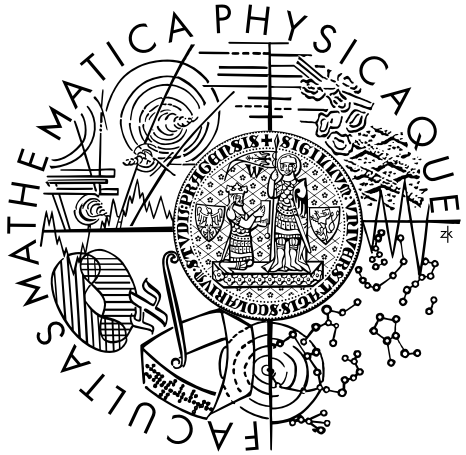
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Martin Krivoš

Numerické modelování proudění vody v nitru ledových těles

Katedra geofyziky

Vedoucí bakalářské práce: prof. RNDr. Ondřej Čadek, CSc.

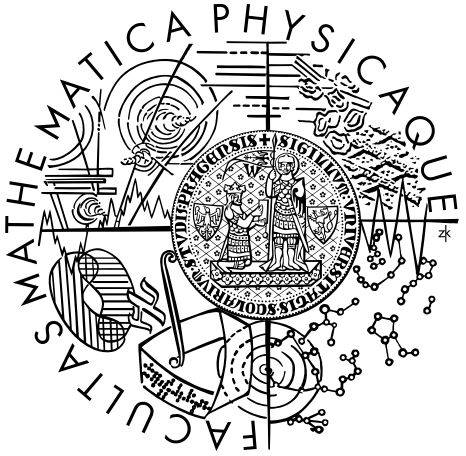
Studijní program: Fyzika

Studijní obor: Obecná Fyzika

Praha 2016

Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS



Martin Krivoš

Numerical modeling of liquid water flows in ice bodies' interiors

Department of Geophysics

Supervisor of the bachelor thesis: prof. RNDr. Ondřej Čadek, CSc.
Study programme: Physics
Study branch: General Physics

Prague 2016

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ..... signature of the author

Title: Numerical modeling of liquid water flows in ice bodies' interiors

Author: Martin Krivoš

Department: Department of Geophysics

Supervisor: prof. RNDr. Ondřej Čadek, CSc., Department of Geophysics

Abstract: We studied the flow induced by water jets in the subsurface oceans on the Solar system moons - Europa and Enceladus. In water plumes of Enceladus Cassini spacecraft detected small silica particles with radii $\approx 6 - 9$ nm. As shown by experiments, these particles grow in size with time spent in the ocean. The small size of particles suggests that the material transport from the the jets on the oceanic floor to the source of the plume at the moon's surface is highly efficient. In the thesis we investigate the characteristic transport time by solving the Navier-Stokes equation for incompressible fluid. For this purpose we have developed a Fortran program in two-dimensional Cartesian geometry based on the finite-difference staggered-grid method. Another program, using the second order Runge-Kutta method, was written to reconstruct the trajectories of the particles in the ocean. Using these tools we estimated the effectiveness of material transport under different conditions, namely presence of global lateral flow, width of the water jet, the Reynolds number and the number of jets.

Keywords: Navier-Stokes equation, numerical model, icy satellites

Název práce: Numerické modelování proudění vody
v nitru ledových těles

Autor: Martin Krivoš

Katedra: Katedra geofyziky

Vedoucí bakalářské práce: prof. RNDr. Ondřej Čadek, CSc., Katedra geofyziky

Abstrakt: Študovali sme tečenie v podpovrchovom ľadovom oceáne poháňané vodnými tryskami na mesiacoch Slnečnej sústavy - Európa a Enceládus. Vo vodných plumách Enceláda detekovala sonda Cassini malé silikátové častice s polomerom $\approx 6 - 9$ nm. Experimentmi bolo ukázané, že tieto častice rastú s časom stráveným v oceáne. Malé rozmery častíc teda naznačujú, že materiálový transport od trysiek na dne oceána k vodným plumám v ľadovej slupke je veľmi efektívny. V tejto práci skúmame charakteristické transportné časy riešením Navier-Stokesovej rovnice pre nestlačiteľnú kvapalinu. Pre tieto účely sme vyvinuli program v jazyku Fortran v dvojdimenzionálnej kartézskej geometrii založený na metóde konečných krokov v sieti staggered-grid. Ďalší program, užívajúci metodu Runge-Kutta druhého rádu, bol vyvinutý na rekonštrukciu trajektórií častíc v oceáne. S týmito pomôckami sme odhadli efektívnosť transportu pri rôznych podmienkach, ako je prítomnosť postranného globálneho toku, šírka vodnej trysky, Reynoldsové číslo a počet trysiek.

Klíčová slova: Navier-Stokesova rovnice, numerický model, ledové měsíce

I would like to thank my supervisor, prof. RNDr. Ondřej Čadek, CSc. for his endless support, patience and inspiration during my work.

Contents

Introduction

Since the beginning of time, people have been searching for extraterrestrial life. Water, as one of the main ingredients attracted attention of many scientists. Hence objects containing this unique chemical compound are of a big interest even today. Naturally, bodies in our Solar system are the easiest to observe. In this work we will focus especially on the Saturn's moon Enceladus and Jupiter's moon Europa. On these, we have not only confirmed the mere presence of the water, but discovered whole subsurface oceans (Dougherty et al., 2009).

In four most well-known Jupiters moons - so called Galilean moons, named after Galileo Galilei, Europa is the smallest one. Its surface is cowered with ice crust, with little to no atmosphere containing even oxygen. During the Galileo spacecraft flyby, the measurements of the magnetic field suggest a salty ocean beneath the surface (Pappalardo et al. 2009). Future missions to Europa are in progress to confirm this hypothesis. Surface of Europa is shown in Fig. 1.

Enceladus, discovered in 1789 by Willian Herschel, is natural satellite of planet Saturn, orbiting in its E Ring. The Voyager spacecrafts on their flybys revealed ice crust covering the whole surface with a possibility of large water reservoir under it near the south pole. Later, in 2015 Cassini spacecraft gravitational experiments confirmed the presence of the global ocean, separating ice crust and its core (Thomas et al., 2015). On top of that, water plumes, emitting water particles, were discovered near the region of the south pole - Fig. 1 - right. Composition studies of these plumes revealed a presence of small silica particles (mainly SiO_2), with radii not larger than $\approx 6 - 9$ nm. This limited size has a big impact on our understanding about the ongoing processes in the global ocean. On Earth reconstruction of the ocean environment shows, that these silica particles grow constantly in size with time and cannot sustain in the ocean longer than few months without exceeding the measured size as shown by Hsu et al. (2015). Theory supports a model, where a water jet on the ocean-core interface drives a transport of these silica particles to the water plumes in the ice crust. The possibility of such process is the study of this thesis.

In the following work, we will use a mathematical model to simulate a situation, where a water jet is present at the oceanic floor. The object is to prove, that such situation may lead to stable and fast enough transport of silica particles from the oceans bottom to the plumes. Flow of the subsurface oceans is described by the Navier-Stokes equation. Because solutions in three dimensions are numerically difficult, we reduce our problem to only two dimensions.

In this model, we prescribe water jets on the bottom boundary and we will monitor the speed and effectiveness of the transport in the fluid based on changes in different parameters like the Reynolds number and width of the jet. We believe that this experiment may contribute to understanding of this process.

The structure is as follows: The first chapter involves mathematical model and definition of the Navier-Stokes equation. In second chapter we discretize these equations and make numerical integration. The third chapter shows the results of our simulations and their discussion. In the final section we summarise the conclusion.

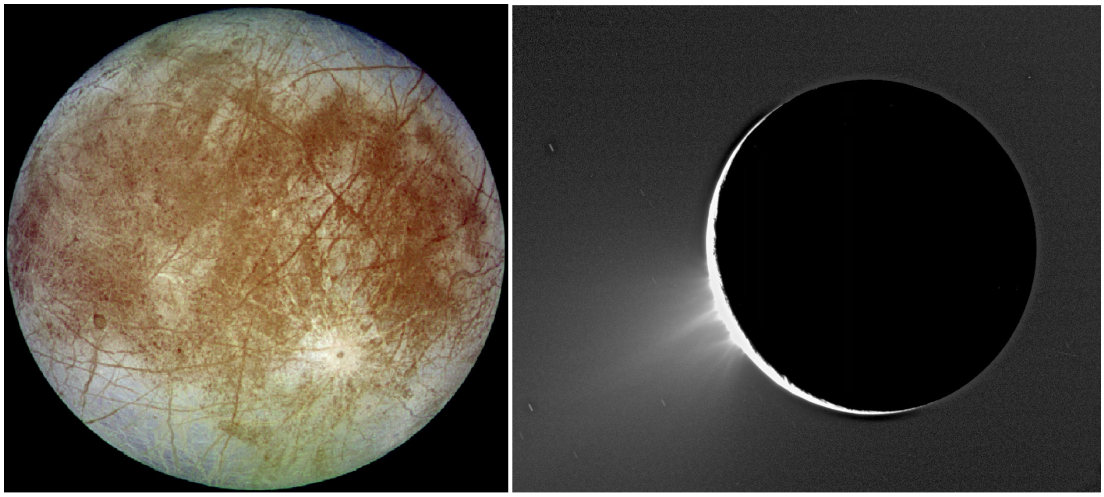


Figure 1: Left: Europa's smooth icy surface with reddish-brown patterns (of unknown composition) as seen by Galileo spacefraft during its second orbit around Jupiter in 1996 at range of 677,000 kilometers. Right: Water plumes ejecting from south polar region of Enceladus, backlit by the sun, as taken by Cassini spacefraft in November 2005. Pictures from JPL/NASA archives.

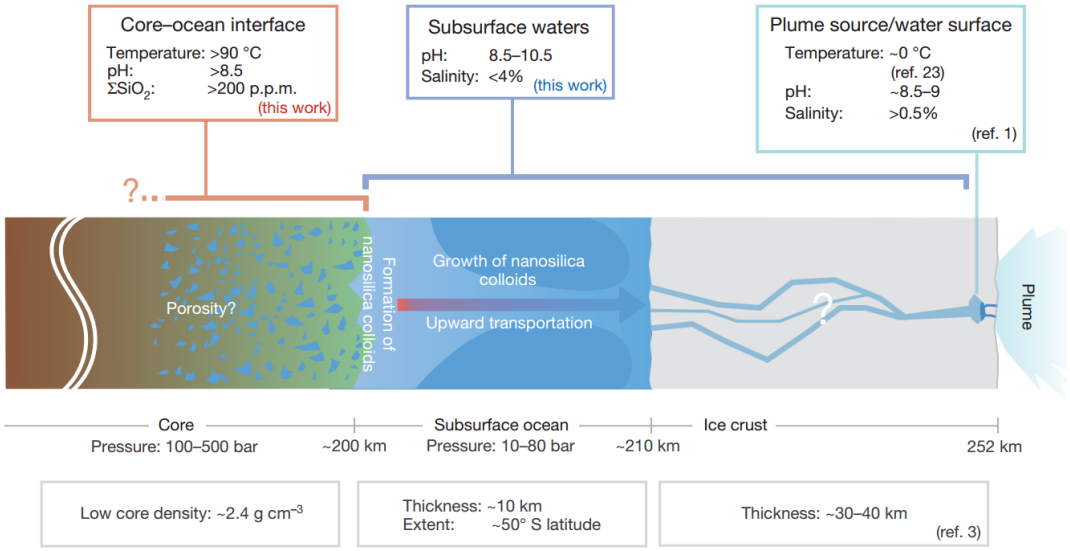


Figure 2: Sketch from Hsu et al. (2015) showing the inner structure of the Enceladus. Middle (blue) area representing ocean will be of interest in this work.

1. Mathematical model

To simulate the flow of water inside Enceladus's internal ocean, we approximated the region under consideration by a two-dimensional rectangular box and developed a Fortran program to calculate the flow in the domain induced by jets at the oceanic floor. The flow of the fluid is modelled using the Navier-Stokes equation:

$$-\nabla p + \eta\nabla^2\vec{v} + \rho\vec{f} = \rho\vec{v}\,\nabla\cdot\vec{v} + \rho\frac{\partial\vec{v}}{\partial t}, \tag{1.1}$$

where p is pressure, $\vec{v}$ is velocity, f is external force, t is time, ρ is density and η is viscosity. The continuity equation for an incompressible fluid is:

$$\nabla\cdot\vec{v} = 0. \tag{1.2}$$

1.1 Dimensionless Navier-Stokes equation and Reynolds number

The solutions of the Navier-Stokes equation depends on the combination of parameters ρ, v, L and η, also known as the *Reynolds number*:

$$Re = \frac{\rho v L}{\eta}, \tag{1.3}$$

where L is a characteristic linear dimension. This means, that for instance, if we replace one fluid by another with a doubled viscosity and a doubled density, the character of the flow does not change provided that, the other parameters (v and L) are changed so that the Reynolds number is the same.

With this information it is useful to alter eq. (1.1), to the final form with only one free parameter – the Reynolds number. For this purpose we define characteristic size D, characteristic velocity V and dimensionless spatial coordinates x_i' and temporal dimensionless time t':

$$x_i = Dx_i'; \qquad t = \frac{D}{V}t'. \tag{1.4}$$

The derivatives with respect to the dimensionless variables are then defined as:

$$\frac{\partial}{\partial x_i'} = D\frac{\partial}{\partial x_i}; \qquad \frac{\partial}{\partial t'} = \frac{D}{V}\frac{\partial}{\partial t}, \tag{1.5}$$

and the Navier-Stokes equation 1.1 reads

$$-\frac{1}{D}\nabla' p + \frac{\eta}{D^2}\nabla'^2\vec{v} + \rho\vec{f} = \frac{\rho}{D}\vec{v}\,\nabla'\vec{v} + \frac{\rho V}{D}\frac{\partial\vec{v}}{\partial t'}, \tag{1.6}$$

where the prime in ∇' means diferentiation with respect to dimensionless variables. By multipliing eq. (1.6) by factor $\frac{D}{\rho V^2}$, we get

$$-\nabla'\frac{p}{\rho V^2} + \frac{\eta}{\rho D V}\nabla'^2\frac{\vec{v}}{V} + \frac{D\vec{f}}{V^2} = \frac{\partial}{\partial t'}\frac{\vec{v}}{V} + \frac{\vec{v}}{V}\nabla'\frac{\vec{v}}{V}, \tag{1.7}$$

which can be expressed in terms of Reynolds number (1.3) as follows:

$$-\nabla' p' + \frac{1}{Re}\nabla'^2 \vec{v'} + \vec{f'} = \vec{v'}\,\nabla' \vec{v'} + \frac{\partial \vec{v'}}{\partial t'}, \tag{1.8}$$

where $p' = \frac{p}{\rho V^2}$. Removing the primes we obtain:

$$-\nabla p + \frac{1}{Re}\nabla^2 \vec{v} + \vec{f} = \vec{v}\,\nabla \vec{v} + \frac{\partial \vec{v}}{\partial t}. \tag{1.9}$$

Equation (1.9) is the final form of Navier-Stokes equation which we will use in the following sections.

1.2 Two-dimensional rectangular area and boundary conditions

Because of time demanding computations and available computer memory, we have simplified the problem by restricting our model into two dimensions only. As the depth of Encaladus's ocean is much smaller compared to its horizontal extent, we have chosen a rectangular domain as it best fits the real situation and, at the same time it is computationally feasible. The top boundary in our model represents the base of the icy crust while the bottom boundary mimics the ocean floor. The conditions considered on the boundaries are illustrated in Fig. 1.1. We use the Dirichlet boundary condition at the top and bottom boundary which corresponds to liquid-solid interface. At the side boundaries we either consider no-slip or free-slip.

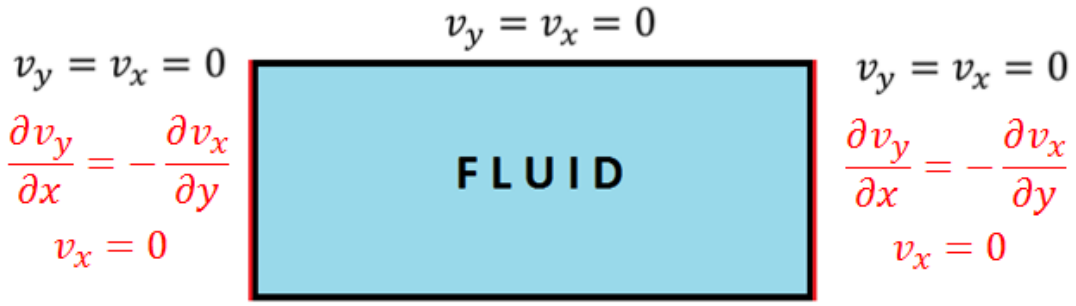


Figure 1.1: Boundary conditions. Black colour represents no-slip condition and red colour is free-slip variation. Bottom boundary is shown in Fig. 1.2.

Bottom boundary - the water jet

With Dirichlet condition on the bottom boundary we are able to prescribe the velocities of the water jet. Two types of jets which are shown in Fig. 1.2 will be used. They will always be located in the middle of the computational domain and will be accompanied symmetrically with two water drains with same size, but halved velocities (so that the continuity equation holds).

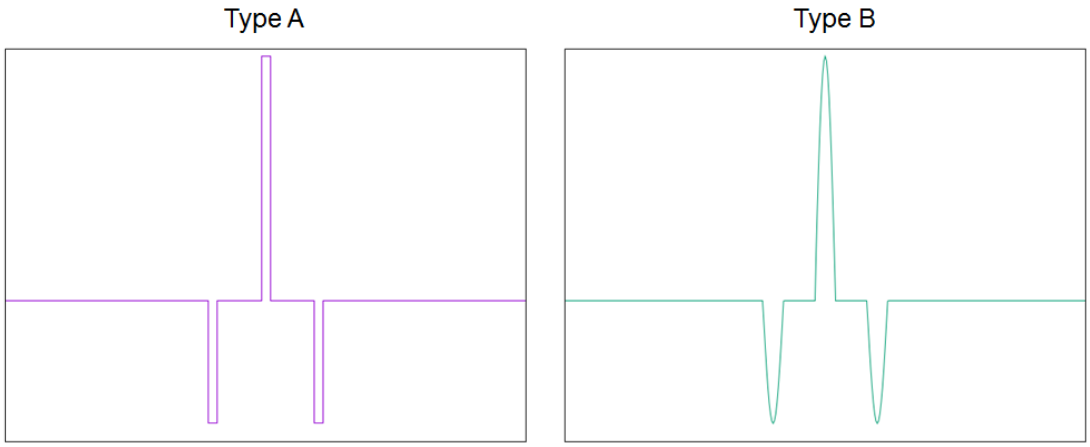


Figure 1.2: Velocities of the bottom boundary for jet type A and B. Type A: Water jet has dimensionless width $w = 1/150$ and velocity $v = 1$. Type B: Velocity has a sine-like character where width w is parameter that can be changed.

2. Numerical representation

Since the Navier-Stokes equation cannot be solved analytically, we need to reformulate eq. (1.9) with condition (1.2) into a numerically suitable format. In this chapter we show how to transform these equations using a finite-difference method in space, their numerical integration in time and that this formulation yields a numerically stable solution.

2.1 Staggered grid

For our purposes, we have chosen quite simple, but effective grid to mesh our rectangular computational domain.("Staggered grid", for more details see Thomas et al., 2015). The grid is shown in Fig. 2.1. Note, that there are four types of points (nods) all distinguished by colour and shape. Each type represents some quantity. The green and pink circles contain values of y- and x-component of the velocity, respectively. The orange squares contain pressure and the blue squares are left empty. To obtain desired value in different nod, we use arithmetic mean of closest values. For example: to obtain x-component of velocity in orange square, we use averaged value from the nod to the left and the nod to the right(in Fig. 2.1 denoted by red arrows). To obtain value of x-component of velocity in green circles, we use averaged value of four closest points (in Fig. 2.1 denoted by purple arrows).

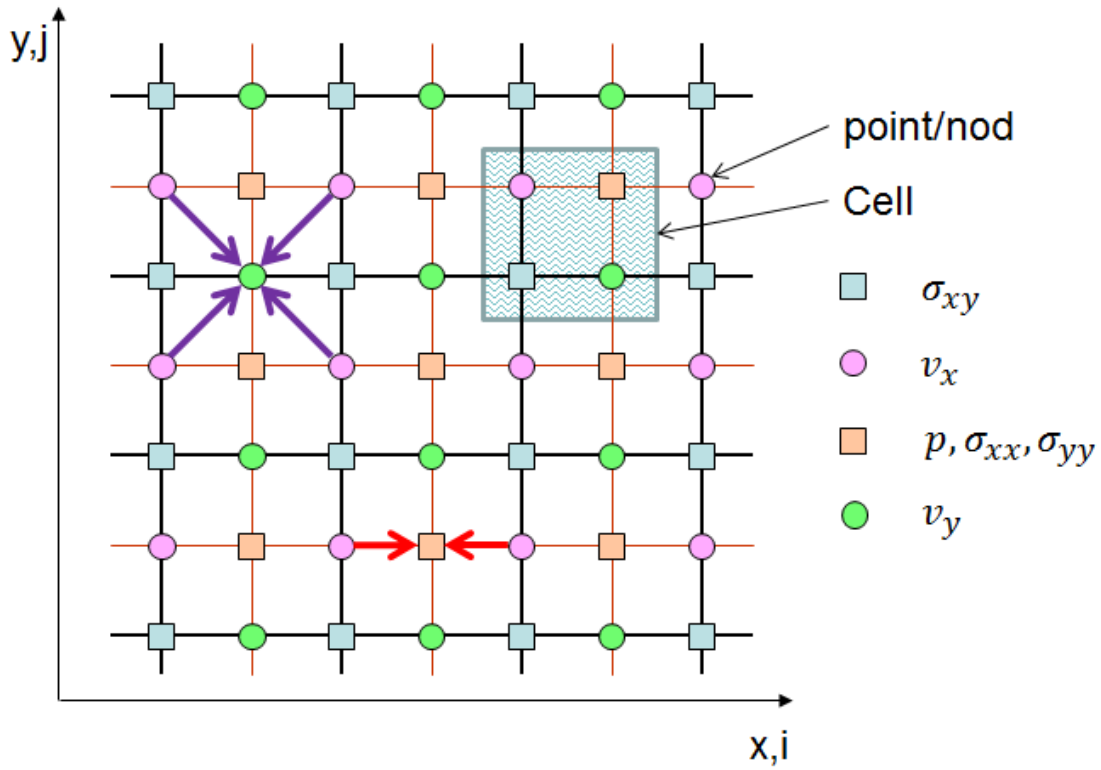


Figure 2.1: Staggered grid layout.

The set of all four symbols forms a cell (marked by a light blue square in Fig. 2.1). Each cell is defined by a pair of indexes (i,j) increasing along the x-

and y-axes. Cells are numbered in each direction starting from the lower left corner. The number of these cells defines the resolution of the grid. In numerical formalism, we abandon continuous axes x and y and we replace them by discrete axes i and j.

Each cell contains three unknowns and to obtain a unique solution, we need three equations for each cell. These are the Navier-Stokes equations for x- and y-component and the continuity equation. Boundary cells must be treated separately and we will discuss them later in section 2.2.2.

For further reading, it is important to note, that for numerical purpose, the Navier-Stokes equation (1.9) is expressed in terms of two first-order partial differential equations, namely the momentum equation

$$-\nabla p + \eta \nabla \cdot \boldsymbol{\sigma} + \rho \vec{f} = \rho \vec{v} \nabla \cdot \vec{v} + \rho \frac{\partial \vec{v}}{\partial t}, \tag{2.1}$$

where σ is the deviatoric part of the Cauchy stress tensor.

The constitutive equation for a newtonian incompressible fluid

$$\boldsymbol{\sigma} = \eta \left[\nabla \vec{v} + (\nabla \vec{v})^T \right], \tag{2.2}$$

Components σ_{xx} and σ_{yy} of this tensor are discretized in orange squares (together with pressure) while blue squares contain the values of unknown σ_{xy}. Since the stress components can be expressed in terms of velocities (see eqs. 2.6, 2.7 and 2.8), they are not considered as independent unknowns, but rather auxiliary variables facilitating the staggered grid implementation.

2.1.1 Analytical to numerical

The transformation of analytical equations into numerical formulas will be demonstrated in the following example where we will calculate partial derivative of the y-component of the velocity v_y with respect to x - see Fig. 2.2. Here we use a first order differential scheme i.e. we take two values next to each other and divide their difference by their distance:

$$\frac{\partial v_y}{\partial x} = \frac{v^{i,j} - v^{i-1,j}}{\Delta x}. \tag{2.3}$$

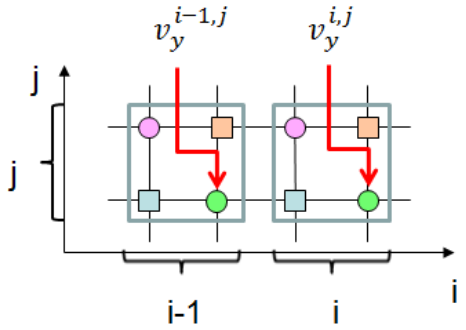


Figure 2.2: Numerical derivative example.

The other equations are expressed in an analogical manner.

2.2 Numerical discretization

The notation of variables and their derivatives corresponding to time step n and a grid cell (i,j) is shown in Fig. 2.3.

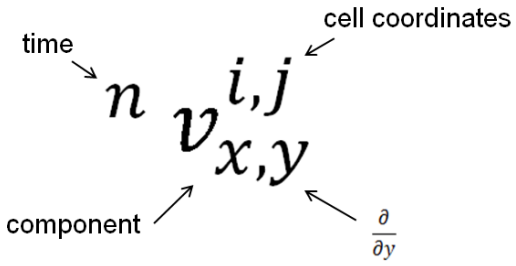


Figure 2.3: Indexes of quantities in numerical equations. In this figure partial derivative with respect to y of x-component of the velocity at time step n in cell coordinate (i,j) is shown.

2.2.1 Spatial equations

In this section we will show spatial discretization of equations. We omit the partial derivative $\frac{\partial \vec{v}}{\partial t}$ and its discretization will be discussed later in section 2.2.3.

Navier-Stokes equations

We express the momentum equation (2.1) in cartesian components:

$$-\frac{\partial p}{x}+\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xy}}{\partial y}+\rho f_x=\rho v_x\frac{\partial v_x}{\partial x}+\rho v_y\frac{\partial v_x}{\partial y}+\rho\frac{\partial v_x}{\partial t}, \tag{2.4}$$

$$-\frac{\partial p}{y}+\frac{\partial \sigma_{xy}}{\partial x}+\frac{\partial \sigma_{yy}}{\partial y}+\rho f_y=\rho v_x\frac{\partial v_y}{\partial x}+\rho v_y\frac{\partial v_y}{\partial y}+\rho\frac{\partial v_x}{\partial t}, \tag{2.5}$$

where we have used:

$$\sigma_{xx}=\frac{2}{Re}\frac{\partial v_x}{\partial x}, \tag{2.6}$$

$$\sigma_{yy}=\frac{2}{Re}\frac{\partial v_y}{\partial y}, \tag{2.7}$$

$$\sigma_{xy}=\frac{1}{Re}\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right). \tag{2.8}$$

In our application, no external forces are present and the flow is only driven by the influx prescribed at the boundary. Hence we set $f_x = f_y = 0$ in eqs. (2.4) and (2.5). In addition, cells in our grid will be uniform and square-shaped, meaning that $\Delta x = \Delta y = const.$, where this constant will be chosen from size of the domain and number of cells.

The discretization of eqs. (2.4) and (2.5) on a staggered grid (section 2.1.1) leads to the following algebraic equations:

$$-\frac{p^{i,j}-p^{i-1,j}}{\Delta x}+\frac{\sigma_{xx}^{i,j}-\sigma_{xx}^{i-1,j}}{\Delta x}+\frac{\sigma_{xy}^{i,j+1}-\sigma_{xy}^{i,j}}{\Delta y}=v_x\frac{v_x^{i+1,j}-v_x^{i-1,j}}{2\Delta x}+v_y\frac{v_x^{i,j+1}-v_x^{i,j-1}}{2\Delta y}, \tag{2.9}$$

$$-\frac{p^{i,j}-p^{i,j-1}}{\Delta y}+\frac{\sigma_{yy}^{i,j}-\sigma_{yy}^{i,j-1}}{\Delta y}+\frac{\sigma_{xy}^{i+1,j}-\sigma_{xy}^{i,j}}{\Delta x}=v_x\frac{v_y^{i+1,j}-v_y^{i-1,j}}{2\Delta x}+v_y\frac{v_y^{i,j+1}-v_y^{i,j-1}}{2\Delta y}. \tag{2.10}$$

Discretizing the formulas (2.6), (2.7) and (2.8), substituing them in (2.9) and (2.10) we finally obtain:

$$\begin{aligned}&\frac{1}{\Delta x}p^{i-1,j}-\frac{1}{\Delta x}p^{i,j}+\frac{2}{Re}\frac{1}{\Delta x^2}v_x^{i-1,j}-\frac{2}{Re}\left(\frac{2}{\Delta x^2}+\frac{1}{\Delta y^2}\right)v_x^{i,j}+\frac{2}{Re}\frac{1}{\Delta x^2}v_x^{i+1,j}+\\&+\frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j-1}+\frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j+1}-\frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i,j}+\frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i-1,j}-\\&-\frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i-1,j+1}+\frac{2}{Re}\frac{1}{\Delta x\Delta y}v_y^{i,j+1}=v_x\frac{v_x^{i+1,j}-v_x^{i-1,j}}{2\Delta x}+v_y\frac{v_x^{i,j+1}-v_x^{i,j-1}}{2\Delta y},\end{aligned} \tag{2.11}$$

$$\begin{aligned}&\frac{1}{\Delta y}p^{i,j-1}-\frac{1}{\Delta y}p^{i,j}+v_x^{i+1,j}\frac{1}{Re}\frac{1}{\Delta x\Delta y}-\frac{1}{Re}v_x^{i+1,j-1}\frac{1}{\Delta x\Delta y}+\frac{1}{Re}v_y^{i+1,j}\frac{1}{\Delta x^2}-\\&-\frac{2}{Re}v_y^{i,j}\left(\frac{1}{\Delta x^2}+\frac{2}{\Delta y^2}\right)-v_x^{i,j}\frac{1}{Re}\frac{1}{\Delta x\Delta y}+\frac{2}{Re}\frac{1}{\Delta x\Delta y}v_x^{i,j-1}+\frac{1}{Re}\frac{1}{\Delta x^2}v_y^{i-1,j}+\\&+\frac{2}{Re}\frac{1}{\Delta y^2}v_y^{i,j+1}+\frac{2}{Re}\frac{1}{\Delta y^2}v_y^{i,j-1}=v_x\frac{v_y^{i+1,j}-v_y^{i-1,j}}{2\Delta x}+v_y\frac{v_y^{i,j+1}-v_y^{i,j-1}}{2\Delta y}.\end{aligned} \tag{2.12}$$

Continuity equation

Compared to the Navier-Stokes equation, the continuity equation (1.2) is simple, because it contains only velocities and that is why its discretization is straightforward. By using the same formal steps as above, we get:

$$\frac{v_x^{i+1,j}-v_x^{i,j}}{\Delta x}+\frac{v_y^{i,j+1}-v_y^{i,j}}{\Delta y}=0. \tag{2.13}$$

2.2.2 Points at the border and boundary conditions

In Fig. 2.4, the blueish area marks the computational domain and the thick black lines denote its boundaries. In order to obtain a complete set of numerical equations over the computational domain, we also need nodes outside the domain. These nodes are required to express the boundary conditions and the spatial derivatives in the partial differential equations at the boundaries. Since we want to keep the homogeneous structure of the cells everywhere (both inside and outside the domain), we will also define the nodes which are not needed for diretization of equations, but they allow the cell structure to be uniform. The values

in these nodes can be chosen arbitrarily and associated with the formal equation which is not linked to the rest of the equations. These nodes are crossed in Fig. 2.4. and labelled as formal equations.

We will first discuss the no-slip boundary condition, which requires that $v_x = v_y = 0$. If the node with the appropriate velocity component lays directly on the border, we apply the boundary condition of type 1.(see Fig. 2.4). If it is not the case, we prescribe at the boundary the average value obtained from velocity components outside and inside the domain. This is referred as the boundary condition of type 2 in Fig. 2.4.

The free-slip boundary condition is evaluated in analogical manner(is not shown in Fig. 2.4).

The Navier-Stokes equations (2.11) and (2.12) have to be slightly altered at the boundaries. Recalling equations (2.6) and (2.7), we can see that if some of the stress components must be expressed outside the border, we would require another row/column of cells outside the computational domain. To avoid this, we define a new variable $\tilde{p}$ that represents the traction force at the boundary of the domain. It is a sum of p and σ_{xx} or σ_{yy} (which one is present). We abandoned overall uniformity of the cells. Some orange squares will represent $\tilde{p}$ instead of p (these are denoted with horizontal strip in Fig. 2.4.

The discretized Navier-Stokes equations at the boundaries can be generally expressed as follows:

$$\begin{aligned}\frac{1}{\Delta x}p^{i-1,j} - \frac{1}{\Delta x}p^{i,j} + \frac{\mathbf{2}}{\boldsymbol{Re}}\frac{\mathbf{1}}{\boldsymbol{\Delta x^2}}\boldsymbol{v_x^{i-1,j}} - \frac{2}{Re}\left(\frac{\mathbf{2}}{\Delta x^2} + \frac{1}{\Delta y^2}\right)v_x^{i,j} + \frac{\mathbf{2}}{\boldsymbol{Re}}\frac{\mathbf{1}}{\boldsymbol{\Delta x^2}}\boldsymbol{v_x^{i+1,j}} + \\ + \frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j-1} + \frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j+1} - \frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i,j} + \frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i-1,j} - \\ - \frac{1}{Re}\frac{1}{\Delta x\Delta y}v_y^{i-1,j+1} + \frac{2}{Re}\frac{1}{\Delta x\Delta y}v_y^{i,j+1} = v_x\frac{v_x^{i+1,j} - v_x^{i-1,j}}{2\Delta x} + v_y\frac{v_x^{i,j+1} - v_x^{i,j-1}}{2\Delta y},\end{aligned} \tag{2.14}$$

$$\begin{aligned}\frac{1}{\Delta y}p^{i,j-1} - \frac{1}{\Delta y}p^{i,j} + v_x^{i+1,j}\frac{1}{Re}\frac{1}{\Delta x\Delta y} - \frac{1}{Re}v_x^{i+1,j-1}\frac{1}{\Delta x\Delta y} + \frac{1}{Re}v_y^{i+1,j} - \\ - \frac{1}{\Delta x^2}\frac{2}{Re}v_y^{i,j}\left(\frac{1}{\Delta x^2} + \frac{\mathbf{2}}{\Delta y^2}\right) - v_x^{i,j}\frac{1}{Re}\frac{1}{\Delta x\Delta y} + \frac{2}{Re}\frac{1}{\Delta x\Delta y}v_x^{i,j-1} + \frac{1}{Re}\frac{1}{\Delta x^2}v_y^{i-1,j} + \\ + \frac{\mathbf{2}}{\boldsymbol{Re}}\frac{\mathbf{1}}{\boldsymbol{\Delta y^2}}\boldsymbol{v_y^{i,j+1}} + \frac{\mathbf{2}}{\boldsymbol{Re}}\frac{\mathbf{1}}{\boldsymbol{\Delta y^2}}\boldsymbol{v_y^{i,j-1}} = v_x\frac{v_y^{i+1,j} - v_y^{i-1,j}}{2\Delta x} + v_y\frac{v_y^{i,j+1} - v_y^{i,j-1}}{2\Delta y}.\end{aligned} \tag{2.15}$$

Where the facultative terms (used only at specific boundaries) are marked in colours. The appropriate algebraic equations are obtained using the following recipe:

- Look at Fig. 2.4 and identify of which colour is the border where you want to write the equation.
- Drop the fraction typed in this colour in eqs. (2.14) or (2.15).
- In the equation where you dropped the fraction in the previous step, change the digit 2, typed in pink, to 1.

- According to Fig. 2.4 replace p with appropriate $\tilde{p}$.
- The equation you obtain following these steps is the Navier-Stokes equation, you will use at the chosen boundary (the other one is unchanged).

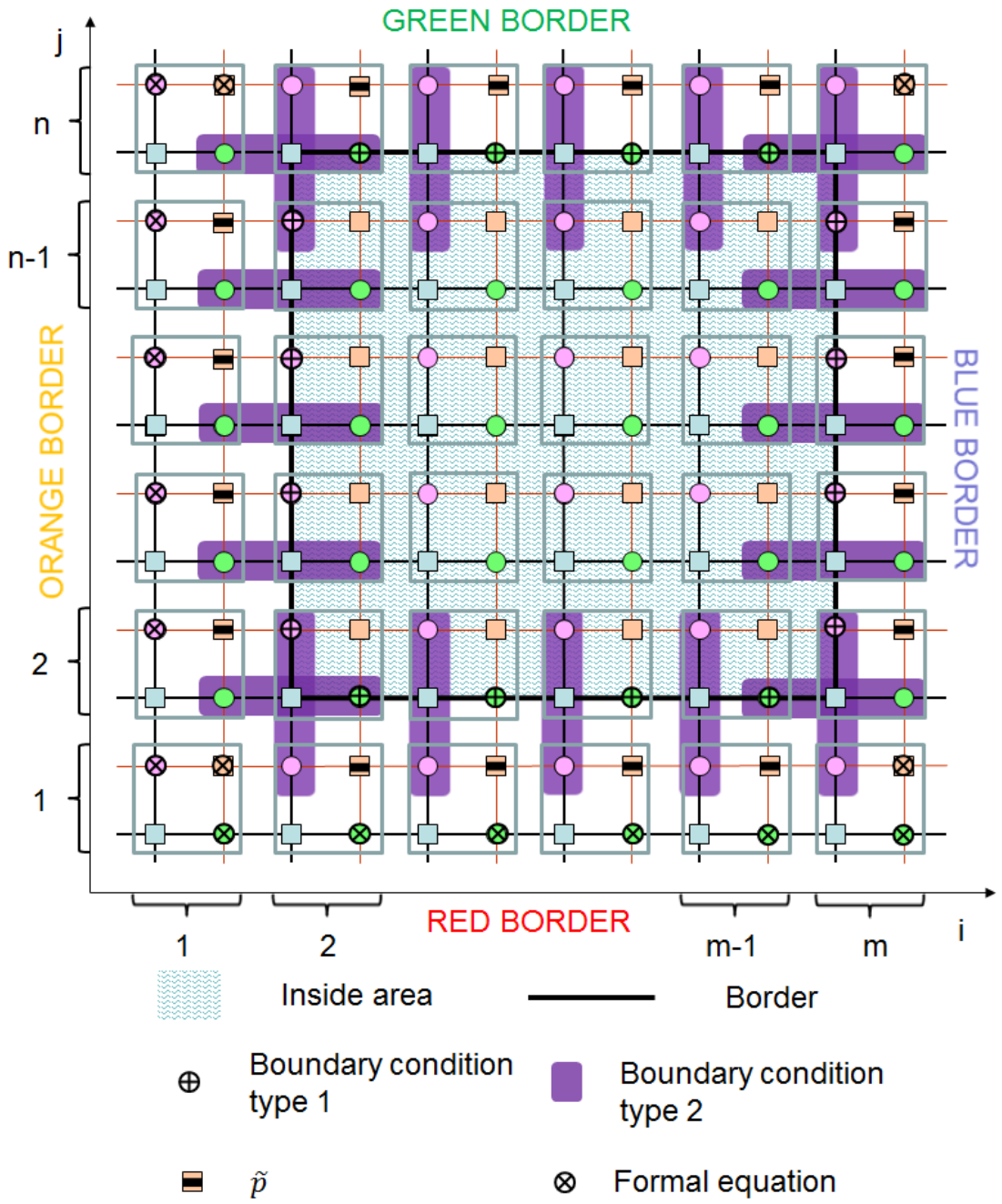


Figure 2.4: Scheme of boundary conditions for $m \times n$ area (for description see text).

It is important to note, that the cell with coordinates (2,2) is located on two boundaries and it will miss two fractions, unlike all the other points, that will miss only one.

2.2.3 Time equations

Navier-Stokes equations

After discretizing the spatial Navier-Stokes equation into eqs. (2.11) and (2.12), we add time derivative $\frac{\partial \vec{v}}{\partial t}$ which was omitted in previous step:

$$\frac{1}{\Delta x}p^{i-1,j} - \frac{1}{\Delta x}p^{i,j} + \frac{2}{Re}\frac{1}{\Delta x^2}v_x^{i-1,j} - \frac{2}{Re}\left(\frac{2}{\Delta x^2} + \frac{1}{\Delta y^2}\right)v_x^{i,j} + \frac{2}{Re}\frac{1}{\Delta x^2}v_x^{i+1,j} + \\ + \frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j-1} + \frac{1}{Re}\frac{1}{\Delta y^2}v_x^{i,j+1} - \frac{1}{Re}\frac{1}{\Delta x \Delta y}v_y^{i,j} + \frac{1}{Re}\frac{1}{\Delta x \Delta y}v_y^{i-1,j} - \\ - \frac{1}{Re}\frac{1}{\Delta x \Delta y}v_y^{i-1,j+1} + \frac{2}{Re}\frac{1}{\Delta x \Delta y}v_y^{i,j+1} = v_x\frac{v_x^{i+1,j} - v_x^{i-1,j}}{2\Delta x} + v_y\frac{v_x^{i,j+1} - v_x^{i,j-1}}{2\Delta y} + \\ + \frac{{}^{n}v_x^{i,j} - {}^{n-1}v_x^{i,j}}{\Delta t}, \quad (2.16)$$

$$\frac{1}{\Delta y}p^{i,j-1} - \frac{1}{\Delta y}p^{i,j} + v_x^{i+1,j}\frac{1}{Re}\frac{1}{\Delta x \Delta y} - \frac{1}{Re}v_x^{i+1,j-1}\frac{1}{\Delta x \Delta y} + \frac{1}{Re}v_y^{i+1,j}\frac{1}{\Delta x^2} - \\ - \frac{2}{Re}v_y^{i,j}\left(\frac{1}{\Delta x^2} + \frac{2}{\Delta y^2}\right) - v_x^{i,j}\frac{1}{Re}\frac{1}{\Delta x \Delta y} + \frac{2}{Re}\frac{1}{\Delta x \Delta y}v_x^{i,j-1} + \frac{1}{Re}\frac{1}{\Delta x^2}v_y^{i-1,j} + \\ + \frac{2}{Re}\frac{1}{\Delta y^2}v_y^{i,j+1} + \frac{2}{Re}\frac{1}{\Delta y^2}v_y^{i,j-1} = v_x\frac{v_y^{i+1,j} - v_y^{i-1,j}}{2\Delta x} + v_y\frac{v_y^{i,j+1} - v_y^{i,j-1}}{2\Delta y} + \\ + \frac{{}^{n}v_y^{i,j} - {}^{n-1}v_y^{i,j}}{\Delta t}. \quad (2.17)$$

Time step

Instead of setting one time step that will not change through the whole run of the simulation, we determine its best value after each individual step. We do this in the following way:

1. We find the nod with the maximum speed in our computational domain.
2. Based on already known size of cells $\Delta x = \Delta y$, we calculate time needed for this fastest particle to cross one cell:

$$t_{max} = \Delta x / v_{max}.$$

3. Now we reduce this amount by a chosen constant $c \in (0, 1)$:

$$t_{step} = c\ t_{max}.$$

Constant c in given interval ensures, that no particle travels more than one cell width during one time step. Value of $c = 0.5$ was chosen.

2.2.4 Numerical integration

In previous sections we have created equations that can be written into our Fortran program. In this section we will enhance them to improve their stability.

Linearisation of non-linear term $\vec{v}\nabla\cdot\vec{v}$

One of the greatest complexities in Navier-Stokes equation is the non-linear term – $\vec{v}\nabla\cdot\vec{v}$. When linearised, a linear solver could be used to find the solution.

One of the methods is shift in time for velocity:

$$ {}^{n}\vec{v}\nabla\cdot{}^{n}\vec{v} \rightarrow^{n-1} \vec{v}\nabla\cdot{}^{n}\vec{v}. \tag{2.18}$$

Equations (2.16) and (2.17) will change to (we do not show the other parts):

$$ {}^{n-1}v_x\frac{{}^{n}v_x^{i+1,j} -^{n} v_x^{i-1,j}}{2\Delta x} +^{n-1} v_y\frac{{}^{n}v_x^{i,j+1} -^{n} v_x^{i,j-1}}{2\Delta y}, \tag{2.19}$$

$$ {}^{n-1}v_x\frac{{}^{n}v_y^{i+1,j} -^{n} v_y^{i-1,j}}{2\Delta x} +^{n-1} v_y\frac{{}^{n}v_y^{i,j+1} -^{n} v_y^{i,j-1}}{2\Delta y}. \tag{2.20}$$

Upwind scheme

Upwind scheme is computational method for discretization of partial differential equations and is used in places, where propagation of information with distinct direction is present. In our case, information is flow of the fluid with its velocity $\vec{v} = (v_x, v_y)$ which will give us direction of propagation.
We define for x-component

$$ {}^{n-1}v_x^{+} = max({}^{n-1}v_x, 0); \qquad {}^{n-1}v_x^{-} = min({}^{n-1}v_x, 0), \tag{2.21}$$

and

$$ v_{x,x}^{-} = \frac{{}^{n}v_x^{i,j} -^{n} v_x^{i-1,j}}{\Delta x}; \qquad v_{x,x}^{+} = \frac{{}^{n}v_x^{i+i,j} -^{n} v_x^{i,j}}{\Delta x}, \tag{2.22}$$

$$ v_{x,y}^{-} = \frac{{}^{n}v_x^{i,j} -^{n} v_x^{i,j-1}}{\Delta y}; \qquad v_{x,y}^{+} = \frac{{}^{n}v_x^{i,j+1} -^{n} v_x^{i,j}}{\Delta y}. \tag{2.23}$$

For y-component the definitions are similar.
Applying above definitions to equations (2.19) and (2.20) we obtain:

$$ (2.19) \rightarrow^{n-1} v_x^{+}v_{x,x}^{-} +^{n-1} v_x^{-}v_{x,x}^{+} +^{n-1} v_y^{+}v_{x,y}^{-} +^{n-1} v_x^{-}v_{x,y}^{+}, \tag{2.24}$$

$$ (2.20) \rightarrow^{n-1} v_x^{+}v_{y,x}^{-} +^{n-1} v_x^{-}v_{y,x}^{+} +^{n-1} v_y^{+}v_{y,y}^{-} +^{n-1} v_x^{-}v_{y,y}^{+}. \tag{2.25}$$

It is important to note, that upwind scheme played a major role in our computer program. Without this scheme, our program faced many instabilities and usually ended with unreliable results. Upon applying this method, we have gained huge stability improvements.

Stabilization by including formal compressibility

In computer binary representation, not all decimal numbers can be expressed perfectly, therefore small numerical error is almost always present. The right hand side of the continuity equation (1.2) is equal to zero which gives very little freedom to rounding errors of the derivatives on the left hand side and program

may become unstable. To remove this behaviour, we add small compressibility to the fluid. The continuity equation will be:

$$\nabla \cdot \vec{v} + \varepsilon p = 0. \tag{2.26}$$

We will denote to ε as compressibility coefficient, because of its physical meaning. Our fluid will be considered as incompressible, because the coefficient ε is very small and has no visible physical effect.

We have had run few tests to estimate our optimal value of ε to be $\varepsilon = 10^{-5}$.

2.3 Solver

With set resolution $n \times m$, equations (2.16), (2.17) and (2.13) form a system of $N := 3 * m * n$ linear equations with N unknowns which we solve using already mentioned Fortran program. Using matrix formalism the system can be written by the following equation:

$$\boldsymbol{A} \cdot \vec{x} = \vec{b}, \tag{2.27}$$

where b is right hand side vector of dimension $1 \times N$, x is the vector of unknowns also of dimension $1 \times N$ and $\boldsymbol{A}$ is band matrix of dimension $N \times N$ with $KU = 3*n$ upper-diagonal and $KL = 3 * n + 1$ sub-diagonal elements. Each line in matrix $\boldsymbol{A}$ represents one equation and each column one variable. Procedure dgbsv from LAPACK library and Intel® Fortran Compiler were used to find the solution of (2.27).

2.4 Computation of particle trajectories and markers densities

We are mainly interested in the amount of particles propelled through the fluid, rather than overall flow. For this purpose we need to reconstruct trajectories of individual particles and focus on those, coming from the water jet. We have written new program in Python programming language which reconstructs trajectories, based on results of our Navier-Stokes program, using Runge-Kutta second order method.

Rather then following each particle one by one, it is more useful to inject many and monitor how their density in the domain changes with time. To see if the transport of the silica-particles is effective, we picked two areas of interest which are shown in Fig. 2.5. We release testing particles (markers) in small segment 2 cells wide and 2 cells above the bottom boundary.

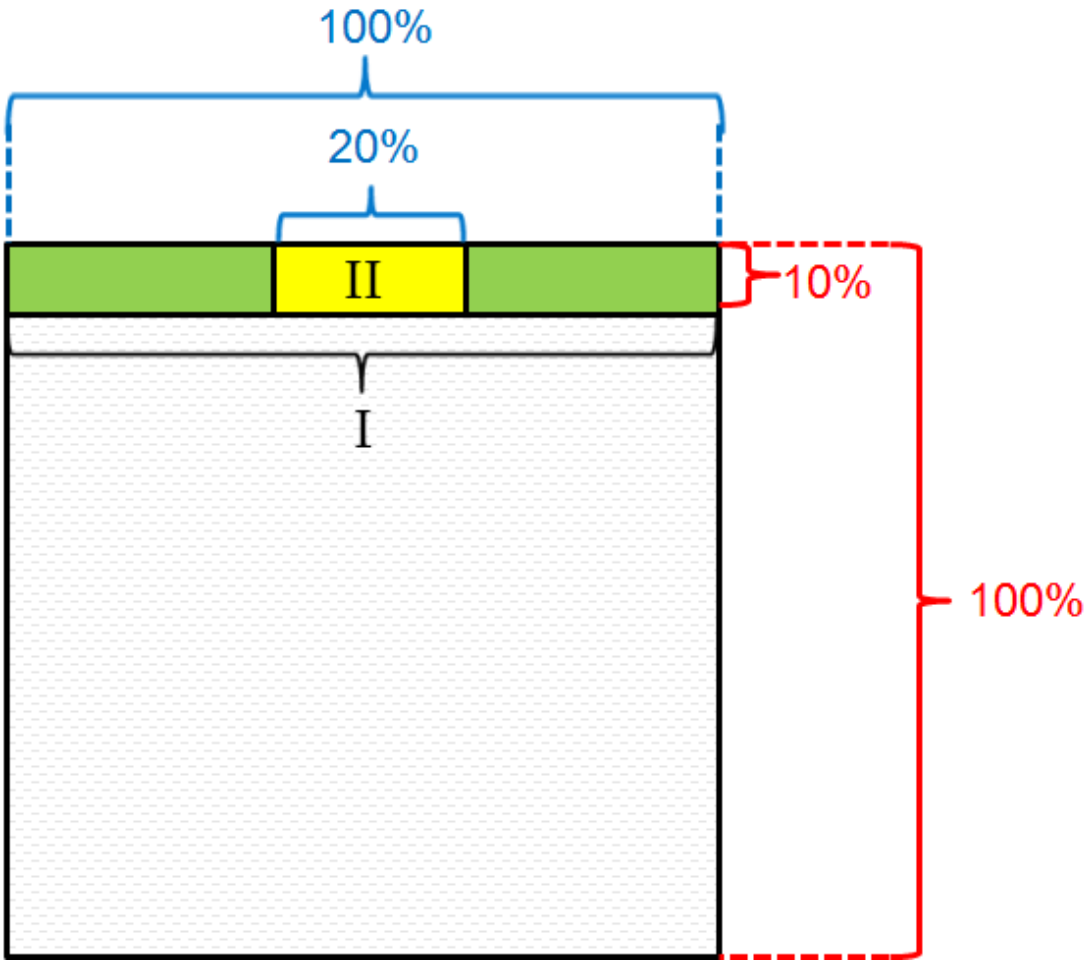


Figure 2.5: Scheme of areas of interest for densities of markers. Area I (green and yellow) is on tenth of high and spans through the whole width of the computational domain - this represents the whole area where plumes may origin in the ice crust. Area II (yellow) is also on tenth of hight, but width is only one fifth of the width and is located directly above the water jet in the middle.

3. Results and discussion

Our results are shown separately for bottom boundary of type A and B (see Fig. 1.2). The reader should be also familiar with Fig. 2.5.

3.1 Bottom boundary of type A

Our first simulations were in rectangular box (with dimensionless width equal to 4 and dimensionless height equal to 1), where we could monitor wide area of the ocean and it best approximated its height:width ratio. From Fig. 3.1 we see that used resolution 300×75 is not satisfactory. With higher resolution, computational time increase would be too demanding, hence we abandoned rectangular shape and focused purely on the flow in the middle using square computational domain with dimensionless height equal to width (=1).

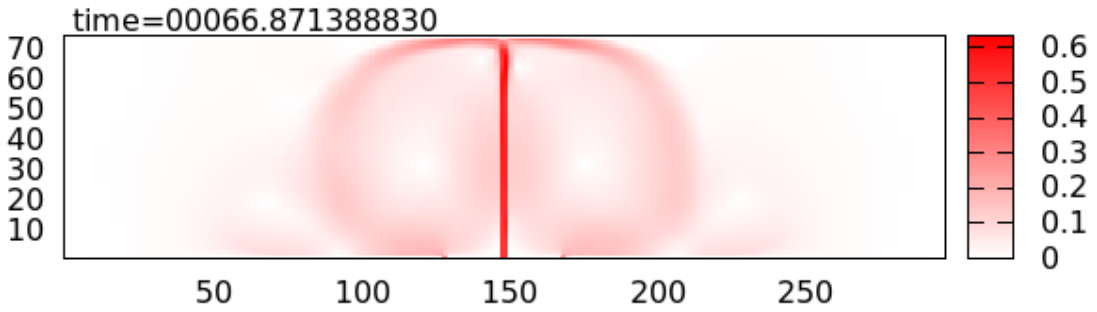


Figure 3.1: Snapshot of velocity from rectangular problem with resolution 300×75, Reynolds number $Re = 10^9$, no-slip on sides and water jet type A. As in all following snapshots, axes denote cell numbering. Considering the depth of the ocean to be $\approx (10 - 100)$ km, our water jet has $width \approx (130 - 1300)$ m which is much more than is expected. We have to increase the resolution to reduce this value.

As stated in section 1.1, character of the flow depends only on the Reynolds number. Fig. 3.2 demonstrates how L^2-norms change with the Reynolds number in square computational domain. In Fig. 3.3 and Fig. 3.4 velocities in several time steps are shown for $Re = 10^9$ and $Re = 10^4$ respectively. We have used our Python program to reconstruct trajectories of particles, where several of them are shown in Fig. 3.5. Of our primary interest is to demonstrate, that sufficient percentage of these particles reaches top border in adequate time. In Fig. 3.6 we show how concentration of 1000 particles propel thought the fluid.

In real situation, several water jets, one next to another is more likely to appear. Therefore we added 2 water jets of type A assymetrically around our main jet from the previous problem. Snapshots of velocities are shown is Fig. 3.7 and concentration of markers in Fig. 3.8.

Up till now, we considered our ocean to be stationary, without any global effects. It is very likely, that slap forces from Saturn create some global lateral flow in the ocean. To simulate this, we prescribed 100-times and 50-times smaller (compared to main jet) parabolic flow in left-to-right direction. Fig. 3.9

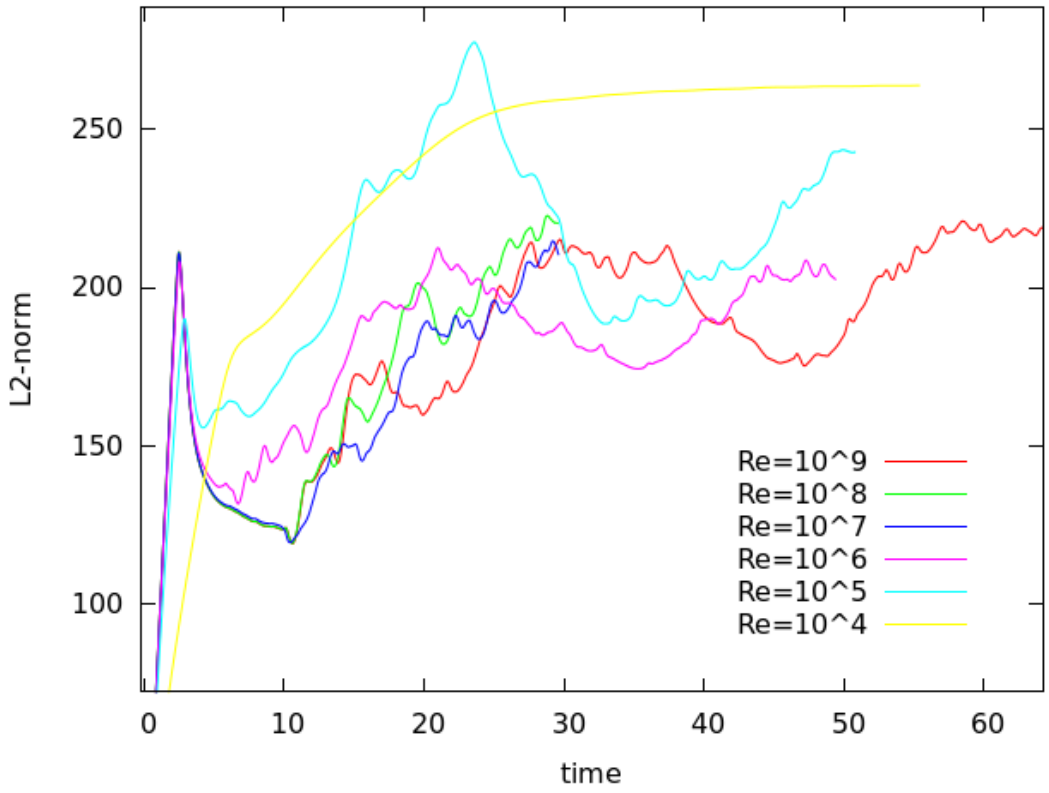


Figure 3.2: L^2-norms for different values of the Reynolds number in square box with resolution 150×150, no-slip boundary condition on sides and A type water jet. We can see, that for higher values of the Reynolds number $Re \geq 10^4$ many maxima/minima occur. This behaviour is caused by turbulences. L^2-norm for smaller Reynolds numbers like 10^4(yellow) is very smooth which corresponds with laminar flow.

shows snapshots of velocities from these simulations. In Fig. 3.10 densities of 100 markers for Area of type I and II are shown.

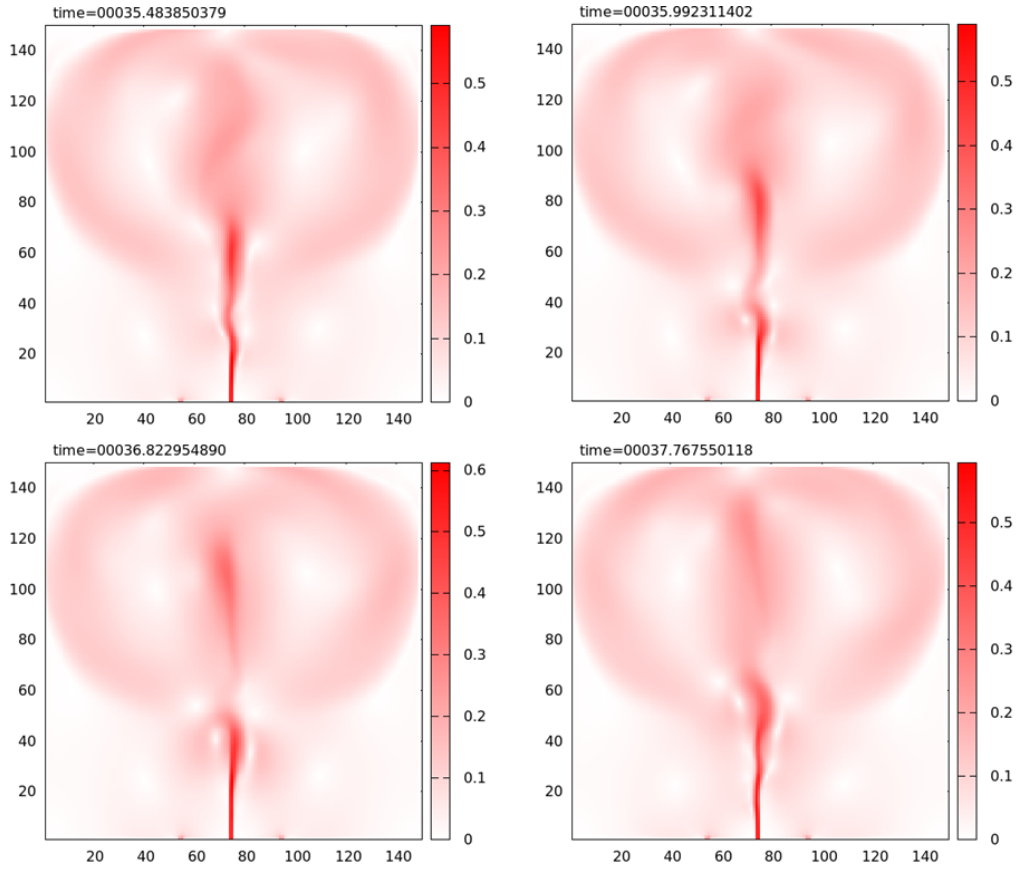


Figure 3.3: Four snapshots of velocities in square computational domain with resolution 150×150, no-slip boundary conditions on sides, water jet of type A and Reynolds number $Re = 10^9$. This figure demonstrates the turbulent flow in our simulation. In this figure a dissipation of continuous stream via turbulences can be demonstrated.

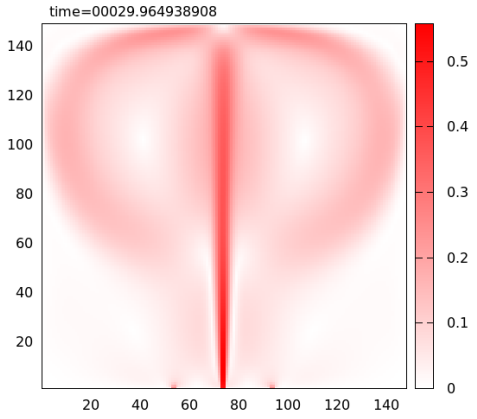


Figure 3.4: Same situation as Fig. 3.3 with $Re = 10^4$. As can be seen in Fig. 3.2, the L^2 norm of this problem converges without any oscillations which correspondes with laminar flow. In this figure only the snapshot of the final stabilized state is displayed.

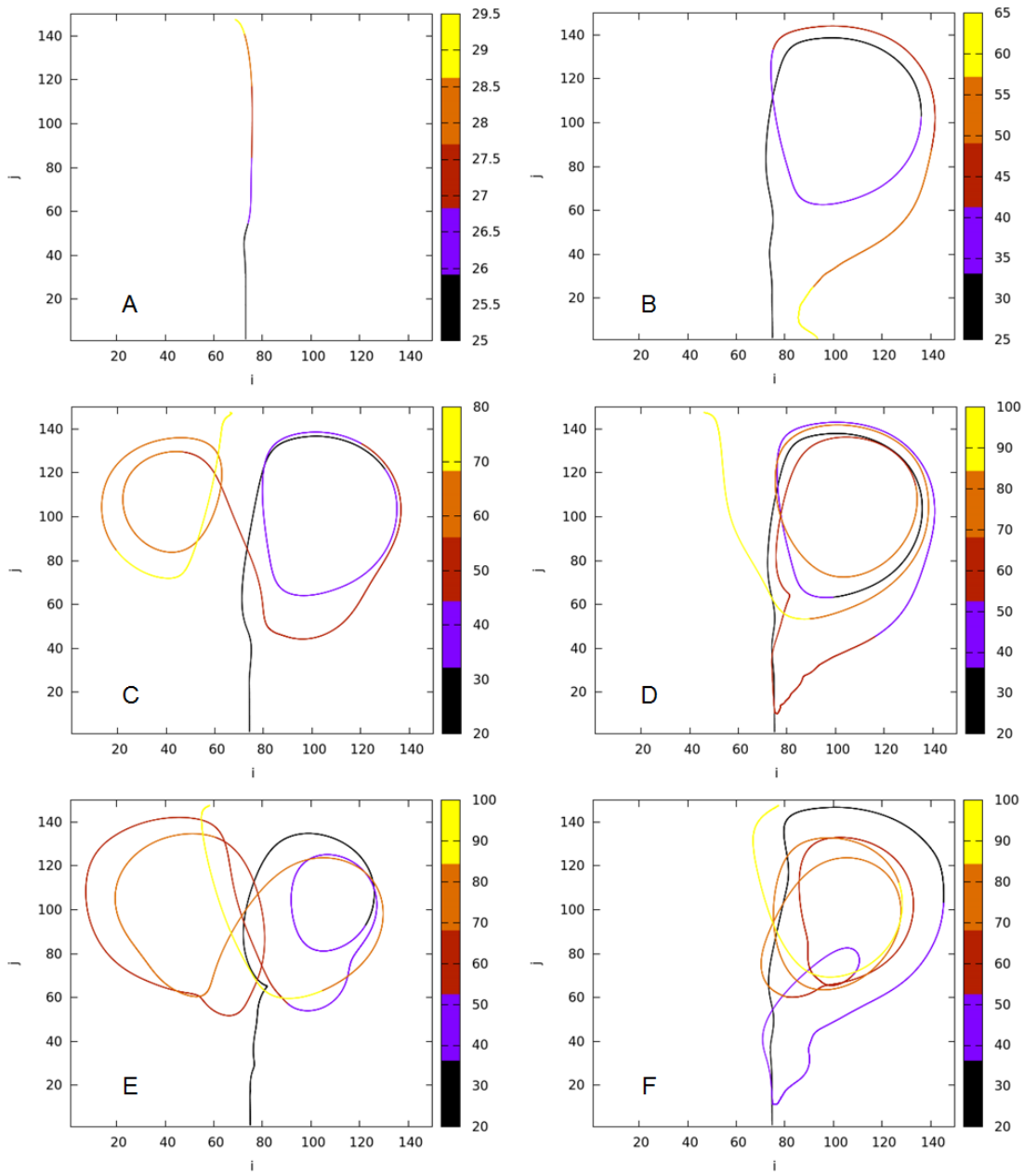


Figure 3.5: Six trajectories from problem in Fig. 3.3. Colour denotes time interval in which particle travelled corresponding segment of the trajectory. Particle A is example of very fast particle which moves through the fluid avoiding any turbulences. Examples C and E are particles that are caught in the circular vertex, but manage to escape after few turns. Particles B and D leave the domain through the water drain on the bottom boundary, rest is propelled over the top border (to the plume in ice crust).

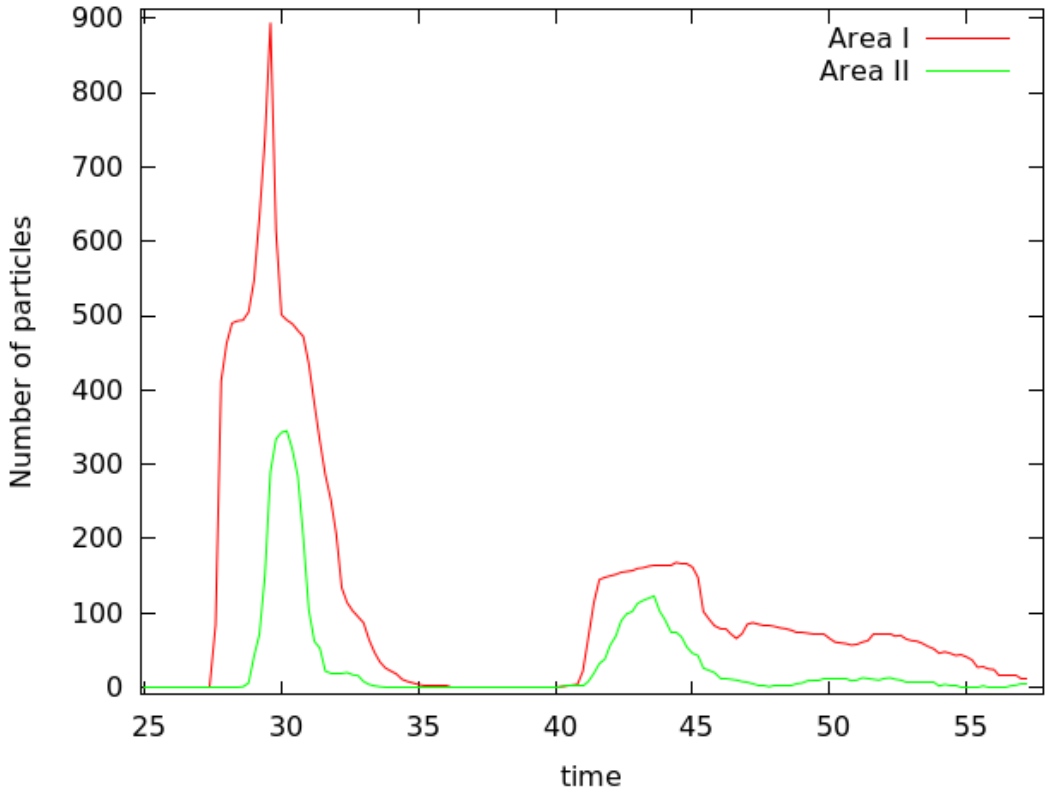


Figure 3.6: Densities of 1000 particles in problem from Fig. 3.3 in areas I and II. All trajectories were released near water jet in dimensionless time $t = 25$. We can clearly identify first maximum which is a global maximum for both curves at $t \approx 30$. This is time when our particles reached top border for the first time. We can also see that approximately 90% of the particles reached the upper ice crust. Later maxima correspond with particles that were circulating in the domain(like particles B,C,D,E,F in Fig. 3.5).

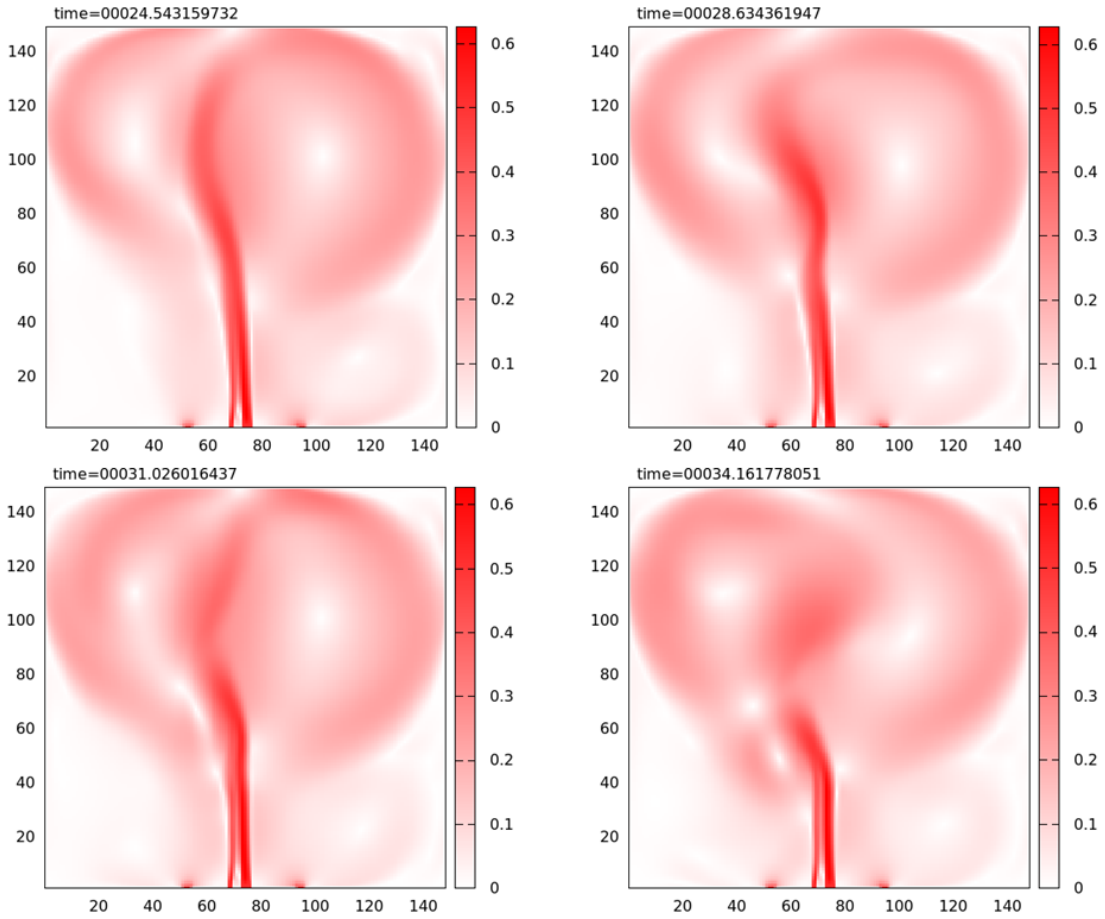


Figure 3.7: Velocities of assymetrical problem in square area with resolution 150×150, $Re = 10^9$, no-slip on sides and water jet type A. In 2 cells distance to the right from main water jet, another water jet of the type A is prescribed and 3 cells to the left another one. Drains are made 3 cells wide to satisfy the continuity equation. For densities of trajectories see Fig. 3.8

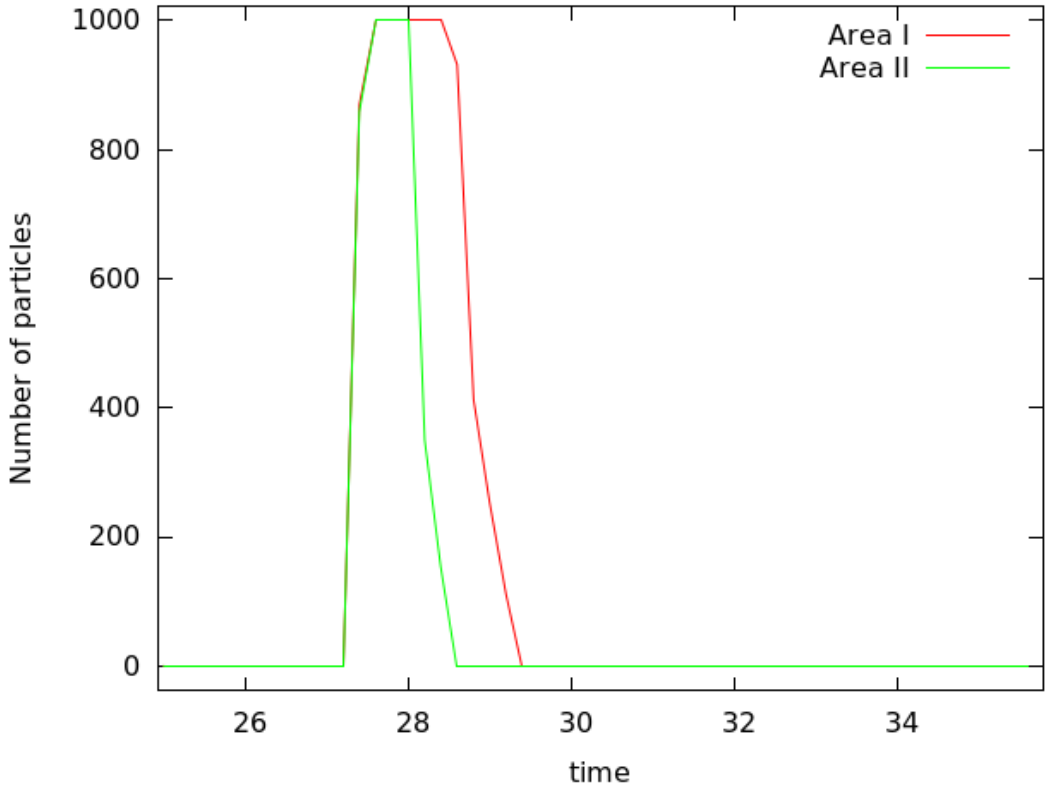


Figure 3.8: Densities for 1000 particles in areas I and II for problem in Fig. 3.7 released at dimensionless time $t = 25$. Particles reached the top border slightly sooner compared to single jet (see Fig. 3.6). Clearly, two extra jets create more stable stream in the middle.

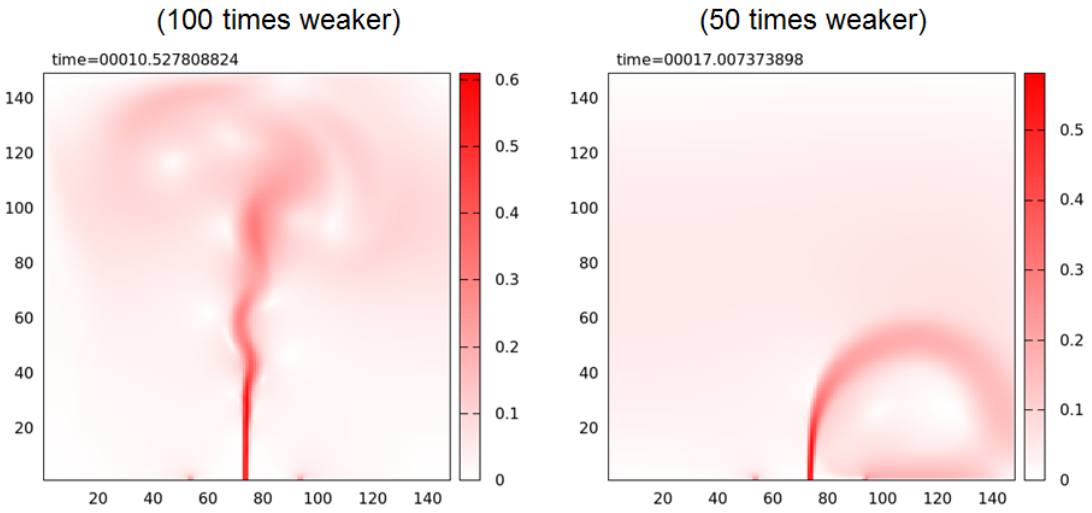


Figure 3.9: Velocities of problem with slap forces 100- and 50-times slower than our water jet of type A. This simulation is for square domain 150×150 and $Re = 10^9$. We see that 50-times weaker global flow bends our jet and does not allow any continuous transport to form between the oceanic floor and ice crust – snapshot to the right is the final state. On the other hand, 100-times slower lateral flow allows formation of continuous stream as can be seen on the left snapshot. See Fig. 3.10 for densities of markers.

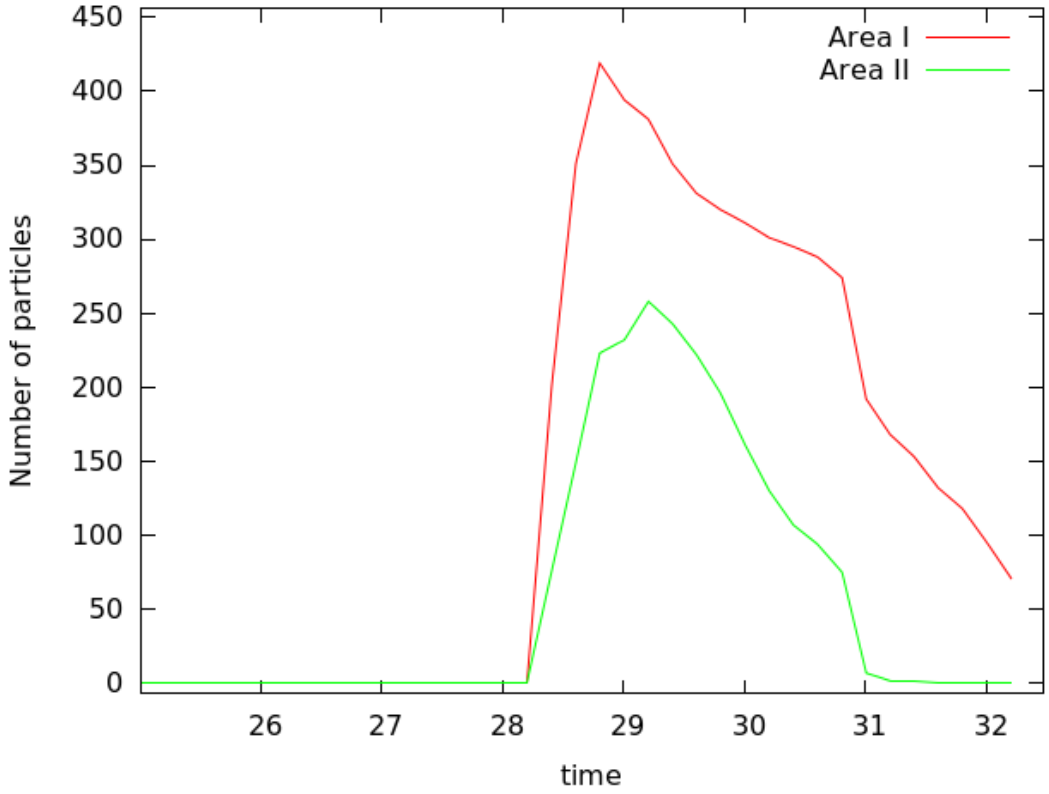


Figure 3.10: Densities for 1000 particles in Area I and II for left problem from Fig. 3.9 (100-times weaker lateral flow). As comapred to problem without the lateral flow (Fig. 3.6), time required for particles to reach the ice crust is approximately the same, but amount of particles is about the half. This shows that even very slow lateral flow can effect the transport. For the problem on the right (50-times weaker lateral flow), none of the testing particles reached area I or II.

3.2 Bottom boundary of type B

In our square problem, zero velocity condition on the sides is slowing down the flow and does not correspond with approximations in bigger global ocean. Free-slip condition is much more judicious option. Water jet of type B is more real case of the water jet.

In the following we show, that the amount of transported silica particles from the ocean floor to the plumes in the ice crust depends on the width of the prescribed jet. Table 3.1 summarizes all widths we have used in both numerical width (in cells) and responsive real width in meters (for ocean with depth D=10km). We compare only the narrowest and the widest cases which are in Fig. 3.11 and Fig. 3.12. The most important results are in Fig. 3.13 and Fig. 3.14, where concentration of markers in area of type 1 and 2 respectively are shown for different widths.

Table 3.1: Table of tested widths of water jet.

Test #	Numerical[# of cells]	Real[meters]
1	6	400
2	8	533
3	10	666.7
4	16	1066.7

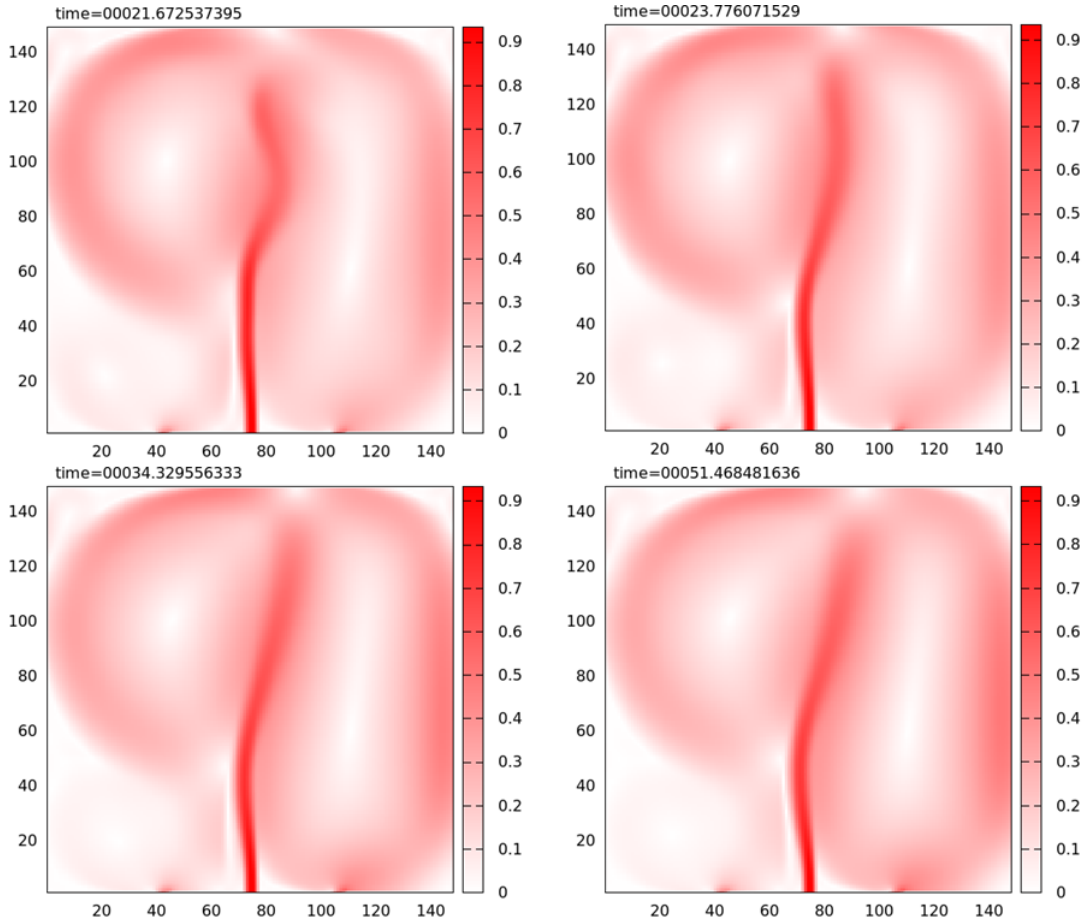


Figure 3.11: Snapshots of velocities in square computational domain with resolution 150×150, $Re = 10^9$, free-slip side boundary condition on sides and type B bottom boundary with dimensionless width $w = 6$. Because the jet is relatively large, the simulation reaches semi-stationary state with little turbulences on the top border. The stream is slightly tilted to the right side from small numerical assymmetry in location of the prescribed jet. Compare with Fig. 3.12 for much larger jet.

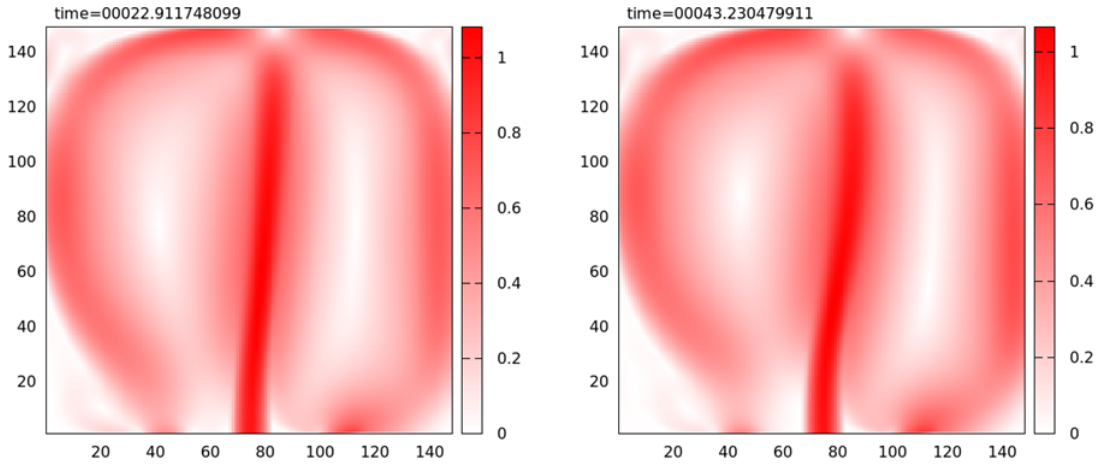


Figure 3.12: Same situation as in Fig. 3.11, but with dimensionless width $w = 16$. This very large jet stabilize in shown position with almost none turbulences, thus only two snapshots are shown.

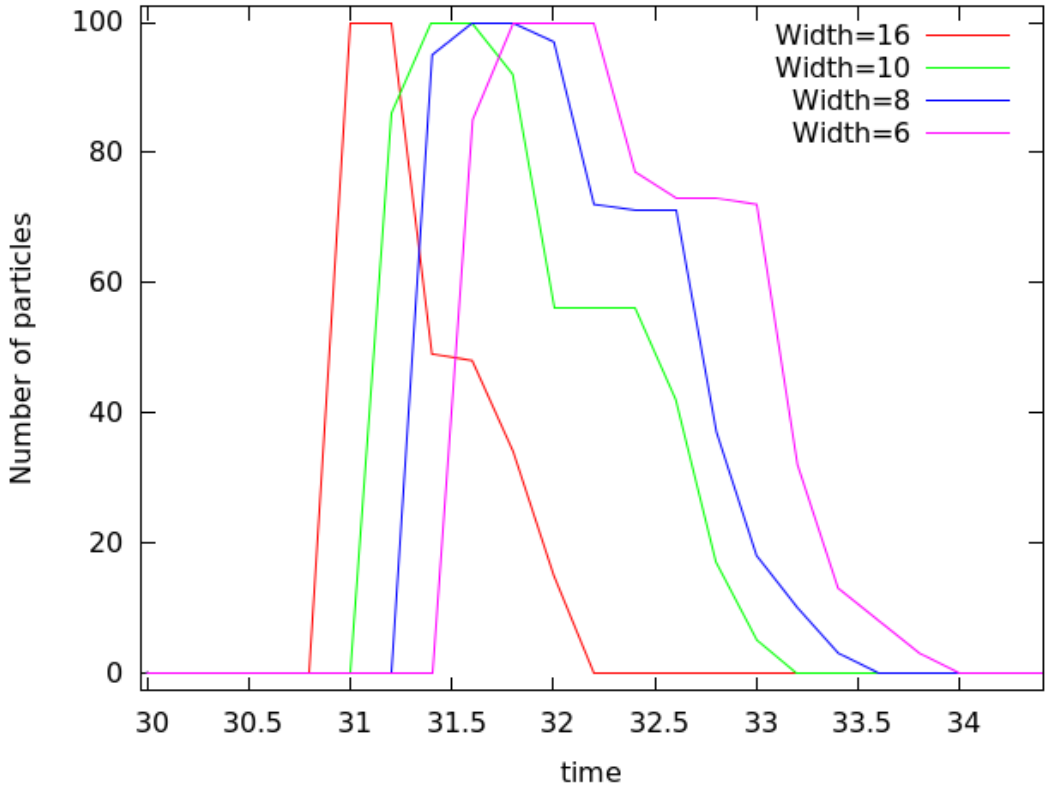


Figure 3.13: Density of 100 trajectories in area of type I. For type II see Fig. 3.14. Particles were released at dimensionless time $t = 30$ directly above water jet. For widest jet the particles reached top the fastest. This is because the largest jet undergoes the least amount of turbulences and instabilities. For the narrowest jet the time is the longest.

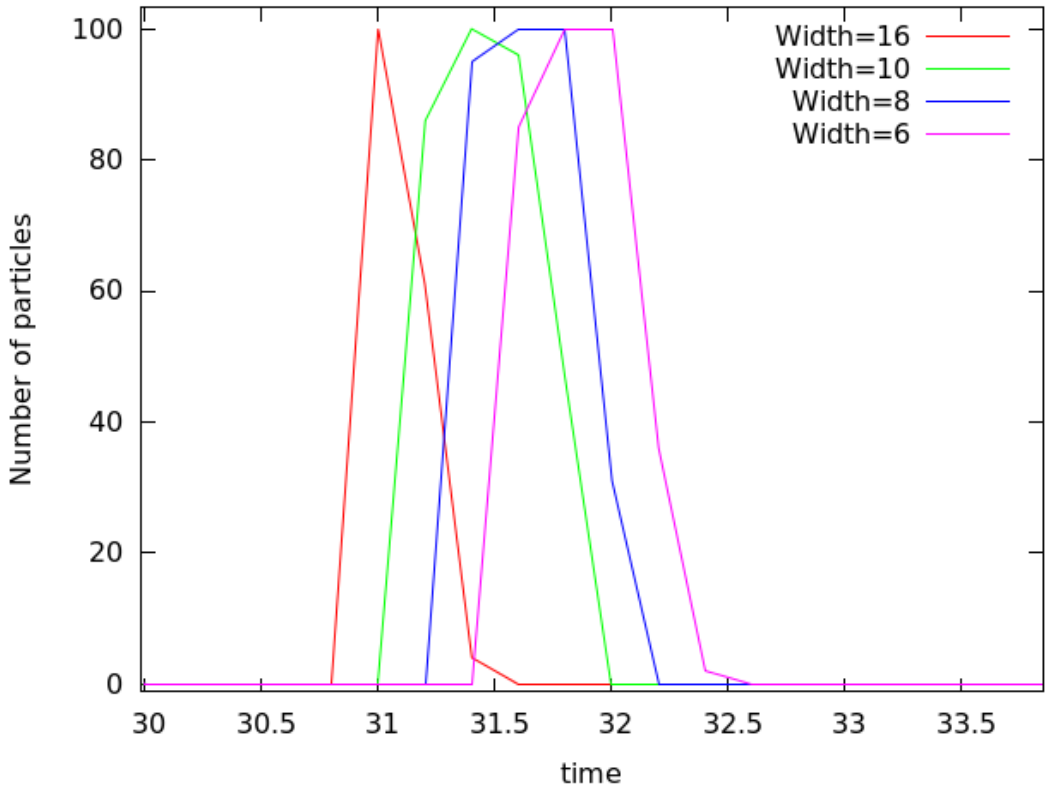


Figure 3.14: Same situation as in Fig. 3.13, but for area of type II.

3.3 Discussion

We did series of simulations in simplified 2D cartesian geometry which purpose was to rate the flow in subsurface ocean induced by water jets on the ocean floor and estimate durations of nano-silica particles transport.

Results that are shown in chapter 3 show, that the transport of nano-silica particles in the ocean is very effective. Even when the depth of the ocean is ($D = 50$ km) and velocity in jet is very low($v = 1$ m/s), the transport takes only 97 hours. This time is less, than is needed by geochemical model from Hsu et al. (2015). We did not notice much dependence on the Reynolds number, when the jet is sufficiently small and when the lateral flow on the ocean is much smaller than water jet. Sufficient amount of concentration of nano-silica particles near top boundary even when the lateral flow is not stronger than 1% of the water jet.

Simulations shown in chapter 3 show first step to realistic flow in the subsurface ocean on Enceladus or Europa. We would like to point out, that our simulation is simplified and is neglecting many effects which may have effect on the material transport. The most important simplifications (from most important) :

- The width of the water jets is unknown, hence it is very likely that their size is much smaller than those we have used. On Earth, the ocean objects that supports hydrothermal convection (black smokers) have widths in meters, while our narrowest size is ≈ 67 m. It is most likely, that with very narrow jets, the diffusion of the nano-silica particles would be much bigger and their transport less effective. With limited amount of data it is possible that the hydrothermal circulations on Enceladus is different from that one on the Earth, where (and probabily also on Europa), the hydrothermal circulation is induced by magnetism. The main source of heat on Enceladus is slap-heating. This heating has long-wavelength character and similar may be the hydrothermal circulation. If so, the depth of the ocean chosen in this work is only lower estimation.

- The resolution is not sufficiently high. It does not allow us to prescribe even narrower jets (see point above).

- Our model is only two dimensional and Cartesian. In 3D geometry the transport time would probably be bigger.

- In our model, we work with the traditional Navier-Stokes equation and we do not include turbulences via RANS (Reynols Averaged Navier-Stokes equation). Character of the fluid is given only with chosen geometry and the Reynolds number. The effect of turbulent viscosity is not included in our model.

- We assume, that the water in the ocean is homogeneous in all the domain and have constant salinity. We omit the thermal convection which would most likely only empower the transport, because the water from the jet can have more than 90° than is estimated average temperature of the ocean.

Conclusion

The object of this work was to show that there exists sufficient transport of silica particles from the ocean floor to the top in subsurface ocean of Saturn's moon Enceladus or Jupiter's moon Europa. In order to test the hypothesis we developed a Fortran program that allows us to simulate the flow of the fluid in cartesian 2D computational domain induced with water jet on the bottom. Program uses staggered grid and can solve the Navier-Stoke equation up to resolution of 300×75 or 150×150. Hence it can effectively simulate the flow for the Reynolds number up to $Re = 10^9$. We did series of tests for different shapes of the domain, different boundary conditions and different water jets. We have also written another program in Python programming language, to reconstruct the trajectories of the particles. We have used this program to compute density of silica particles right below the upper ice crust.

We showed, that when the water jet is approximately 150 times smaller than the depth of the ocean, the transport of the particles from the ocean floor to the top is very effective and transport times are lower than those limited by the geochemical model from *Hsu, (2015)*.

The limits for our model which we set in discussion, suggests next possibilities for the following research. Our work presents the first estimation to quantify the speed of the transport of the nano-silica particles in the subsurface ocean on the ice moons.

Despite many simplifications, we believe that our work is useful first step towards the research of this phenomenon which may have big impact for astrobiological potential on Enceladus or Europa.

Bibliography

Hsu, H.W., Postberg, F., Sekine, Y., Shibuya, T., Kempf, S., Horanyi, M., Juhasz, A., Altobelli, N., Suzuki, K., Masaki, Y., Kuwatani, T., Tachibana, S., Sirono, S., Mogaras-Klostermeyer, G., and Srama, R. (2015). Ongoing hydrothermal activities within Enceladus, *Nature*, **519**, 207-210, doi: 10.1038/nature14262.

Dougherty, M.K., Esposito, L.W., and Stamatios, M.K. (2009). *Saturn from Cassini-Huygens*, Spriner, New York.

Pappalardo, R.T., McKinnon, W.B., and Khurana, K.(2009). *Europa*, Univ. Arizona Press, Tucson.

Thomas , P.C., Tajeddine, R., Tiscareno, M.S., Burns, J.A., Joseph, J., Loredo, T.J., Helfenstein, P., and Porco, C. (2015). Endeladus's measured physical liberation requires a global subsurface ocean, *Icarus*, **264**, 37-47, doi 10.1016/j.icarus.2015.08.037.

Gerya, T. (2010). *Introduction to Numerical Geodynamic Modelling*, Cambridge Univ. Press., Cambridge.

List of Tables

Attachments

Attachment 1: CD with developed computer programs