

## 3 Newtonian incompressible flow

In this chapter, we derive the variational formulation of the Navier–Stokes equations (see section 3.1) and we also describe a fully implicit and fully coupled formulation to solve these equations using the finite element method and implicit time integrators (see section 3.2). We developed a C++ code using this approach; our implementation will be described on chapter 6. The implementation is validated using the problem of a lid–driven cavity flow (see section 3.3). Despite this being a widely studied problem in computational fluid dynamics, it makes up the basic theoretical and computational framework that we use in the next chapters.

### 3.1 Variational formulation

The basic idea in any numerical method for a differential equation is to discretize the continuous problem to obtain a discrete problem described by a system of algebraic equations with a finite number of unknowns, which can be solved numerically. The classical numerical method to solve partial differential equations is the finite difference method in which the discrete problem is obtained approximating derivatives by difference quotients involving the values of the unknown at certain, finite number, points.

The discretization process using the finite element method is completely different. In the finite element method the given differential equation is written as an equivalent variational problem. In our case, the differential equations are the Navier–Stokes equations for Newtonian incompressible fluids stated in the previous chapter (see equations 2-1 and 2-2).

Before writing the variational formulation of the Navier–Stokes equations, we must be careful with the regularity properties of its solution in order to assure the mathematical consistency of the formulation. Let us say that the velocity and pressure fields,  $\vec{u}_f$  and  $p_f$ , are the solution of the variational problem in a given fluid domain  $\Omega_f$ . We shall assume that they belong to the following function space:

$$\mathbb{C}_f := \mathbb{V}_f \times \mathbb{P}_f := \{(\vec{u}_f, p_f) \mid \vec{u}_f \in \mathbb{V}_f \text{ and } p_f \in \mathbb{P}_f\} \quad (3-1)$$

were  $\mathbb{V}_f$  and  $\mathbb{P}_f$  are defined as:

$$\mathbb{V}_f := \{\vec{u}_f \in \mathbb{H}^1(\Omega_f) \mid \vec{u}_f|_{\partial\Omega_f} = \vec{u}_{\partial\Omega_f}\}$$

$$\mathbb{P}_f := \{p_f \in \mathbb{H}^0(\Omega_f)\}$$

and  $\mathbb{H}^k(\Omega_f)$  is the Sobolev space:

$$\mathbb{H}^k(\Omega_f) = \{w \mid w \in L_2(\Omega_f), \frac{\partial w}{\partial x} \in L_2(\Omega_f); \dots; \frac{\partial^k w}{\partial x} \in L_2(\Omega_f)\}$$

The definition of the Sobolev space says that if a function  $w$  belongs to  $\mathbb{H}^k(\Omega_f)$  then  $w$  and all its derivatives up to some order  $k$  have a finite  $L_2(\Omega_f)$  norm. Observe that the solution space 3-1, is an infinite dimensional space.

The velocity and pressure solutions are required to have different regularity properties and 3-1 says that the velocity field must respect a prescribed Dirichlet condition  $\vec{u}_{\partial\Omega_f}$  on the domain's boundary  $\partial\Omega_f$ . To ensure the solution unicity, we may also require the pressure scalar field to respect a prescribed constraint on  $\Omega_f$ .

Now, let  $\vec{\phi} \in \mathbb{V}_f$  be an arbitrarily chosen function. We can write the variational formulation of the momentum conservation equation 2-1 as:

$$\int_{\Omega_f} \left( \rho_f \frac{D\vec{u}_f}{Dt} - \nabla \cdot \boldsymbol{\sigma}_f - \vec{g} \right) \cdot \vec{\phi} d\Omega_f = 0 \quad (3-2)$$

We rewrite the previous equation using the following tensorial identity:

$$\boldsymbol{\sigma}_f : \nabla \vec{\phi} = \nabla \cdot (\boldsymbol{\sigma}_f \cdot \vec{\phi}) - (\nabla \cdot \boldsymbol{\sigma}_f) \cdot \vec{\phi}, \quad (3-3)$$

were the tensorial operation written on the left side of the previous equality 3-3 is defined as follows:

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^d \sum_{j=1}^d a_{ij} b_{ij}$$

where  $d$  is the dimension of the simulation domain, in our case  $d = 2$ . Observe that the operation results on a scalar value. The variational formulation of the momentum equation now becomes:

$$\int_{\Omega_f} \left( \rho_f \frac{D\vec{u}_f}{Dt} - \vec{g} \right) \cdot \vec{\phi} d\Omega_f + \int_{\Omega_f} \boldsymbol{\sigma}_f : \nabla \vec{\phi} - \nabla \cdot (\boldsymbol{\sigma}_f \cdot \vec{\phi}) d\Omega_f = 0 \quad (3-4)$$

In order to obtain the final momentum equation of the variational problem, we must use the divergence theorem on the last integral term in the previous equation, which leads to:

$$\int_{\Omega_f} \left( \rho_f \frac{D\vec{u}_f}{Dt} - \vec{g} \right) \cdot \vec{\phi} d\Omega_f + \int_{\Omega_f} \boldsymbol{\sigma}_f : \nabla \vec{\phi} d\Omega_f - \int_{\partial\Omega_f} \vec{f} \cdot \vec{\phi} d\partial\Omega_f = 0 \quad (3-5)$$

where  $\vec{f} = \vec{n}_f \cdot \boldsymbol{\sigma}_f$  represents a force acting on  $\partial\Omega_f$  and is closely related with the physical behavior of the boundary conditions that we are interested to impose on the flow simulation.

The variational form of the continuity equation 2-2 is derived analogously now using an arbitrarily chosen element  $\chi \in \mathbb{P}_f$  :

$$\int_{\Omega_f} (\nabla \cdot \vec{u}_f) \chi d\Omega_f = 0 \quad (3-6)$$

The variational problem equivalent to the differential formulation of the Navier–Stokes equations is finally stated as: find  $\vec{u}_f \in \mathbb{V}_f$  and  $p_f \in \mathbb{P}_f$  such that  $\forall \vec{\phi} \in \mathbb{V}_f$  and  $\forall \chi \in \mathbb{P}_f$ ,

$$\begin{aligned} \int_{\Omega_f} \left( \rho_f \frac{D\vec{u}_f}{Dt} - \vec{g} \right) \cdot \vec{\phi} d\Omega_f + \int_{\Omega_f} \boldsymbol{\sigma}_f : \nabla \vec{\phi} d\Omega_f - \int_{\partial\Omega_f} \vec{f} \cdot \vec{\phi} d\partial\Omega_f &= 0 & \text{in } \Omega_f \\ \int_{\Omega_f} (\nabla \cdot \vec{u}_f) \chi d\Omega_f &= 0 & \text{in } \Omega_f \end{aligned} \quad (3-7)$$

### 3.2

#### Fully coupled and implicit discretization

The idea to obtain a problem that can be solved numerically using the finite element method is to replace the infinite dimensional solution space  $\mathbb{C}_f$  by a subspace  $\mathbb{C}_f \subset \mathbb{C}_f$ , of finite dimension.

The choice of the finite dimensional space  $\mathbb{C}_f$  is essentially the choice of the finite element discretization and is influenced by the variational formulation, regularity properties of the exact solution, accuracy requirements, etc. It is well known that variational formulations associated with constraints (as the incompressible Navier–Stokes equations) may lead to severe numerical problems and to ensure the existence and uniqueness of the approximated solution in the subspace  $\mathbb{C}_f$ , we must verify the Babuska–Brezzi condition (1, 2, 3). Let us define the following finite dimensional space:

$$\mathbb{C}_f := \{(\vec{u}_f, p_f) \mid \vec{u}_f \in \mathbb{V}_f \text{ and } p_f \in \mathbb{P}_f\} \quad (3-8)$$

were  $\mathbb{V}_f$  and  $\mathbb{P}_f$  are defined as:

$$\mathbf{V}_f := \{\vec{\mathbf{u}}_f \in \mathbb{P}_4(\Lambda_f) \times \mathbb{P}_4(\Lambda_f) \mid \vec{\mathbf{u}}_f|_{\partial\Omega} = \vec{u}_{\partial\Omega}\}$$

$$\mathbf{P}_f := \{\mathbf{p}_f \in \mathbb{P}_2(\Lambda_f)\}$$

$\mathbb{P}_4(\Lambda_f)$  and  $\mathbb{P}_2(\Lambda_f)$  are spaces of piecewise polynomials in two variables with maximal degree 4 and 2 defined over some discretization  $\Lambda_f$  of the domain  $\Omega_f$ . More precisely, we use biquadratic and/or quadratic spaces to approximate the velocity field, and bilinear and/or linear spaces for pressure approximation.

We can use the finite dimensional solution space  $\mathbf{C}_f$  to rewrite the infinite dimensional variational problem 3-7 as the following finite dimensional problem: find  $\vec{\mathbf{u}}_f \in \mathbf{V}_f$  and  $\mathbf{p}_f \in \mathbf{P}_h$  such that  $\forall \vec{\phi} \in \mathbb{P}_4(\Lambda_f)$  and  $\forall \chi \in \mathbb{P}_2(\Lambda_f)$ ,

$$\begin{aligned} \sum_{\tau \in \Lambda_f} \int_{\tau} \left( \rho_f \frac{D\vec{\mathbf{u}}_f}{Dt} - \vec{g} \right) \cdot \vec{\phi} \, d\tau + \sum_{\tau \in \Lambda_f} \int_{\tau} \boldsymbol{\sigma}_f : \nabla \vec{\phi} \, d\tau - \sum_{\epsilon \in \partial\Lambda_f} \int_{\epsilon} \vec{f} \cdot \vec{\phi} \, d\epsilon = 0 \\ \sum_{\tau \in \Lambda_f} \int_{\tau} (\nabla \cdot \vec{\mathbf{u}}_f) \chi \, d\tau = 0 \end{aligned} \tag{3-9}$$

This problem is equivalent to a large nonlinear system of algebraic equations, and the hope is that a solution  $(\vec{\mathbf{u}}_f, \mathbf{p}_f) \in \mathbf{C}_f$  is a sufficiently good approximation for the solution of the original differential problem.

In practice, the spatial discretization process is based on building the partition  $\Lambda_f$  of the given fluid's domain  $\Omega_f$  into a finite number of sub-domains, and consider a set of basis functions of the finite solution space. The sub-domains are in fact triangles and/or quadrangles and, as we said before, they are called *elements*. The set of basis functions is build as follows: suppose that for the given finite dimensional space, there is a finite set of points which uniquely determines any function in the space by its values at these points. The set of functions in the space that take on a nonzero value (more precisely one) at exactly one of the points form a basis for the finite solution space, called the *nodal or Lagrange basis*. Figure 3.1 shows the Lagrange basis for a piecewise linear space of functions defined over the interval  $[0,1]$ , discretized with 5 one-dimensional elements.

Let us denote by  $(\vec{\phi}_i, \chi_j)$  an element in the Lagrange basis of the finite solution space, where  $\vec{\phi}_i \in \mathbb{P}_4(\Lambda_f) \times \mathbb{P}_4(\Lambda_f)$  and  $\chi_j \in \mathbb{P}_2(\Lambda_f)$  with  $i \in (1 \dots \#\mathbb{P}_4(\Lambda_f) \times \mathbb{P}_4(\Lambda_f))$  and  $j \in (1 \dots \#\mathbb{P}_2(\Lambda_f))$ , where  $\#V$  is the dimension of the vector space  $V$ . Using the Lagrange basis, the velocity and pressure solution can be written as:

$$\vec{\mathbf{u}}_f = \sum_i \mathbf{u}_i \vec{\phi}_i \quad \text{and} \quad \mathbf{p}_f = \sum_j \mathbf{p}_j \chi_j$$

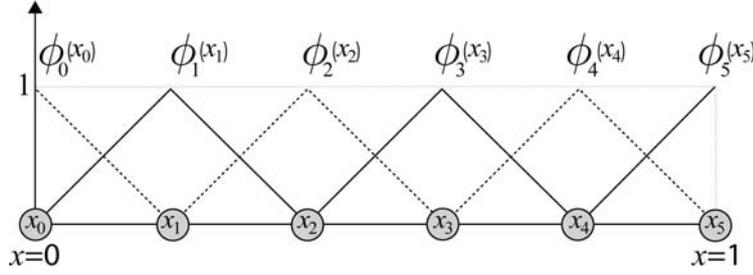


Figure 3.1: Lagrange basis for a piecewise linear space of functions defined over the interval  $[0,1]$ , discretized by 5 elements.

Moreover, we observe that in the discrete problem 3-9, the elements  $\vec{\phi}$  and  $\chi$  were arbitrarily chosen. In particular, they can be any element in the Lagrange basis of the solution space. This choice leads to a system of non-linear time-variable differential equations, whose solution gives the coefficients  $u_i$  of the velocity and  $p_j$  of the pressure approximations written in the Lagrange basis  $(\vec{\phi}_i, \chi_j)$ . Using the Lagrange basis, the stress tensor definition and writing explicitly the derivative operators, we can rewrite 3-9 in Cartesian coordinates:

$$\sum_{\tau \in \Lambda_f} \int_{\tau} \rho_f \frac{\partial u^x}{\partial t} \phi_i + \rho_f \left[ u^x \frac{\partial u^x}{\partial x} + u^y \frac{\partial u^x}{\partial y} \right] \phi_i - g^x \phi_i + \left[ -p_f + 2\mu_f \frac{\partial u^x}{\partial x} \right] \frac{\partial \phi_i}{\partial x} + \mu_f \left[ \frac{\partial u^x}{\partial y} + \frac{\partial u^y}{\partial x} \right] \frac{\partial \phi_i}{\partial y} d\tau - \sum_{\epsilon \in \partial \Lambda_f} \int_{\epsilon} f^x \phi_i d\epsilon = 0$$

$$\sum_{\tau \in \Lambda_f} \int_{\tau} \rho_f \frac{\partial u^y}{\partial t} \phi_i + \rho_f \left[ u^x \frac{\partial u^y}{\partial x} + u^y \frac{\partial u^y}{\partial y} \right] \phi_i - g^y \phi_i + \left[ -p_f + 2\mu_f \frac{\partial u^y}{\partial y} \right] \frac{\partial \phi_i}{\partial y} + \mu_f \left[ \frac{\partial u^x}{\partial y} + \frac{\partial u^y}{\partial x} \right] \frac{\partial \phi_i}{\partial x} d\tau - \sum_{\epsilon \in \partial \Lambda_f} \int_{\epsilon} f^y \phi_i d\epsilon = 0$$

$$\sum_{\tau \in \Lambda_f} \int_{\tau} \left( \frac{\partial u^x}{\partial x} + \frac{\partial u^y}{\partial y} \right) \chi d\tau = 0$$

(3-10)

where the first two equations are the coordinate functions of the momentum conservation equation, and the last equation is the continuity equation. We use the notation  $\vec{v} = (v^x, v^y)$  for the coordinate functions of a vector field  $\vec{v}$ .

In our finite elements C++ code, we use biquadratic quadrangular elements and/or quadratic triangular elements to approximate the velocity field, and bilinear quadrangular elements and linear triangular elements for the pressure approximation. This choice allow us to use tri-quad meshes without losing the solution's regularity in edges shared by triangles and quadrangles (see figure 3.2). Moreover, previous works (2, 1, 3) proves that the chosen elements satisfy the Babuska-Brezzi condition.

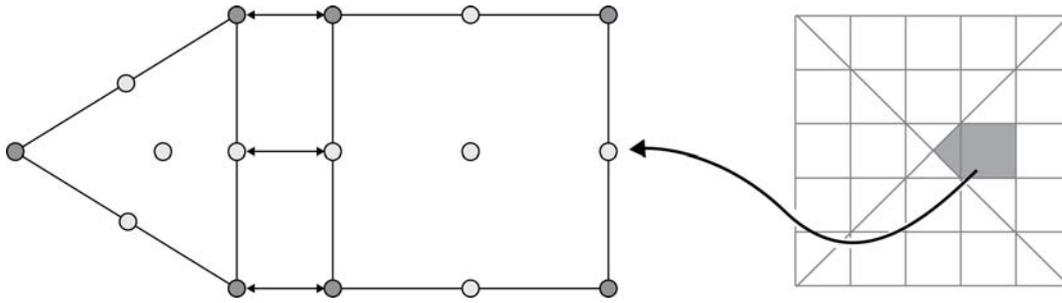


Figure 3.2: Mesh with quadrangles and triangles: we use elements that ensure the solution's continuity at the edges shared by triangles and quadrangles. The dark nodes are used for the pressure's approximation, and the dark and light nodes are used to build the velocity's solution space.

We use an implicit Euler method to perform the time integration that appears on the momentum coordinate function equations in 3-10. We use an implicit integrator since, in many cases, the problems arising from incompressible Newtonian flow applications are stiff and the use of an explicit method requires impractically small time steps to keep the error in the result bounded. For such problems, to achieve a given accuracy, it takes much less computational time to use an implicit method with larger time steps. We denote the time step by  $\delta t$ .

While explicit time integration methods compute the physical unknowns at a later time using the solution obtained at the current time, an implicit method finds it by solving an equation involving both the current solution and the later time unknowns. If  $s^{(t)}$  is the current solution and  $s^{(t+1)}$  is the physical unknowns at the later time, then for implicit Euler method:

$$\frac{\partial s}{\partial t} = f(t, s) \Rightarrow \frac{s^{(t+1)} - s^{(t)}}{\delta t} = f(t, s^{(t+1)}) \Rightarrow s^{(t)} = s^{(t+1)} - f(t, s^{(t+1)})\delta t$$

which clearly indicates that we must solve a system of equations to obtain the physical values at the later time  $s^{(t+1)}$ .

In our approach, we wrote the system of equations given by the variational formulation of the Navier–Stokes equations 3-9 and by the system obtained from the implicit Euler time integration into a single fully coupled system of non-linear equations, and we solved it using Newton's method. Newton's method is an extremely powerful and fast iterative technique since it has in general quadratic convergence, i.e., the error is essentially squared at each iteration. The method's algorithm requires the construction of the Jacobian matrix  $\mathbf{J}$  and the residue vector  $\vec{r}$  associated to the nonlinear-system at each iteration of the procedure. The residue and Jacobian computations are detailed in the Appendix A. Newton's method is computed using the procedure:

**Algorithm 1** Newton's method iterative procedure

---

```

 $s^{(t+1)} \leftarrow (x_1, x_2, x_3, \dots, x_k)$  // Initial guess.
while  $|s^{(t+1)} - s^{(t)}| < \epsilon$  do
   $\mathbf{J}(\delta s^{(t+1)}) = -\vec{r}(s^{(t)})$ 
   $s^{(t+1)} = s^{(t)} + \delta s^{(t+1)}$ 
end while

```

---

Observe that algorithm 1 requires an initial guess to start the iterative procedure. If the given initial value is too far from the desired solution, Newton's method may fail to converge. For this reason, Newton's method is often referred to as a local technique. There are several techniques that help the determination of good initial guesses. In our software, the initial guess at each time step is computed as a linear extrapolation of the solution obtained by two previous time steps.

Finally, to solve each iteration of Newton's method we use the `IML++` library (10) that provides sparse matrix representation together with iterative methods to solve efficiently linear systems.

### 3.3 Code validation

The lid-driven cavity flow is probably one of the most studied problems in the field of computational fluid dynamics. The simplicity of the geometry of the cavity flow makes the problem easy to code and to apply boundary conditions. Although the problem looks simple in many ways, the flow in a cavity is complex with counter rotating vortices appearing at the corners of the cavity.

For the reasons described above, the driven cavity flow is a benchmark problem for numerical methods in terms of accuracy, efficiency and so on. In the literature it is possible to find numerous studies on the driven cavity flow (11, 12, 13, 16, 36, 37) that guided us during the validation our code.

We divided our tests in two groups. The first group of tests aims to evaluate the software robustness with respect to the mesh resolution and fluids properties in steady simulations. The second group aims to analyze the robustness of the unsteady solutions in relation to the time discretization. The basic setup for a lid-driven cavity flow is shown on figure 3.3. For all results we plot the velocity profiles in the vertical line that pass through the center of the domain and render the velocity solution. Velocity vectors with size near to zero are painted in blue, and the ones with norm near to the lid's velocity are colored in red.

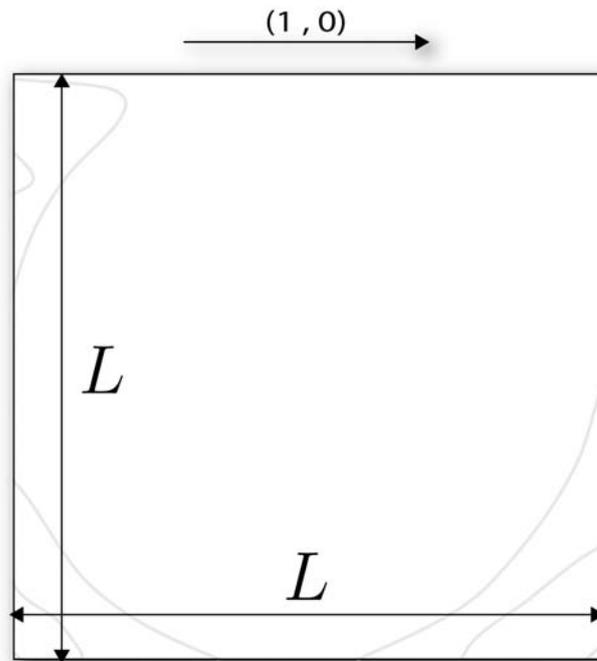


Figure 3.3: Lid-driven cavity flow setup: the cavity is a closed box with no-slip boundary condition in the fixed walls and a moving lid with constant horizontal velocity. In our test  $L = 1 \text{ cm}$  and the lid velocity  $1 \frac{\text{cm}}{\text{ds}}$ .

The flow characteristics in the lid-driven cavity flow problem is usually described in terms of the dimensionless *Reynolds number*. It gives a measure of the ratio of inertial forces to viscous forces and consequently quantifies the relative importance of these two types of forces for given flow conditions. In that way, the Reynolds number is an important parameter that describes whether flow conditions lead to laminar or turbulent flow. The Reynolds number in the cavity flow problem is calculated using the following expression:

$$Re = \frac{Lu_l\rho}{\mu}$$

where  $u_l$  is the lid velocity and  $L$  is the *characteristic length* that describes the domain geometry and is a matter of convention. Usually, it is chosen as the length of the object that the flow is going through or around. In the cavity flow problem, it is chosen as the length of the cavity.

**Steady tests:** Figures 3.4, 3.5 and 3.6 show results obtained by steady lid-driven cavity flow simulations obtained using our software. We tested the robustness of our implementation in relation to the discretization of the cavity domain  $\Omega$ . We ran this validation test for setups with different Reynolds numbers. In the first one, the Reynolds number was 10 (see Figure 3.4), in the second the Reynolds was set 100 (see Figure 3.5) and in the third the

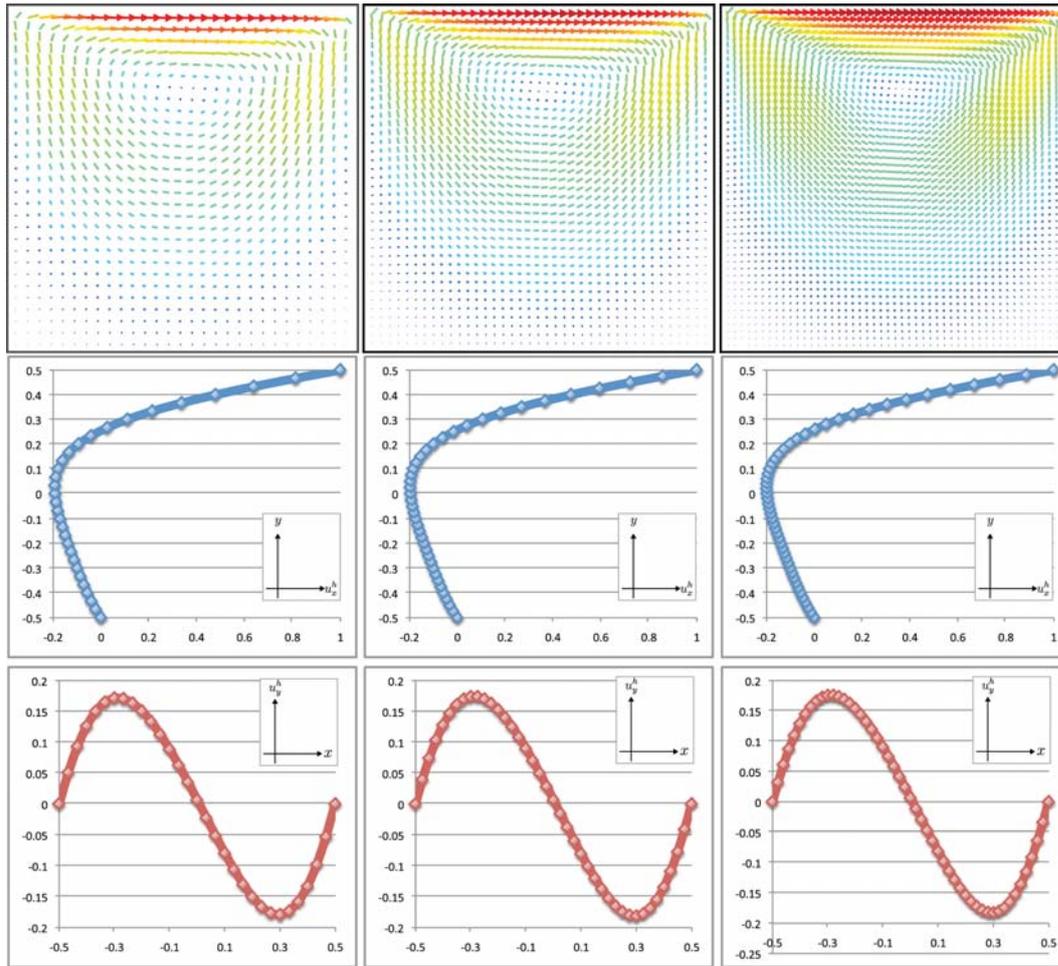


Figure 3.4: Lid-driven cavity flow validation: steady case with Reynolds number 10 and three different meshes with 225, 400 and 625 elements.

Reynolds was 1000 (see Figure 3.6). In all this test cases, we performed the simulation using quadrangular meshes with 225, 400 and 625 elements. The results, for all mesh resolutions, are shown in each image reading the columns from left (lowest resolution) to right (highest resolution).

It is important to observe that, in each image, the flow behavior is virtually independent of the mesh resolution. The horizontal and vertical velocities as function of  $y$  and  $x$  respectively are virtually the same for all the meshes used. As expected, the vertical velocity profile, as function of  $x$  is almost symmetric at low Reynolds number. The symmetry is lost as we increase the Reynolds number value (as we can see in Figures 3.5 and 3.6). The flow and the velocity profiles behavior agree with the results in the literature, as can be seen in the work of (11).

Finally, we can see that regions near the upper left and bottom corners, which are regions of counter rotating vortices, are easier observed as the Reynolds number is increased. Moreover, the position of the center of the large

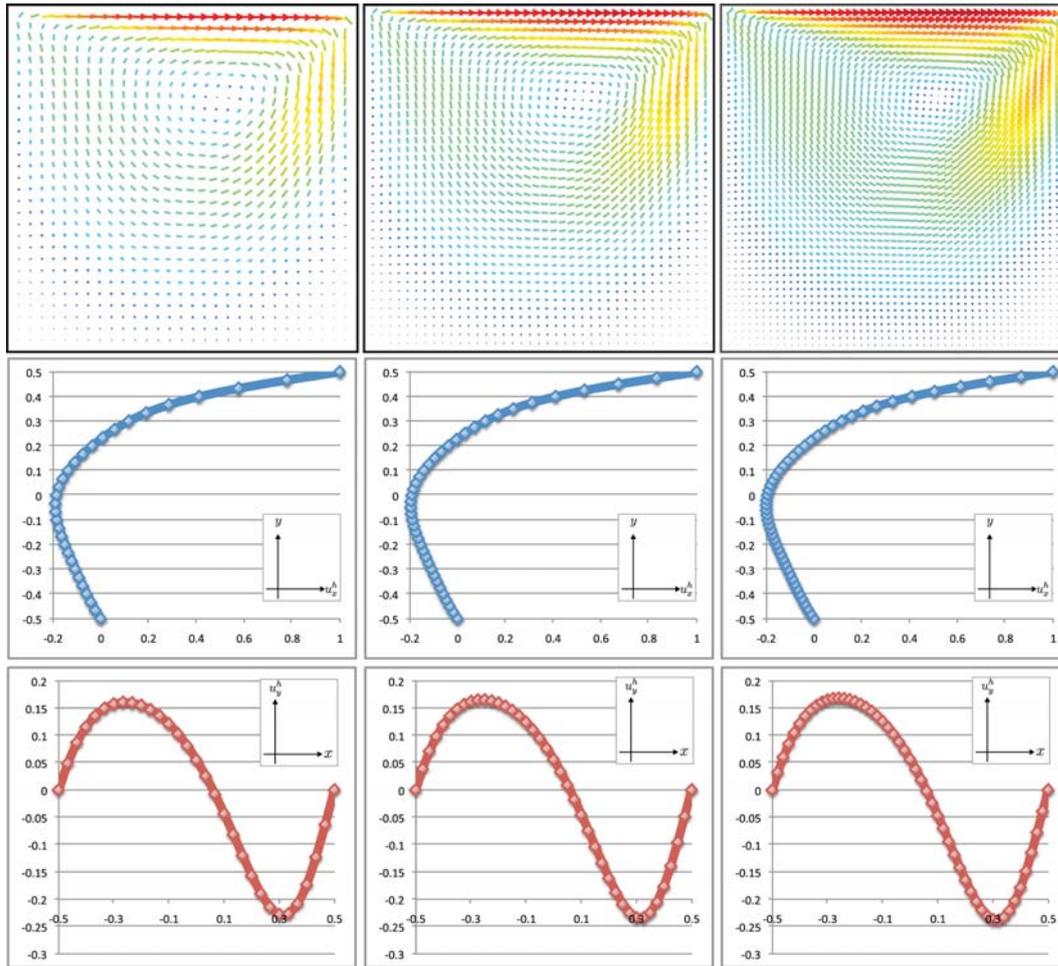


Figure 3.5: Lid-driven cavity flow validation: steady case with Reynolds number 100 and three meshes with 225, 400 and 625 elements.

vortex also depends on the Reynolds numbers. These patterns were already expected and had good agreement with previous studies in lid-driven flows.

In order to numerically compare our results with previous works, the velocity profiles obtained in Erturk et. al (2009) (11) in a lid-driven cavity simulation using Reynolds 1000 is plotted on figure 3.6 (see the black curves in each velocity profile).

**Unsteady tests:** Figures 3.7, 3.8 and 3.9 show results obtained by unsteady lid-driven cavity flow simulations obtained using our software. We used this test to evaluate the robustness of our software in relation to the time discretization. We run this validation tests using a fixed mesh resolution and Reynolds number set to 1000. We used a quadrangle mesh with 625 quadrangular elements. Figure 3.7 shows the results obtained using a time step  $\delta t = 0.01$ , and Figure 3.8 shows the pictures obtained using the time step  $\delta t = 0.001$ . The columns of unsteady results show the flow solution when the time was 5, 15, 30 and 50 deciseconds.

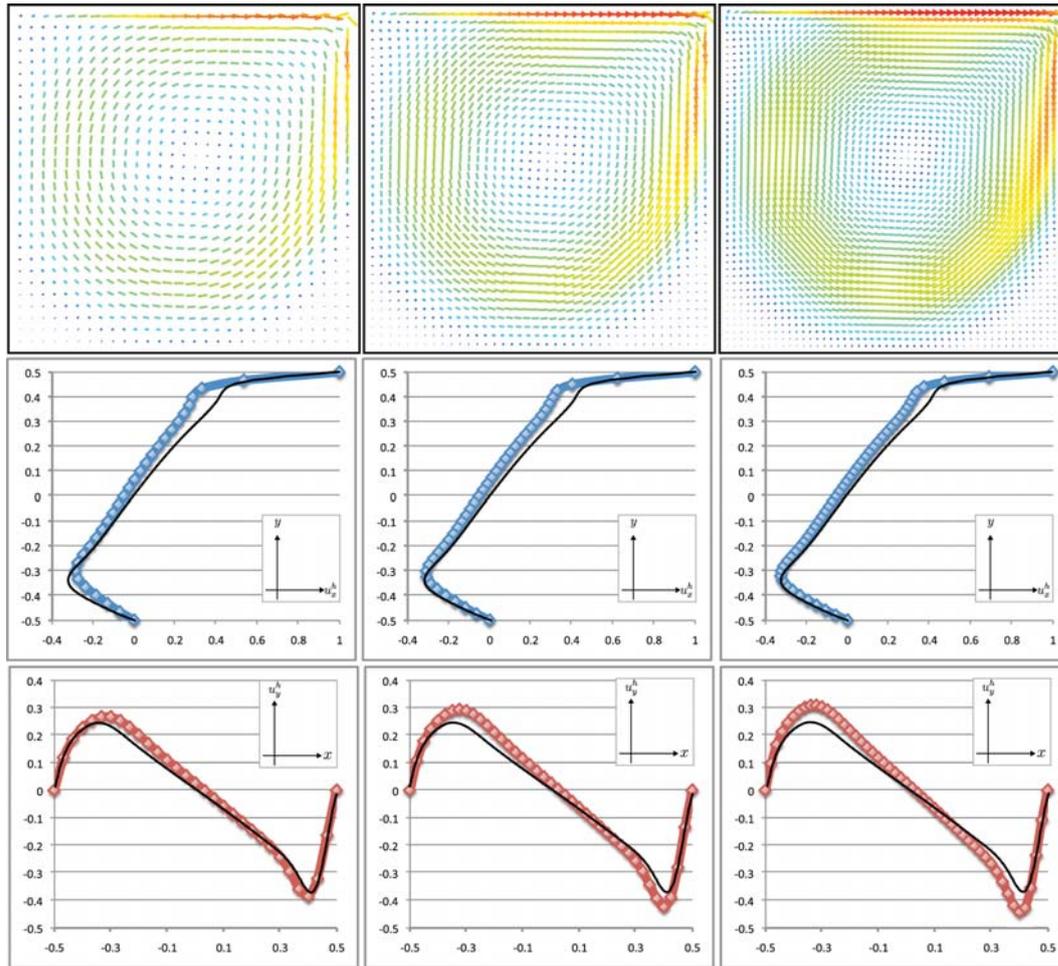


Figure 3.6: Lid-driven cavity flow validation: Steady case with Reynolds number 1000 and three meshes with 225, 400 and 625 elements.

If we analyze the velocity profiles for the unsteady simulations (second and third row of each figure), we see that they always converge to the steady case when the simulations reach sufficiently high execution times. It can be verified if we observe that the results showed on the last columns of Figures 3.7 and 3.8 reproduce exactly the same flow obtained by the steady cavity problem with the same Reynolds numbers that are shown in Figure 3.6. If we analyze the rendering of the velocity field we also can observe this convergence behavior on the evolution of the fluid flow and the formation of the center and the counter rotating vortices. Observe that the velocity final profiles obtained by the unsteady simulation with Reynolds number set to 1000 are very close to the ones by Erturk et. al (2009) (11).

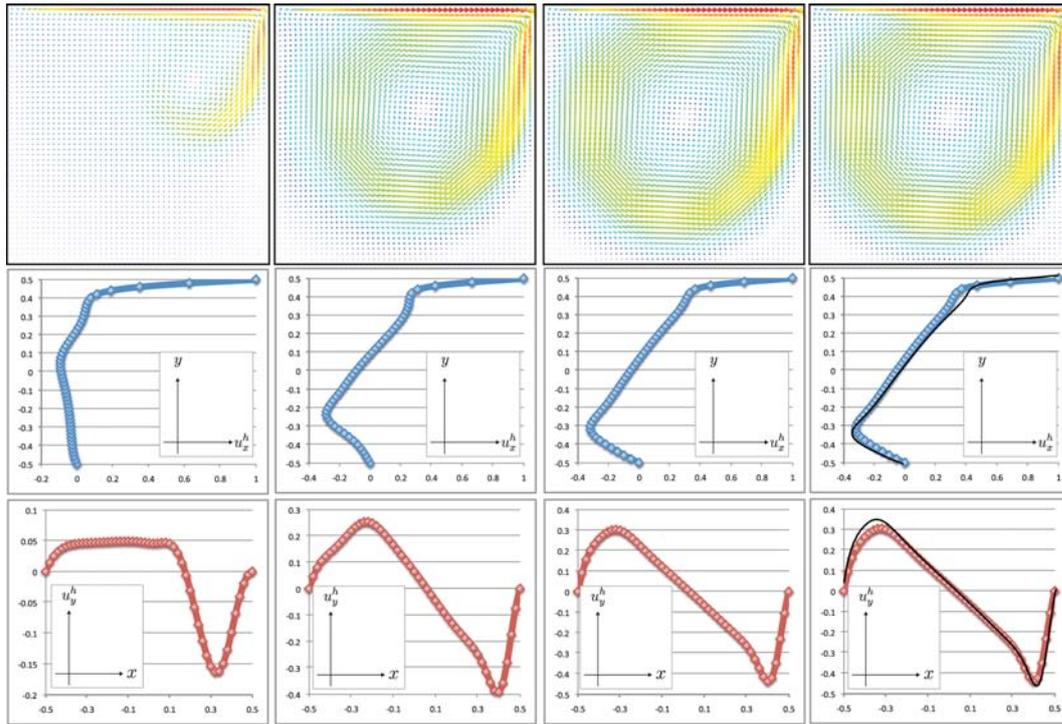


Figure 3.7: Lid-driven cavity flow validation: unsteady case with Reynolds number 1000, mesh with 625 elements and time step 0.01. The columns shows the simulation after 5, 15, 30 and 50  $ds$  of simulated time.

Figure 3.9 shows the last validation of our software using the cavity test. Our objective in this test was to show that the software is also robust when using higher mesh resolutions and solving flows at Reynolds number larger than 1000. In this test case, the quadrangular mesh was built using 1600 elements, and Reynolds number was set to 4000. It is important to mention that we are still solving a laminar flow, even at high Reynolds number at which the flow may already be turbulent as in the last example of figure 3.9.

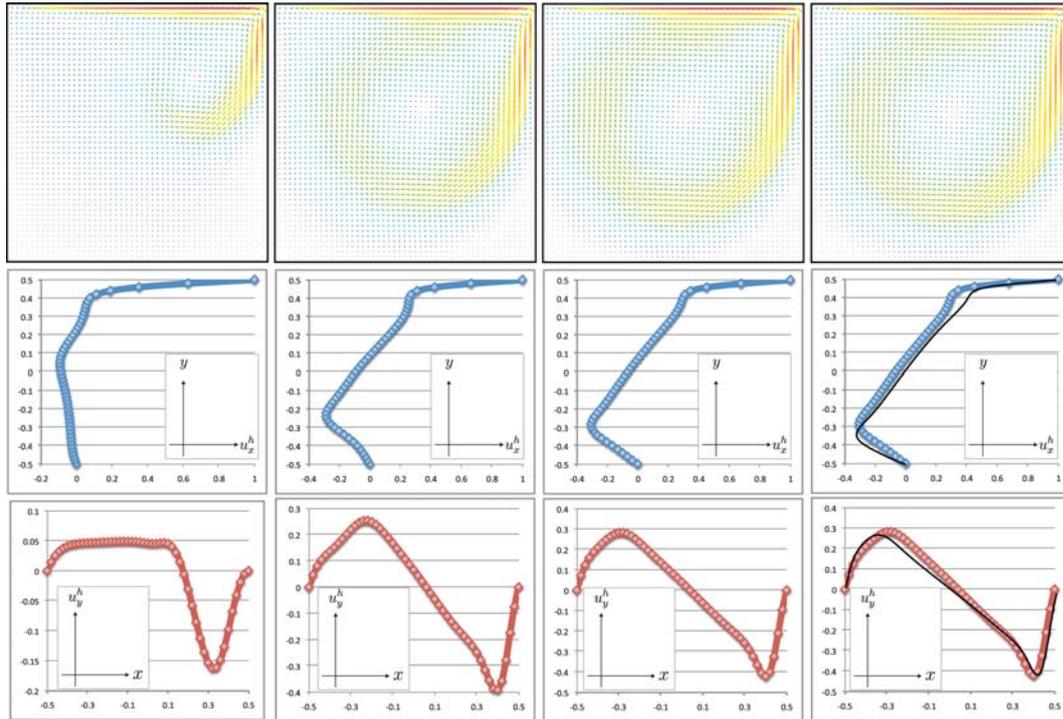


Figure 3.8: Lid-driven cavity flow validation: unsteady case with Reynolds number 1000, mesh with 625 elements and time step 0.001. The columns shows the simulation after 5, 15, 30 and 50  $ds$  of simulated time.

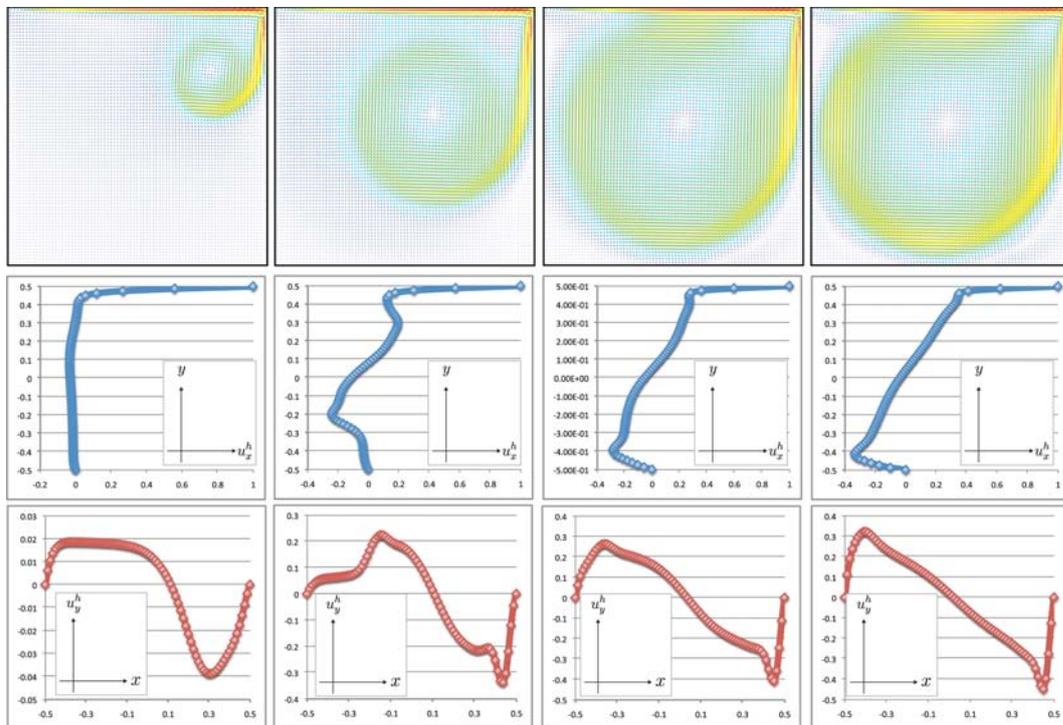


Figure 3.9: Lid-driven cavity flow validation: unsteady case with Reynolds number 4000, mesh with 1600 elements and time step 0.01. The columns shows the simulation after 5, 15, 30 and 50  $ds$  of simulated time.