij}(1)] \triangle M^n_{i-\frac{1}{2},j} -$$

$$- [1 - D^n_{ij}(2)] \triangle M^n_{i,j-\frac{1}{2}} - [1 - D^n_{ij}(3)] \triangle M^n_{i+\frac{1}{2},j} - [1 - D^n_{ij}(4)] \triangle M^n_{i,j+\frac{1}{2}} \Big) \Big\} / \left\{ \rho^{n+1}_{ij} \triangle x \triangle y \right\}, \tag{2.9}$$

where $D^n_{ij}(k)$ 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.

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

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 $C\frac{\triangle t}{\triangle s} < 1$, 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 $A\frac{\triangle x}{\triangle t}$,where A is constant from equation (1.24). For method of large particle numerical viscosity is $\varepsilon_x = \frac{1}{2}|u| \triangle x, \quad \varepsilon_y = \frac{1}{2}|v| \triangle y$.

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:

$$q = \nu \frac{\triangle u}{\triangle x} = BC\rho \triangle u, \tag{2.11}$$

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

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

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

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

with a similar form for $\tilde{v}_{ij}^n$ y-component of velocity. Computations for this model (discussed below) show that results are consistent for $\mu \geq 0.9$ results for such smaller viscosities on pictures B.23 start time and continue on 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

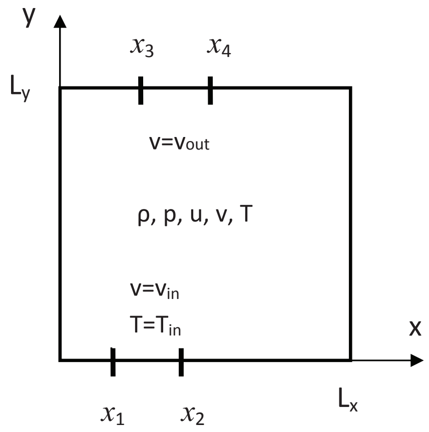
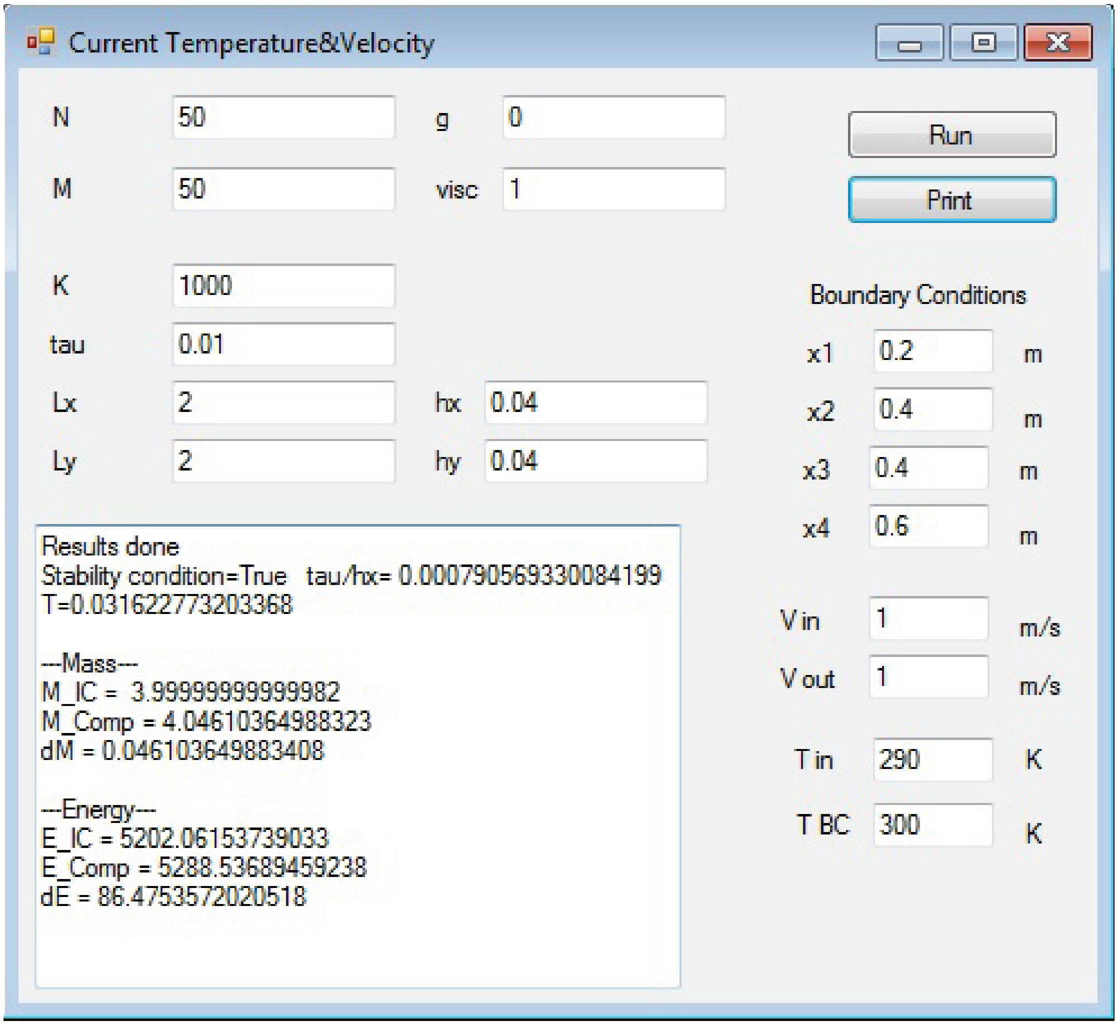


Figure 3.1: Test model 1

First test model is to observe rectangular area with two"no-leak" boundaries and on the bottom boundary area has condition with stream of higher temperature vertically flowing in with velocity v_{in} between x_1 and x_2 coordinates and on top boundary has condition of flow out with the velocity v_{out} between x_3 and x_4 coordinates. 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.



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

Boundary Conditions

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:

$$\begin{cases} u(0,y,t) = u(L_x,y,t) = 0, 0 \le y \le L_y \\ v(x,L_y,t) = \begin{cases} 0, 0 \le x \le x_3, x_4 \le x \le L_x \\ v_{out}, x_3 \le x \le x_4, \end{cases} \\ v(x,0,t) = \begin{cases} 0, 0 \le x \le x_1, x_2 \le x \le L_x \\ v_{in}, x_1 \le x \le x_2, \end{cases} \\ T(x,0,t) = T_{in} \\ \rho(x,0,t) = \rho_0 \\ p(x,0,t) = p_0 \end{cases} \tag{3.1}$$

Initial conditions are:

$$\begin{cases} \mathbf{V} = 0 \\ T = T_0 \\ \rho = \rho_0 \\ p = \frac{R}{M}\rho T \end{cases} \tag{3.2}$$

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 12s for model with initial temperature inside $T_0 = 290$K and temperature of influent flow $T_{in} = 300$ can be seen on pictures B.1, B.2 and B.3. It is seen that temperature inside rose for 294K at 2.5s, what is seen on figures with temperature distribution, density and pressure distribution conform physical interpretation of the modeled process.

3.1.1 Effects of steps in model

For analyzing method lets look at effects of different parameters on test model 1, for benchmark lets take results for model with parameters length of side of domain is $L_x = L_y = 2$m, for influent current coordinates of opening $x_1 = 0.2$m, $x_2 = 0.4$m and velocity $v_{in} = 1$m/s, for effluent current coordinates of opening $x_1 = 0.4$m, $x_2 = 0.6$m, $v_{out} = 1$m/s,value of density and energy carries from previous layer, time step $\tau = 0.008$, control time $t = 0,316$s, due to low velocities values artificial viscosity is added $\mu = 1$. Current passing through the left side of the domain and formed a swirl at the right can be seen on results for chosen benchmark on figure B.5.

First series of tests are to compare results for benchmark B.5 with results for the same physical model, but with different space steps (bigger and smaller than for chosen model) and time step is considered fixed at $\tau = \tau_0$. Results on figures B.6 shows result for bigger cell approximation in particular $h = 0.2$m and $h = 0.067$m. For $N = 30$ results very close to benchmark, but further increasing in space steps ($N = 10, 5, ..$) lead for loss of accuracy. However it is showed advantage of method that even for such large particles method is stable and gives acceptable results which can be used for fast first computation to estimate behavior of flow and if necessary for more precise computing can be used smaller particles. Comparatively small number of cell for domain allows to reduce computational time. Results on figures B.7 shows result for smaller cell approximation in particular $h = 0.02$m and $h = 0.028$m shows no further improvement in results, so computations can be done with lower amount of cell and obtain the same physical picture. From this series of test can be concluded that to star chose space step better from higher h, and continue until desired accuracy is obtained.

Next couple of comparison are made with fixed space step $h = h_0$ and for different time steps (bigger and smaller than chosen benchmark). Results on figure B.8 shows that both of them close to chose benchmark and first one is twice smaller then second. So for faster computation time step should be chosen starting from stability condition without compromising accuracy. Also results of comparison with τ_0 ($|\mathbf{V} - \mathbf{V_0}|$) on figure B.9 for $\tau = 0.005$ and the same graph was obtained for $\tau = 0.01$, both of them demonstrate very small differences ($\sim 10^{-3}$).

3.1.2 Artificial viscosity effects

Also this model is helpful to show work of method for bigger velocities where there is no necessity in artificial viscosity. Results on picture B.10 obtained for velocity 10m/s without artificial viscosity, only numerical viscosity is present , and for comparison the same problem computed with viscosity equal one on picture B.11, where Reynolds number $Re = \rho V l / \mu$ is equal 200. This is showed work of method for velocities bigger than in convection model without necessity in artificial viscosity.

3.2 Test model 2

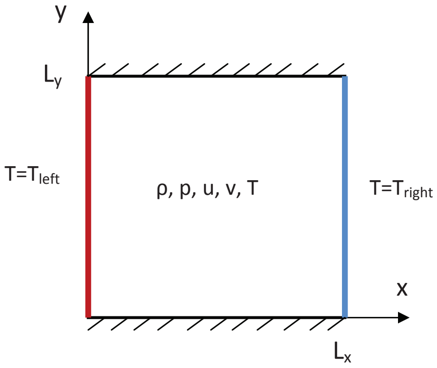


Figure 3.3: Test model 2

Second test model is to observe rectangular domain, with initial conditions assuming to be equilibrium state of this area with gravity taken into account.

$$\begin{cases} \mathbf{V} = 0 \\ T = T_0 \\ \rho = \rho_0 e^{-\frac{Mg}{RT}y} \\ p = \dfrac{R}{M}\rho T \end{cases} \tag{3.3}$$

and inner temperature T_{in}. In this model, fluid is assumed viscous in order to observe flow direction better, and Navier-Stokes equations (1.3) are solved instead of Euler's (1.4) in this case. 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

$T_{right} < T_{in}$.

$$\begin{cases} u(0,y,t) = u(L_x,y,t) = 0, 0 \le y \le L_y \\ v(x,0,t) = v(x,L_y,t) = 0 \\ T(0,y,t) = T_{left} \\ T(L_x,y,t) = T_{right} \end{cases} \tag{3.4}$$

In result,expecting to observe convection in this area. Results for different times are presented on figures B.12 where flow goes straight form left to right due to compressibility assumptions, B.13 and final stabilized convection on figure B.14 at 31s after launch. dM and dE are differences between initial and final values of mass and energy. Results show expected convection, air streams warms and rise near hot boundary and air cools and go down near cold boundary forming convectional roll taking 31s from launch with given above initial conditions to forming a roll.

3.3 Main model

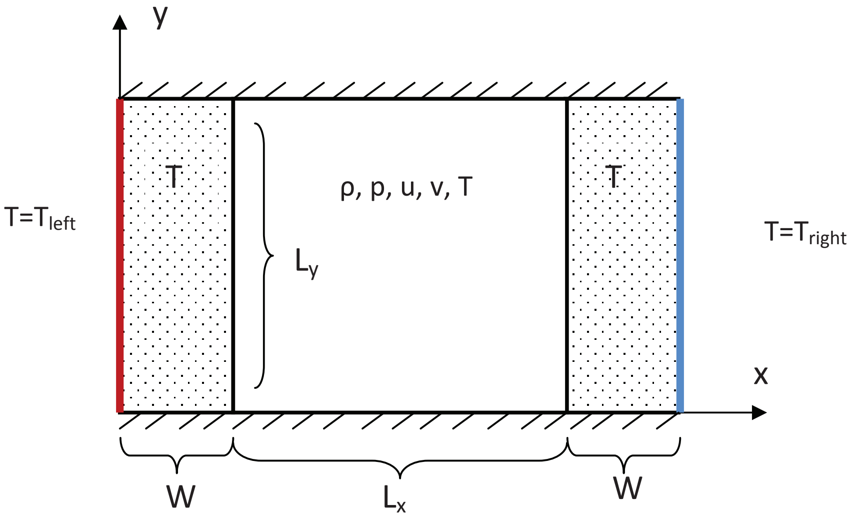


Figure 3.4: Main model

The main model based on test model 2 with modifications at boundaries. For this model at every side of domain added two rectangular domain representing role of glass walls. At left boundary of left window temperature of environment assumed constant and equal T_{left}, the same at right boundary of right window temperature of environment is $T_{right} < T_{left}$.

In this model heat equation computed to obtain temperature distribution $s(x, y, t)$ inside glass area:

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

where k is thermal diffusivity of material. So using explicit finite differences to approximate heat equation, one get formula:

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

In places of connection between glass and air, one need to obtain a proper boundary conditions. There is a different 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 particulary, 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]

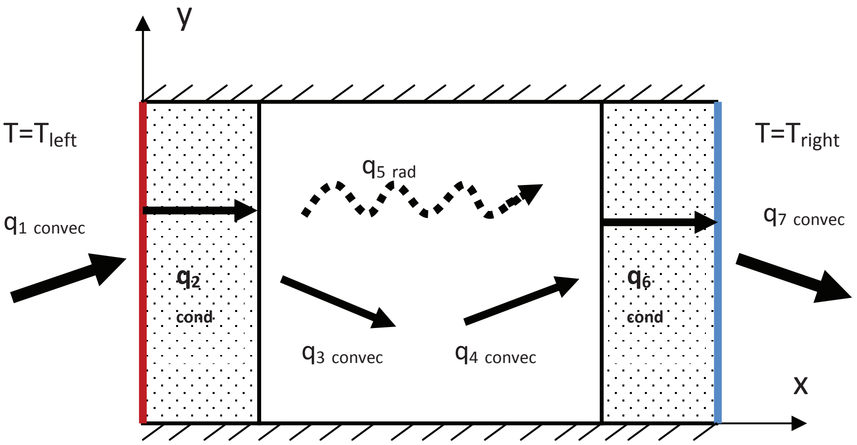


Figure 3.5: Types of heat transfer

Figure 3.5 shows different types of heat transfer discussed above: q_1 is free convection from air to the glass, q_2 is conduction through the left glass, q_3 is free convection from the glass to the air space, q_4 is free convection from the air space to the glass, q_5 is net radiation exchange between the inner surface of the left glasses, q_6 is conduction through the right glass, q_7 is free convection from the glass to the air. Main model is described all this heat exchanges, except radiation q_5.

Initial condition for air space domain is the same as in previous test model (3.3). For glass areas initial temperature distributions are $s^0_{ij} = T_0$. Top and bottom boundary of glasses assumed perfectly insulated:

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

Inner and outer surface boundary conditions computed using boundary condition:

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

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 can be seen on figures B.15 temperature distribution in glass and air domains and B.16 velocity field inside air domain for time 0.3s, times to establish convection is longer than in test model 2 due to time necessary to glasses to warmed/cooled and create differences of temperature at air domain boundaries. Final results at time 171s where convection established on figure B.18 and temperature distribution on B.17. Computations were conducted with presence of artificial viscosity equal to one. However as it was tested viscosity can't be decreased lower than 0.92, results for such low viscosity on figure B.23. Comparison of result for different parameters can be seen on picture B.19 for different conductivity, picture B.20 for different viscosity coefficient, on B.21 for different parameters of temperature outside of glass and on picture B.22 results for different mesh size.

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. In general system of equations describing motion of fluid was formed and known as Navier-Stokes equations for viscous fluid or Euler's equation for inviscid. Also when modeling gas or fluid dynamics equation of state uses to complete system and ensure that all parameter of system such as pressure p, density ρ and temperature T is described by the law of behavior of gases. Different assumptions can be applied to problem in order to make computations simpler, such as incompressibility, or look at a particular process, for example, adiabatic process. Depending on type and complexity of model process system of motion of fluid - Navier-Stokes system usually 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 are 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 perform computations using the method of large particles. This is a finite volume method was developed based on idea of combining benefits of Euler 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.

Obtained results represents three models for which method of large particles was applied in this work. For the method of large particles computational software package was developed using $C\#$ programming language, code provides results in text files, which then used in drawing program, written in $Matlab$ for obtaining visualized pictures of data. First model describes rectangular cavity with open segments on top and bottom boundaries, current of velocity V_{in} and slightly higher temperature flows in through bottom open segment and current of the same velocity is on top boundary. Thus the model represent enclosure with some current passing through. Obtained results show correspondence with physical meaning. Second model describes rectangular encloser with set temperatures at left and right boundaries ($T_{left} > T_{in} > T_{right}$). Initially air inside set at equilibrium state, from launch convection start forming and after some time stabilized convection is observed.

Third model is a main model for this work, represents method of large particles applied for two dimensional convection problem in rectangular domain with finite thickness walls. Model intended to represent processes in window. Walls considered participating in system through heat exchange with air inside and outside of walls using Newton's law of cooling as a boundary condition. Obtained results coincide with physical representation of convection process. Future work in the direction of extension of numerical model may include extension of the model to three dimensions, also can be included radiative component of heat exchange for more precise approximation of heat transfer in system, for comparison purposes model can be computed using other classical methods.

References

Bibliography

[1] Amaya J., Cabrit O., Poitou D., Cuenot B., ElHafi M. Unsteady coupling of Navier-Stokes and radiative heat transfer solvers applied to an anisotheral 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] Belotserkovskii 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. Computatuional 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 Mechanichs.*, 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 $TglassRight$ is a matrix for values of function $u(x, y, t)$ at nodes of mesh. The same procedure for left glass, where $TglassLeft$ 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 $p^n_{i\{+/-\}\frac{1}{2},j}$ and p{p/m}yHalf for $p^n_{i,j\{+/-\}\frac{1}{2}}$; uXX, uYY are for u_{xx}, u_{yy}; u{p/m}Half is for $u^n_{i\{+/-\}\frac{1}{2},j}$; v_eu, u_eu and E_eu are for $\tilde{u}^n_{ij}, \tilde{v}^n_{ij}$ and $\tilde{E}^n_{ij}$.

```
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 $M^n_{i\{+/-\}\frac{1}{2},j}$ 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=> 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=> 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

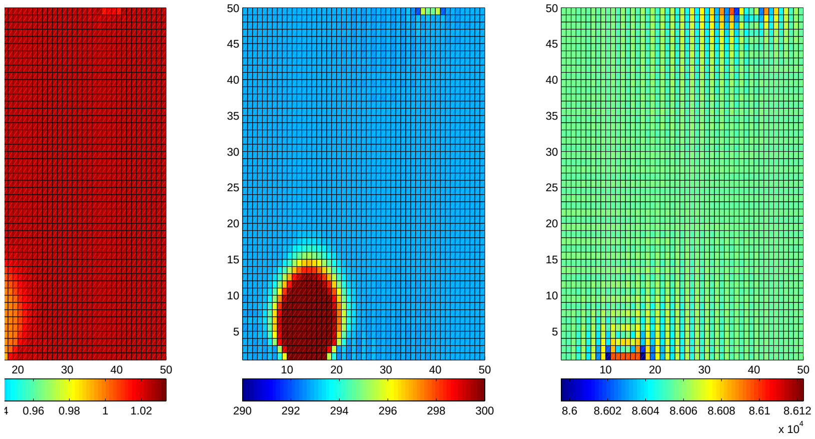


Figure B.1: Test model 1, graphs for density (left) ρ, temperature (middle) T and pressure (right) p at time 2.52s for $N = 50$, $h_0 = 0.04$m, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2$m, $V_{in} = V_{out} = 1$m/s, $x1 = 0.2$m, $x2 = 0.4$m, $x3 = 1.4$m, $x4 = 1.6$m

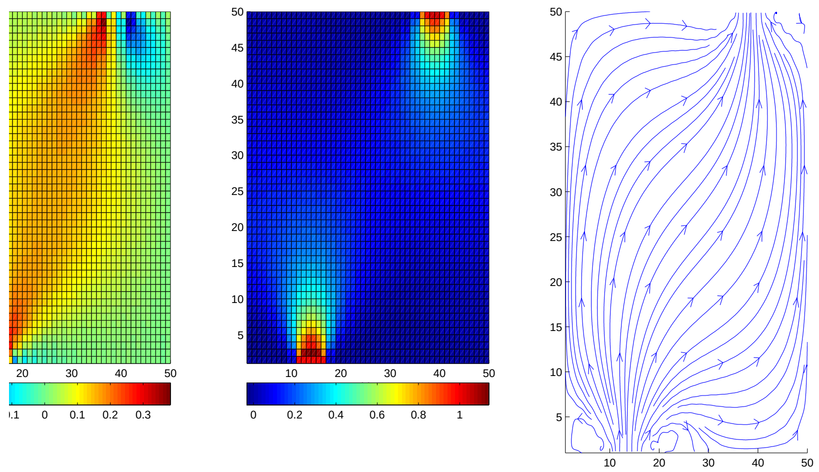


Figure B.2: Test model 1, graphs for velocity field at time 2.52s

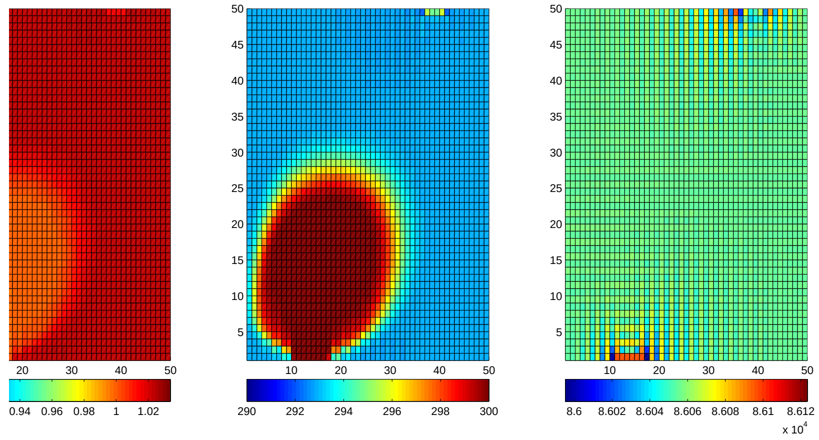


Figure B.3: Test model 1, graphs for density (left) ρ, temperature (middle) T and pressure (right) p at time 12s

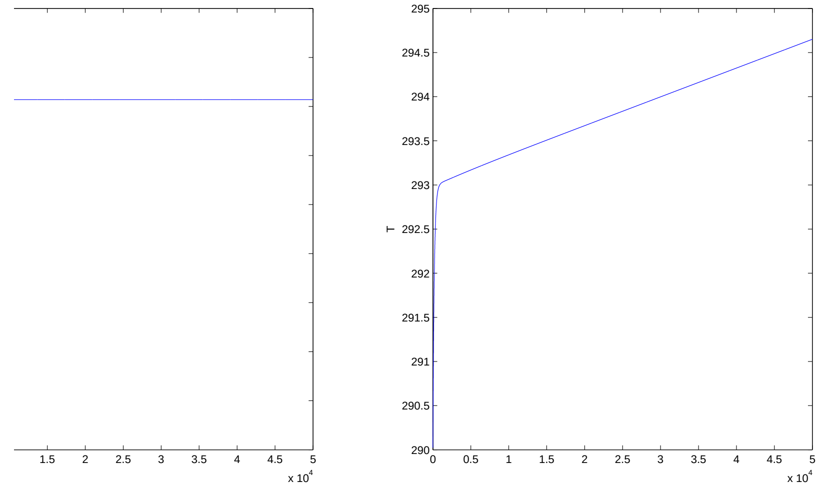


Figure B.4: Test model 1, graphs for average temperature (left) and energy (right)

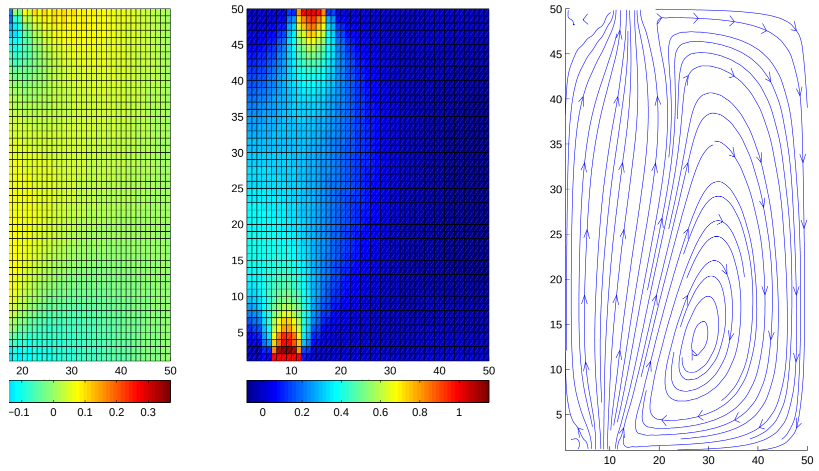


Figure B.5: Test model 1, graphs for x and y components of velocity field and velocity field for $N = 50$, $h_0 = 0.04$m, $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2$m, $V_{in} = V_{out} = 1$m/s, $x1 = 0.2$m, $x2 = 0.4$m, $x3 = 0.4$m, $x4 = 0.6$m

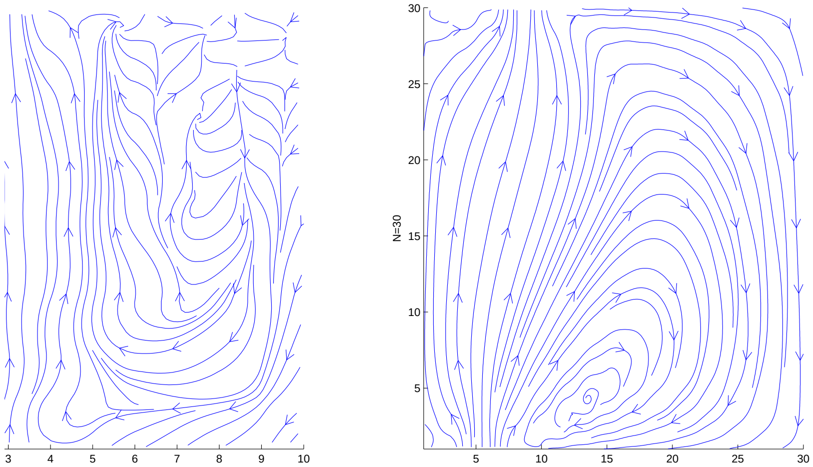


Figure B.6: Test model 1, graphs for bigger space steps shows velocity field, where $N = 10$, $h = 0.2\text{m} > h0$ (left) and $N = 30$, $h = 0.067\text{m} > h0$ (right), $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2\text{m}$, $V_{in} = V_{out} = 1\text{m/s}$, $x1 = 0.2\text{m}$, $x2 = 0.4\text{m}$, $x3 = 0.4\text{m}$, $x4 = 0.6\text{m}$

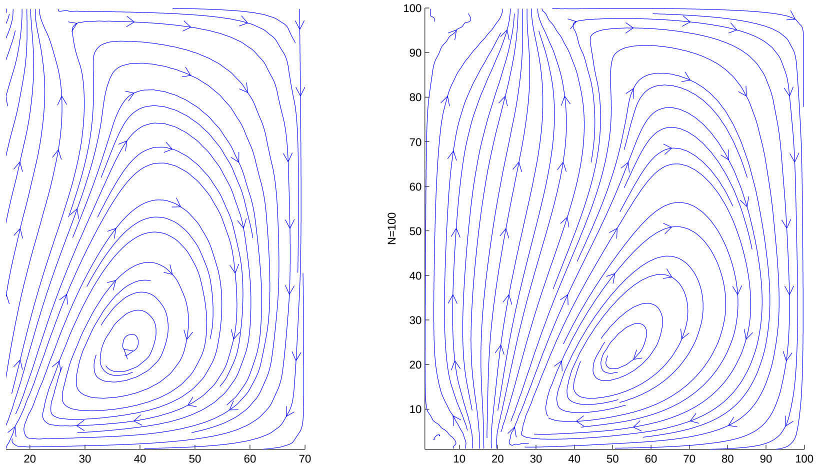


Figure B.7: Test model 1, graph for smaller space steps shows velocity field, where $N = 70$, $h = 0.028\text{m} < h0$ (left) and $N = 100$, $h = 0.02\text{m} < h0$ (right), $\tau_0 = 0.008$, $\mu = 1$, $L_x = L_y = 2\text{m}$, $V_{in} = V_{out} = 1\text{m/s}$, $x1 = 0.2\text{m}$, $x2 = 0.4\text{m}$, $x3 = 0.4\text{m}$, $x4 = 0.6\text{m}$

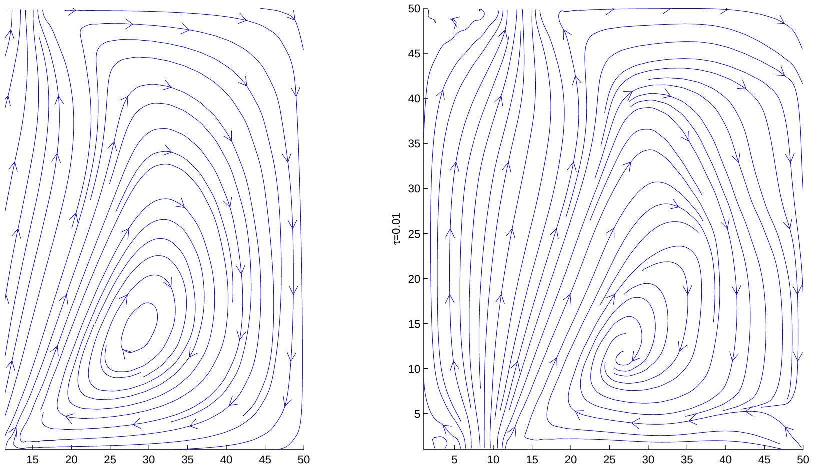


Figure B.8: Test model 1, graphs for different time steps shows velocity fields for τ = 0.005 < τ_0 and τ = 0.01 > τ_0, N = 50, h = $h0$ = 0.04, μ = 1, L_x = L_y = 2m, V_{in} = V_{out} = 1m/s, x_1 = 0.2m, x_2 = 0.4m, x_3 = 0.4m, x_4 = 0.6m

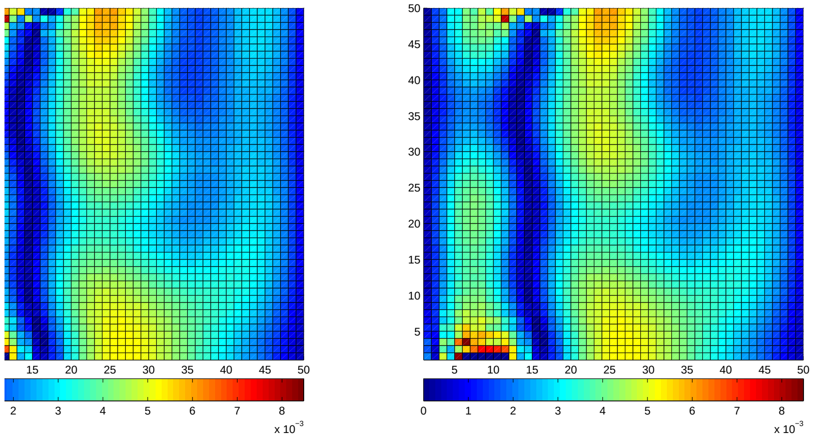


Figure B.9: Test model 1, graphs for differences of x and y components of velocity field for N = 50, h = $h0$ = 0.04m, τ = 0.01 & τ_0, μ = 1, L_x = L_y = 2m, V_{in} = V_{out} = 1m/s, x_1 = 0.2m, x_2 = 0.4m, x_3 = 0.4m, x_4 = 0.6m, where left is $|u - u_0|$ x component and right is $|v - v_0|$ x component differences

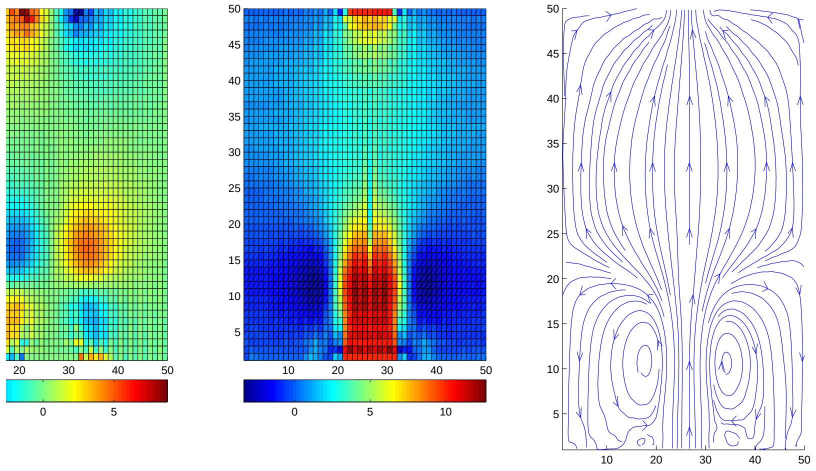


Figure B.10: Test model 1, graphs for x and y components of velocity field and velocity field for computations without artificial viscosity $\mu = 0$, $N = 50$, $h = 0.4$m, $\tau = 0.008$, $L_x = L_y = 20$m, $V_{in} = V_{out} = 10$m/s, $x_1 = 8$m, $x_2 = 12$m, $x_3 = 8m$, $x_4 = 12$m

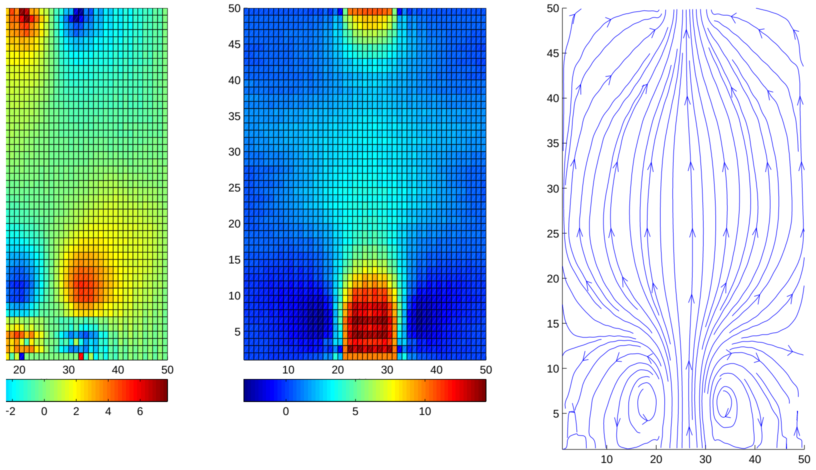


Figure B.11: Test model 1, graphs for x and y components of velocity field and velocity field for computations with artificial viscosity $\mu = 1$ with other parameters without change $N = 50$, $h = 0.4$m, $\tau = 0.008$, $L_x = L_y = 20$m, $V_{in} = V_{out} = 10$m/s,$x_1 = 8$m, $x_2 = 12$m, $x_3 = 8m$,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

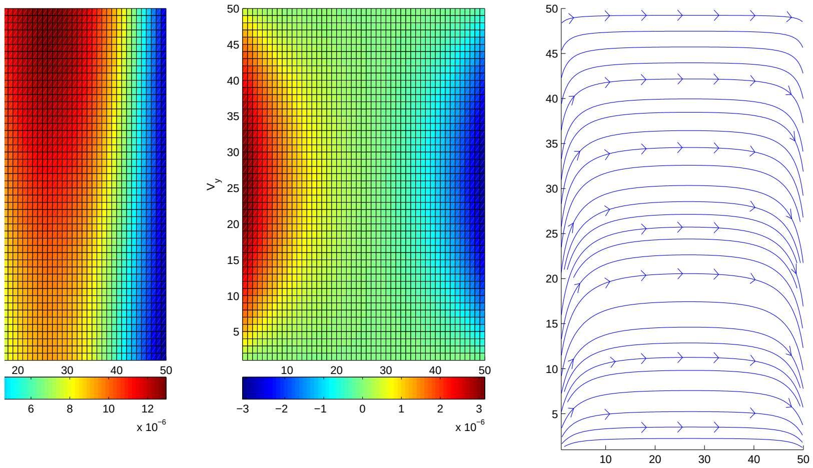


Figure B.12: Test model 2,graphs for velocity at time 10s, K = 300000, dM = $4,55E-15$, dE = $423,15$

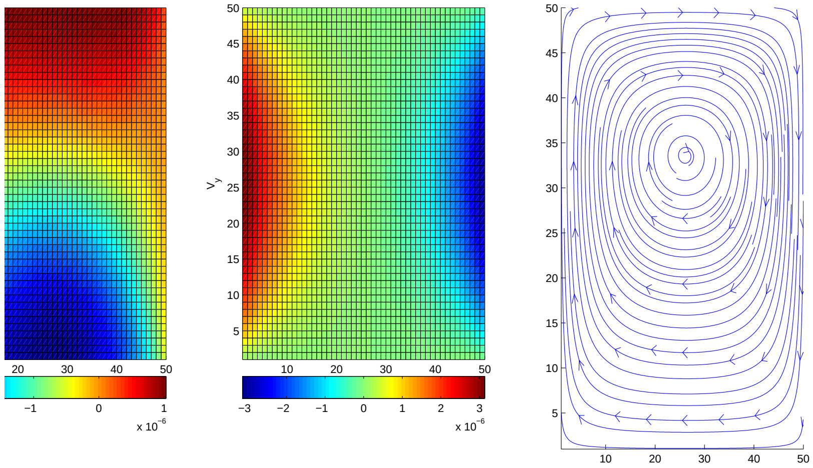


Figure B.13: Test model 2,graphs for velocity at time 18, 9s, K = 600000, dM = $5,44E-15$, dE = $423,14$

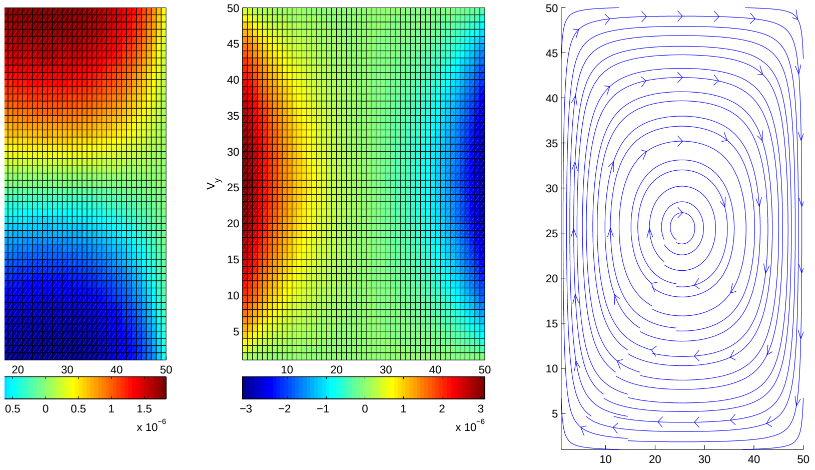


Figure B.14: Test model 2,graphs for velocity at time 31s, $K = 1000000$, $dM = 4,55E - 15$, $dE = 423,14$

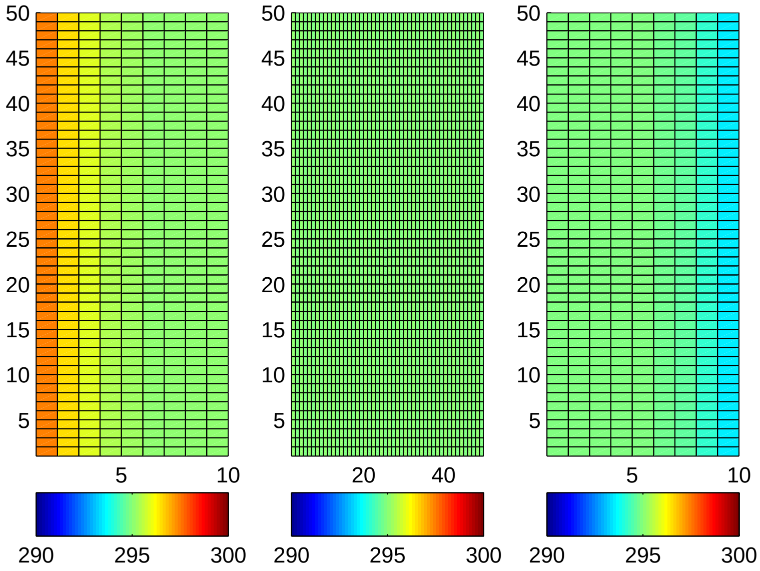


Figure B.15: Main model, start temperature distribution at time 0.3s

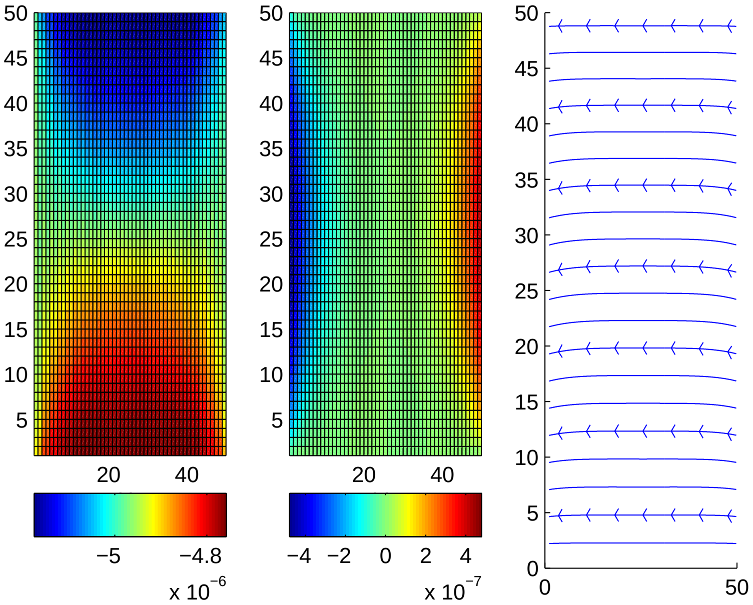


Figure B.16: Main model, start velocity field at time 0.3s

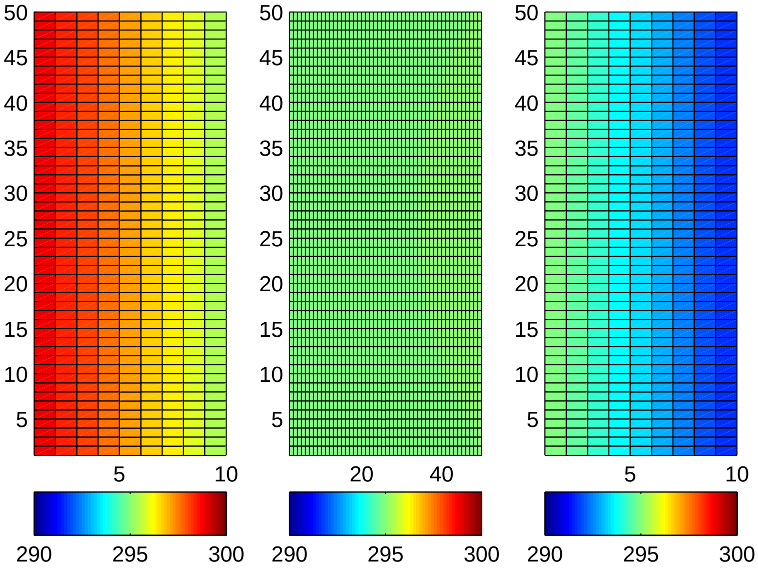


Figure B.17: Main model, final result for temperature distribution at time 171s

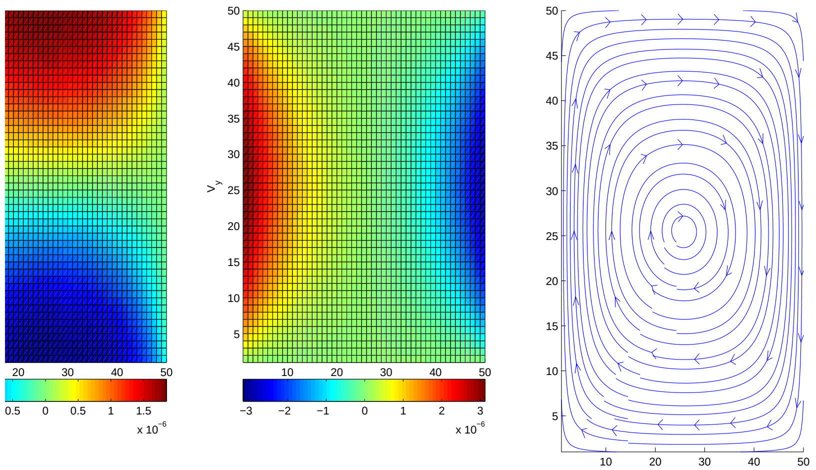


Figure B.18: Main model, final result for velocity field inside air space at time 171s

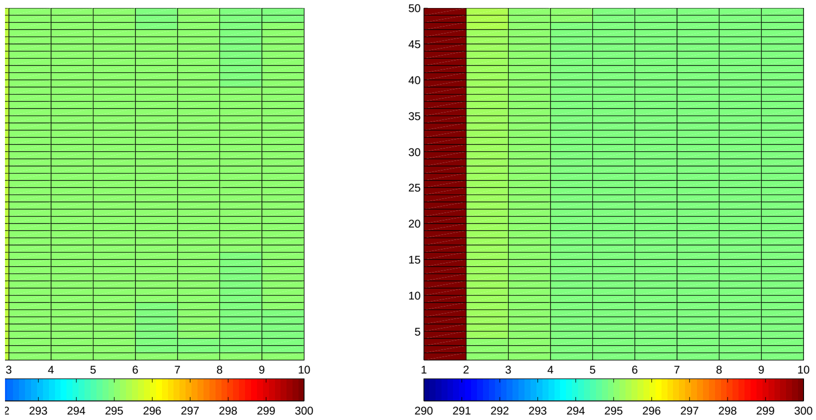


Figure B.19: Main model, comparison of result for different glass conductivity at time 41s, on left picture for conductivity $k = 3.4E-7$ is lower than on right $k = 3.4E-4$

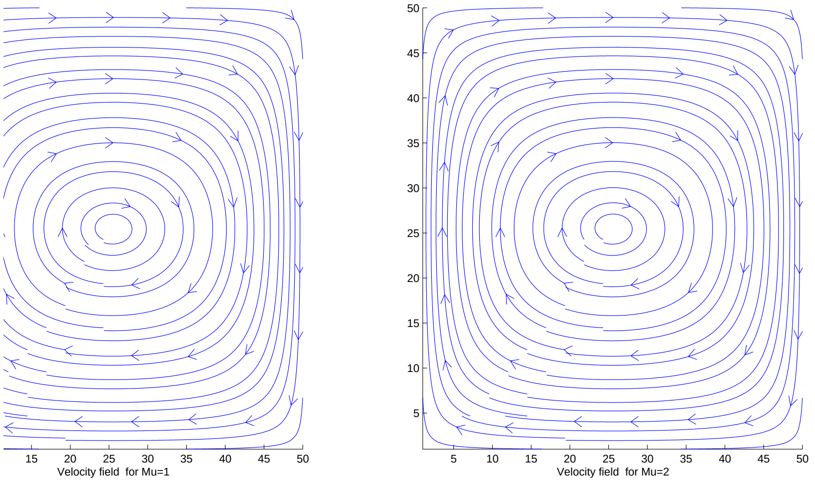


Figure B.20: Main model, comparison of velocity field result for different viscosity at time 41s, on left picture viscosity equal one and on right viscosity equal two

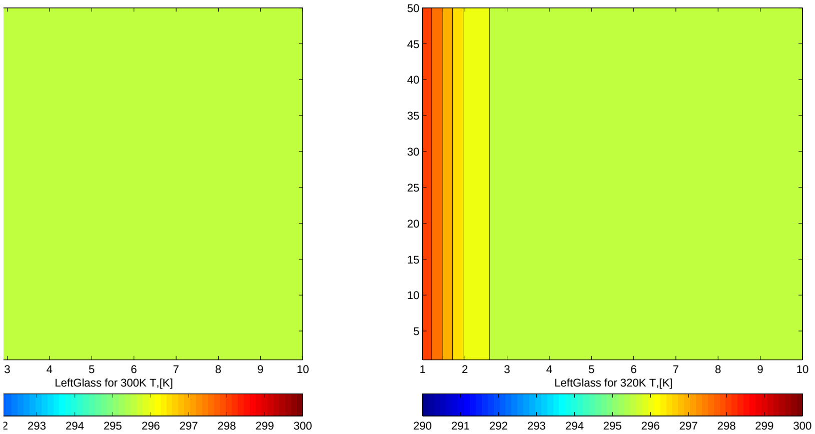


Figure B.21: Main model, comparison of temperature distribution inside glasses result for different temperature at left boundary at time 41s, on left picture outside temperature is $300K$ and on right outside temperature is $320K$

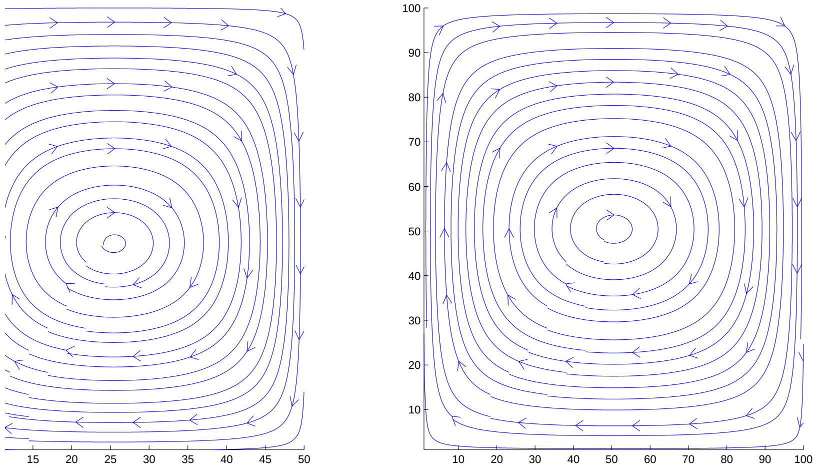


Figure B.22: Main model, comparison of vector field result for different mesh size at time 25s, on left picture outside temperature is mesh size 50 and on right is 100

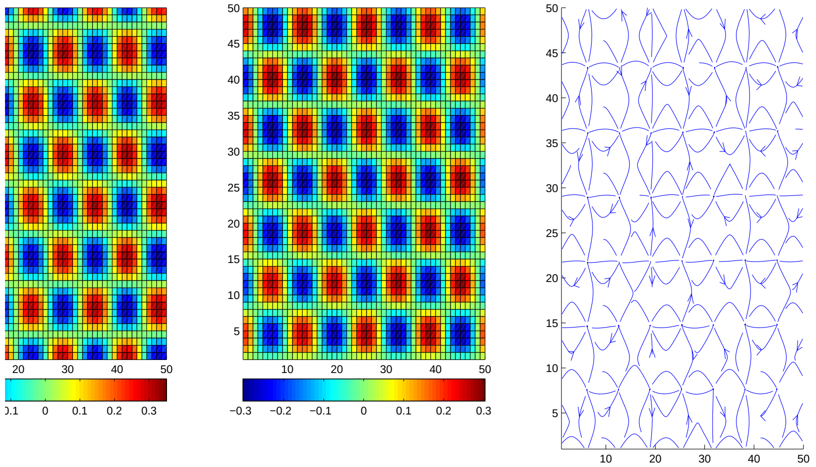


Figure B.23: Main model, result for velocity field inside air space for insufficient viscosity