THE NAVIER-STOKES VOIGHT MODEL AND CONVERGENCE TO EQUILIBRIUM AND STATISTICAL EQUILIBRIUM

by

Ali Koseoglu

BS in Mathematics, Ege University, Turkey, 2007

Submitted to the Graduate Faculty of

the Arts and Sciences in partial fulfillment

of the requirements for the degree of

MS in Mathematics

University of Pittsburgh

2011

UNIVERSITY OF PITTSBURGH

ARTS AND SCIENCES

This thesis was presented

by

Ali Koseoglu

It was defended on

JULY 22 2011

and approved by

Prof. William J. Layton, University of Pittsburgh

Prof. Paul Gartside, University of Pittsburgh

Prof. Juan J. Manfredi, University of Pittsburgh

Thesis Advisor: Prof. William J. Layton, University of Pittsburgh

THE NAVIER-STOKES VOIGHT MODEL AND CONVERGENCE TO EQUILIBRIUM AND STATISTICAL EQUILIBRIUM

Ali Koseoglu, M.S.

University of Pittsburgh, 2011

This thesis tests accelerating convergence to steady state and statistical equilibrium for the Navier-Stokes Voight (NSV) Model motivated by the Spin-Up phase of Ocean Climate Models. First, by adding a new parameter to Navier-Stokes equations, the NSV model is determined. Test conditions are identified for spin up, attaining equilibrium, and statistical equilibrium. Convergence is then analyzed and the Finite Element Method with Backward Euler discretization is programmed using Free FEM++ to simulate the NSV model. Two problems with a known exact solution are used for a square domain. One problem with unknown solution and one problem with known exact solution in circle domain are also tested for convergence of the method. We find that the adding parameter may not accelerate convergence to equilibrium or statistical equilibrium.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to thank everyone who made this thesis possible. Not only those who helped with mathematical tools and programming details, but also those who provided continuous support during the formation of this thesis.

Beyond everything else, I feel eternally indebted to my advisor Prof. William Layton, for his ecouragement, invaluable ideas and suggestions. His passion, enthusiasm and support make this thesis possible.

I wish to thank all of the committee members, Prof. Paul Gartside and Prof. Juan J. Manfredi for all their valuable suggestions and for being part of my thesis committee.

It is a pleasure to thank Osman R. Isik from whom I have learned so much especially in terms of mathematical tools and skills.

Other key people I also must acknowledge are Ross Ingram and Nate Mays for their high support in FreeFem++ without any hesitation and teaching programming skills.

I also would like to awknwledge theTurkish Government for their financial support during my graduate study.

My sincere and very special thanks goes out my parents Husnu and Nuray Koseoglu, and my wife's parents Mustafa and Gulten Dulger, who have supported me during my graduate studies.

Lastly, but most importantly, I want to thank my dear wife, Duygu who has dedicated herself to me and my dreams. I cannot suficiently thank her enough.

To my wife, Duygu...

1.0 INTRODUCTION

1.1 NAVIER STOKES VOIGHT MODEL

One important moment in the history of fluid flow occurred in 1687 through the work of one of the greatest scientist, Isaac Newton. That moment came to prosperity specifically via his book Philosophiae Naturalis Principia Mathematica. He stated that the shear stress between layers is proportional with the velocity gradient, which is perpendicular to those layers as long as the flow is smooth, parallel and monotone. In the year of 1738, Bernoulli contributed to Newton's work by proving that the acceleration is also proportional to the gradient of pressure. Then, Leonhard Euler derived his equations for the flow of incompressible and frictionless fluids ($\nu = 0$). In 1821, Navier introduced the viscosity parameter in viscous flows but there was still something incomplete: the physical meaning of parameter ν. After 24 years, Sir George Gabriel Stokes used absolute viscosity, in other words, stated that ν is the viscosity of the fluid. That resulted in what we know the Navier-Stokes Equations (NSE):

$$\frac{\partial u(x,t)}{\partial t} + u(x,t) \cdot \nabla u(x,t) = -\nabla p(x,t) + \upsilon \nabla^2 u(x,t) + f \qquad (1.1)$$

If we look at closer:

$$\frac{\partial u(x,t)}{\partial t} \quad : \quad \text{Unsteady acceleration}$$

$$u(x,t) \cdot \nabla u(x,t) \quad : \quad \text{Convective acceleration}$$

$$-\nabla p(x,t) \quad : \quad \text{Pressure gradient}$$

$$\upsilon \nabla^2 u(x,t) \quad : \quad \text{Viscosity}$$

$$f \quad : \quad \text{Body force.}$$

where x is a spatial point, t is time, $u(x,t)$ is the velocity vector field and v is the constant dynamic viscosity.

As the years evolved, Navier-Stokes Equations were modified. In 1934 J. Leray ([24],[25]) studied the regularizations of the NSE and in 1973 Oskolov ([28]) studied the Navier-Stokes Voight Model for the viscoelastic incompressible fluid. This model is known as the 3D Navier-Stokes Voight Model of viscoelastic fluids. The model is given by

$$
\begin{aligned}
-\alpha^2 \Delta u_t + u_t - v\Delta u + u \cdot \nabla u + \nabla p &= f, \\
\nabla \cdot u &= 0,
\end{aligned}
$$

where α is the regularization parameter, $\nabla \cdot u = 0$ is the incompressibility (divergence free) condition (see also ([5], [6])). This model is globally well posed ([28], [29]) and in Kalantarov and Titi ([20]), it is shown that it has a finite dimensional global attractor.

My thesis was greatly inspired by works that we have mentioned above. We will be using NSV Model further taking α instead of α^2 where $\alpha > 0$. The model presented below will be tested with Backward Euler Method as we will look how adding $-\alpha\Delta u_t$ to NSE affects the convergence behavior of the NSE-steady state solution.

$$
\begin{aligned}
-\alpha \Delta u_t + u_t - v\Delta u + u \cdot \nabla u + \nabla p &= f, \\
\nabla \cdot u &= 0,
\end{aligned}
\tag{1.2}
$$

where α is a positive parameter.

1.2   SPIN UP AND STATISTICAL EQUILIBRIUM

The universe is hiding considerable knowledge from us. It is waiting to be discovered. One of the most interesting elements about the universe is climate. Indeed, to better understand climate we need to utilize the Nature's own language and that is undeniably *Mathematics*. In 1967, Kirk Bryn [Bryan and Cox, [3]] introduced that the first Ocean General Circulation Model that is the root of many Ocean Models now currently being used. As the years passed, many ocean models have been undergoing various developments. In order to initialize a

climate model simulation, an initial velocity (that is in statistical equilibrium) is needed. It is then taken as an initial condition for a climate model evolution. This interesting process is called "spin up" and is defined as the time taken for an ocean model to reach statistical equilibrium with mean data. The main fact here is without spin up, the model cannot be studied.

Mostly, the researchers concentrated on stable equilibrium. Constantin, Foias and Temam ([13]) dealt with the problem numerically. They found the existence of stable equilibrium was a result of numerically computed stable equilibrium solution. Stability of solutions and error estimates were studied by Heywood and Rannacher ([18]). However, the factor of *time* imposed another problem. Attaining a spin up state needs very long time scales. One example is in paleoclimatology. Using the Community Climate System Model Version 3 [Collins et al., 2006 ([11])], for the Cenomanian-Turonian epoch, the model is used forward in time for 2000 years [Alexandre et al., 2010 ([1]), see ([2]) for clear explanation]. Another example is Ocean General Circulation Model that takes approximately 8000 years for a spin up run with an ocean depth of 5000m ([2]).

Since the spin up phase costs huge amounts of CPU time and takes a long real time, methods need acceleration. Bryan ([4]), Klinger ([22]) and Khatiwala and co-workers ([21]) used so-called distorted physics method, exponential extrapolation and so-called matrix-method, respectively, to accelerate convergence to equilibrium solution (see more in ([2])).

With this idea we will test NSV in (1.2). First we will count number of steps that the equilibrium required. This is measured by

$$
\left\| \frac{u^{n+1} - u^n}{\Delta t} \right\| \leq \epsilon \qquad \text{(Test 1)}
$$

$$
\left\| \frac{u^{n+1} - u^n}{\Delta t} \right\| + \left\| \nabla \frac{u^{n+1} - u^n}{\Delta t} \right\| \leq \epsilon
$$

where ϵ is the $TOLERANCE$.

Second we will test accelerating convergence to statistical equilibrium. Let's look at this closer. If u and p is a smooth solution of (1.1), taking inner product with u and integrating over domain Ω and integrating in time T the energy equation of the NSE becomes:

$$
\frac{1}{2} ||u(T)||^2 + \int_0^T \upsilon ||\nabla u(t)||^2 dt = \frac{1}{2} ||u_0||^2 + \int_0^T (f(t), u(t)) dt
$$

3

This equation has the interpretation:

$$\text{Kinetic Energy } (T) + \text{Total Energy Dissipated over } [0, T]$$

$$= \text{ Initial Kinetic Energy } + \text{Total Power Input over } [0, T]$$

If we divide the integrals with T then they become time averages and we have

$$\frac{\frac{1}{2}||u(T)||^2 - \frac{1}{2}||u_0||^2}{T} + \frac{1}{T}\int_0^T v||\nabla u(t)||^2 dt = \frac{1}{T}\int_0^T \int_\Omega f(x,t) \cdot u(x,t) dx dt$$

This equation is statistical steady state if

$$\frac{1}{T}\int_0^T v||\nabla u(t)||^2 dt = \frac{1}{T}\int_0^T \int_\Omega f(x,t) \cdot u(x,t) dx dt$$

Thus, statistical steady state occurs if

$$\left| \frac{\frac{1}{2}||u(T')||^2 - \frac{1}{2}||u_0||^2}{T'} \right| \leq \epsilon \text{ for all } T' \geq T$$

If we apply this to the (1.2) from the term $-\alpha \Delta u_t$, there will be an extra term coming from $(-\alpha \Delta u_t, u)$ and integrated over time T, statistical equilibrium becomes

$$\left| \frac{\frac{1}{2}||u(T')||^2 - \frac{1}{2}||u_0||^2}{T'} + \alpha\frac{\frac{1}{2}||\nabla u(T')||^2 - \frac{1}{2}||\nabla u_0||^2}{T'} \right| \leq \epsilon \text{ for all } T' \geq T \qquad \text{(Test 2)}$$

2.0   PRELIMINARIES

In this chapter we introduce the basic facts that will be used for the development of this thesis. We use definitions and notations as in Layton ([23]).

Let Ω denotes a bounded, open and connected domain in $\mathbb{R}^2$ that the fluid flows. The Hilbert space $L^2(\Omega)$ is the most important tool in fluid dynamics. If we look at closer, we will see that velocity with the total kinetic energy forms the function space $L^2(\Omega)$. Let ρ_0 be the constant density and u be the velocity for a fluid with domain Ω.

$$Kinetic\ Energy = \frac{1}{2}mass \ \times \ velocity^2$$

which can be presented as

$$Total\ Kinetic\ Energy = \frac{1}{2}\int_\Omega |u|^2\,dx.$$

So, as stated in Layton ([23]), the space $L^2(\Omega)$ is just the set of all velocity fields with finite kinetic energy.

2.1   THE HILBERT SPACE $L^2(\Omega)$ AND $X := H_0^1(\Omega)$

Definition 1. $(L^2(\Omega)$ functions$)$. $L^2(\Omega)$ denotes the set of all Lebesgue measurable functions $p : \Omega \to \mathbb{R}$ with

$$\int_\Omega |p|^2\,dx < \infty.$$

5

An inner product on $L^2(\Omega)$ and its produced norm can be defined as

$$(p, q) := \int_\Omega p(x)q(x)dx$$

and the norm on the continuous functions $C^0(\Omega)$ is,

$$\|p\| := \left[\int_\Omega |p|^2 \, dx\right]^{1/2}$$

respectively. In this thesis, $L^2(\Omega)$ norm and the inner product denoted as $\|\cdot\|$ and $(\cdot, \cdot)$ respectively. We also give the definitions of velocities and tensors with finite kinetic energy.

Definition 2. *(velocities)* $L^2(\Omega)^d = \{v = (v_1, ..., v_d) : \Omega \to \mathbb{R}^d : each\ component\ v_j \in L^2(\Omega)$ *where* $j = 1..d\}$ *and*

$$\|v\| = \|v\|_{L^2(\Omega)^d} := [\|v_1\|^2 + \|v_2\|^2 + ... + \|v_d\|^2]^{\frac{1}{2}}.$$

Definition 3. *(tensors)* $L^2(\Omega)^{d \times d} := V = \{(V_{ij}), i, j = 1, ..., d : V_{ij} \in L^2(\Omega)\}, i.e., \|V\| < \infty, where$

$$\|V\| := \left[\sum_{i,j=1}^d \|V_{ij}\|^2\right]^{\frac{1}{2}}$$

The notations that we use for complete, normed, linear $L^2(\Omega)$ Hilbert space with inner product $(\cdot, \cdot)$ are

$$(p, q) \quad : \quad = \int_\Omega p(x)q(x)dx \text{ for } p, q : \Omega \to \mathbb{R} \text{ and } p, q \in L^2(\Omega),$$

$$(u, v) \quad : \quad = \int_\Omega \sum_{i=1}^d u_i v_i dx \text{ for } u, v \in L^2(\Omega)^d,$$

$$(S, T) \quad : \quad = \int_\Omega \sum_{i,j=1}^d S_{ij}(x)T_{ij}(x)dx \text{ for } S, T \in L^2(\Omega)^{d \times d}$$

where the norm of a vector u in $L^2(\Omega)$ is

$$\|u\| = \sqrt{(u, u)} \ \forall u \in L^2(\Omega).$$

While a fluid is flowing, each section of fluid applies a force to another section of fluid so that this battle generates local changes in velocity and makes complex patterns in a domain of fluid. This fact is represented mathematically with the first derivatives of u. Since the velocity is in $L^2(\Omega)$, its gradient should be in $L^2(\Omega)^{d \times d}$

6

Definition 4. *Let $d = \dim(\Omega) = 2$ or 3. If $u = u_{ij}$, $i = 1, ..., d$, then ∇u is the $d \times d$ matrix of all possible first derivatives of u,*

$$(\nabla u)_{ij} = \frac{\partial u_j}{\partial x_i}, i, j = 1...d, \;\; and \;\; \|\nabla u\|^2 = \sum_{i,j=1}^{d} \left\| \frac{\partial u_j}{\partial x_i} \right\|^2.$$

Let u be a $C^1(\Omega)$ function vanishing on $\partial\Omega$. Then,

$$\|u\|_X := [\|u\|^2 + \|\nabla u\|^2]^{\frac{1}{2}}$$

is a norm which is induced by an inner product,

$$(u, v)_X := (u, v) + (\nabla u, \nabla v),$$

where $(\nabla u, \nabla v) = \sum_{i,j=1}^{d} \left(\frac{\partial u_i}{\partial x_j}, \frac{\partial v_i}{\partial x_j} \right)$.

Definition 5. *$X = H_0^1(\Omega)$ is the closure in $\|\cdot\|_X$ of*

$$\{v : \Omega \to \mathbb{R}^d : v \in C^1 \, and \, v = 0 \;\, on \;\, \partial\Omega\}$$

Now, we are ready to define our spaces. We will seek the velocity u and the pressure p in the region with the space

$$
\begin{aligned}
X \;\; &: \;\; = H_0^1(\Omega) = \left\{ v \in L^2(\Omega) : \nabla v \in L^2(\Omega)^{d \times d} \text{ and } v\,|_{\partial\Omega} = 0 \right\} \\
Q \;\; &: \;\; = L^2(\Omega)^d = \left\{ q : q \in L^2(\Omega) : \int_\Omega q = 0 \right\}
\end{aligned}
$$

2.2   THE NAVIER-STOKES EQUATIONS

Consider the flow of a fluid in a region Ω in $\mathbb{R}^2$. The Navier-Stokes equations (NSE) describe the motion of fluid flow as a continuum model. They are based on the conservation of mass and conservation of momentum laws. For compressible and incompressible fluids, NSE is the most accepted model. For the incompressible case the equation is given by

$$u_t + u \cdot \nabla u + \nabla(\frac{p}{\rho_0}) = \frac{\mu}{\rho_0}\Delta u \ , \ \text{in} \ \Omega. \tag{2.1}$$

Here ρ_0 is the density, the parameter μ dynamic viscosity so that

$$\frac{\mu}{\rho_0} := \nu = \ \text{kinematic viscosity.} \tag{2.2}$$

and replacing the pressure by a scaled pressure p:

$$p = \frac{p}{\rho_0}$$

we give the definition of time dependent incompressible NSE:

Definition 6. *The time dependent NSE are given by*

$$u : \Omega \times [0,T] \to \mathbb{R}^d \ \ , \ \ p : \Omega \times [0,T] \to \mathbb{R}$$

satisfying

$$
\begin{aligned}
u_t + u.\nabla u - \nu \Delta u + \nabla p &= f\ (x,t) \ for\ x \in \Omega,\ 0 < t \leq T, \tag{2.3}\\
\nabla \cdot u &= 0,\ x \in \Omega\ \ for\ 0 < t \leq T,\\
u &= 0\ \ on\ \ \partial\Omega\ \ for\ \ 0 \leq t \leq T,\\
u(x,0) &= u_0(x)\ \ for\ \ x \in \Omega
\end{aligned}
$$

and the usual normalization condition that $\int_\Omega p(x,t)dx = 0$ for $0 < t \leq T$.

where u denotes the fluid velocity, ν is the viscosity, p denotes the pressure, $f(x,t)$ is the body force, $\nabla \cdot u$ is the incompressible condition and $u_0 \in L^2(\Omega)$ is a divergence free condition. The derivation of these equations can be found in Chorin and Marsden ([10]). Here, we want to give more information about viscosity as described in Layton ([23]). Since the flow occurs in a domain Ω, there must be a characteristic length L of the flow geometry. If we rescale the variables in (2.1), where L is the reference length and V is the reference speed of flow,

$$
\begin{aligned}
\Omega^* \quad &: \quad = \frac{\Omega}{L} \\
x^* \quad &: \quad = \frac{x}{L} \\
u^* \quad &: \quad = \frac{u}{V} \\
t^* \quad &: \quad = \frac{Vt}{L} \\
p^* \quad &: \quad = \frac{p - p_0}{\rho_0^* V^2}
\end{aligned}
$$

After rescaling the values in (2.1) the equation becomes

$$
u_{t^*}^* + u^* \cdot \nabla u^* + \nabla p^* = \left(\frac{\mu}{\rho_0 V L}\right)\Delta^* u^*
$$

We define this dimensionless parameter as Reynolds Number and represent it as

$$
\mathrm{Re} := \frac{\rho_0 V L}{\mu}
$$

Generally the term VL is taken as 1 and from (2.2), the equation become

$$
\nu = \frac{1}{\mathrm{Re}}
$$

The equation in (2.3) has a solution. Although there is an open question about strong solution vs. weak solution in 3D. In 2D it is clearly shown that a weak solution is actually a strong solution in Manica ([26]). Definitions and proof of this also can be found in many books such as Galdi ([16]) and Shor ([30]). We give the definition of strong solution as in Layton ([23])

Definition 7. (u, p) *is a strong solution of (2.3) if* $u \in L^2(0, T; X) \cap L^\infty(0, T; L^2(\Omega))$ *and*

1. $u : [0,T] \to X$ is a differentiable map with $u_t \in L^2(0,T;X)$ and $p : (0,T] \to Q$ is an integrable map with $p \in L^2(0,T;Q)$

2. For all $t' \in (0,T]$, (u,p) satisfies

$$\int_0^{t'} \left[(u_t, v) + (u \cdot \nabla u, v) + v(\nabla u, \nabla v) - (p, \nabla \cdot v) \right] dt' = \int_0^{t'} (f, v) \, dt'$$

for all $v \in L^2(0,T;H_0^1(\Omega)) \cap L^\infty(0,T;L^2(\Omega))$ and $\int_0^{t'} (q, \nabla \cdot u) \, dt' = 0$ for all $q \in L^2(0,T;L_0^2(\Omega))$.

3. $u_0 \in V$ and $\|u(t) - u_0\| \to 0$ as $t \to 0$.

4. $u \in L^4(0,T;X)$.

In Chapter 3, we will use the trilinear form in Section 2 and 3. We give the definition with referring Manica([26])

Definition 8. *(Skew-symmetric trilinear form) On $X \times X \times X$, $b^* : X \times X \times X \to \mathbb{R}$ is defined as*

$$b^*(u, v, w) = \frac{1}{2} (u \cdot \nabla v, w) - \frac{1}{2} (u \cdot \nabla w, v).$$

3.0 FINITE ELEMENT METHOD

Finite Element Method (FEM) is a method that gives numerical solution of field problems. This method takes a structure of a field and splits it into several pieces. Then, connect pieces with nodes and with this process the field quantity becomes interpolated over the structure. This method has several advantages in terms of wide variety of engineering problems especially in fluids.

- FEM is a powerful method that is capable to handle very complex geometry
- It can also handle complex restraints so that makes indeterminate structures solvable
- Although FEM acquires approximate solution, it is the fastest and closest solution in its own field
- Number of degrees-of-freedom is finite.

3.1 FINITE ELEMENT SPACE

FEM is the approach that we will be using in this thesis in order to solve NSV Model. Assume the problem has a known velocity $u(x, y)$ defined in a region Ω and also assume $u^h(x, y)$ is an approximate velocity representing $u(x, y)$ with a finite number of degrees of freedom.

In two dimensions, approximating a velocity is achieved by introducing a triangulation $T^h(\Omega)$ and defining $u^h(x, y)$ on each triangle with a small number of degrees of freedom. To begin constructing the approximate surface a triangulation $T^h(\Omega)$ is constructed satisfying a few basic conditions as stated in Layton ([23]):

11

- Conforming: The triangles are all edge to edge; meaning a vertex of one triangle cannot lie on the edge of another.

- Nondegeneracy: The triangles are not close to straight line segments. This is measured in different ways. It is common to ask that the smallest angle in the triangulation be bounded away from either zero or the largest from 180 degrees.

- The boundary is followed appropriately: Generally this means that (i) the boundary of the computational domain is within the targeted error of the boundary of the real domain, and (ii) no triangle has all three vertices on a part of the boundary where Dirichlet boundary conditions are imposed.

Once a mesh is generated, either by direct input or automatically, the possible approximations on that mesh must be selected. We are considering the approximate velocity to be globally continuous over each triangle and the nodes are just the vertices of $T^h(\Omega)$. The superscript h represents the triangle fineness measure:

$$h = \max_{K \in \tau^h} d(K).$$

The Finite Element Method is formed by Galerkin approximation and the proper choice of finite dimensional space X^h. Once we have a partial difference equation, we can solve it by simply solving its equivalent variational formulation. The fact of using this formulation rather than pointwise differential equation leads the stability and the power of FEM. Generally, given a bounded linear function $F \in X^*(dual\ space\ of\ X)$ the problem is to find $u \in X$ satisfying,

$$a(u, v) = F(v), \ \forall v \in X. \tag{3.1}$$

The form of (3.1) is continuous and coercive. The Galerkin method begins by selecting a finite dimensional subspace $X^h \subset X$. The Galerkin approximation $u^h \in X^h$ is the solution of the equations: find $u^h \in X^h$ satisfying

$$a(u^h, v^h) = F(v^h), \ \forall v^h \in X^h.$$

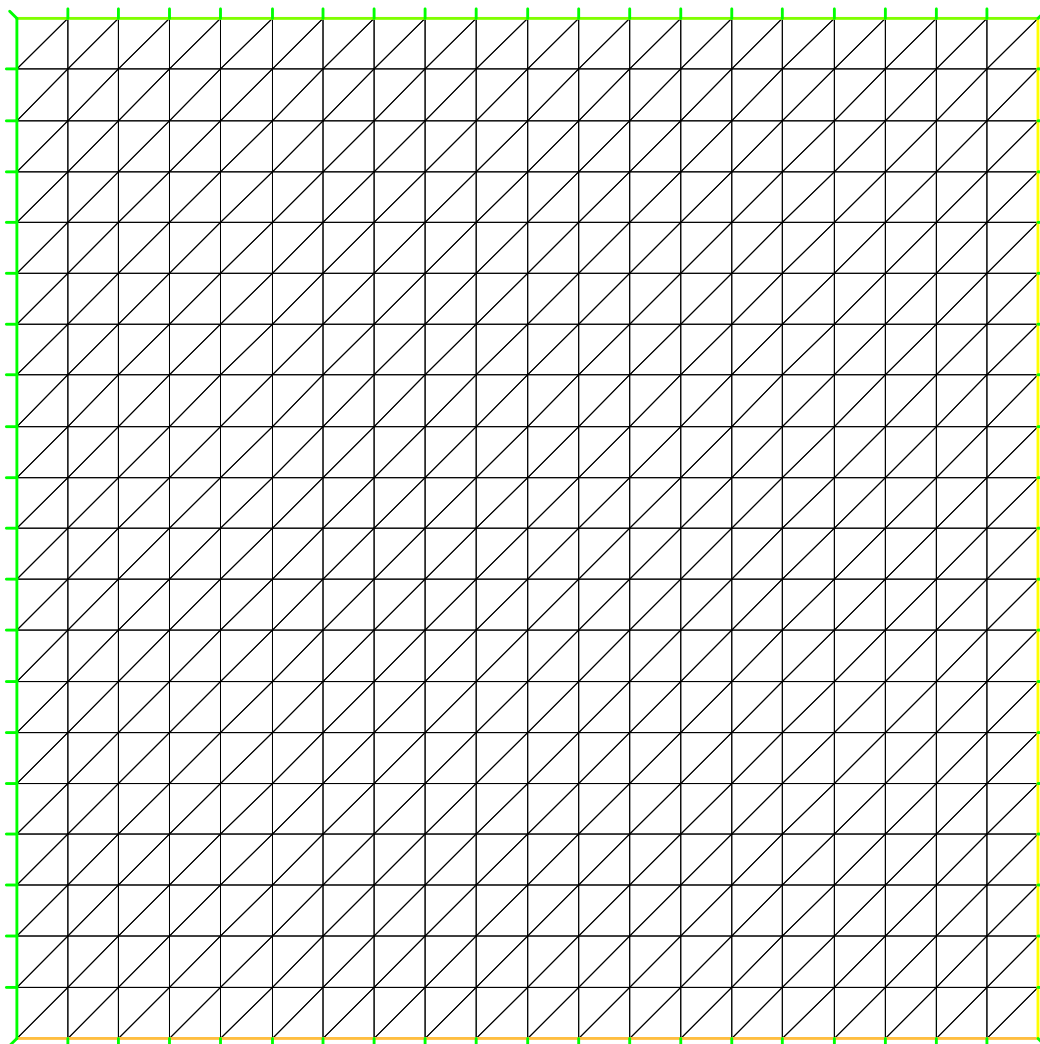The complete convergence theory of Galerkin method can be found in many Finite Element books such as Layton([23])

Figure 1: Triangulated mesh in a square domain (-1,1)x(-1,1)

3.2 FINITE ELEMENT SOLUTION OF NS-VOIGHT MODEL AND EQUILIBRIUM PROBLEM

Now, we are ready to discretize our model for spin up problem. Recall our model:

$$
\begin{aligned}
u_t - \alpha \Delta u_t + u.\nabla u - \nu \Delta u + \nabla p &= f(x,t) \text{ for } x \in \Omega, \ 0 < t \le T, \quad (3.2) \\
\nabla \cdot u &= 0, \ x \in \Omega \ \text{ for } 0 < t \le T, \\
u(x,0) &= u_0(x) \ \text{ for } \ x \in \Omega
\end{aligned}
$$

Let $X^h \subset X$ and $Q^h \subset Q$ be respectively the velocity and pressure finite element spaces which satisfy the following LBB^h (discrete inf-sup) condition,

$$
\inf_{q \in Q^h} \sup_{v \in X^h} \frac{(q, \nabla \cdot v)}{|v|_1 \|q\|} \ge C > 0.
$$

The approximate velocity and pressure are represented as maps

$$
\begin{aligned}
u^h &: \ [0,T] \to X^h \ , \\
p^h &: \ (0,T] \to Q^h
\end{aligned}
$$

for all $v^h \in X^h$ and $q^h \in Q^h$

$$
\begin{aligned}
\left(u_t^h, v^h\right) + \alpha \left(\nabla u_t, \nabla v^h\right) + b^* \left(u^h, u^h, v^h\right) + v \left(\nabla u^h, \nabla v^h\right) - \left(p^h, \nabla \cdot v^h\right) &= \left(f, v^h\right) \\
\left(\nabla \cdot u^h, q^h\right) &= 0 \\
\left(u^h(\cdot, 0) - u_0, v^h\right) &= 0 \ .
\end{aligned}
$$

Under LBB^h, select v^h to be the set of discretely divergence free functions and defined as $V^h := \left\{ v^h \in X^h : \left(q^h, \nabla \cdot v^h\right) = 0, \ \forall q^h \in Q^h \right\}$. The variational formulation of (3.2) is, find $u \in X, p \in Q$ and $\forall v \in X$ such that,

$$
\begin{aligned}
(u_t, v) + \alpha \left(\nabla u_t, \nabla v\right) + (u \cdot \nabla u, v) + v \left(\nabla u, \nabla v\right) - (p, \nabla \cdot v) &= (f, v) \\
(\nabla \cdot u, q) &= 0 \ , \ \forall q \in Q
\end{aligned}
$$

14

or using trilinear form which is equivalent to

$$(u_t, v) + b^*(u, u, v) + v(\nabla u, \nabla v) - (p, \nabla \cdot v) + \alpha(\nabla u_t, \nabla v) = (f, v) \qquad (3.3)$$

$$(\nabla \cdot u, q) = 0, \ \forall q \in Q.$$

In this thesis, since we are looking at accelerating convergence to equilibrium, we need to give the equilibrium problem associated with (3.2). When $t \to \infty$, the following equilibrium problem is to find u_∞, p_∞ satisfying

$$u_\infty \cdot \nabla u_\infty - \nu \Delta u_\infty + \nabla p_\infty = f(x) \text{ for } x \in \Omega \qquad (3.4)$$

$$\nabla \cdot u_\infty = 0, \ x \in \Omega$$

$$u_\infty = 0, \text{ on } \partial\Omega,$$

$$\int_\Omega p_\infty dx = 0.$$

Finite element approximations u_∞^h and p_∞^h satisfy the equations

$$v(\nabla u_\infty^h, \nabla v^h) + b^*(u_\infty^h, u_\infty^h, v^h) - (p_\infty^h, \nabla \cdot v) = (f_\infty, v^h) \ \forall v^h \in X$$

$$(\nabla \cdot u_\infty^h, q^h) = 0 \ \forall q^h \in Q$$

and the variational formulation of equilibrium problem is, find u_∞, p_∞ such that

$$v(\nabla u_\infty, \nabla v) + b^*(u_\infty, u_\infty, v) - (p_\infty, \nabla \cdot v) = (f_\infty, v) \ \forall v \in X$$

$$(\nabla \cdot u_\infty, q) = 0 \ \forall q \in Q$$

15

3.3   FULLY-DISCRETE APPROXIMATION

The semi-discrete FEM reduces the NSV model to a large, stiff system of ordinary differential equations in time. This system must still be solved by an appropriate time-stepping scheme. Now, let us consider a full-discretization of (3.2) via Backward Euler linearization time-stepping by following a similar development as for the semi-discrete formulation. Let $k = \Delta t > 0$ be time-step. The algorithm given below computes $u_1^h, u_2^h, \ldots, p_1^h, p_2^h, \ldots$ where $u_j^h(x) \cong u(x, t_j)$, $p_j^h(x) \cong p(x, t_j)$ and $t_j = jk$.

Algorithm 9. *(BELFEM) For each $n = 1, 2, \ldots, M - 1$, find $(u_{n+1}^h, p_{n+1}^h) \in X^h \times Q^h$ satisfying*

$$
\left(\frac{u_{n+1}^h - u_n^h}{\Delta t}, v^h \right) + \alpha \left(\frac{\nabla u_{n+1}^h - \nabla u_n^h}{\Delta t}, \nabla v^h \right) \tag{3.5}
$$
$$
+ b^* \left(u_n^h, u_{n+1}^h, v^h \right) + v \left(\nabla u_{n+1}^h, \nabla v^h \right) - \left(p_{n+1}^h, \nabla \cdot v^h \right)
$$
$$
= \left(f_{n+1}, \nabla \cdot v^h \right)
$$
$$
\left(\nabla \cdot u_{n+1}^h, q^h \right) = 0, \quad \forall q^h \in Q^h.
$$

In the next chapter, we will try to specify whether adding the term " $- \alpha \Delta u_t$" increase the accelerating convergence to steady state or not. We also will try to specify the statistical equilibrium and spin-up problem, i.e., adding " $- \alpha \Delta u_t$" to NSE satisfies $\left\| \dfrac{u_{n+1} - u_n}{\Delta t} \right\| < \epsilon$ *(tolerance)* with the time reduction faster than NSE.

3.3.1   Accelerating convergence to steady-state

Although the focal point of this thesis lies on more numerical approach. However we find it essential to include the theory that highlights the convergence rate analysis of to steady state solution of NSE and NSV. If we do time discretization of 2.3 we get

$$
\frac{u^{n+1} - u^n}{\Delta t} + u^{n+1} \cdot \nabla u^{n+1} - \nu \Delta u^{n+1} + \nabla p^{n+1} = f^{\,n+1} \tag{3.6}
$$

To find the convergence rate to steady-state solution of NSE, let's subtract (3.6) from (3.4), adding subtracting the term $u^{n+1} \cdot \nabla u_\infty$ and writing $e^{n+1} = u_{n+1} - u_\infty$ give us

$$\frac{e^{n+1} - e^n}{\Delta t} + u_{n+1} \cdot \nabla e^{n+1} + e^{n+1} \cdot \nabla u_\infty - v\Delta e^{n+1} + \nabla p_{n+1} - \nabla p_\infty = 0.$$

Multiplying by e^{n+1} gives

$$\left(\frac{e^{n+1}-e^n}{\Delta t}, e^{n+1}\right) + \left(u_{n+1} \cdot \nabla e^{n+1}, e^{n+1}\right) + \left(e^{n+1} \cdot \nabla u_\infty, e^{n+1}\right)$$

$$-v(\Delta e^{n+1}, e^{n+1}) + (\nabla p_{n+1}, e^{n+1}) - (\nabla p_\infty, e^{n+1}) = 0$$

Using skew-symmetry $(u \cdot \nabla v, w) = -(u \cdot \nabla w, v)$ and the trick that $x^2 - xy = \frac{x^2 - y^2 + (x-y)^2}{2}$,

$$\frac{\left\|e^{n+1}\right\|^2 - \left\|e^n\right\|^2 + \left\|e^{n+1} - e^n\right\|^2}{2\Delta t} + \left(e^{n+1} \cdot \nabla u_\infty, e^{n+1}\right) \tag{3.7}$$

$$-v\left(\Delta e^{n+1}, e^{n+1}\right) + \left(\nabla (p_{n+1} - p_\infty), e^{n+1}\right) = 0$$

By the divergence theorem

$$(\Delta u, v) = \int_\Omega \Delta u \cdot v.dx = \int_\Omega \nabla u \cdot \nabla v dx + \int_{\partial\Omega} \nabla u v \tilde{n} ds.$$

Since $u = 0$ on the boundry $\partial\Omega$,

$$(\Delta u, v) = \int_\Omega \nabla u \cdot \nabla v dx \tag{3.8}$$

Similarly, since u vanishes on $\partial\Omega$

$$(\nabla p, v) = \int_\Omega \nabla p \cdot v dx = -\int_\Omega p \nabla \cdot v dx. = 0 \tag{3.9}$$

Deleting $\left\|e^{n+1} - e^n\right\|^2$, using (3.8) and (3.9) equation becomes

$$\frac{\left\|e^{n+1}\right\|^2}{2\Delta t} + v\left(\nabla e^{n+1}, \nabla e^{n+1}\right) \leq -\left(e^{n+1} \cdot \nabla u_\infty, e^{n+1}\right) + \frac{\left\|e^n\right\|^2}{2\Delta t}.$$

Using continuity of the trilinear form,

$$|u \cdot \nabla v, w| \leq N \left\|\nabla u\right\| \left\|\nabla v\right\| \left\|\nabla w\right\|, \quad N \text{ is a finite constant}$$

and the bound on u^∞

$$\left\|\nabla v\right\| \leq \nu^{-1} \left\|f\right\|_* \quad \text{where } \left\|f\right\|_* = \sup_{v \in V} \frac{(f, v)}{\left\|\nabla v\right\|}$$

inequality holds and become

$$\frac{\left\|e^{n+1}\right\|^2}{2\Delta t} + v\left\|\nabla e^{n+1}\right\|^2 \le Nv^{-1}\left\|f\right\|_* \left\|\nabla e^{n+1}\right\|^2 + \frac{\left\|e^n\right\|^2}{2\Delta t}.$$

rearranging them

$$\left\|e^{n+1}\right\|^2 + v2\Delta t\left(1 - \frac{N}{v^2}\left\|f\right\|_*\right)\left\|\nabla e^{n+1}\right\|^2 \le \left\|e^n\right\|^2. \tag{3.10}$$

Using Poincaré–Friedrichs' inequality

$$\left\|u\right\| \le C_{pf}\left\|\nabla u\right\| \text{ where } C_{pf} \text{ is a positive constant}$$

and under the small data condition, $\frac{N}{v^2}\left\|f\right\|_* \le \eta < 1$ and $1 - \frac{N}{v^2}\left\|f\right\|_* > 0$,

$$\Rightarrow \quad \left\|e^{n+1}\right\|^2 + \Delta t\gamma\left\|e^{n+1}\right\|^2 \le \left\|e^n\right\|^2 \quad , \quad \gamma = 2v\left(1 - \frac{N}{v^2}\left\|f\right\|_*\right)C_{pf}^{-2} \tag{3.11}$$

$$\Rightarrow \quad \left\|e^{n+1}\right\|^2 \le \left(\frac{1}{1+\gamma\Delta t}\right)\left\|e^n\right\|^2 \tag{3.12}$$

$$\Rightarrow \quad \left\|e^{n+1}\right\| \le \left(\frac{1}{1+\gamma\Delta t}\right)^{n/2}\left\|e^0\right\| \quad , \quad \beta = -\frac{\ln\left(\frac{1}{1+\gamma\Delta t}\right)}{\Delta t} > 0,$$

$$\Rightarrow \quad \left\|e^{n+1}\right\| \le e^{-\frac{n\Delta t\beta}{2}}\left\|e^0\right\|.$$

Thus, the error sequence converge to zero with the speed $O(e^{-\frac{n\Delta t\beta}{2}})$.

Let us construct similar steps for (3.2). Time discretization of (3.2) is

$$\frac{u_{n+1} - u_n}{\Delta t} + u_{n+1}\cdot\nabla u_{n+1} - v\Delta u_{n+1} + \nabla p_{n+1} - \alpha\Delta\left(\frac{u_{n+1} - u_n}{\Delta t}\right) = f$$

$$\nabla\cdot u_{n+1} = 0.$$

we only have extra term

$$-\alpha\Delta\left(\frac{u_{n+1} - u_n}{\Delta t}\right)$$

Using (3.8), the same trick in (3.7) and the fact that $e^{n+1} = u_{n+1} - u_\infty$ and then multiplying e^{n+1} finally gives

$$-\alpha\Delta\left(\frac{u_{n+1} - u_n}{\Delta t}\right) = \alpha\frac{\left\|\nabla e^{n+1}\right\|^2 - \left\|\nabla e^n\right\|^2 + \left\|\nabla(e^{n+1} - e^n)\right\|^2}{2\Delta t}$$

18

If we add this to inequality in (3.10), the inequality in (3.11) becomes

$$\left\| e^{n+1} \right\|^2 + \Delta t \gamma \left\| e^{n+1} \right\|^2 + \alpha(\left\| \nabla e^{n+1} \right\|^2 - \left\| \nabla e^n \right\|^2 + \left\| \nabla(e^{n+1} - e^n) \right\|^2) \leq \left\| e^n \right\|^2 \qquad (3.13)$$

If the term

$$\alpha(\left\| \nabla e^{n+1} \right\|^2 - \left\| \nabla e^n \right\|^2 + \left\| \nabla(e^{n+1} - e^n) \right\|^2) \qquad (3.14)$$

is positive each time step, then we can write

$$\alpha(\left\| \nabla e^{n+1} \right\|^2 - \left\| \nabla e^n \right\|^2 + \left\| \nabla(e^{n+1} - e^n) \right\|^2) = \sigma \left\| e^{n+1} \right\|^2, \ \sigma > 0$$

then (3.13) will be

$$(1 + \sigma) \left\| e^{n+1} \right\|^2 + \Delta t \gamma \left\| e^{n+1} \right\|^2 \leq \left\| e^n \right\|^2$$

and finally

$$\left\| e^{n+1} \right\|^2 \leq \left(\frac{1}{1 + \sigma + \gamma \Delta t} \right) \left\| e^n \right\|^2 .$$

Since the coefficient in the right hand side is smaller than the coefficient $(\frac{1}{1+\gamma \Delta t})$ in (3.12), the rate of the convergence to zero is faster, i.e., $u_{n+1}^{NSV} \to u_\infty$ is faster than $u_{n+1}^{NSE} \to u_\infty$. If the term in (3.14)

$$\alpha(\left\| \nabla e^{n+1} \right\|^2 - \left\| \nabla e^n \right\|^2 + \left\| \nabla(e^{n+1} - e^n) \right\|^2)$$

is not positive for each step, it cannot be guaranteed for accelerating converge to equilibrium and statistical equilibrium. Next chapter we will see this in numerical experiments.

4.0 NUMERICAL EXPERIMENTS AND RESULTS

In this chapter, we test some examples to see behavior of convergence of our model. We use the software $FreeFEM++([17])$ to run numerical tests with taking tolerance $\epsilon = 10^{-6}$ as a stopping criterion. We give the $FreeFEM++$ codes for each example and results of our test are given with tables and supported with the figures of velocity and pressure. Since we also look some problems with known solutions we will look at 4 tests:

$$
\begin{aligned}
\left\|u^h - u_\infty\right\| &< \epsilon & \text{(Test 1)} \\
\left\|\frac{u^{n+1} - u^n}{\triangle t}\right\| &\leq \epsilon & \text{(Test 2)} \\
\left\|\frac{u^{n+1} - u^n}{\triangle t}\right\| + \left\|\nabla\frac{u^{n+1} - u^n}{\triangle t}\right\| &\leq \epsilon & \text{(Test 3)} \\
\left|\frac{\frac{1}{2}||u(T')||^2 - \frac{1}{2}||u_0||^2}{T'} + \alpha\frac{\frac{1}{2}||\nabla u(T')||^2 - \frac{1}{2}||\nabla u_0||^2}{T'}\right| &\leq \epsilon & \text{(Test 4)}
\end{aligned}
$$

All tests are calculated in L^2 $norm$. Tests are made with different values of $Reynolds$ $numbers$, h and dt. Tables summarize the most convenient results so that one can check the results easily.

4.1 THE PROBLEM 1

First problem is tested for a known exact solution in a square domain $\Omega = (-1, 1) \times (-1, 1)$ where the boundary approximation is exact. The uniform mesh is obtained by dividing Ω into squares and drawing a diagonal each square in same direction. We choose the exact solution by:

20

$$
\begin{aligned}
u_1(x, y, t) &= e^{-2vt}(-\cos(x)\sin(y)) \\
u_2(x, y, t) &= e^{-2vt}\sin(x)\cos(y) \\
p(x, y, t) &= 0
\end{aligned}
$$

where initial conditions are for $t = 0$:

$$
u_0 = (-\cos(x)\sin(y), \sin(x)\cos(y))
$$

and the equilibrium for $t \to \infty$

$$
u_\infty(x, y) = 0
$$

Here is the code for Square Domain with true solution of NSE:

```
///————————————————————————————————————
/* SquareExact.edp –linearization with Backward Euler Time Discrization
u_t + u .* grad(u) - nu div(grad(u)) + grad(p0) = f,
div(u)=0 in omega
u=(exp(-2*nu*t)*(-cos(x)*sin(y)),exp(-2*nu*t)*(sin(x)*cos(y))) on the boundry
time discrization;
(U(n+1)-U(n))/dt+U(n).*grad[U(n+1)]-nu*div(grad[U(n+1)]+grad[P(n+1)]=f(n+1)
Ali Koseoglu
*/
//*****************************************************************//
//*****************************************************************//
real T0 = 0.0; // initial time
real Tf = 10.0; // final time
real dt = 0.1; // time step size
int maxits=(Tf-T0)/dt;
real t; // time step counter
real Re=1.; // Reynolds
real nu = 1./Re; // viscosity
```

```
real alpha=0.; // accelerating parameter
real tol=1.0e-6;
real energyerror,stepspinup,spinuperror,energyerror1,spinuperror1;
real eqerror,eqgraderror,eqerror1,eqgraderror1;
real step,stepeq,stepgradeq;
int numsteps,numstepseq,numstepsgradeq,c;
int numstepspinup;
int a1=1;
int a2=1;
int a3=1;
int a4=1;
string test="not small enough or blow up";
// define macros
macro grad(u) [dx(u),dy(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //
macro dot(u1,u2,v1,v2) (u1*v1+u2*v2) //
macro ugradv1(u1,u2,v1) (u1*dx(v1)+u2*dy(v1)) //
macro cc(u1,u2,v1,v2,w1,w2) (ugradv1(u1,u2,v1)*w1+ugradv1(u1,u2,v2)*w2) //
macro cch(u1,u2,v1,v2,w1,w2) (0.5*(cc(u1,u2,v1,v2,w1,w2)-cc(u1,u2,w1,w2,v1,v2))) //
macro contract(u1,u2,v1,v2)(dx(u1)*dx(v1)+dx(u2)*dx(v2)
                            +dy(u1)*dy(v1)+dy(u2)*dy(v2))//
// define the triangulated mesh
mesh Th=square(20,20,[2*x-1,2*y-1]); //h=1/10
//plot(Th,wait=1,ps="Th.eps",value=true);
func f1 = 0;
func f2 = 0;
func g1 =exp(-2*nu*t)*(-cos(x)*sin(y));
func g2 =exp(-2*nu*t)*(sin(x)*cos(y));
func uinf1=0;
func uinf2=0;
```

```
// create the FE velocity space Vh of continuous piecewise quadratics
//and pressure space Ph of continuous piecewise linears
fespace Vh(Th,P2);
fespace Ph(Th,P1);
// define the FE functions
Vh u1,u2,u1old,u2old,v1,v2,u1err,u2err,U01,U02,u1ex,u2ex;
Ph p,q;
// define the variational formulation of NSE with adding alpha
problem NSE([u1,u2,p],[v1,v2,q]) =
int2d(Th)(
(1/dt)*dot(u1,u2,v1,v2)
+(1/dt)*alpha*contract(u1,u2,v1,v2)
+ cch(u1old,u2old,u1,u2,v1,v2)
+ nu*contract(u1,u2,v1,v2)
- p*div(v1,v2)
+ q*div(u1,u2))
-int2d(Th)(
(1/dt)*dot(u1old,u2old,v1,v2)
+(1/dt)*alpha*contract(u1old,u2old,v1,v2)
+ dot(f1,f2,v1,v2))
+ on(1,2,3,4,u1=g1,u2=g2);
u1 =-cos(x)*sin(y);
u2 = sin(x)*cos(y);
U01 =-cos(x)*sin(y);
U02 = sin(x)*cos(y);
// begin time stepping loop
c=0;
t=T0;
while (t < Tf)
{
```

```
u1old = u1;

u2old = u2;

t = t+dt;

// solve NSE output=[u1,u2,p]

NSE;

//

u1err=u1-u1old;

u2err=u2-u2old;

u1ex=abs(uinf1-u1);

u2ex=abs(uinf2-u2);

if (t>1)

{

eqerror =(1/dt)*sqrt(int2d(Th)(dot(u1err,u2err,u1err,u2err)));

eqgraderror=(1/dt)*sqrt(int2d(Th)(contract(u1err,u2err,u1err,u2err))) + eqerror;

energyerror=(1/t)*0.5*(int2d(Th)(dot(u1,u2,u1,u2))

            -int2d(Th)(dot(U01,U02,U01,U02)))

              +(1/t)*0.5*alpha*(int2d(Th)(contract(u1,u2,u1,u2))

              -int2d(Th)(contract(U01,U02,U01,U02)));

spinuperror=sqrt(int2d(Th)(u1ex^2+u2ex^2));

if ((abs(energyerror) < tol )&(energyerror!=0)&(a1==1))

{     numsteps=c;

 energyerror1=abs(energyerror);

              a1=2;}

if ((eqerror<tol)&(eqerror!=0)&(a2==1))

{ numstepseq=c;

    eqerror1=eqerror;

            a2=2;}

if ((eqgraderror < tol )&(eqgraderror!=0)&(a3==1))

{       eqgraderror1=eqgraderror;

    numstepsgradeq=c;
```

```
                              a3=2;}
    if ((spinuperror < tol )&(eqgraderror!=0)&(a4==1))
    { numstepspinup=c;
        spinuperror1=spinuperror;
                    a4=2;}
    }
    c=c+1;
    plot([u1,u2],wait=0,value=true);
    //if((t>1)&(spinuperror<tol)&(eqerror<tol)&(energyerror < tol )&(eqgraderror < tol
))
    //{break;}
    } // end while loop
    // plot the computed pressure and velocity
    plot(p,fill=1,wait=1,ps="pressurefinal1.eps",value=true);
    plot([u1,u2],wait=1,ps="velocityfinal1.eps",value=true);
    // print number of steps and values
    cout << "**********************************************************"<<endl;
    cout << "Time Interval = [0,"<<Tf<<"] dt="<<dt<<" max iterations="<<maxits
<<endl;
    if (energyerror1==0)
    {cout << "energy error for < Tol =" << test <<endl;
    numsteps=c;}
    else
    {cout << "energy error for < Tol =" << energyerror1 <<endl;}
    cout << "energy error at "<<Tf<<" is =" << abs(energyerror) <<endl;
    if (eqerror1==0)
    {cout << "equilubrum error for <Tol =" << test <<endl;
    numstepseq=c;}
    else
    {cout << "equilubrum error for <Tol =" << eqerror1 <<endl;}
```

25

cout << "equilubrum error at "<<Tf<<" is =" << eqerror <<endl;

if (eqgraderror1==0)

{cout << "equilubrumgrad error for < Tol=" << test <<endl;

numstepsgradeq=c;}

else

{cout << "equilubrumgrad error for < Tol =" << eqgraderror1 <<endl;}

cout << "equilubrumgrad error at "<<Tf<<" is =" << eqgraderror <<endl;

if (spinuperror1==0)

{cout << "spin-up error for < Tol =" << test <<endl;

numstepspinup=c;}

else

{cout << "spin-up error for < Tol =" << spinuperror1 <<endl;}

cout << "spin-up error at "<<Tf<<" is =" << spinuperror <<endl;

cout << "****if steps = "<<c<<" then max steps is exceeded!!! ****" <<endl;

cout << "number of steps for energy error steps=" <<numsteps<<endl;

cout << "number of steps for equilibrum steps=" <<numstepseq<<endl;

cout << "number of steps for equilibrumgrad steps=" <<numstepsgradeq<<endl;

cout << "stopping time for spinup error steps=" <<numstepspinup<<endl;

We ran the code for different values of h and α, starting with $h = \frac{1}{10}$ and $\alpha = 0$ where $\frac{1}{10} < h < \frac{1}{40}$ and $0 \leq \alpha \leq 1$. For different Re, different *time intervals* such as $[0, 10]$ to $[0, 80]$ and different dt where 0.1 to 0.0001. For this test all we got is, for the values of $\alpha > 0$, number of steps to reach equilibrium and statistical equilibrium are increasing. We give the table of the number of steps for equilibrium tests and values of $L^2 - norm$ of the energy. We pick the values for *final time* = 20, $dt = 0.1$, Re = 1 and viscosity $v = 1$ for the Table 1. Here, *tolerance* = 10^{-6} and for Test 4, it is the value of Test 4 at *final time*. Note that for $\alpha = 0$, it is the model of NSE in (2.3).

Table 1: Test results for Problem 1, h=1/10

value	number of steps			Value
α	$Test\ 1$	$Test\ 2$	$Test\ 3$	$Test\ 4$
0	70	74	79	0.0396647
0.1	70	74	79	0.0517317
0.2	70	74	79	0.0637988
0.3	108	112	119	0.0758658
0.4	115	119	130	0.0879329
0.5	165	169	181	0.0999999
0.6	93	96	107	0.112067
0.7	114	116	135	0.124134
0.8	116	117	131	0.136201
0.9	148	148	170	0.148268
1	exceeded	exceeded	exceeded	0.155149

As we see in Table 1, as the α is getting bigger, number of steps that NSV model reaches equilibrium and value of statistical equilibrium are clearly increasing. After trying for small and big α's we concluded that convergence is getting worse as the α increases. Later, we tried it for small h and the results were very similar, α makes convergence the worse. We will see it also in the plots.

The plots of the computed pressure and velocity for each value of h at the final time are displayed in the following figures starting with Figure 2.
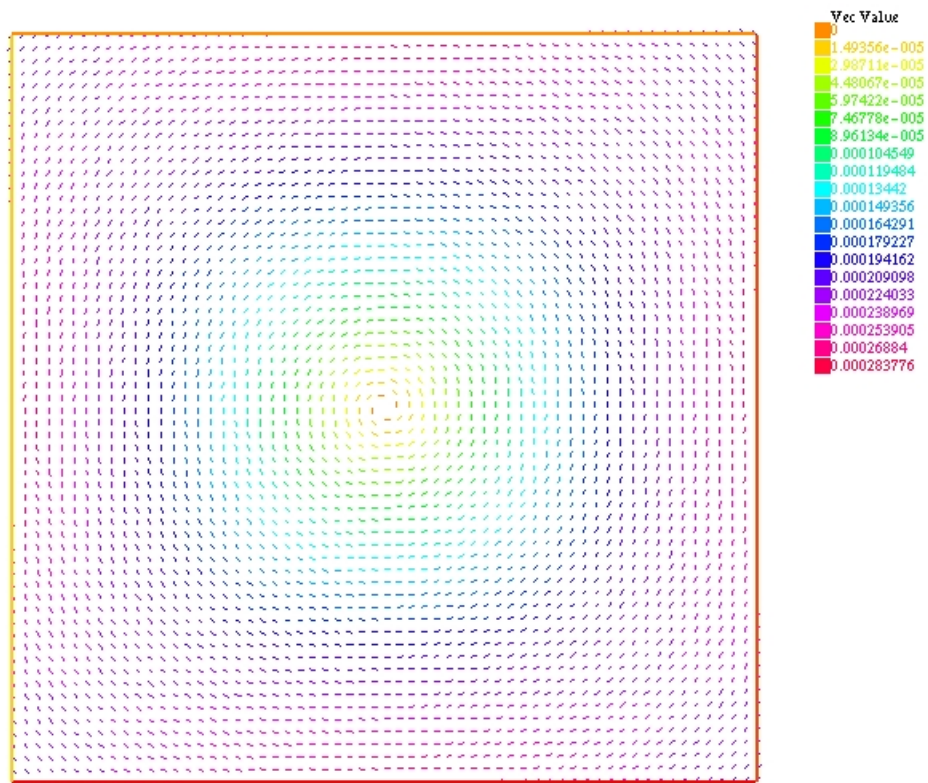

Figure 2: Problem 1, h =1/10, $alpha = 0$, Computed Velocity

Here we see the velocity vectors for the NSE as in(2.3). Velocity vectors are getting smaller and the NSE reaches equilibrium. Notice that the "*vec value*"s at the right top of the picture. We are expecting that these values will increase as the α is getting bigger.
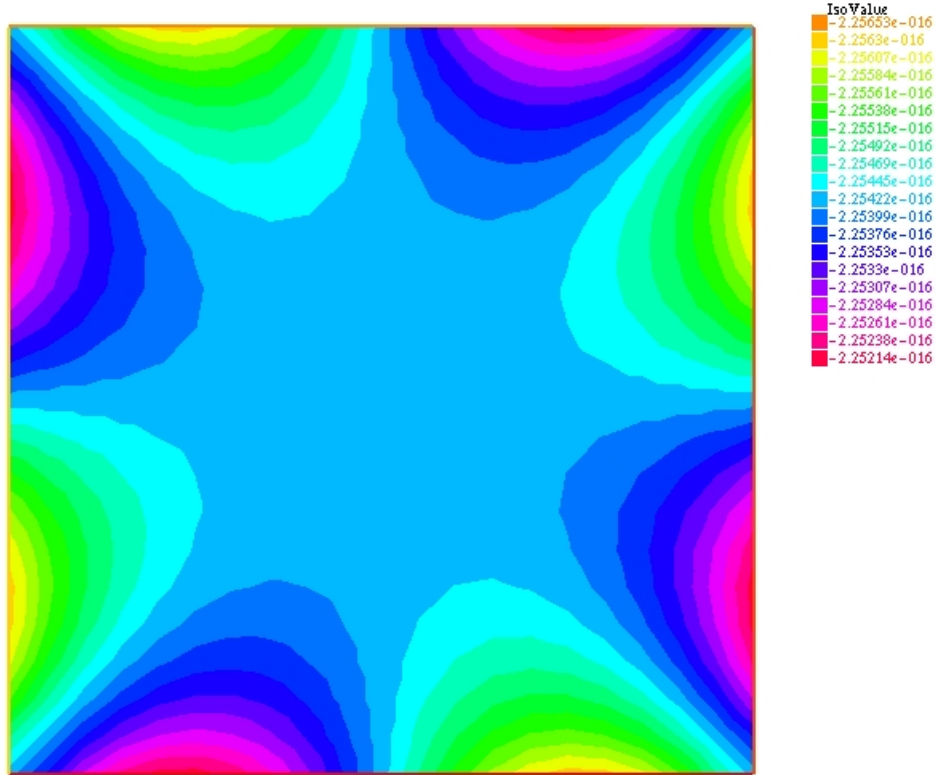
IsoValue
-2.25653e-016
-2.2563e-016
-2.25607e-016
-2.25584e-016
-2.25561e-016
-2.25538e-016
-2.25515e-016
-2.25492e-016
-2.25469e-016
-2.25445e-016
-2.25422e-016
-2.25399e-016
-2.25376e-016
-2.25353e-016
-2.2533e-016
-2.25307e-016
-2.25284e-016
-2.25261e-016
-2.25238e-016
-2.25214e-016

Figure 3: Problem 1, h =1/10, $alpha = 0$, Computed Pressure

This figure is the computed pressure p for the NSE model. The reason that plot is like square-shaped is the value of h. Also notice that the values of "$Iso\ value$" at the top right of the picture. We are expecting that these values will increase as the α is getting bigger though h is getting smaller.
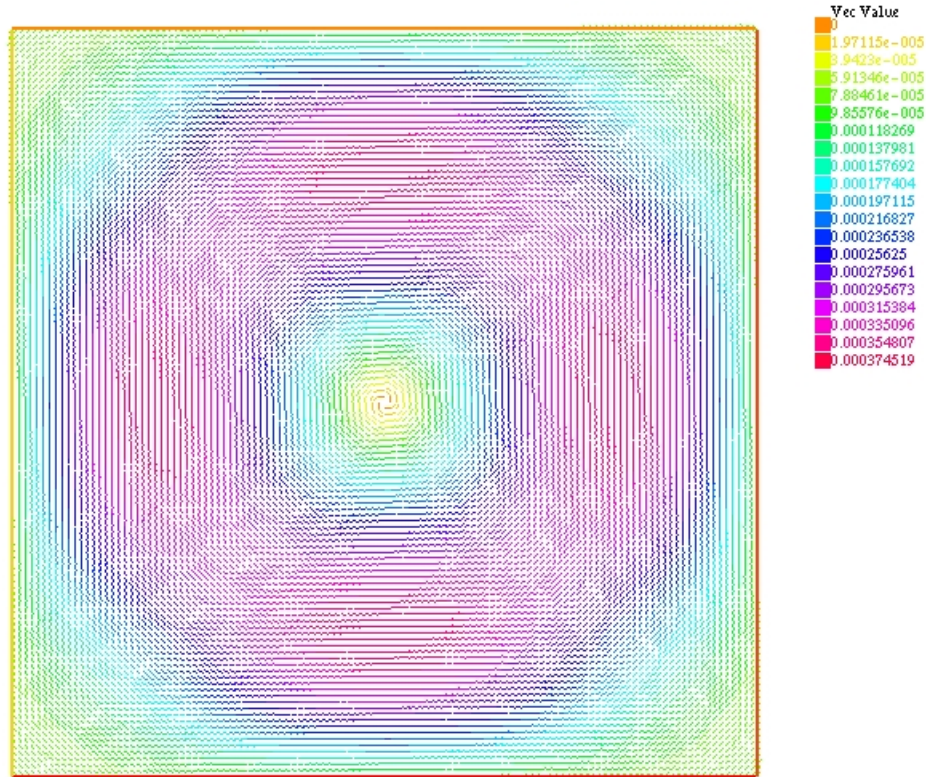
Figure 4: Problem 1, h =1/20, $alpha = 0.5$, Computed Velocity

We see the velocity vectors for the NSV as in(3.2). As we see here the velocity vectors getting smaller, the NSV reaches equilibrium. Note that the density of the vectors. When h is getting smaller, figure comes out more detailed. There is also one thing we should pay attention the "*vec values*" right top of the picture. Although h is twice smaller, values of velocity vectors increasing when we compare to $Figure$ 2. We will see it other pictures too.
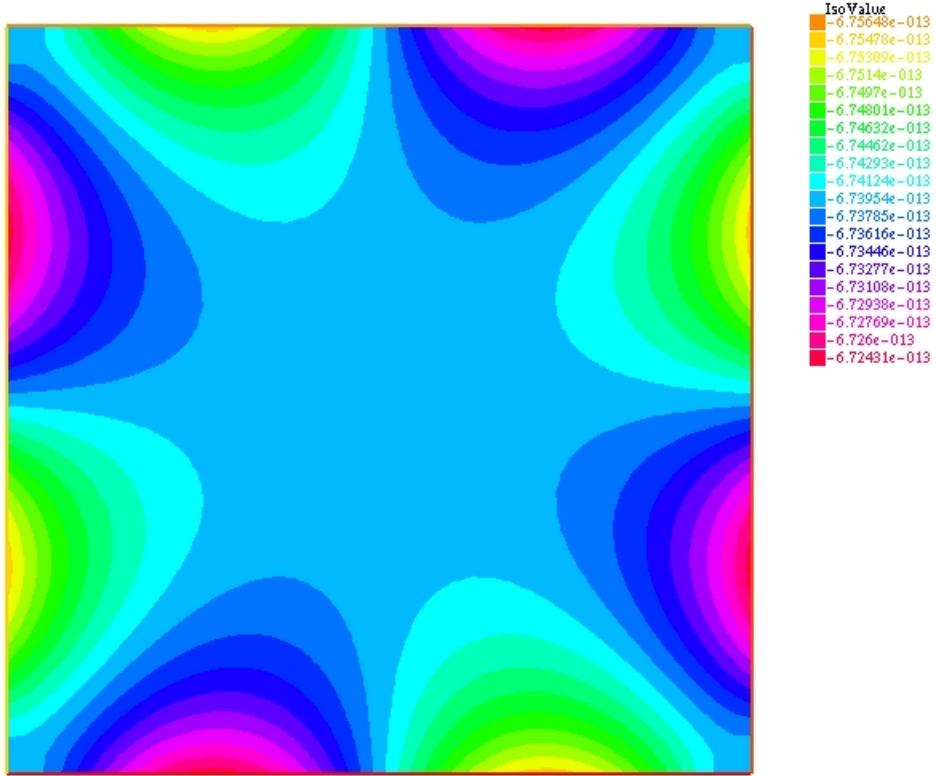
Figure 5: Problem 1, h =1/20, $alpha = 0.5$, Computed Pressure

This figure is the computed pressure p for NSV model for $h = \frac{1}{10}$. Since the value of h is twice smaller, the plot comes out smoother. Besides, if we look at the "Iso Values", they are bigger than the $Figure$ 3. If α was 0, because of the h, these values would be smaller, but this α make them bigger which is also we can see from the picture clearly
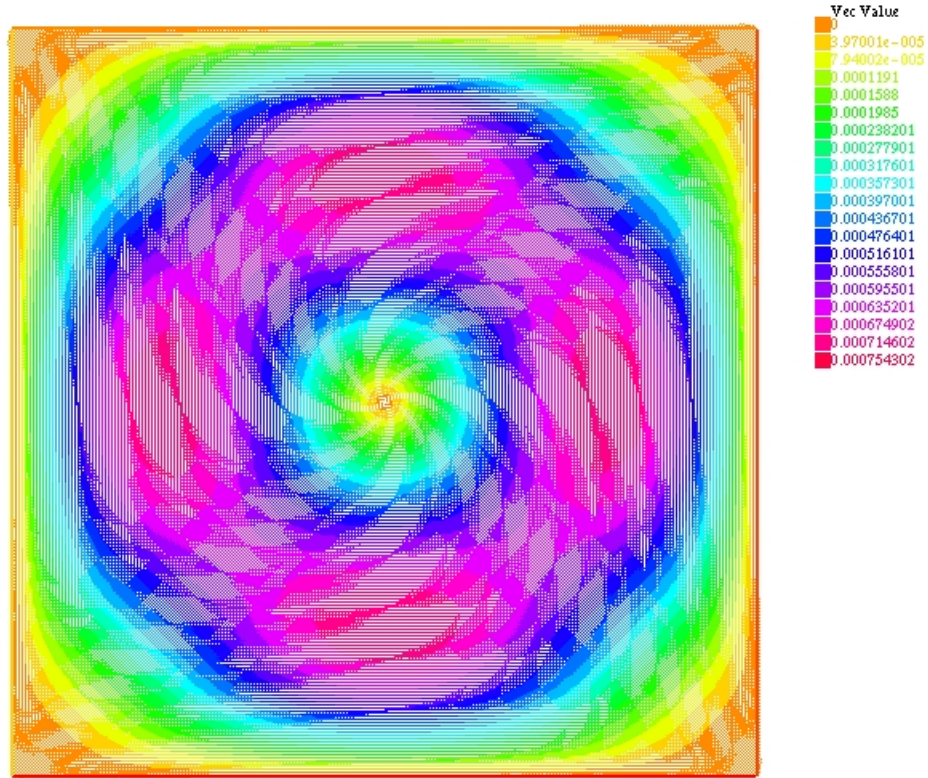
Figure 6: Problem 1, h =1/40, $alpha = 1$, Computed Velocity

We see the velocity vectors for the NSV as in(3.2). As we see here, the velocity vectors getting smaller, the NSV reaches equilibrium but a little slower. Note that the density of the vectors, when h is getting smaller, figure comes out more detailed but "$vec\ values$" are getting bigger when we compare to $Figure\ 4$ because of α. If we pay attention the value of $h = \frac{1}{40}$, we can clearly see that α makes convergence worse.
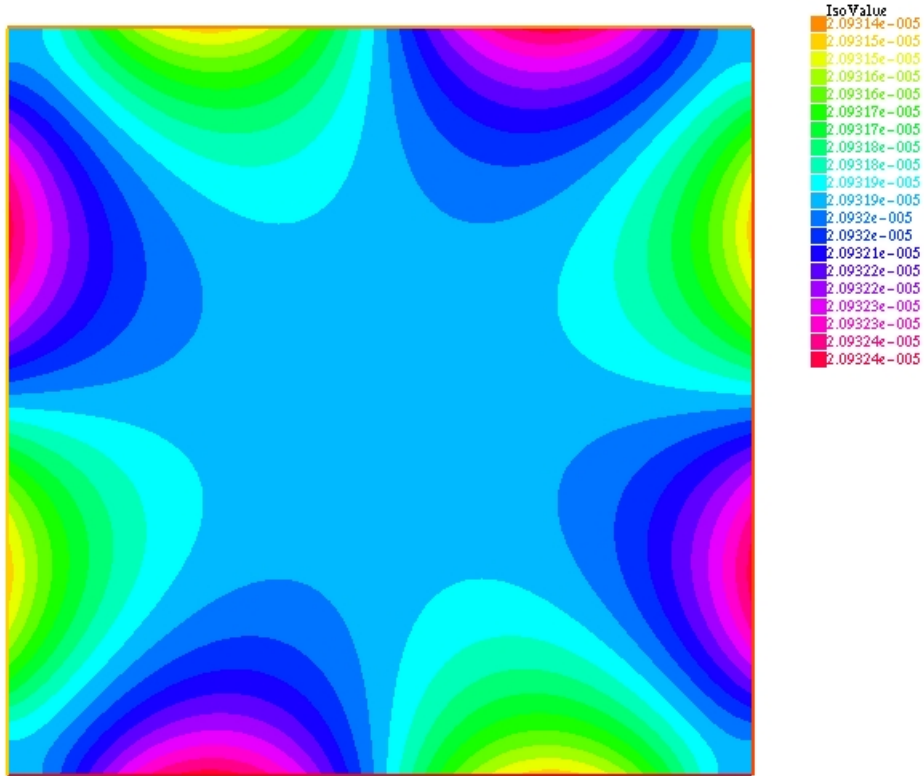
Figure 7: Problem 1, h =1/40, $alpha = 1$, Computed Pressure

This figure is the computed pressure p for NSV model for the $\alpha = 1$. Since the value of h is smaller, the plot comes out with perfect smoothness. Also the Iso value's are getting bigger which means that big α makes convergence worse no matter pressure or velocity vectors and no matter h.

4.2   THE PROBLEM 2

Second problem is tested for a true solution in a square domain $\Omega = (-1, 1) \times (-1, 1)$ with periodic boundary conditions. Our report in this case is performed using Taylor-Green vortices on the square as a true solution, Taylor-Green vortices are a solution of the NSE with driving force $f = 0$, given by:

$$
\begin{aligned}
u_1(x, y, t) &= e^{-2N^2\pi^2 vt}(-\cos(N\pi x)\sin(N\pi y)) \\
u_2(x, y, t) &= e^{-2N^2\pi^2 vt}\sin(N\pi x)\cos(N\pi y)) \\
p(x, y, t) &= -\frac{1}{4}e^{-2N^2\pi^2 vt}\cos(2N\pi x)\cos(2N\pi y)
\end{aligned}
$$

we choose $N = 2$ as in Connors([12]) and $\nu = 1$, where initial conditions are

$$
u_0 = (-\cos(x)\sin(y), \sin(x)\cos(y))
$$

and the equilibrium for $t \to \infty$

$$
\begin{aligned}
u_\infty(x, y) &= 0 \\
p_\infty(x, y) &= 0
\end{aligned}
$$

Code is same as Problem 1, just changing the functions. We picked longer time scales such as $[0, 20], [0, 30]$ and $[0, 40]$ with $dt = 0.01$. For these time scales, the bigger $0 \le \alpha \le 1$ and $100 \le Re \le 1000$ makes convergence worse. Mostly, our $tests$ exceeded the max number of steps and then blow up, so we give the results for time interval $[0, 10], 0 \le \alpha \le 0.5, Re = 1$ and $dt = 0.01$ to see progress with Table 2.

Table 2: Test results for Problem 2, h=1/10, T=10,

value	number of steps			Value
α	$Test\ 1$	$Test\ 2$	$Test\ 3$	$Test\ 4$
0	69	90	103	0.0997447
0.1	455	484	510	0.88844
0.2	536	571	615	1.67713
0.3	exceeded	exceeded	exceeded	2.16677
0.4	exceeded	exceeded	exceeded	3.2544
0.5	blow up	blow up	blow up	blow up

In Table 2, the bigger α makes our convergence really bad this time. After the exceeded steps, it makes them blow up. We also wanted to show for one example of long time scale. Table 3 shows results for $time\ interval : [0, 30]$ and $dt = 0.01$

Table 3: Test results for Problem 2, h=1/10, T=30,

value	number of steps			Value
α	$Test\ 1$	$Test\ 2$	$Test\ 3$	$Test\ 4$
0	69	90	103	0.0332815
0.1	455	484	510	0.296443
0.2	536	571	615	0.559604
0.3	1449	1486	1593	blow up
0.4	exceeded	exceeded	exceeded	blow up
0.5	blow up	blow up	blow up	blow up

As we see in Table 3, nothing much change. After $\alpha = 0.5$ we could not find a value which does not cause a blow up.

We want to give some figures to show velocity vectors when $h = \frac{1}{10}, \frac{1}{20}$ and $\alpha = 0, 0.2, 0.4$.
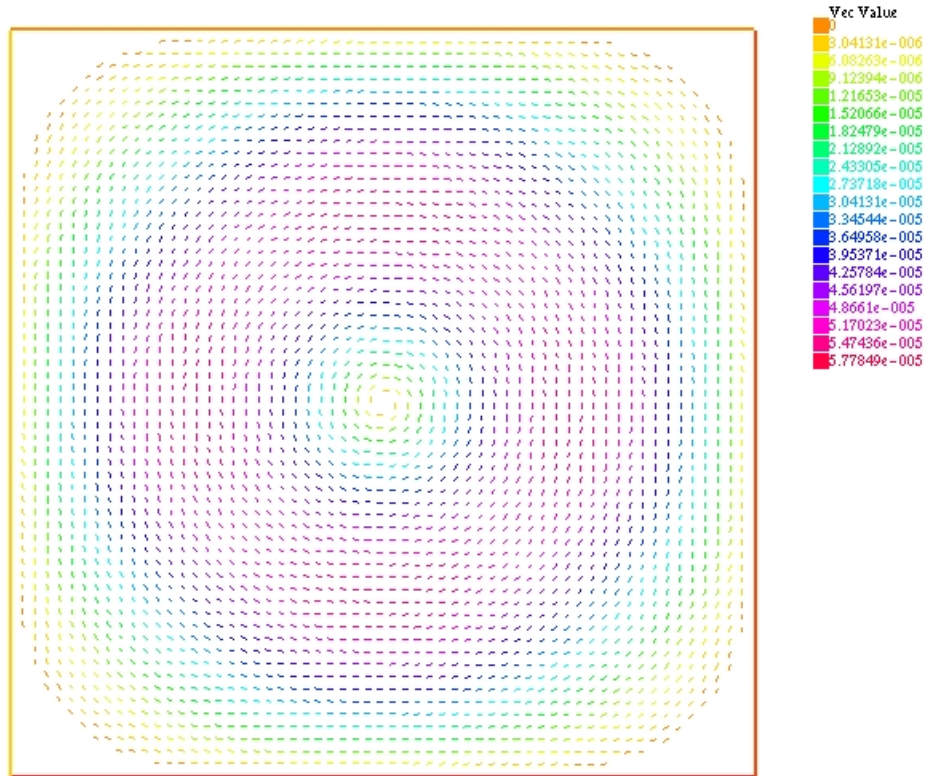

Figure 8: Problem 2, Re=1, h =1/10, $alpha = 0$, Computed Velocity

When $\alpha = 0$ and $h = \frac{1}{10}$, as we see the picture, velocity vectors go to equilibrium. Next picture, we will see that how the figure changes when h=$\frac{1}{20}$ and $\alpha = 0.2$.
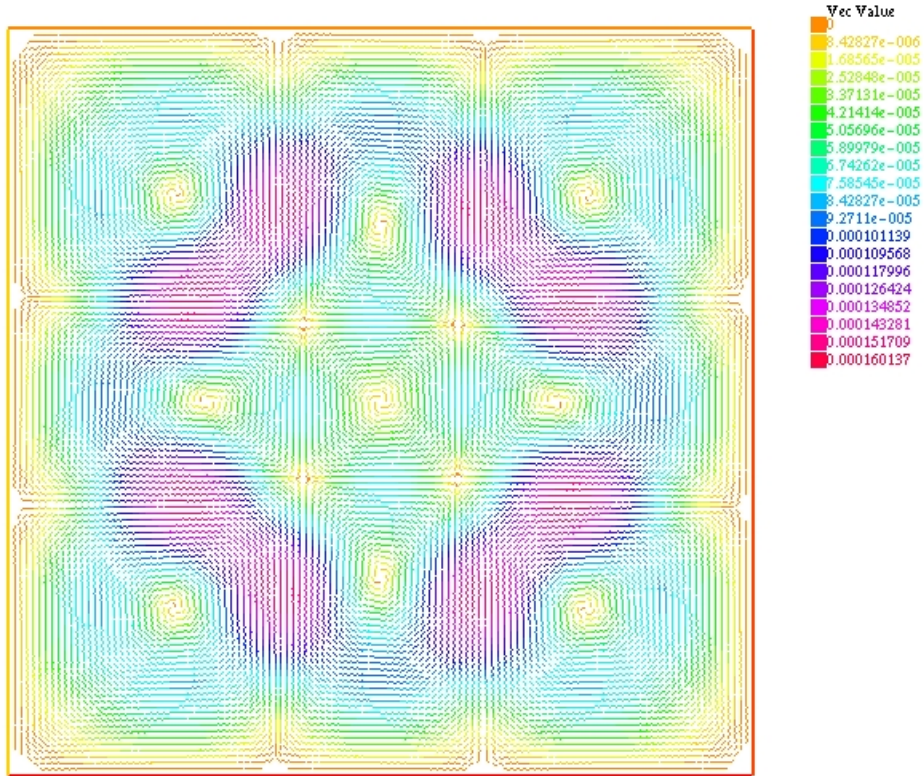
Figure 9: Problem 2, Re=1, h =1/20, $alpha = 0.2$, Computed Velocity

Very small increase of α makes "vec values" bigger than the $\alpha = 0$ NSE case when we decrease the $h = \frac{1}{20}$. When we tried for large α's with this h, the results were worse. In other words, when $h = \frac{1}{20}$ and α is large, say $\alpha > 0.5$, tests are starting to blow up. One can test this with using the code that is given. For convenience, we only gave results for $h = \frac{1}{10}$ in the Table 2 and Table 3.
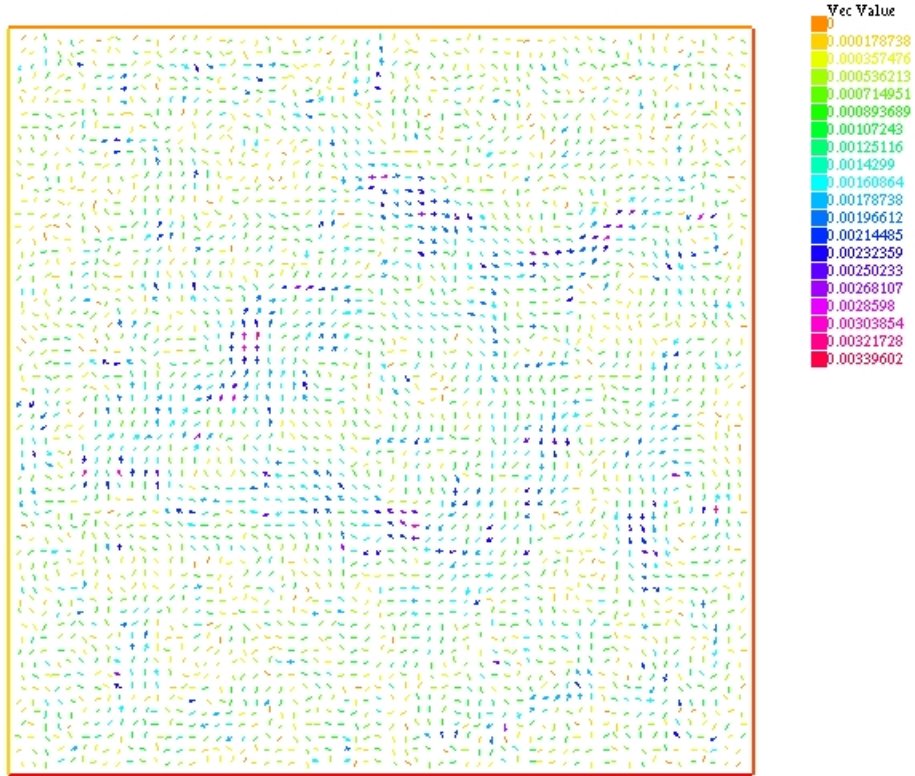
Figure 10: Problem 2, Re=1, h =1/10, $alpha = 0.4$, Computed Velocity at Final Time

As we see here this picture shows us how α changes the values and the velocity vectors. When $T = 10$, the values show us it does not reach the equilibrium for given tolerance $\epsilon = 10^{-6}$ when we compare to values at Figure 8.

4.3   THE PROBLEM 3

The third problem is tested for an unknown solution in a circle domain Ω with $radius = 1$. We choose the initial values for $t = 0$:

$$u_0 = (-y, x),$$

with the body force

$$f(x, y) = 2 < -y, x >$$

38

The code for this problem is:

```
///——————————————————————————————————————
/*CircularNoSolution.edp –linear with Backward Euler Time Discrization
u_t + u .* grad(u) - nu div(grad(u)) + grad(p0) = f,
div(u)=0 in omega
u=0 on the boundary of omega
time discrization;
(U(n+1)-U(n))/dt+U(n).*grad[U(n+1)]-nu*div(grad[U(n+1)]+grad[P(n+1)] =f(n+1)
Ali Koseoglu
//****************************************************************//
//****************************************************************//
real T0 = 0.0; // initial time
real Tf = 10.0; // final time
real dt = 0.1; // time step size
real t;
real reynolds=500.;
real nu =1./reynolds; // viscosity
real alpha=0.;
real tol=1.0e-6;
real energyerror,stepspinup,spinuperror,energyerror1,spinuperror1;
real eqerror,eqgraderror,eqerror1;
real step,stepeq,stepgradeq;
int numsteps,numstepseq,numstepsgradeq,c;
int numstepspinup;
c=0;
int a1=1;
int a2=1;
int a3=1;
int a4=1;
real PRESSURESTABLIZE = 1.0e-12;
```

```
// define macros
macro grad(u) [dx(u),dy(u)] //
macro div(u1,u2) (dx(u1)+dy(u2)) //
macro dot(u1,u2,v1,v2) (u1*v1+u2*v2) //
macro ugradv1(u1,u2,v1) (u1*dx(v1)+u2*dy(v1)) //
macro cc(u1,u2,v1,v2,w1,w2) (ugradv1(u1,u2,v1)*w1+ugradv1(u1,u2,v2)*w2) //
macro cch(u1,u2,v1,v2,w1,w2) (0.5*(cc(u1,u2,v1,v2,w1,w2)-cc(u1,u2,w1,w2,v1,v2)))//
macro contract(u1,u2,v1,v2) (dx(u1)*dx(v1)+dx(u2)*dx(v2)
                            +dy(u1)*dy(v1)+dy(u2)*dy(v2)) //
// define the triangulated mesh
border C(t=0,2*pi){x=cos(t);y=sin(t);}
border obs(t=0,2*pi){x=0.5+0.1*cos(t);y=0.1*sin(t);}
mesh Th = buildmesh(C(50)+obs(-15));
savemesh(Th,"mesh.msh");
func f1 =-2*y;
func f2 = 2*x;
// create the FE velocity space Vh of continuous piecewise quadratics and
pressure space Ph of continuous piecewise linears
fespace Vh(Th,P2);
fespace Ph(Th,P1);
// define the FE functions
Vh u1,u2,u1old,u2old,v1,v2,u1err,u2err,U01,U02,u1ex,u2ex;
Ph p,q,pold;
// define the variational formulation of NSE with adding alpha
problem NSE([u1,u2,p],[v1,v2,q]) =
int2d(Th)(
(1/dt)*dot(u1,u2,v1,v2)
+(1/dt)*alpha*contract(u1,u2,v1,v2)
+ cch(u1old,u2old,u1,u2,v1,v2)
+ nu*contract(u1,u2,v1,v2)
```

```
- p*div(v1,v2)
+ q*div(u1,u2)
+ PRESSURESTABLIZE * p * q)
-int2d(Th)(
(1/dt)*dot(u1old,u2old,v1,v2)
+(1/dt)*alpha*contract(u1old,u2old,v1,v2)
+ dot(f1,f2,v1,v2)
+ PRESSURESTABLIZE * pold * q)
+ on(C,u1=0,u2=0)+on(obs,u1=0,u2=0);
//initial values
u1 =-y;
u2 =x;
U01=-y;
U02=x;
// begin time stepping loop
c=1;
t=0;
while (t < Tf)
{
u1old = u1;
u2old = u2;
pold=p;
t = t+dt;
// solve for [u1,u2,p]
NSE;
//
u1err=u1-u1old;
u2err=u2-u2old;
if (t>1)
{
```

```
eqerror=(1/dt)*sqrt(int2d(Th)(dot(u1err,u2err,u1err,u2err)));

eqgraderror=(1/dt)*sqrt(int2d(Th)(contract(u1err,u2err,u1err,u2err))) + eqerror;

energyerror=(1/t)*0.5*(int2d(Th)(dot(u1,u2,u1,u2))

           -int2d(Th)(dot(U01,U02,U01,U02)))

           + (1/t)*0.5*alpha*(int2d(Th)(contract(u1,u2,u1,u2))

           -int2d(Th)(contract(U01,U02,U01,U02)));

if ((abs(energyerror) < tol )&(energyerror!=0)&(a1==1))

{ numsteps=c;

energyerror1=abs(energyerror);

a1=2;}

if ((eqerror<tol)&(eqerror!=0)&(a2==1))

{ numstepseq=c;

eqerror1=eqerror;

    a2=2;}

    if ((eqgraderror < tol )&(eqgraderror!=0)&(a3==1))

{ eqgraderror1=eqgraderror;

    numstepsgradeq=c;

a3=2;}

}

}

c=c+1;

plot([u1,u2],wait=0,value=true);

//if ((eqerror!=0)&(eqerror<tol)&(energyerror < tol )&(eqgraderror < tol ))

//{break;}

} // end while loop

// plot the computed pressure

plot(p,fill=1,wait=1,ps="pressurefinal1.eps",value=true);

// print number of steps and values

cout << "*******************************************************"<<endl;
```

```
cout << "Time Interval = [0,"<<Tf<<"] dt="<<dt<<" max iterations="<<maxits
<<endl;
    if (energyerror1==0)
    {cout << "energy error for < Tol =" << test <<endl;
    numsteps=c;}
    else
    {cout << "energy error for < Tol =" << energyerror1 <<endl;}
    cout << "energy error at t="<<Tf<<" is =" << abs(energyerror) <<endl;
    if (eqerror1==0)
    {cout << "equilubrum error for <Tol =" << test <<endl;
    numstepseq=c;}
    else
    {cout << "equilubrum error for <Tol =" << eqerror1 <<endl;}
    cout << "equilubrum error at t="<<Tf<<" is =" << eqerror <<endl;
    if (eqgraderror1==0)
    {cout << "equilubrumgrad error for < Tol=" << test <<endl;
    numstepsgradeq=c;}
    else
    {cout << "equilubrumgrad error for < Tol =" << eqgraderror1 <<endl;}
    cout << "equilubrumgrad error at t="<<Tf<<" is =" << eqgraderror <<endl;
    cout << "****if steps = "<<maxits<<" then max steps is exceeded!!! ****" <<endl;
    cout << "number of steps for energy error steps=" <<numsteps<<endl;
    cout << "number of steps for equilibrum steps=" <<numstepseq<<endl;
    cout << "number of steps for equilibrumgrad steps=" <<numstepsgradeq<<endl;
```

We ran the code for different values of h and α, starting with $h = 0.172742$ and $\alpha = 0$ where $0 \leq \alpha \leq 10$. Also we ran the code for different *Reynolds number*, different *final time* and different dt. For this test all we got is, for the values of $\alpha > 0$, the value of statistical equilibrium is decreasing until $\alpha = 1.6$ then starts to increase again. We give the table of the values number of steps for equilibrium tests and $L^2 - norm$ of the energy. We pick the

values for $final\ time = 10$ and $dt = 0.01$. Also, Re = 500 and viscosity $v = \frac{1}{500}$. Here, $tolerance = 10^{-6}$ and the values that in the Table 4 are the values at $final\ time$. Note that for $\alpha = 0$ it is the model of NSE in (2.3).

The results are given for $h = 0.172742$ and different values of α :

Table 4: Test results for Problem 3, Re=500, T=10, h=0.172742

	Test 4		
α	value at T	α	value at T
0	0.705566	0.9	0.668468
0.1	5.43379	1	0.541023
0.2	3.51808	1.1	0.430156
0.3	2.52956	1.2	0.331935
0.4	1.92563	1.3	0.243577
0.5	1.51939	1.4	0.163061
0.6	1.22492	1.5	0.088882
0.7	0.999073	1.6	0.019898
0.8	0.818238	1.7	0.044775

Table 4 shows that this test in inconclusive. The simulations did not reach statistical equilibrium by $T = 10$ for any value of α. Just to be sure, let us to see $Re = 1$ case for $\alpha = 0$, $\alpha = 1$, and $\alpha = 1.6$.

Table 5: Test results for Problem 3, Re=1, T=10, h=0.172742

	Test 4	
α	#steps	value at T
0	exceeded	0.0772927
1	exceeded	0.378334
1.6	exceeded	0.559005

Here, we see the effect of *Reynolds Number*. This test also inconclusive.

We the plot the computed pressure and velocity for value of $\alpha = 0$ and $\alpha = 1.6$ at the

final time for the $Re = 1$. We will see the effect of α with the pictures. Here is our mesh and following that are the figures:
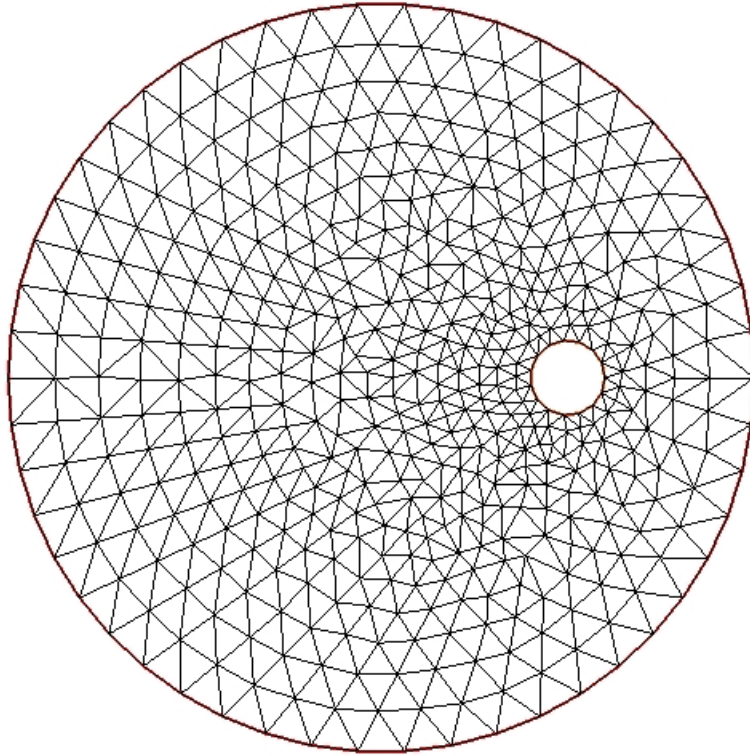

Figure 11: Unit circle mesh C(50)+obs(-15)

Here is the example of our mesh. For this mesh $h = 0.174722$ roughly because of obstacle.
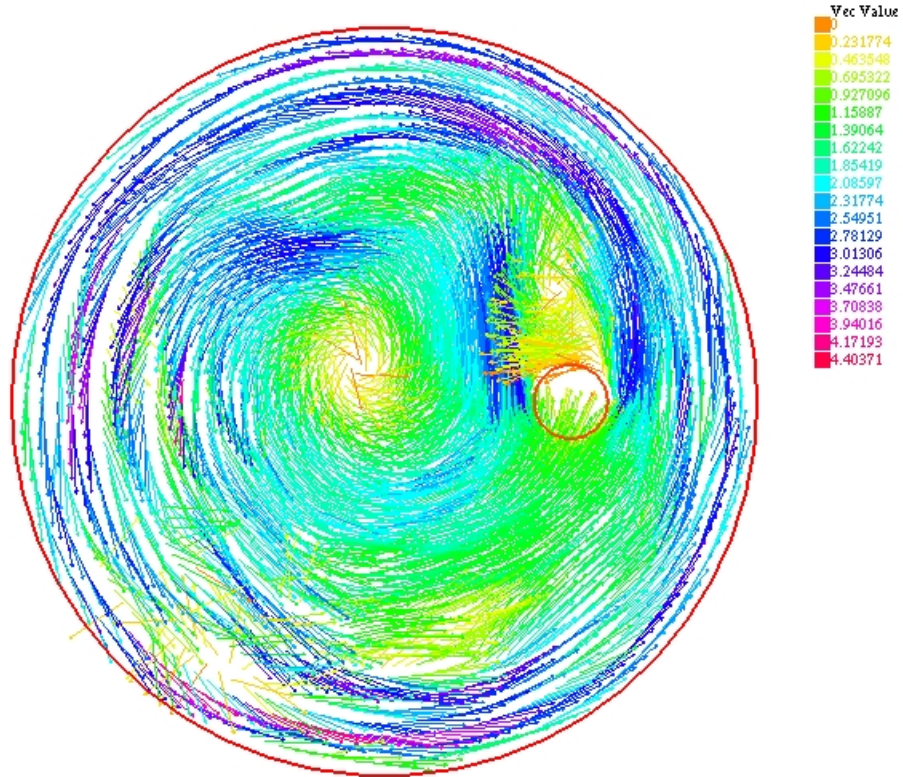
Figure 12: Problem 3, h =0.172742, $alpha = 0$, Computed Velocity at Final Time

As we see here for $\alpha = 0$, velocity vectors are big and complicated. The reason is $Re = 500$. For a given tolerance ϵ, statistical equilibrium test (Test 4) passes over the maximum number of steps.
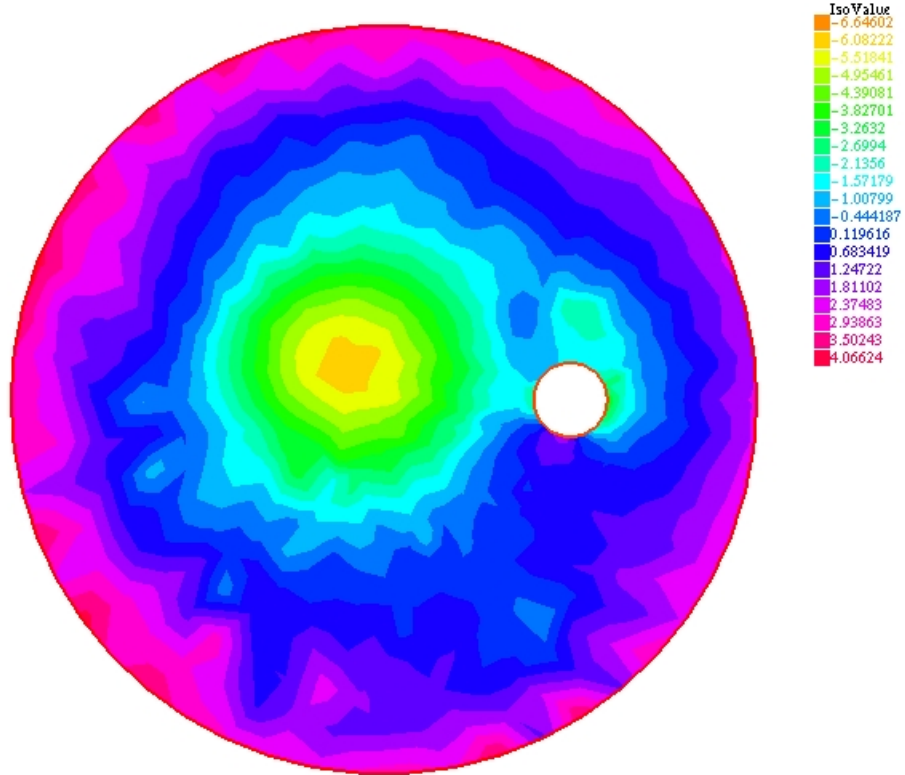
Figure 13: Problem 3, h =0.172742, $alpha = 0$, Computed Pressure at Final Time

Here, we see the calculated pressure at final time. Since we do not know the true solution of pressure, we cannot make a certain statement. But, since we know velocity vectors come big for the case $\alpha = 0$ and pressure is clearly depending on velocity errors, we might say the values of Iso Value's are bad. Next, we will see the pictures of velocity vectors and pressure for $\alpha = 1.6$.
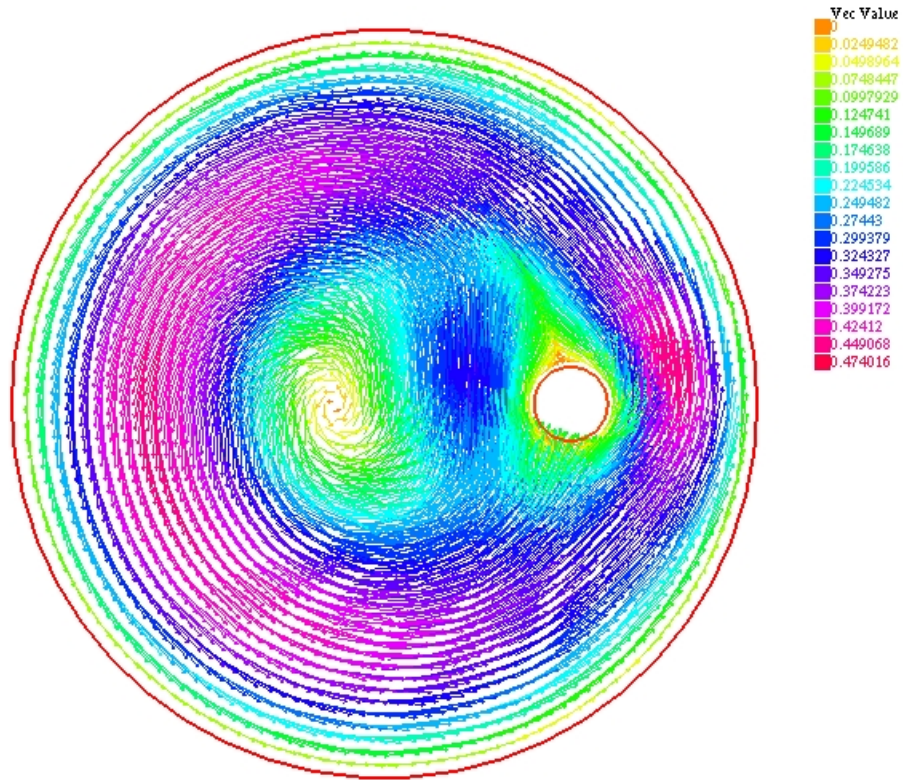
Figure 14: Problem 3, h =0.172742, $alpha = 1.6$, Computed Velocity at Final Time

For the value of $\alpha = 1.6$, velocity vectors are smaller than the Figure 12. But as we see in the Table 4, statistical equilibrum test going worse. So we can say this test is inconclusive.
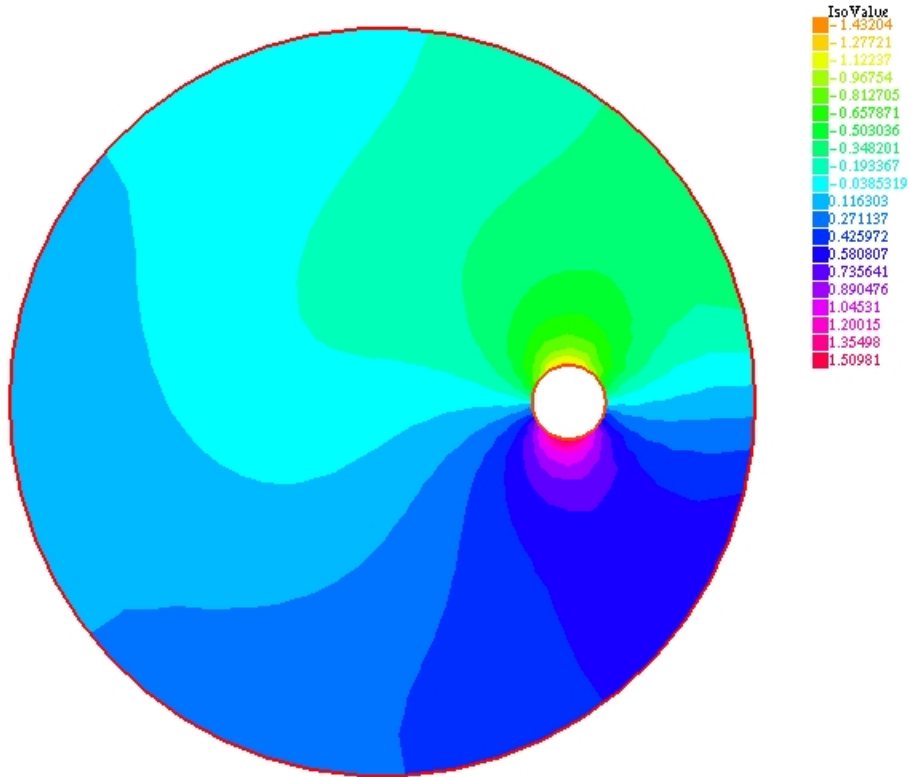
Figure 15: Problem 3, h =0.172742, $alpha = 1.6$, Computed Pressure at Final Time

The pressure seems to smooth, and when we compare to case for $\alpha = 0$, "$IsoValue$" of pressure is smaller. What we concluded here is even the pictures come nicer for the case $\alpha = 1.6$, we know statistical equilibrium goes really bad as the α increases. The results that we find now bring the question of reliability of α. We need to test one more example to see reliability of α.

4.4   THE PROBLEM 4

The fourth problem is tested for a true solution in a circle domain with $radius = 1$. We test this case using no obstacle and unit circle has zero boundary conditions. True solutions are:

$$
\begin{aligned}
u_1(x, y, t) &= 2^{-t}(1 - x^2 - y^2)y \\
u_2(x, y, t) &= -2^{-t}(1 - x^2 - y^2)x \\
p(x, y, t) &= -\frac{1}{6}2^{-2t}((1 - x^2 - y^2)^3 - \frac{1}{4}) \\
f(x, y, t) &= 2^{-t}(\ln 2(1 - x^2 - y^2) - 8\nu) < y, -x >
\end{aligned}
$$

As in Connors([12]) we take $\nu = 10^{-2}$, where initial condition are for $t = 0$:

$$
u_0 = ((1 - x^2 - y^2), -(1 - x^2 - y^2))
$$

and the equilibrium fot $t \to \infty$

$$
\begin{aligned}
u_\infty(x, y) &= 0 \\
p_\infty(x, y) &= 0
\end{aligned}
$$

The Code is same as Problem 3, just changing the functions. We picked longer time scales such as $[0, 20], [0, 30]$ and $[0, 40]$ with $dt = 0.01$. For these time scales, the bigger $0 \leq \alpha \leq 10$ and $100 \leq Re \leq 1000$ makes convergence sometimes bad sometimes good. Mostly, our $tests$ exceeded the max number of steps, so, we give the results for time interval $[0, 10]$, $0 \leq \alpha \leq 1.6$, $Re = 1$ and $dt = 0.01$ to see progress with Table 6. We also give the values at T, because even the numbers of steps are exceeded, values at final time are increasing. For this example, the bigger α makes $Test1$, $Test2$, $Test3$ and $Test4$ worse.

Table 6: Test results for Problem 4, Re=1, T=10, h =0.174722

α	Test 1		Test 2		Test 3		Test 4	
	#steps	value at T	#steps	value at T	#steps	value at T	#steps	value at T
0	exceeded	0.000493437	exceeded	0.000343239	exceeded	0.00171807	exceeded	0.0130760
0.1	exceeded	0.000532255	exceeded	0.000370241	exceeded	0.00185289	exceeded	0.0338352
0.2	exceeded	0.000577702	exceeded	0.000401854	exceeded	0.00201068	exceeded	0.0545937
0.3	exceeded	0.000631635	exceeded	0.000439370	exceeded	0.00219784	exceeded	0.0753522
0.4	exceeded	0.000696676	exceeded	0.000484612	exceeded	0.00242343	exceeded	0.0961107
0.5	exceeded	0.000776620	exceeded	0.000540188	exceeded	0.00270039	exceeded	0.116869
0.6	exceeded	0.000876896	exceeded	0.00060959	exceeded	0.00304610	exceeded	0.137627
0.7	exceeded	0.00100451	exceeded	0.000696544	exceeded	0.00347929	exceeded	0.158386
0.8	exceeded	0.00116680	exceeded	0.000803461	exceeded	0.00401234	exceeded	0.179144
0.9	exceeded	0.00136928	exceeded	0.000929834	exceeded	0.00464330	exceeded	0.199902
1	exceeded	0.00161338	exceeded	0.00107154	exceeded	0.00535221	exceeded	0.220660
1.1	exceeded	0.00189518	exceeded	0.00122127	exceeded	0.00610276	exceeded	0.241417
1.2	exceeded	0.00220528	exceeded	0.00136973	exceeded	0.00684888	exceeded	0.262174
1.3	exceeded	0.00252972	exceeded	0.00150696	exceeded	0.00754077	exceeded	0.282931
1.4	exceeded	0.00285136	exceeded	0.00162357	exceeded	0.00813124	exceeded	0.303687
1.5	exceeded	0.00315142	exceeded	0.00171153	exceeded	0.00857977	exceeded	0.324442
1.6	exceeded	0.00341091	exceeded	0.00176459	exceeded	0.00885460	exceeded	0.345198

Table 6 shows us how α makes the deviation from equilibrium or statistical equilibrium at $T = 10$ increase as α increases. After trying for small and big α's we concluded that convergence is getting worse as the α increases. Later, we tried it for small h and the results were very similar, α makes convergence the worse. Here we can say that adding the term $-\alpha\Delta u_t$ to the NSE is not reliable. Now, we also see that in the pictures.
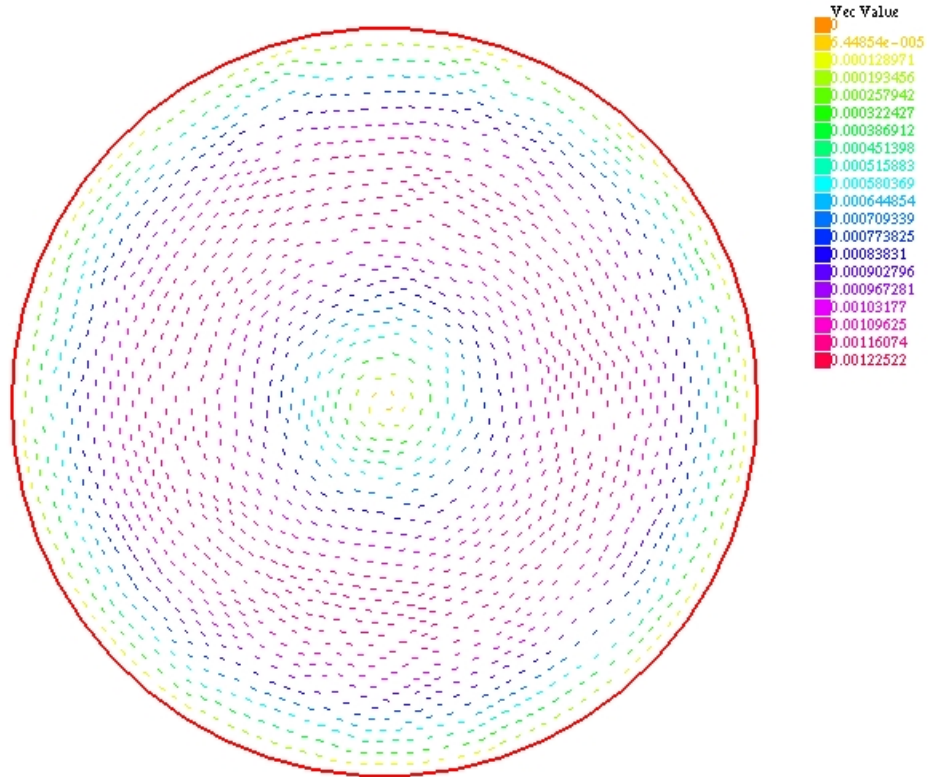
Figure 16: Problem 4, h =0.174722, $alpha = 1$, Computed Velocity at Final Time

We see the velocity vectors for the case $\alpha = 1$. In the Table 6, it was shown that for large α, all tests got worse. We will see difference between velocity vectors in Figure 18.
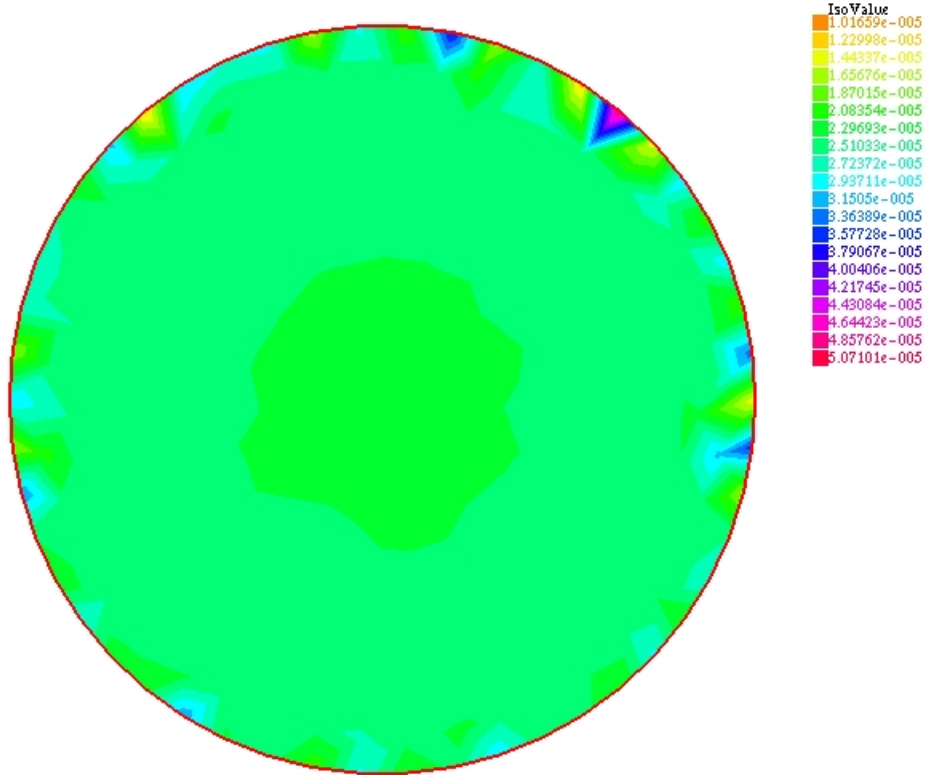
Figure 17: Problem 4, h =0.174722, $alpha = 1$, Computed Pressure at Final Time

This figure is the computed pressure p and .has an interesting color in center. Since the value of pressure at $T = \infty$ should be zero, this is not a good solution of pressure.
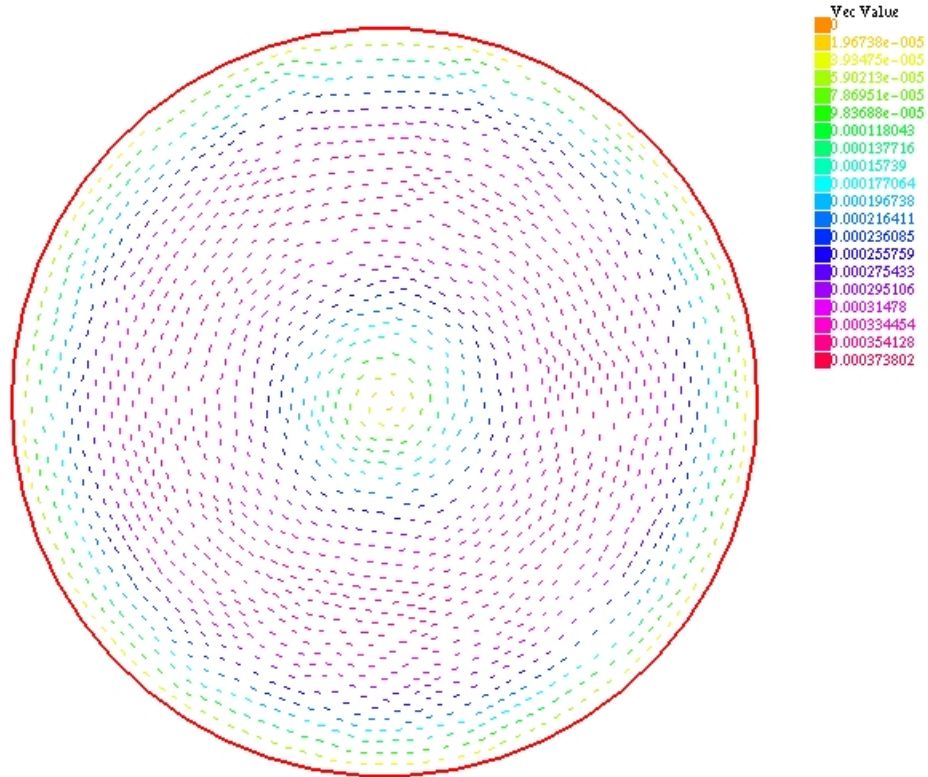
Figure 18: Problem 4, h =0.174722, $alpha = 0$, Computed Velocity at Final Time

For the value of $\alpha = 0$, velocity vectors are smaller than the $\alpha = 0$ case. This was what we expected from the Table 6.
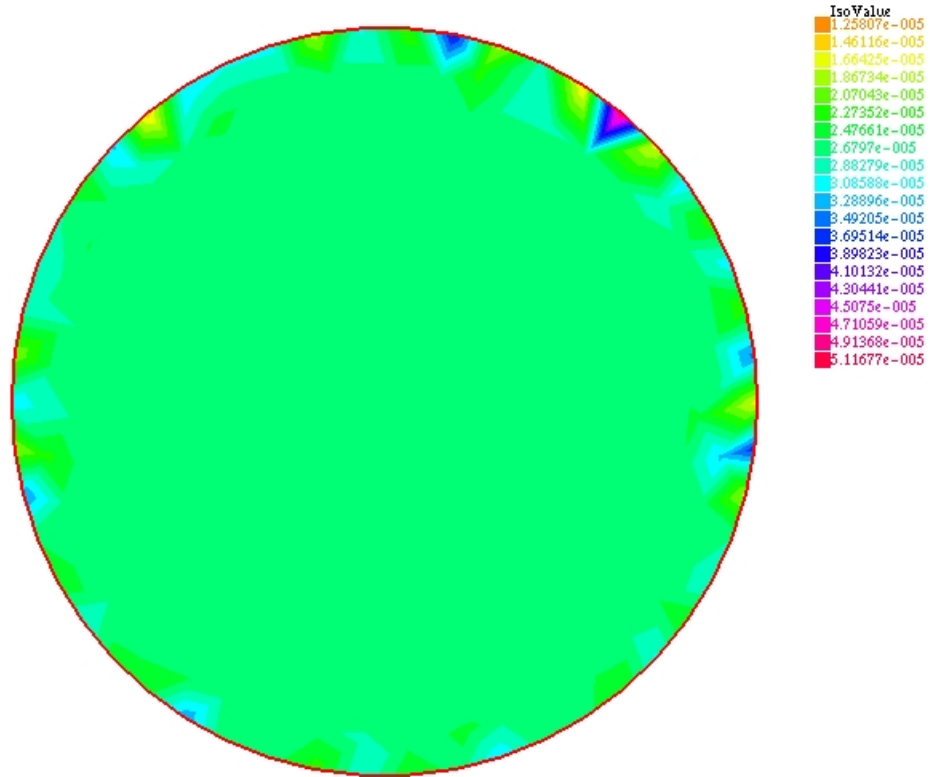
Figure 19: Problem 4, h =0.174722, $alpha = 0$, Computed Pressure at Final Time

If we look at the Iso Values, pressure values are bigger than the $\alpha = 0$ case. We can conclude that the test is inconclusive.

5.0   SUMMARY AND CONCLUSIONS

A modification of NSE model was made and tested for attaining equilibrium, statistical equilibrium and convergence speed. We used true solutions for Problem 1, 2 and 4 and an unknown solution for Problem 3. For each problem, we used different domains to see as much as possible. After using different time steps, different *Reynolds number* and final time, we picked most convenient results to show in tables and pictures so one can easily see using the same code and same parameters.

For Problem 1, we see that adding $-\alpha\Delta u_t$ to the time dependent NSE (2.3), for $\alpha > 0$, big or small, cannot make convergence to equilibrium faster. The values that we picked for α are failed. We showed the results for most suitable values.

For Problem 2, we used a similar problem as Problem 1 with same domain. Picking small α, makes convergence worse and big α exhibits a blow up of a solution. The values that we picked for α failed. Since we showed the results only for $0 \leq \alpha \leq 0.5$.

For Problem 3, we changed our domain to unit circle and tried to find a solution with a body force. We saw that, as the α getting bigger, the deviation from statistical equilibrium a little decreased but then started to increase. This test was inconclusive. The case of $\alpha = 1.6$ did not work for Problem 1 and 2.

For Problem 4, unit circle without obstacle and with a true solution, we saw that as the α got bigger, all tests got worse. The values that we picked for α are failed. Also the case of $\alpha = 1.6$ did not work for this problem too. This test was also inconclusive.

After these tests, we get a result that, the adding term $-\alpha\Delta u_t$ to the NSE, is not reliable. Mostly, values of α makes convergence worse for the problems that we used. As a conclusion, the $\alpha > 0$ may not accelerate convergence to equilibrium or statistical equilibrium.

BIBLIOGRAPHY

[1] J. T. Alexandre, E. Tuenter, G. A. Henstra, K. J. van der Zwan, R. S. W. Van de Wal, H. A. Dijkstra, and P. L. De Boer. Mid-Cretaceous North Atlantic nutrient trap: Black shales and OAEs. Paleoceanography, in press, 2010. doi: 10.1029/2010PA001925.

[2] E. Bernsen, A New Approach to the Spin-up Problem in Ocean–Climate Models, Ph.D Thesis, 2010

[3] K. Bryan and M. D. Cox. A numerical investigation of the oceanic general circulation. Tellus XIX, 19(1):54–80, 1967

[4] K. Bryan. Accelerating the convergence to equilibrium of ocean-climate models. Journal of Physical Oceanography, 14:666–673, 1984

[5] C. Cao, D. Holm, and E. Titi, On the Clark- model of turbulence: global regularity and long-time dynamics, Journal of Turbulence, 6 (2005), pp. 1–11.

[6] Y. Cao, E. M. Lunasin, E. S. Titi, "Global well-posedness of the three-dimensional viscous and inviscid simplified Bardina turbulence models", Comm. in Math. Sciences, 4 (2006), 823–884.

[7] S. Chen, C. Foias, D. Holm, E. Olson, E. Titi, and S. Wynne, Camassa–Holm equations as closure model for turbulent channel and pipe flow, Phys. Rev. Lett., 81 (1998), pp. 5338–5341.

[8] S. Chen, C. Foias, D. Holm, E. Olson, E. Titi, and S. Wynne, The Camassa–Holm equations and turbulence, Phys. D, 133 (1999), pp. 49–65.

[9] A. Cheskidov, D. Holm, E. Olson, E. Titi, On a Leray- model of turbulence, Royal Soc. A, Mathematical, Physical and Engineering Sciences, 461 (2005), pp. 629–649.

[10] A. J. Chorin and J. E. Marsden. A Mathematical Introduction to Fluid Mechanics. Springer-Verlag, New-York, 1993.

[11] W. D. Collins, C. M. Bitz, M. L. Blackmon, G. B. Bonan, C. S. Bretherton, J. A. Carton, P. Chang, S. C. Doney, J. J. Hack, T. B. Henderson, J. T. Kiehl, W. G. Large, D. S. McKenna, B. D. Santer, and R. D. Smith. The Community Climate System Model Version 3 (CCSM3). Journal of Climate, 19(11):2122–2143, 2006. doi: 10.1175/JCLI3761.1.

[12] J. Connors, Convergence Analysis and Computational Testing of the Finite Element Discretization of the Navier-Stokes Alpha Model, 2008

[13] P. Constantin, C. Foias, R. Temam, On the large time Galerkin approximation of the NSE, SINUM 21 (1984) 615-634

[14] M.A. Ebrahimi, M. Holst, and E. Lunasin, The Navier–Stokes-Voight Model for image inpainting, arXiv:0901.4548v3, 2009

[15] Encyclopædia Britannica, "Navier-Stokes equation." Encyclopædia Britannica Online. Encyclopædia Britannica, 2011

[16] G.P. Galdi. An introduction to the Navier-Stokes initial-boundary value problem. In G.P. Galdi, J.G. Heywood, and R. Rannacher, editors, Fundamental Directions in Mathematical Fluid Dynamics, pages 1 -70. BirkhÄauser, 2000.

[17] F. Hecht, A. LeHyaric and O. Pironneau. Freefem++ version 3.13-2, 2011. http://www.freefem.org/ff++.

[18] J. G. Rannacher, Finite Element Approximation of the nonstationary Navier-Stokes problem, Part II: Stability of solutions and error estimates uniform in time, SIAM J. Numer. Anal. 23 (4) (1986) 750-777

[19] A. Ilyin, E. Lunasin, and E. Titi, A modified-Leray- subgrid scale model of turbulence, Nonlinearity, 19 (2006), pp. 879–897

[20] V. Kalantarov and E. S. Titi, Global attractors and estimates of the number of degrees of freedom of determining modes for the 3D Navier-Stokes-Voight equations. arXiv.0705.3972v1, 2007.

[21] S. Khatiwala, M. Visbeck, and M. A. Cane. Accelerated simulation of passive tracers in ocean circulation models. Ocean Modelling, 9:51–69, 2005.

[22] B. A. Klinger. Acceleration of general circulation model convergence by exponential extrapolation. Ocean Modelling, 2:61–72, 2000

[23] W. Layton, Introduction to the Numerical Analysis of Incompressible Viscous Flows, Siam, 2008

[24] J. Leray, Sur les mouvements d'une liquide visqueux emplissant l'espace, Acta Math., 63(1934), 193-248.

[25] J. Leray, Essay sur les mouvements plans d'une liquide visqueux que limitent des parois, J. Math. Pur. Appl., Paris Ser. IX, 13(1934), 331-418.

[26] C. C. Manica, Numerical Methods in Turbulence, Ph.D Thesis, 2006

[27] E. Olson and E. Titi, Viscosity versus vorticity stretching: global well-posedness for a family of Navier–Stokes-alpha-like models, Nonlinear Anal., 66 (2007), pp. 2427–2458.

[28] A. P. Oskolkov, The uniqueness and solvability in the large of the boundary value problems forthe equations of motion of aqueous solutions of polymers, Zap. Naucn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (LOMI), 38 (1973), pp. 98–136

[29] A. P. Oskolkov, On the theory of Voight fluids, Zap. Naucn. Sem. Leningrad. Otdel. Mat. Inst. Steklov (LOMI), 96 (1980), pp. 233–236

[30] H. Sohr. The Navier-Stokes Equations, An Elementary Functional Analytic Approach. BirhÄauser Advanced Texts. BirkhÄauser Verlag Basel, Boston, Berlin, 2001.