NSIKE - An Incompressible Navier-Stokes Solver for Unstructured Meshes

Medic Gorazd, Bijan Mohammadi

INRIA
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

NSIKE - an incompressible Navier-Stokes solver for unstructured meshes

Gorazd Medić and Bijan Mohammadi

No 3644

March 1999

THÈME 4

ISSN 0249-6399

INRIA
ROCQUENCOURT

NSIKE - an incompressible Navier-Stokes solver for unstructured meshes

Gorazd Medić * and Bijan Mohammadi †

Thème 4 — Simulation et optimisation
de systèmes complexes

Projet M3N

Rapport de recherche n ° 3644 — March 1999 — 66 pages

Abstract: A new solver for 3D incompressible Navier-Stokes equations is developed. The standard $k-\varepsilon$ turbulence model is implemented for turbulent flow simulation. The incompressible Navier-Stokes solver is based on Chorin's projection method. with finite element discretization. The computation of intermediate velocity and turbulence variables is done explicitely and the stabilization of convection terms is based on PSI (positive streamwise invariant) residual distribution scheme. The Poisson problem for pressure is solved using conjugate gradient technique. In the case of turbulent flow simulations the wall-laws are used for solid boundaries. Test cases presented are 2D flat plate, cavity, backward facing step, circular cylinder and 3D cavity, backward facing step, both in laminar and turbulent regime.

Key-words: incompressible, Navier-Stokes, three-dimensional, $k-\varepsilon$, wall-laws

(Résumé : tsvp)

* e-mail: Gorazd.Medic@inria.fr
† e-mail: Bijan.Mohammadi@inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

NSIKE - un solveur pour les équations de Navier-Stokes incompressibles pour des maillages non-structurés

Résumé : Un nouveau solveur pour les équations de Navier-Stokes incompressibles en trois dimensions a été developpé. Le modèle $k - \varepsilon$ standard a été implémenté pour la simulation des écoulements turbulents. Le solveur est basé sur la méthode de projection de Chorin, en utilisant une discretisation par éléments finis. Le calcul de la vitesse intermediare, k et ε est explicite. La stabilisation du terme d'advection est basé sur la technique de *residual distribution* (le schéma PSI). Le problème de Poisson pour la pression est résolu par la méthode des gradients conjugés. Des lois de paroi sont utilisées pour le calcul des couches limites turbulents. Les cas tests présentés sont : la plaque plane, la cavité, la marche et le cylindre circulaire en 2D, et la cavite et la marche en 3D.

Mots-clé : incompressible, Navier-Stokes, modèle $k - \varepsilon$, lois de paroi

Contents

1 INTRODUCTION

The motivation for the development of a new solver for 3D incompressible Navier-Stokes equations comes basically from our previous experiences in turbulent flow simulations with the compressible solver NSC2KE [28]. The need for a compact, simple (explicit) 3D incompressible solver appeared with our intention of testing different turbulence models for 3D (quasi)stationary incompressible turbulent flows. In that sense, we used widely the experiences accumulated throughout the development and utilization of previously mentioned compressible flow solver.

The new solver is based on the projection method ([2], [3]). The standard $k-\varepsilon$ turbulence model is implemented for turbulent flow simulation. The stabilization of the convection terms is via PSI (Positive Streamwise Invariant) residual distribution scheme ([12], [20]). The Poisson problem for pressure is solved using conjugate gradient technique. In the turbulent flow simulations, the wall-laws are used for the solid boundaries, with the requirement $u \cdot n = 0$ treated in the strong sense.

The resolution of the problem for the intermediate velocity (and of the k and ε equations) in a completely explicit fashion, leads to an extremely small-memory-size (we do not memorize coefficient matrices) and easy-to-handle code with not more than 4000 lines of code (3D code with $k-\varepsilon$ turbulence model).

Furthermore, the CPU times for the simulations carried both in 2D and 3D are reasonable (e.g. 12 hours overnight computation for a steady 3D turbulent problem, with approximately 50000 nodes, on a workstation with a theoretical MFlops peak performance $\approx 10 MFlops$).

We present the numerical results for typical laminar and turbulent test configurations, such as: flat plate, cavity, flow over a step, etc.

2 PROJECTION METHOD

2.1 Incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations are

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \nabla \cdot S + f, \tag{1}$$

$$\nabla \cdot u = 0, \tag{2}$$

where $S = \nu(\nabla u + \nabla u^T)$, $u(x) \in R^d$ is the velocity vector, $p(x) \in R$ the kinematic pressure (divided by density), $\nu(x) \in R$ the kinematic viscosity, $f(x) \in R^d$ is a given body force, $(\nabla u)_{ij} = u_{i,j}$ is the gradient of u, $u \cdot \nabla u$ is $u_i \partial_i u$ and d is the spatial dimension of the problem (here $d = 2$ or 3).

These equations are to be solved in a domain $\Omega \times (0, T_{NS})$, where Ω is an open connected bounded set in R^d with a smooth boundary $\partial\Omega$, subjected to the typical boundary conditions:

- specified velocity (Dirichlet boundary condition):

$$u = w \text{ on } \Gamma_1, \tag{3}$$

- specified tractions (Neumann boundary condition):

$$-p + (S \cdot n \cdot n) = F_n \text{ and } (S \cdot n \cdot s) = F_s \text{ on } \Gamma_2, \tag{4}$$

where $\Gamma_1 \cup \Gamma_2 = \partial\Omega$, n and s represent the outward unit normal and corresponding unit tangent, respectively, and F_n and F_s are the normal and tangential components of the specified boundary 'traction' ($F_n = F_s = 0$ is commonly used when Γ_2 represents an 'outflow' boundary), and initial condition

$$u(x, 0) = u_0(x), \tag{5}$$

where it is required that

$$\nabla \cdot u_0 = 0 \text{ in } \Omega. \tag{6}$$

Remark

Equation (1) implies the following pressure Poisson equation (PPE):

$$\nabla^2 p = \nabla \cdot (f - u \cdot \nabla u) \text{ in } \Omega, \tag{7}$$

with

$$\frac{\partial p}{\partial n} = n \cdot (\nabla \cdot S + f - \frac{\partial u}{\partial t} - u \cdot \nabla u) \text{ on } \Gamma_1 \tag{8}$$

$$p = (S \cdot n \cdot n) - F_n \text{ on } \Gamma_2. \tag{9}$$

2.2 Projection method for Navier-Stokes equations

The projection methods for solving the incompressible Navier-Stokes were introduced by Chorin [2] in 1968, and were since then improved and analyzed by a number of researchers ([3], [8],[5],[6],[7],[10], [9]).

The idea behind the projection method is to decouple the solution for u and p in the original problem (1)-(6) in $\Omega \times (0, T_{NS})$, into a sequential solution of problems for approximate v and $p(T)$, which are *good* approximations of the solution of the original problem. The general framework for the projection method is the following:

(a) Given u_0 with $\nabla \cdot u_0 = 0$, approximate the pressure gradient $\nabla p(t)$, and solve the momentum equations up to a 'projection time' $t = T$, without taking care of the divergence-free constraint, for the *intermediate velocity* $\tilde{u}$.

(b) Perform the projection of the intermediate velocity $\tilde{u}$ onto the subspace of divergence-free vector fields. The result, divergence-free velocity field v, $\nabla \cdot v = 0$, is obtained from the intermediate velocity using the relation $v = \tilde{u} - \nabla\varphi$. This step is equivalent to: Solve for φ

$$\Delta\varphi = \nabla \cdot \tilde{u} \text{ in } \Omega, \tag{10}$$

adding the appropriate boundary conditions (in the case of Dirichlet boundary conditions for u on Γ: $\frac{\partial\varphi}{\partial n} = n \cdot (\tilde{u} - v)$), and update the velocity field

$$v = \tilde{u} - \nabla\varphi \text{ in } \bar{\Omega}, \tag{11}$$

(c) Accepting v as the physical velocity at time $t = T$, determine $p(T)$, and one projection cycle is completed.

The choice of boundary conditions both for the intermediate velocity $\tilde{u}$ and for φ is very important, and it implies the use of the pressure field p_0 and the rate of the change of the pressure field $\dot{p}_0$, that, then again, implies the need for the resolution of two more Poisson problems (for p_0 and $\dot{p}_0$ (see [3]).

The cost behind this general scheme for a projection method lies in the need for solving the three Poisson problems per projection cycle. Simpler schemes can be derived substituting the computation of $p(T)$ by solving the Poisson pressure equation with the estimation of $p(T)$ from φ (the error estimates later will partly justify this step). Furthermore, ignoring the compatibility conditions for the boundary conditions, i.e. simply applying the prescribed boundary conditions for the physical velocity u on the intermediate velocity $\tilde{u}$ (and afterwards not performing the projection $v = \tilde{u} - \nabla\varphi$ on Γ), simplifies the boundary conditions avoiding the computations of ∇p and $\nabla\dot{p}$ on Γ. Further justification of the usage of these schemes can be found in [3] and [9].

The two simplest algorithms which are actually programmed are presented further on.

2.2.1 Algorithm 1

The simplest projection scheme, corresponding to the original Chorin's scheme.

(0) Given u_0 with $\nabla \cdot u_0 = 0$,

(1) Solve for $\tilde{u}(T)$, with $\tilde{u}_0 = u_0$ at $t = 0$, from

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u} \cdot \nabla \tilde{u} = \nabla \cdot \tilde{S} + \tilde{f}, \text{ in } \Omega, \tag{12}$$

$$\tilde{u} = w \text{ on } \Gamma_1, \tag{13}$$

$$(\tilde{S} \cdot n \cdot n) = F_n \text{ and } (\tilde{S} \cdot n \cdot s) = F_s \text{ on } \Gamma_2, \text{ for } 0 < t \leq T. \tag{14}$$

(2) Solve for φ from

$$\nabla^2 \varphi = \nabla \cdot \tilde{u}(T) \text{ in } \Omega, \tag{15}$$

$$\frac{\partial \varphi}{\partial n} = 0 \text{ on } \Gamma_1, \tag{16}$$

$$\varphi = -TF_n \text{ on } \Gamma_2. \tag{17}$$

(3) Compute $v = \tilde{u}(T) - \nabla \varphi$ in $\bar{\Omega}$.

(4) Report v, set $t = 0$, $u_0 = v$ in Ω and on Γ_2, and go to step (1). If pressure is required, it is given by $p(T) = \varphi / T$.

2.2.2 Algorithm 2

A projection scheme of higher order in time proposed by Gresho [3].

(0) Given u_0 with $\nabla \cdot u_0 = 0$ and p_0 solution of a corresponding pressure Poisson problem (7),

(1) Solve for $\tilde{u}(T)$, with $\tilde{u}_0 = u_0$ at $t = 0$, from

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u} \cdot \nabla \tilde{u} = \nabla \cdot \tilde{S} + \tilde{f} - \nabla p_0, \text{ in } \Omega, \tag{18}$$

$$\tilde{u} = w \text{ on } \Gamma_1, \tag{19}$$

$$(\tilde{S} \cdot n \cdot n) = F_n + p_0 \text{ and } (\tilde{S} \cdot n \cdot s) = F_s \text{ on } \Gamma_2, \text{ for } 0 < t \leq T. \tag{20}$$

(2) Solve for φ from

$$\nabla^2 \varphi = \nabla \cdot \tilde{u}(T) \text{ in } \Omega, \tag{21}$$

$$\frac{\partial \varphi}{\partial n} = 0 \text{ on } \Gamma_1, \tag{22}$$

$$\varphi = -\frac{T}{2}(F_n + p_0) \text{ on } \Gamma_2. \tag{23}$$

(3) Compute $v = \tilde{u}(T) - \nabla\varphi$ in $\bar{\Omega}$.

(4) Compute $p(T) = p_0 + 2\varphi/T$ in $\bar{\Omega}$.

(5) Report v and p, set $t = 0$, $u_0 = v$ in Ω and on Γ_2, $p_0 = p(T)$ and go to step (1).

2.2.3 Error estimations

The most interesting point connected with the projection method is the question how are the values of v and $p(T)$, i.e the values at the end of the projection cycle, related to the solution of the original problem (1)-(6).

In [3], the rough estimates are presented (see also Appendix 5), and for the Algorithm 2 we have

$$v = u(T) + O(T^3) \qquad \text{in} \quad \Omega, \tag{24}$$

$$\varphi = \frac{T^2}{2}\dot{p_0} + O(T^3) \qquad \text{in} \quad \bar{\Omega}. \tag{25}$$

In the same way, for the Algorithm 1:

$$v = u(T) + O(T^2) \qquad \text{in} \quad \Omega, \tag{26}$$

$$\varphi = Tp_0 + O(T^2) = Tp(T) + O(T^2) \qquad \text{in} \quad \bar{\Omega}. \tag{27}$$

The estimation of pressure, in the Algorithm 1 via $p(T) \approx \varphi/T$ and in the Algorithm 2 via $\varphi \approx \frac{T^2}{2}\dot{p_0}$ using the approximation $p(T) = p_0 + T\dot{p_0} + O(T^2)$, is based on these results.

The analyses presented in [3] and [9] indicate that these estimations are indeed true, even for the actual algorithms that are programmed (that introduce the boundary conditions incompatility), but outside the spurious boundary layer, i.e for $x > \delta$, where $\delta \equiv \sqrt{(\nu T)}$ (this gives a restriction on the projection cycle time T).

2.3 Reynolds averaged Navier-Stokes equations and $k-\varepsilon$ turbulence model

The increase in Reynolds number (nondimensional flow parameter, $Re = \frac{u_\infty L}{\nu}$) leads to the loss of stability of fluid flow, generating more and more complex flow regimes, finally ending in a chaotic flow regime, turbulent flow. Although the Navier-Stokes equations describe even these complicated flow regimes, the direct solution of the original Navier-Stokes equations becomes more and more expensive regarding the scales of the discretization needed to resolve all the flow scales present in turbulent flow, keeping this approach out of hand today. On the other hand, the behavior of some mean flow quantities, and not of small-scale fluctuations themselves, is often more interesting when analyzing the flow. This is in fact the starting point of the turbulence modeling approach, where the turbulent effects are modeled in different ways, and one attempts to solve the equations for the mean flow quantities with the addition of the turbulence model itself.

Introducing the flow variables decomposition $u = U + u'$ and $p = P + p'$, U,P being the mean part and u',p' being the fluctuating or nonrelevant part of the velocity and pressure yields the following so-called Reynolds averaged Navier-Stokes equations:

$$\nabla \cdot U = 0 \tag{28}$$

$$\frac{\partial U}{\partial t} + U \cdot \nabla U + \nabla P - \nabla \cdot S + \nabla \cdot \langle u' \otimes u' \rangle = 0, \tag{29}$$

where $\langle\rangle$ is the averaging (filtering) operator. The last term on the right hand side of (29) contains the complete influence of the fluctuations field on the mean flow (the effects of turbulence). Usually, it is named the Reynolds stress tensor R, and there exist different approaches for its evaluation.

In the standard $k-\varepsilon$ model [25], one assumes (Reynolds hypothesis)

$$R = -\frac{2}{3}kI + \nu_t(\nabla U + \nabla U^T), \tag{30}$$

where $k = \frac{1}{2}\langle |u'|^2 \rangle$ is the kinetic energy of turbulence and ν_t is the so-called turbulent (or "eddy") viscosity defined as $\nu_t = c_\mu \frac{k^2}{\varepsilon}$, where $\varepsilon = \frac{\nu}{2}\langle |\nabla u' + \nabla u'^T|^2 \rangle$ is the dissipation of the kinetic energy of turbulence.

The exact equations for k and ε can be derived, but they contain many unknown terms (the so-called higher order moments), so one has to introduce approximations [26]. The standard equations of the $k-\varepsilon$ model are [25]

$$\frac{\partial k}{\partial t} + U \cdot \nabla k - \nabla \cdot (c_\mu \frac{k^2}{\varepsilon} \nabla k) = \frac{1}{2} R : (\nabla U + \nabla U^T) - \varepsilon, \tag{31}$$

$$\frac{\partial \varepsilon}{\partial t} + U \cdot \nabla \varepsilon - \nabla \cdot (c_\varepsilon \frac{k^2}{\varepsilon} \nabla \varepsilon) = c_1 \frac{\varepsilon}{k} \frac{1}{2} R : (\nabla U + \nabla U^T) - c_2 \frac{\varepsilon^2}{k}, \tag{32}$$

where $c_\mu = 0.09$, $c_\varepsilon = 0.07$, $c_1 = 1.44$ and $c_2 = 1.92$.

Finally, one has to solve the following problem in $\Omega \times (0,T)$:

$$\nabla \cdot U = 0 \tag{33}$$

$$\frac{\partial U}{\partial t} + U \cdot \nabla U + \nabla P - \nabla \cdot S_t = 0, \qquad S_t = (\nu + \nu_t)(\nabla U + \nabla U^T), \tag{34}$$

$$\frac{\partial k}{\partial t} + U \cdot \nabla k - \nabla \cdot (c_\mu \frac{k^2}{\varepsilon} \nabla k) = \frac{1}{2} c_\mu \frac{k^2}{\varepsilon} |\nabla U + \nabla U^T|^2 - \varepsilon, \tag{35}$$

$$\frac{\partial \varepsilon}{\partial t} + U \cdot \nabla \varepsilon - \nabla \cdot (c_\varepsilon \frac{k^2}{\varepsilon} \nabla \varepsilon) = c_1 c_\mu k \frac{1}{2} |\nabla U + \nabla U^T|^2 - c_2 \frac{\varepsilon^2}{k}, \tag{36}$$

with the corresponding boundary conditions on $\partial\Omega$ and initial conditions for $t = 0$.

Remarks

1. The isotropic part of the Reynolds stress tensor $-\frac{2}{3}kI$ has been neglected, considered small compared to the values of pressure. Otherwise, it is possible to consider modified pressure $P' = P + \frac{2}{3}k$

2. We use the values of the constants slightly different to the original values established by Launder [25], i.e. $c_2 = 11/6$ and $c_\varepsilon = 1/1.4225$ (see [29], [30]).

3. The so-called *shear-based* $k-\varepsilon$ model has also been tested and used. The difference with the standard model is in the choice of adding the "turbulent" viscosity ν_t to the molecular viscosity ν in the expression $S_t = (\nu + \nu_t)(\nabla U + \nabla U^T)$ just for the non-diagonal terms.

4. Further on, we use the notation $S_k = \frac{1}{2} c_\mu \frac{k^2}{\varepsilon} |\nabla U + \nabla U^T|^2 - \varepsilon$ and $S_\varepsilon = c_1 c_\mu k \frac{1}{2} |\nabla U + \nabla U^T|^2 - c_2 \frac{\varepsilon^2}{k}$ for the source terms in the equations for k and ε, respectively.

2.3.1 Wall laws

Significant difficulty in the resolution of the above problem comes out from the strong variations of flow variables in the vicinity of $\Gamma_w \in \partial\Omega$, where Γ_w represents the solid wall $u_w = 0$. Furthermore, the standard equations of the $k-\varepsilon$ model are not valid for the region near the solid walls where the viscous effects play more significant role. Different strategies have been developed to overcome these difficulties: the wall-laws approach and the two-layer approach. The computations presented here were carried out using the wall-laws, although the two-layer $k-\varepsilon$ model is also implemented.

The basic idea of the wall-laws approach consists in removing the boundary layers from the computational domain by changing the computational domain Ω in $\Omega_\delta = \Omega - B_\delta$ where B_δ is defined as:

$$B_\delta = \{x - n(x)\lambda : x \in \Gamma_w, \lambda \in (0, \delta(x))\}, \tag{37}$$

where n is the outer unit normal on Γ_w. The computational domain is now Ω_δ and the new boundary Γ_δ replaces Γ_w. In this wall-laws implementation, the parameter δ is given apriori and has the same value everywhere on Γ_w and is kept constant during the computations.

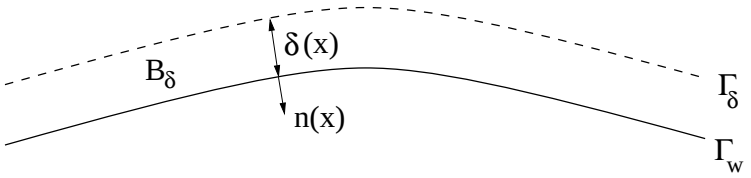


Figure 2: Domain boundary transformation.

The second step in the application of the wall-laws is the derivation of the boundary conditions on Γ_δ. Experiments show some universality in the behavior of turbulent boundary layers. More precisely, the following relations are valid on Γ_δ (for details on the implementation of these boundary conditions, see [29]:

$$U \cdot n = 0, \tag{38}$$

$$(S_t \cdot n \cdot s^*)s^* = -u_\tau^2 s^*, \tag{39}$$

$$k = \frac{u_\tau^2}{\sqrt{c_\mu}}\alpha, \quad \varepsilon = \frac{u_\tau^3}{\kappa\delta} min(1, \alpha + \frac{0.2\kappa(1-\alpha)^2}{\sqrt{c_\mu}}), \tag{40}$$

where $\alpha = min(1, \frac{y^+}{10})$, with $y^+ = \frac{u_\tau \delta}{\nu}$, reproduces the behaviour of k when δ tends to zero (and the correction for the boundary condition for ε is chosen to reproduce the behavior in the near-wall region, see [29]), $S_t = (\nu + \nu_t)(\nabla U + \nabla U^t)$, and the unit tangent s^* is defined as

$$s^* = \frac{(U - (U \cdot n)n)}{|U - (U \cdot n)n|}. \tag{41}$$

The friction velocity u_τ is obtained as the solution of

$$\frac{U \cdot s^*}{u_\tau} = f(u_\tau), \tag{42}$$

where for $f(u_\tau)$ we use the nonlinear Reichardt equation:

$$f_r(y^+) = 2.5log(1 + \kappa y^+) + 7.8(1 - e^{-y^+/11} - \frac{y^+}{11}e^{-0.33y^+}), \tag{43}$$

which presents a generalization of standard log-law.

Remarks

1. The authors have also used convection and pressure gradient corrections to the standard Reichardt equation, developed in [29].

2. The standard log-law is also implemented, i.e.

$$\frac{U \cdot s^*}{u_\tau} = y^+, \text{ for } y^+ < 20, \tag{44}$$

$$\frac{U \cdot s^*}{u_\tau} = \frac{1}{\kappa}log(y^+) + \beta, \tag{45}$$

where $\kappa = 0.41$ and $\beta = 5.5$ are experimentally determined constants.

2.3.2 Two-layer approach

In the region where y^+ is small, the standard $k-\varepsilon$ model used in the high-Reynolds region is no longer valid. Therefore, a two-layer $k-\varepsilon$ model [26] is implemented allowing the integration up to the solid wall. It is done in the following manner:

For $y^+ < 200$ the following transport equation for kinetic energy of turbulence k is used:

$$\frac{\partial k}{\partial t} + U \cdot \nabla k - \nabla \cdot ((\nu + \nu_t)\nabla k) = \frac{1}{2}c_\mu \frac{k^2}{\varepsilon}|\nabla U + \nabla U^T|^2 - Diss, \tag{46}$$

where

$$Diss = \frac{k^{\frac{3}{2}}}{l_\varepsilon}, \quad \nu_t = c_\mu \sqrt{k} l_\mu$$

and l_ε and l_μ are two length scales containing the damping effects in the near wall regions.

$$l_\mu = \kappa c_\mu^{-3/4} y(1 - e^{(\frac{-y^+}{c})}),$$

$$l_\varepsilon = \kappa c_\mu^{-3/4} y(1 - e^{(\frac{-y^+}{2\kappa c_\mu^{-3/4}})})$$

where $c = 70$ and $\kappa = 0.41$ and y^+ is defined as

$$y^+ = \frac{\sqrt{k}y}{\nu},$$

where y is the distance of the current point to the solid wall.

The boundary conditions for velocity on the part of the boundary representing the solid wall are now:

$$U = 0, \quad \text{on } \Gamma_3, \tag{47}$$

while the boundary condition for kinetic energy of turbulence is

$$k = 0, \quad \text{on } \Gamma_3. \tag{48}$$

Remark

In the context of the application of wall-laws, it should be a-posteriori varified that y^+ for a converged solution takes the values for which the wall-laws expressions are valid. However, in the cases where y^+ takes small values, the same problem of validity of $k-\varepsilon$ model arises. A possible solution is to couple the wall-laws with tho layer technique, i.e. to use the approach described above for the part of the domain where $y^+ < 200$.

2.4 Projection method for Reynolds averaged Navier-Stokes equations and $k-\varepsilon$ turbulence model

The introduction of the flow decomposition and the turbulence model leads to the replacement of the original Navier-Stokes problem (1)-(6) with the modified problem (33)-(36) for the mean velocity U, mean kinematic pressure P, kinetic energy of turbulence k and turbulent dissipation ε.

The Algorithm 1 reads now:

(0) Given U_0 with $\nabla \cdot U_0 = 0$, k_0 and ε_0

(1a) Solve for $\tilde{U}(T)$, with $\tilde{U}_0 = U_0$ at $t = 0$, from

$$\frac{\partial \tilde{U}}{\partial t} + \tilde{U} \cdot \nabla \tilde{U} = \nabla \cdot \tilde{S}_t + \tilde{f}, \text{ in } \Omega, \tag{49}$$

$$\tilde{U} = w \text{ on } \Gamma_1, \tag{50}$$

$$(\tilde{S}_t \cdot n \cdot n) = 0 \text{ and } (\tilde{S}_t \cdot n \cdot s) = 0 \text{ on } \Gamma_2, \tag{51}$$

$$\tilde{U} \cdot n = 0 \text{ on } \Gamma_3 \tag{52}$$

$$(\tilde{S}_t \cdot n \cdot s)s = -u_\tau^2 s \text{ on } \Gamma_3, \text{ for } 0 < t \leq T. \tag{53}$$

(1b) Solve for $k(T)$ and $\varepsilon(T)$, with $U(t) = U_0$, $k(0) = k_0$ and $\varepsilon(0) = \varepsilon_0$

$$\frac{\partial k}{\partial t} + U \cdot \nabla k - \nabla \cdot (c_\mu \frac{k^2}{\varepsilon} \nabla k) = \frac{1}{2} c_\mu \frac{k^2}{\varepsilon} |\nabla U + \nabla U^T|^2 - \varepsilon, \text{ in } \Omega, \tag{54}$$

$$\frac{\partial \varepsilon}{\partial t} + U \cdot \nabla \varepsilon - \nabla \cdot (c_\varepsilon \frac{k^2}{\varepsilon} \nabla \varepsilon) = c_1 c_\mu k \frac{1}{2} |\nabla U + \nabla U^T|^2 - c_2 \frac{\varepsilon^2}{k}, \text{ in } \Omega, \tag{55}$$

$$k = k_{\Gamma_1}, \qquad \varepsilon = \varepsilon_{\Gamma_1} \text{ on } \Gamma_1, \tag{56}$$

$$\nabla k \cdot n = 0, \qquad \nabla \varepsilon \cdot n = 0 \text{ on } \Gamma_2, \tag{57}$$

$$k = k_w(u_\tau), \quad \varepsilon = \varepsilon_w(u_\tau) \text{ on } \Gamma_3, \text{ for } 0 < t \leq T. \tag{58}$$

(2) Solve for φ from

$$\nabla^2 \varphi = \nabla \cdot \tilde{U}(T) \text{ in } \Omega, \tag{59}$$

$$\frac{\partial \varphi}{\partial n} = 0 \text{ on } \Gamma_1 \cup \Gamma_3, \tag{60}$$

$$\varphi = 0 \text{ on } \Gamma_2. \tag{61}$$

(3) Compute $V = \tilde{U}(T) - \nabla \varphi$ in $\bar{\Omega}$.

(4) Report V, $k(T)$, $\varepsilon(T)$, set $t = 0$, $U_0 = V$, $k_0 = k(T)$ and $\varepsilon_0 = \varepsilon(T)$ in Ω and on Γ_2, and go to step (1). If pressure is required, it is given by $p(T) = \varphi/T$.

Remarks

1. Instead of $S = \nu(\nabla U + \nabla U^T)$ in the Navier-Stokes equations (and in the projection method algorithms presented), now we have $S_t = (\nu+\nu_t)(\nabla U+\nabla U^T)$, where in this implementation ν_t is computed explicitly, i.e. throughout the projection cycle one takes $\nu_t(t) = c_\mu \frac{k_0^2}{\varepsilon_0}$.

2. Here F_n and F_s are supposed $F_n = F_s = 0$.

3. There is a part of boundary $\Gamma_3 \equiv \Gamma_w$, the solid wall, where the wall-laws are applied ($\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 = \partial\Omega$). We have to compute on Γ_3 the friction velocity u_τ throughout the projection cycle, solving the nonlinear Reichardt equation.

4. Applying the wall-laws on the intermediate velocity $\tilde{U}$, we have, finally, for the velocity at the end of the projection cycle

$$V \cdot n = 0 \text{ on } \Gamma_3 \tag{62}$$

since $\nabla\varphi \cdot n = 0$ on Γ_3, and

$$\begin{aligned} S_t \cdot n &= \tilde{S}_t \cdot n - (\nu + \nu_t)(\nabla(\nabla\varphi) + \nabla(\nabla\varphi)^T) \cdot n \\ &= \tilde{S}_t \cdot n + (\nu + \nu_t)(\nabla n + \nabla n^T) \cdot \nabla\varphi \end{aligned} \tag{63}$$

giving the correction needed to the boundary condition for $\tilde{u}$ to have the right value for $(S_t \cdot n \cdot s)s$ at the end of the projection cycle.

5. In each projection cycle $0 < t \leq T$, in the equations of $k - \varepsilon$ model $U(t)$ is in in this implementation taken as U_0, decoupling in this way the problem of the computation of V and $P(T)$ from the computation of $k(T)$ and $\varepsilon(T)$.

3 FINITE ELEMENT METHOD

The solution of the original problem (1)-(6) for $0 < t \leq T_{NS}$ is now decomposed in the resolution of the sequence of the projection cycle problems for $0 < t \leq T$, using the algorithms presented.

We introduce a partition of the time interval $[0, T_{NS}]$, $t^n = nT$, for $0 \leq n \leq N$, where $T = \frac{T_{NS}}{N}$. We define three sequence of functions $\{v_h^n\}$, $\{\tilde{u}_h^n\}$, $\{p_h^n\}$, such that $(x \rightarrow v_h^n(x))_n (v_h^n \approx v_h(., t^n))$, with the initialization $v_h^0 = u_0$, $\tilde{u}_h^0 = u_0$ and $p_h^0 = p_0$.

In the case of Reynolds averaged Navier-Stokes computations (for turbulent flow computations) we use the notation: $\{V_h^n\}$, $\{\tilde{U}_h^n\}$,$\{P_h^n\}$ $\{k_h^n\}$ and $\{\varepsilon_h^n\}$.

Let $\Omega_h = \cup_j K_j$ be a discretization of the computational domain $\Omega \subset R^d$ by triangles when $d = 2$ and by tetrahedra when $d = 3$, and let $\Omega_h = \cup_i C_i$ be its partition in median dual cells (see Figure 1). Let $\mathcal{E}(K_j)$ be the set of sides of K_j triangle in 2D and the set of faces of K_j tetrahedra in 3D, and let $\{q^i\}_1^N$ be the vertices of our disretization.

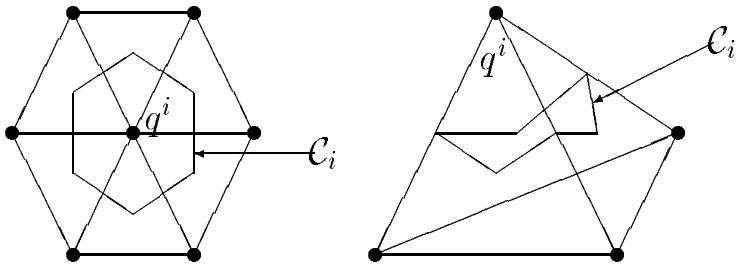


Figure 2: Median dual cells in 2D and 3D.

Let V_h be the set of continuous affine functions on our discretization

$$V_h = \{\psi_h : \Omega_h \rightarrow R, \psi_h \in C^0(\Omega_h) : \forall K_j \in \Omega_h, \psi_h|_{K_j} \in P^1\}. \tag{64}$$

Furthermore, let we introduce the notation

$$(a, b) = \int_{\Omega_h} a_i b_i, \qquad a, b \in L^2(\Omega_h)^d, \tag{65}$$

and

$$(A, B) = \int_{\Omega_h} A_{ij} B_{ij}, \qquad A, B \in L^2(\Omega_h)^{d \times d}. \tag{66}$$

In the following subsection we will discuss the finite element formulation for the solution of problem (33)-(36), i.e. the implementation of projection method, algorithm 1 in particular, for Reynolds averaged Navier-Stokes equations with $k - \varepsilon$ model and wall-laws. The algorithm for the solution of problem (1)-(6) can be easily reconstructed.

3.1 Finite element formulation

STEP 1 - Intermediate velocity and turbulence variables computation:

Let

$$J_{0nh} = \{\phi_h \in V_h^d, \phi_h = 0 \text{ on } \Gamma_{1h}, \phi_h \cdot n_h = 0 \text{ on } \Gamma_{3h}\}, \tag{67}$$

$$W_{0h} = \{w_h \in V_h, w_h = 0 \text{ on } \Gamma_{1h} \cup \Gamma_{3h}\}. \tag{68}$$

Introducing fully explicit time discretization we can write the following Petrov-Galerkin formulation for the intermediate velocity and turbulent variables computation:

Find $\tilde{U}_h^{n+1}$ such that, $\forall \phi_h \in J_{0nh}$

$$(\frac{\tilde{U}_h^{n+1} - V_h^n}{T}, \phi_h) + (V_h^n \cdot \nabla V_h^n, \phi_h) + (S_{t\,h}^n, \nabla \phi_h) - \int_{\Gamma_{3h}} S_{t\,h}^n \cdot n \cdot \phi_h$$

$$+ \sum_{K, K \in \Omega_h} \int_K \left(\frac{\tilde{U}_h^{n+1} - V_h^n}{T} + V_h^n \cdot \nabla V_h^n - \nabla \cdot S_{t\,h}^n \right) \cdot g_1^K(V_h^n, \phi_h) = 0, \tag{69}$$

and

$$\tilde{U}_h^{n+1} - w_h' \in J_{0nh}, \tag{70}$$

where w_h' is an approximation of w' in V_h^d, and w' is an extension of w in $H^1(\Omega)^d$ satisfying $w \cdot n = 0$ on Γ_3.

Find $k_h^{n+1}, \varepsilon_h^{n+1}$ such that, $\forall w_h \in W_{0h}$

$$(\frac{k_h^{n+1} - k_h^n}{T}, w_h) + (V_h^n \cdot \nabla k_h^n, w_h) + (c_\mu \frac{(k_h^n)^2}{\varepsilon_h^n} \nabla k_h^n, \nabla w_h) - (S_{k\,h}^n, w_h) +$$

$$\sum_{K, K \in \Omega_h} \int_K \left(\frac{k_h^{n+1} - k_h^n}{T} + V_h^n \cdot \nabla k_h^n - \nabla \cdot (c_\mu \frac{(k_h^n)^2}{\varepsilon_h^n} \nabla k_h^n) - S_{k\,h}^n \right) g_2^K(V_h^n, k_h^n, w_h) = 0, \tag{71}$$

$$(\frac{\varepsilon_h^{n+1} - \varepsilon_h^n}{T}, w_h) + (V_h^n \cdot \nabla \varepsilon_h^n, w_h) + (c_\varepsilon \frac{(k_h^n)^2}{\varepsilon_h^n} \nabla \varepsilon_h^n, \nabla w_h) - (S_{\varepsilon\,h}^n, w_h) +$$

$$\sum_{K, K \in \Omega_h} \int_K \left(\frac{\varepsilon_h^{n+1} - \varepsilon_h^n}{T} + V_h^n \cdot \nabla \varepsilon_h^n - \nabla \cdot (c_\varepsilon \frac{(k_h^n)^2}{\varepsilon_h^n} \nabla \varepsilon_h^n) - S_{\varepsilon\,h}^n \right) g_2^K(V_h^n, \varepsilon_h^n, w_h) = 0, \tag{72}$$

and

$$k_h^{n+1} - k_h' \in W_{0h}, \qquad \varepsilon_h^{n+1} - \varepsilon_h' \in W_{0h}, \tag{73}$$

where k_h' and ε_h' are the approximations of functions k' and ε' in V_h, where k' and ε' are the functions in $H^1(\Omega)$ that satisfy $k' = k_\Gamma$ on Γ_1, $k' = k_w$ on Γ_3, and $\varepsilon' = \varepsilon_\Gamma$ on Γ_1, $\varepsilon' = \varepsilon_w$ on Γ_3, respectively.

Different stabilized finite element methods have been proposed so far. SUPG (Streamline Upwind Petrov-Galerkin) methods [14], [15] and the recent extensions [16], [17] have been well understood and widely analyzed.

On the other hand, in the context of residual distribution schemes (the so-called fluctuation splitting approach [12], [13]), a number of schemes was developed, e.g. LDA (Low Diffusion) and PSI (Positive Streamwise Invariant) schemes, taking care of the scheme properties like *positivity* and *linearity preservation*. The relation of these schemes with standard stabilized finite element formulations is been analyzed and some interpretions have been proposed [18], [19], [20], where the general residual distribution schemes are rewritten as a Petrov-Galerkin formulation where the stabilizing functions g_1^K and g_2^K are appropriately chosen. If we chose $g_1^K = \tau_K V_h^n \cdot \nabla \phi_h$ and $g_2^K = \tau_K V_h^n \cdot \nabla w_h$ we recover the SUPG (Streamwise upwind Petrov-Galerkin) methods. However, we will chose the stabilization via PSI scheme, and the choice of g_1^K and g_2^K will be discussed later.

STEP 2 - Pressure computation:

Let

$$Q_{0h} = \{\psi_h \in V_h, \psi_h = 0 \text{ on } \Gamma_{2h}\}. \tag{74}$$

The Galerkin finite element formulation of the Poisson problem for φ is:

Find $\varphi_h^{n+1} \in Q_{0h}$ such that, $\forall \psi_h \in Q_{0h}$

$$\int_{\Omega_h} \nabla \varphi_h^{n+1} \cdot \nabla \psi_h = -\int_{\Omega_h} \nabla \cdot \tilde{U}_h^{n+1} \, \psi_h. \tag{75}$$

STEP 3 - Velocity update:

Find V_h^{n+1} such that, $\forall \phi_h \in J_{0nh}$

$$(V_h^{n+1}, \phi_h) = (\tilde{U}_h^n, \phi_h) - (\nabla \phi_h^{n+1}, \phi_h) \tag{76}$$

and

$$\tilde{U}_h^{n+1} - w_h' \in J_{0nh}, \tag{77}$$

where w_h' is an approximation of w' in V_h^d, and w' is an extension of w in $H^1(\Omega)^d$ satisfying $w \cdot n = 0$ on Γ_3.

3.2 Velocity computation

Let $\{\psi_i\}_{1..N}$ be the basis of $V_{0h} = \{\psi_h \in V_h, \psi_h = 0 \text{ on } \Gamma_{1h}\}$, and let $\{\phi_{i,j}\}_{1..N,1..d}$ be the basis of J_{0nh} defined as

$$\left.\begin{array}{l} \phi_{i,1} = n_i\psi_i \\ \phi_{i,2} = s_i^1\psi_i \\ \phi_{i,3} = s_i^2\psi_i \end{array}\right\} \text{ if } q^i \in \Gamma_{3h} \tag{78}$$

$$\phi_{i,j} = \psi_i e_j, \qquad j = 1,..,d \qquad \text{elsewhere} \tag{79}$$

with $s = s^1 + s^2$, s being the unit tangent defined before, and $\{n, s^1, s^2\}$ being the local orthonormal coordinate system with positive orientation, defined for each node q^i. The definition of nodal normal and tangent can pose problems especially for domains containing corners and this problem is emphasized in 3D. We have chosen to compute the nodal normals (and corresponding tangents) as a weighted average of normals of all edges (faces) having the node q_i in common (weighted by the length of edge in 2D, and by the area of face in 3D). This will be explained in more details when explaining the wall-laws implementation. $\{e_j\}$ is the canonical basis of R^d, and $\tilde{U}_h$ can be written in this basis as

$$\tilde{U}_h = \sum_{i=1..N} \sum_{j=1..d} \tilde{U}_{i,j}\phi_{i,j} + w'_h. \tag{80}$$

For determing the system of equations for the computation of nodal values $\tilde{U}_i^{n+1}$ we take the above basis functions as ϕ_h in the formulation (69). We will not apply the stabilization via g_1^K function on the nonstationary term, and that will together with the mass lumping (the Gaussian points of numerical integration coincide with the element nodes) while integrating nonstationary term lead to the decoupling of the discrete equations for $\tilde{U}_i^{n+1}$.

Finally, the function g_1^K defining the stabilization is chosen as $g_1^K = \beta^K(V_h^n, \phi_h) - \phi_h$, where β^K is a function given by PSI scheme. Furthermore, we will use one Gaussian point numerical integration for all diffusion terms in (69) and three nodes as integration points for convection term.

This leads to the following scheme for all nodes not on $\Gamma_{1h} \cup \Gamma_{3h}$.

$$\tilde{U}_{i,j}^{n+1} = V_{i,j}^n - \frac{T_i}{|C_i|} \sum_{K, K\in\Omega_h} (\beta_{i,j}^K \sum_{l=1}^{d+1} (V_{m,j}^n \cdot \nabla\phi_{l,j}) V_{l,j}^n |K| + S_{tK}^n \cdot \nabla\phi_{i,j}{}^K |K|). \tag{81}$$

On Γ_{1h} we apply Dirichlet boundary conditions ($\tilde{U}_i = w'_h(q^i)$) and on Γ_{3h} we have the following scheme ($(\tilde{U}\cdot n)_i^{n+1} = 0$, with the necessity that $(\tilde{U}\cdot n)_i^0 = 0$)

$$(\tilde{U}\cdot s)_i^{n+1} = (V\cdot s)_i^n - \frac{T_i}{|C_i|}s_i^n\cdot(\sum_{K,K\in\Omega_h}(\beta_i^K\sum_{l=1}^{d+1}(V_m^n\cdot\nabla\phi_l)V_l^n|K|$$

$$+S_{tK}^n\cdot\nabla{\phi_i}^K|K|) + \text{ boundary integral on } \Gamma_{3h}). \tag{82}$$

The computation of the boundary integral on Γ_{3h} will be discussed in the subsection that considers the wall-laws implementation.

In the discrete formulations (81) and (82) V_m^n is the mean element velocity from the previous step S_{tK}^n is S_t tensor element value from the previous step. Index l indicates local element node numbering and the correspondance $l \to i$ is defined with the triangulation Ω_h.

The values of $\beta_{i,j}^K$ can be found in [12]:

$$\beta_{i,j}^K = \frac{max(0,b_i)min(0,(V_{i,j}-V_{in,j}^K)(V_{out,j}^K-V_{in,j}^K))}{\sum_{l=1}^{d+1}max(0,b_l)min(0,(V_{l,j}-V_{in,j}^K)(V_{out,j}^K-V_{in,j}^K))}, \tag{83}$$

and

$$V_{in,j}^K = \frac{\sum_{l=1}^{d+1}V_{l,j}min(0,b_l)}{\sum_{l=1}^{d+1}min(0,b_l)}, \qquad V_{out,j}^K = \frac{\sum_{l=1}^{d+1}V_{l,j}max(0,b_l)}{\sum_{l=1}^{d+1}max(0,b_l)}, \tag{84}$$

where $b_i = \frac{1}{d}V_m^n\cdot n_i$ and n_i is the inward unit normal to the triangle edge (tetahedra face) opposite to the node i, multiplied with the length of corresponding triangle edge (the area of tetrahedra face).

The extension for the final step of the same algorithm (velocity update) is straightforward (once more, for all nodes not on Γ_{1h} and Γ_{3h}):

$$V_{i,j}^{n+1} = \tilde{U}_{i,j}^{n+1} - \frac{1}{|C_i|}\sum_{K,K\in\Omega_h}\frac{|K|}{d+1}\nabla\varphi_j^K, \tag{85}$$

where the computation of φ_h is explained below ($\varphi_K \in P^1$).

The time step in the formulations (81) and (82) is computed as

$$T = CFL\min_{q_i}\min_{K,q^i\in K}\frac{\Delta x_K}{|u_K| + \frac{2(\nu+\nu_t)_K}{\Delta x_K}}, \tag{86}$$

where Δx_K is the minimum element height, ν_K is the mean element viscosity, $(\nu_t)_K$ is the mean element turbulent viscosity and $|u_k|$ is the norm of the mean element velocity. We've chosen o apply global time stepping, but for steady flows local time stepping strategy can also be considered. The time step calculation procedure is explained further on in Appendix 4.

3.3 The computation of pressure

Let $\{\psi_i\}_{1..N}$ be the basis of Q_{0h}. Taking into account the boundary conditions, the introduction of the finite element discretization leads to the system

$$A\Phi = F, \tag{87}$$

where A is the positive definite symmetric $n_s \times n_s$ matrix

$$A_{ij} = \int_{\Omega_h} \nabla\psi_j \cdot \nabla\psi_i, \qquad F_i = \int_{\Omega_h} \nabla \cdot \tilde{u}_h(T)\psi_i, \tag{88}$$

and the solution vector is $\Phi = (\varphi_h(q^i))_{i\in\{1,\dots,n_s\}}$, n_s being the number of nodes of the triangulation. The matrix A and vector F are furthermore corrected to take into account the Dirichlet the boundary condition on Γ_{2h}:

$$A_{ij} = 0., \quad i \neq j, \quad A_{ii} = 1., \quad F_i = 0. \quad \text{if} \quad q^i \in \Gamma_{2h}. \tag{89}$$

Conjugate gradient method is applied to the resolve the linear system $A\Phi = F$.

The general preconditioned conjugate gradient method algorithm for the resolution of $N \times N$ linear system is the following:

(0) Initialization
Choose the preconditioning matrix $C \in R^{N\times N}$ (it has to be symmetric positive definite), the stopping criterium (computation precision) ϵ or the maximum number of iterations m_{max}, the initialization Φ^0 (0 for example), and the following initialization for g^0 and h^0: $g^0 = h^0 = C^{-1}(A\Phi^0 - F)$. Set $m = 0$.

(1) Compute:

$$\rho^m = \frac{< g^m, h^m >_C}{< h^m, C^{-1}Ah^m >_C} = \frac{< g^m, h^m >_C}{h^{mT}Ah^m}, \tag{90}$$

$$\Phi^{m+1} = \Phi^m - \rho^m h^m, \tag{91}$$

$$g^{m+1} = g^m - \rho^m C^{-1}Ah^m, \tag{92}$$

$$\gamma^m = \frac{||g^{m+1}||_C^2}{||g^m||_C^2}, \tag{93}$$

$$h^{m+1} = g^{m+1} + \gamma^m h^m, \tag{94}$$

where $< a, b >_C = a^T Cb$ and $||a||_C = (a^T Ca)^{\frac{1}{2}}$.

(2) Check if $||g^{m+1}||_C^2 < \epsilon||g^0||_C^2$, then Φ^{m+1} is the approximated solution of the problem, if not, increase m and return to (1), if $m \leq m_{max}$.

Remarks

1. In the case of Algorithm 1, the initialization for Φ^0 in the $n+1$ global iteration step, is taken as a result from the previous iteration, i.e. $\Phi^0 = P^n T$. In the first iteration, $\Phi^0 = P^0 T$, P^0 chosen to satisfy the conditions for the initialization of the projection scheme.

2. In the case of Algorithm 2, the initialization for Φ^0 is always the same, i.e $\Phi^0 = 0$.

3. The preconditioning matrix C can be chosen as: $C_{ij} = 0., \quad i \neq j, \quad C_{ii} = \frac{1.}{|C_i|}$, where C_i is the median dual cell around the vertex q^i of our triangulation.

4. The discrete pressure P_h^{n+1} is estimated from φ_h using the relations given in the presentation of Algorithm 1 and 2, $P_h^{n+1} = \varphi_h / T$ and $P_h^{n+1} = P_h^n + 2\varphi_h / T^2$, respectively. Here, it is assumed that pressure $P_h \in Q_{0h}$. In the case of Algorithm 1, it can be shown [23] that the projection method acts in a stabilizing manner on the continuity equation, allowing the equal odrer interpolation for V_h and P_h. However, for the nonuniform meshes where element size varies significantly spurious oscillations are possible (the stabilization term has as a coefficient T instead of Δx_K^2. In the spirit of stabilized finite element methods for Navier-Stokes equations ([16], [22], [23]), the stabilization of pressure is proposed through the correction of the pressure p_h^{n+1} computations from φ_h, what is equivalent to the stabilization of the variational formulation of the continuity equation.

5. The previous remark implies that the introduction of P1/P1isoP2 element, instead of P1/P1, would assure less mesh sensitivite pressure computations.

3.4 $k - \varepsilon$ computations

Let $\{\psi_i\}_{1..N}$ be the basis of W_{0h}. Similarly like in the case of the equations for the intermediate velocity, the stabilization (via PSI fluctuation splitting scheme) is applied again only to the convection term (in this case, this violates the consistency more severely). We use the same approach for numerical integration, together with the mass lumping for source terms (Gaussian points coinciding with element nodes). The stabilization of convection term for k_h^{n+1} is defined as $g_2^K = \beta^K(V_h^n, k_n^h, w_h) - w_h$ and for ε_h^{n+1} as $g_2^K = \beta^K(V_h^n, \varepsilon_n^h, w_h) - w_h$.

This results in the following scheme for k_h^{n+1} and ε_h^{n+1} for all nodes q^i not on $\Gamma_{1h} \cup \Gamma_{3h}$ (where we use the Dirichlet boundary conditions and wall-laws) computed by:

$$k_i^{n+1} = k_i^n - \frac{T_i}{|C_i|} \sum_{K, K \in \Omega_h} (\beta_i^K \sum_{l=1}^{d+1} (V_m^n \cdot \nabla \psi_l) k_l^n |K| +$$

$$c_\mu \frac{(k^2)_m^n}{\varepsilon_m^n} \nabla k^n \cdot \nabla \psi_i{}^K |K| + \frac{|K|}{d+1} (S_k)_i^n), \tag{95}$$

$$\varepsilon_i^{n+1} = \varepsilon_i^n - \frac{T_i}{|C_i|} \sum_{K, K \in \Omega_h} (\beta_i^K \sum_{l=1}^{d+1} (V_m^n \cdot \nabla \psi_l) \varepsilon_l^n |K| + c_\varepsilon \frac{(k^2)_m^n}{\varepsilon_m^n} \nabla \varepsilon^n \cdot \nabla \psi_i{}^K |K| + \frac{|K|}{d+1} (S_\varepsilon)_i^n), \quad (96)$$

where the subscript m denotes the mean element values, and the rest is the same as in (81), with the exception of the definition of β_i^K in the discrete formulation for k_i^{n+1}:

$$\beta_i^K = \frac{max(0, b_i) min(0, (k_i - k_{in}^K)(k_{out}^K - k_{in}^K))}{\sum_{l=1}^{d+1} max(0, b_l) min(0, (k_l - k_{in}^K)(k_{out}^K - k_{in}^K))}, \quad (97)$$

with

$$k_{in}^K = \frac{\sum_{l=1}^{d+1} k_l min(0, b_l)}{\sum_{l=1}^{d+1} min(0, b_l)}, \qquad k_{out}^K = \frac{\sum_{l=1}^{d+1} k_l max(0, b_l)}{\sum_{l=1}^{d+1} max(0, b_l)}, \quad (98)$$

and ε_i^{n+1}:

$$\beta_i^K = \frac{max(0, b_i) min(0, (\varepsilon_i - \varepsilon_{in}^K)(\varepsilon_{out}^K - \varepsilon_{in}^K))}{\sum_{l=1}^{d+1} max(0, b_l) min(0, (\varepsilon_l - \varepsilon_{in}^K)(\varepsilon_{out}^K - \varepsilon_{in}^K))}, \quad (99)$$

with

$$\varepsilon_{in}^K = \frac{\sum_{l=1}^{d+1} \varepsilon_l min(0, b_l)}{\sum_{l=1}^{d+1} min(0, b_l)}, \qquad \varepsilon_{out}^K = \frac{\sum_{l=1}^{d+1} \varepsilon_l max(0, b_l)}{\sum_{l=1}^{d+1} max(0, b_l)}, \quad (100)$$

where $b_i = \frac{1}{d} V_m^n \cdot n_i$ and n_i is the inward unit normal to the triangle edge (tetahedra face) opposite to the node i, multiplied with the length of corresponding triangle edge (the area of tetrahedra face).

3.5 The wall-laws implementation

3.5.1 Explicit implementation

In the finite element formulation of Reynolds averaged Navier-Stokes equations, the following boundary integral appears:

$$\int_{\Gamma_{3h}} S_{t\,h}^n \cdot n_h \cdot \phi_h, \quad (101)$$

where $S_{t\,h}^n = (\nu + \nu_t)(\nabla V_h^n + (\nabla V_h^n)^T)$. $S_{t\,h}^n \cdot n_h$ is decomposed over (n, s^*)

$$S_{t\,h}^n \cdot n_h = (S_{t\,h}^n \cdot n_h \cdot n_h) n_h + (S_{t\,h}^n \cdot n_h \cdot s_h^*) s_h^*. \quad (102)$$

The first term on the right hand side of the above equation is in the presented implementation neglected, although it can be calculated explicitely (see [28], [29]).

Introduction of the wall-laws in the formulation leads to

$$\int_{\Gamma_{3h}} (S_t)_h^n \cdot n_h \cdot \phi_h = \int_{\Gamma_{3h}} ((S_t)_h^n \cdot n_h \cdot s_h^*) s_h^* \cdot \phi_h = - \int_{\Gamma_{3h}} (u_\tau)^2 s_h^* \cdot \phi_h, \quad (103)$$

and, finally, to the addition of the following term in the numerical scheme (82) for the nodes $q^i \in \Gamma_{3h}$

$$+ \sum_{K,K\in\Omega_h} \sum_{E\in\mathcal{E}(K),q^i\in E} \frac{|E\cap\Gamma_{3h}|}{d}(u_\tau^E)^2 s_{E,j}^*, \tag{104}$$

Both u_τ^E and s_E^* are computed for all triangles sides E belonging to Γ_{3h} in 2D, and for all tetrahedra faces E belonging to Γ_{3h} in 3D. In these computations we take the mean value of V_h^n on each E, V_E^n, and we solve

$$g(u_\tau^E) = V_E^n \cdot s_E^* - u_\tau^E f(u_\tau^E) = 0, \tag{105}$$

where s_E^* is defined as

$$s_E^* = \frac{V_E^n - (V_E^n \cdot n_E)n_E}{|V_E^n - (V_E^n \cdot n_E)n_E|}. \tag{106}$$

Regarding the wall-laws for k_i^{n+1} and ε_i^{n+1}, since u_τ is computed for triangle sides in 2D, and tetrahedra faces in 3D, to compute the nodal values (for the node q^i) we compute the average value of u_τ for all triangle sides on the boundary Γ_{3h} having the node q^i as common in 2D (weighted by triangle side lenghts), and for all tetrahedra faces on the boundary Γ_{3h} having the node q^i as common in 3D (weighted by tetrahedra face areas):

$$k_i^{n+1} = \frac{\sum_{K,K\in\Omega_h}\sum_{E\in\mathcal{E}(K),q^i\in E}|E\cap\Gamma_{3h}|k_E^{n+1}}{\sum_{K,K\in\Omega_h}\sum_{E\in\mathcal{E}(K),q^i\in E}|E\cap\Gamma_{3h}|}, \tag{107}$$

$$\varepsilon_i^{n+1} = \frac{\sum_{K,K\in\Omega_h}\sum_{E\in\mathcal{E}(K),q^i\in E}|E\cap\Gamma_{3h}|\varepsilon_E^{n+1}}{\sum_{K,K\in\Omega_h}\sum_{E\in\mathcal{E}(K),q^i\in E}|E\cap\Gamma_{3h}|}, \tag{108}$$

where $k_E = k_E(u_\tau)$ and $\varepsilon_E = \varepsilon_E(u_\tau)$ are defined by (40).

3.5.2 u_τ problem resolution

The nonlinear equation for u_τ:

$$h(u_\tau) = U \cdot s - u_\tau f(u_\tau) = 0, \tag{109}$$

is resolved applying the Newton method,

$$u_\tau^{l+1} = u_\tau^l - \frac{h^l(u_\tau)}{h^l(u_\tau)'}. \tag{110}$$

The initialization of the algorithm for u_τ was chosen as $u_\tau^0 = \sqrt{\frac{\nu U\cdot s}{\delta}}$.

4 NUMERICAL TESTS IN LAMINAR REGIME

We have carried out a number of test case computations in laminar regime, both in 2D and in 3D.

The test cases analyzed in 2D were: flat plate flow, flow in a cavity, flow over a backward facing step and flow past a circular cylinder.

In 3D, we analyzed the following test cases: flow in a cavity and flow over a backward facing step, where the 2D domain was extended in $z-$direction (we have used rather thin domains) with imposing the symmetry boundary conditions:

$$u \cdot n = 0, \qquad S \cdot n = 0. \tag{111}$$

Basically, these test cases served to verify the 3D code in obtaining the results identical to those from 2D computations.

The first tests were conducted for the basic 2D laminar test cases. Here, the behavior of the code was analyzed for the flows with high Reynolds number values, as a preparation for further turbulent flow computations.

The flat plate flow computations were verified comparing the friction along the plate with the Blasius law and the velocity profile in the boundary layer with the Blasius solution [51]. The Reynolds number chosen was rather high, $Re = 33000$.

The cavity flow, because of multiple separation in the corners when increasing Reynolds number, is rather difficult to compute [58], [59]. Here we have compared the results with the results available in the literature up to $Re = 10000$.

The flow over a backward facing step with the recirculations appearing both on the bottom and upper wall when increasing Reynolds number, presents the further more complex step in laminar high Reynolds flows simulations. Here, the main recirculation lenght is compared with experimental results [56] up to $Re = 800$.

Finally, we computed the steady laminar flow over a circular cylinder ([54], [11]) as a preparation for further unsteady computations.

3D cavity flow is analyzed for $Re = 400$ and compared with adequate 2D simulation.

For 3D flow over a backward facing step, we have chosen $Re = 150$ where, according to experiments, the flow is supposed to stay two-dimesnional. In that way, we could compare it with our previously conducted two-dimensional simulation.

4.1 2D flat plate

The flat plate test case represents a very simple test case, and there are well documented experimental results, as well as analytic solutions for the friction coefficient distribution and velocity profiles, developed by Blasius (see [51]).

The problem configuration is shown in Figure 3, corresponding to a flow in a rectangular domain $\Omega = [-0.2, 1.] \times [0, 0.1]$, and the boundary conditions are defined as in the figure, where we impose an uniform inlet profile $u_p = (1., 0.)$, on the plate we set $u = 0$, and the symmetry boundary condition is standard: $u \cdot n = 0$ and $S \cdot n = 0$, as well as it is the outflow boundary conditions $F_n = 0$ and $F_s = 0$. Reynolds number defining the flow is $Re = \frac{u_\infty L}{\nu} = 33000$.

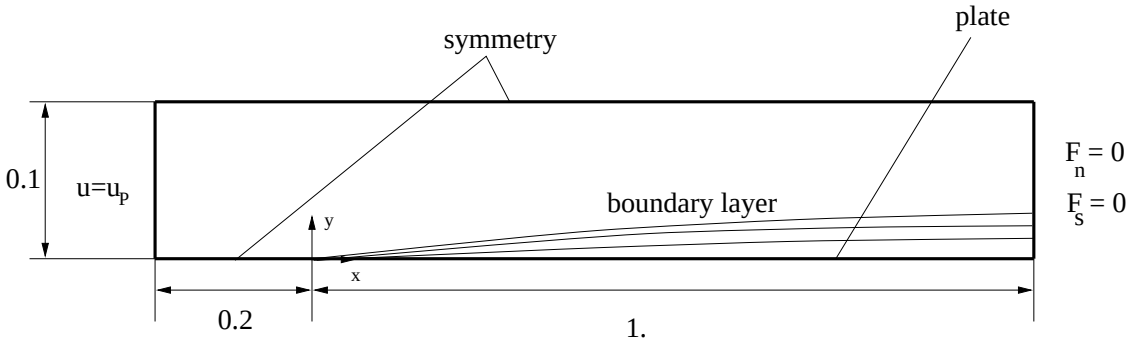


Figure 3: Problem description.

We have tested several uniform and nonuniform meshes (see Figure 4). Typical flow pattern (isovalues of $|u|$ and p) are shown in Figures 5 and 6.

The results presented are the comparison of the computed values of the friction coefficient $c_f = \frac{\tau_w}{\frac{1}{2}\rho_\infty |u_\infty|^2} = \frac{S \cdot n \cdot s}{\frac{1}{2}\rho_\infty |u_\infty|^2}$ on the plate with the values given with the Blasius law (the analytical solution): $c_f = \frac{0.664}{\sqrt{\frac{\nu}{u_\infty x}}}$ for different meshes.

Horizontal velocity profiles in the cross-sections defined with $x = 0.2$ and $x = 0.9$ computed on the mesh denoted by MESH3 (see Figure 4) are also compared with the Blasius solution values.

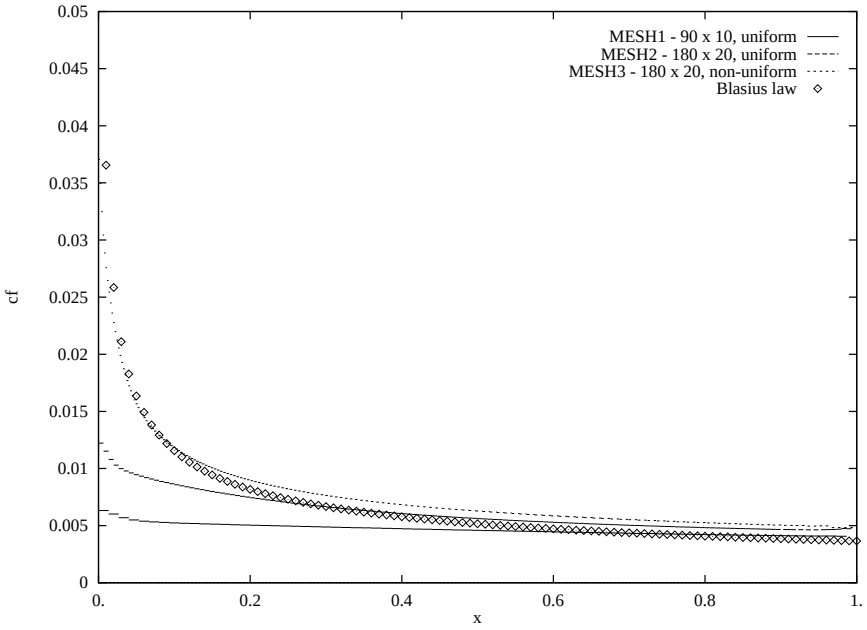


Figure 4: Friction coefficient c_f along the plate computed with different meshes.

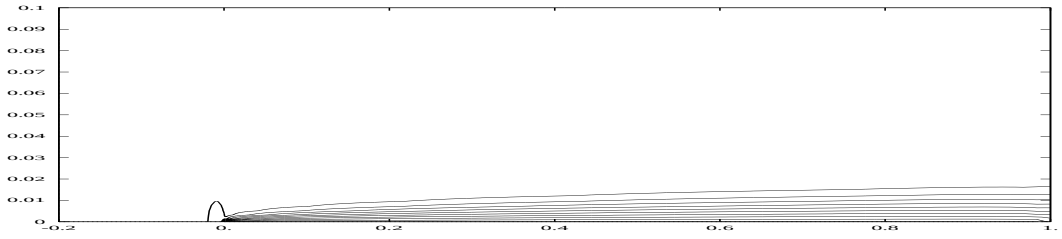


Figure 5: Isolines of $|u|$, MESH3.

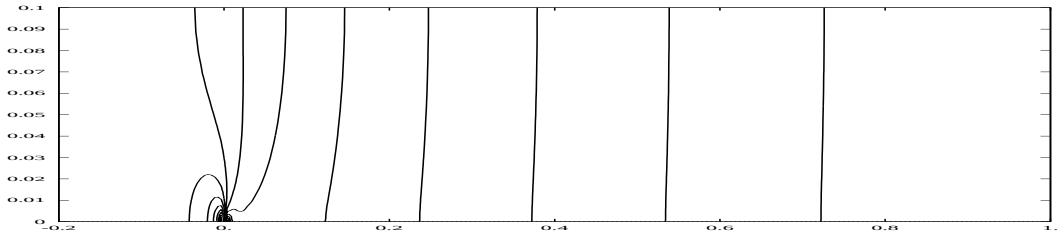


Figure 6: Isolines of pressure p, MESH3.

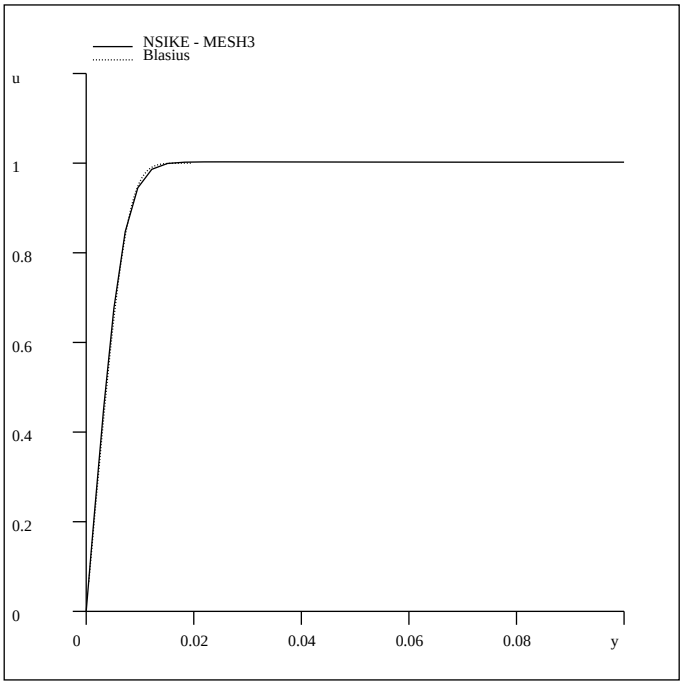


Figure 7: Horizontal velocity profile at $x = 0.2$ computed with the finest mesh MESH3, comparison with Blasius solution.

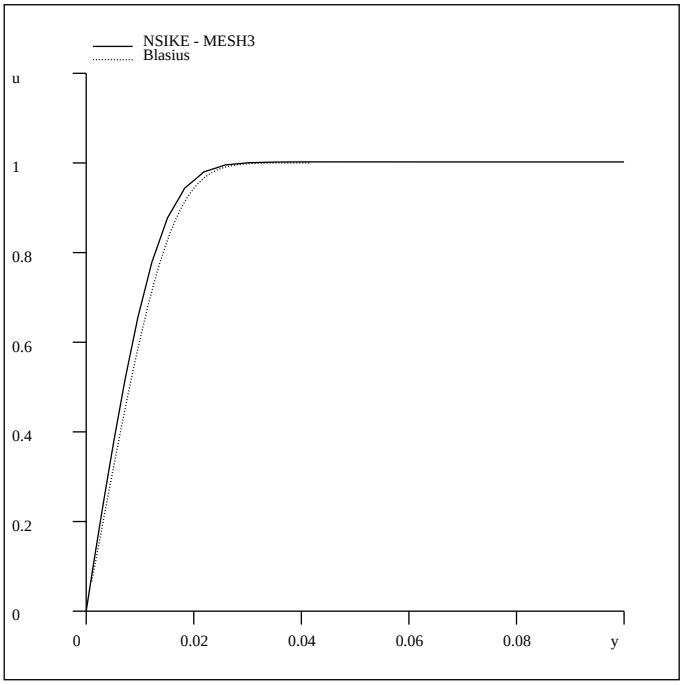


Figure 8: Horizontal velocity profile at $x = 0.9$ computed with the finest mesh MESH3, comparison with Blasius solution.

4.2 2D cavity

The cavity flow represents a classical Navier-Stokes solver benchmark test, and there exists a variety of numerical results for different flow regimes. The problem configuration is shown in Figure 9, corresponding to a flow in a square domain $\Omega = [0,1] \times [0,1]$ with the boundary conditions defined as in the figure.

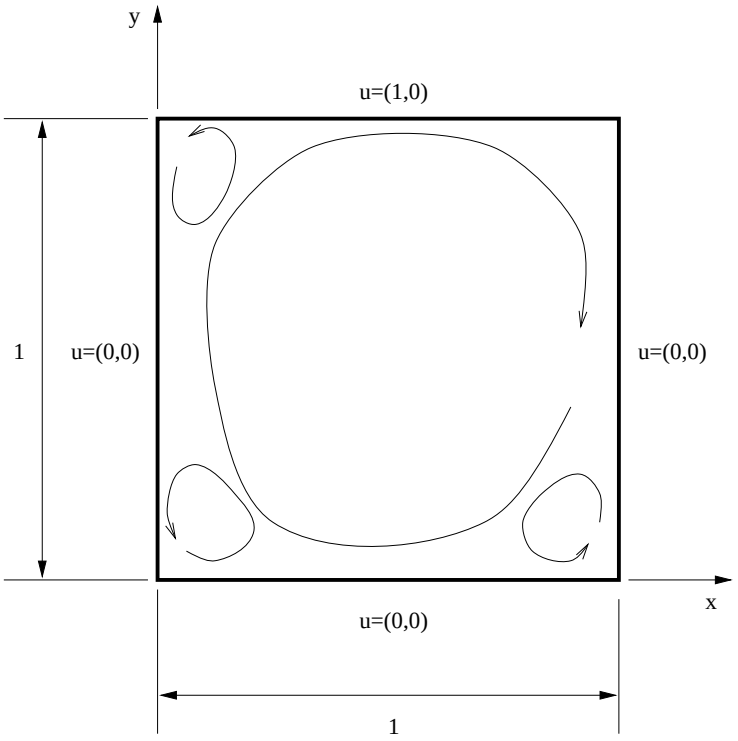


Figure 9: Problem description.

The computations are carried out for the flow regimes corresponding to the Reynolds number values $Re = 1, 100, 400, 1000, 5000, 10000$.

We have used three non-uniform meshes, MESH0 : 441 nodes, 800 elements, MESH1 : 2601 nodes, 5000 elements, MESH2 : 10201 nodes, 20000 elements. All the results presented are obtained using MESH1, except the results presented for Reynolds number $Re = 10000$, where we have used finer mesh MESH2.

The results of present computations are compared with the results obtained by Ghia et al [58] (obtained using the meshes with 129×129 and 257×257 nodes). In particular we are interested in the position of the center of primary vortex in the cavity.

Furthermore, we present the streamlines for different Reynolds number.

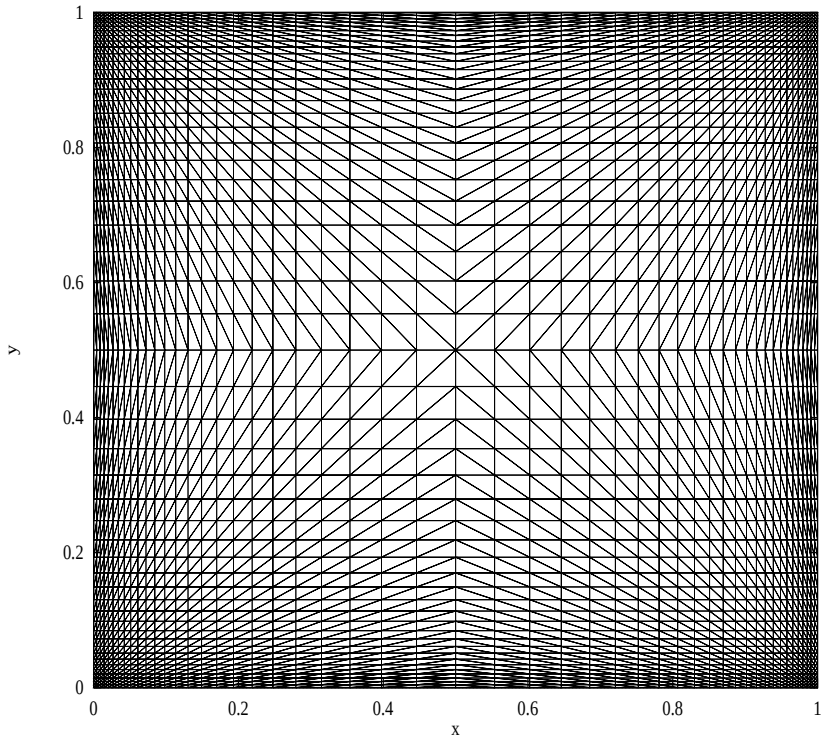
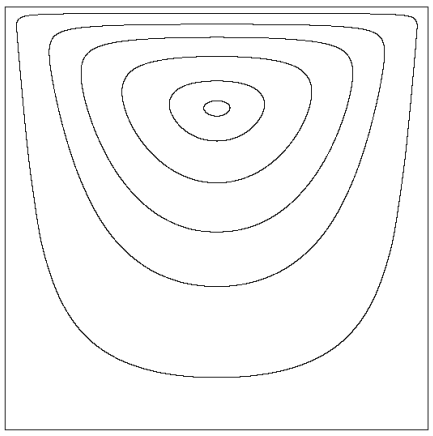


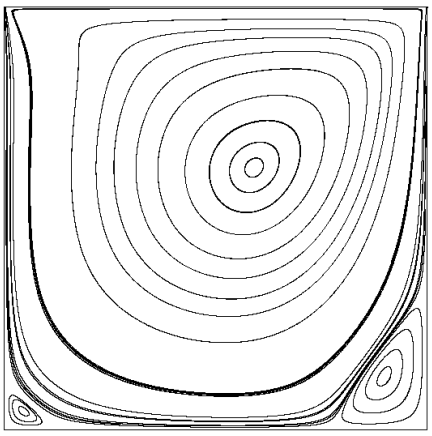
Figure 10: Computational mesh, MESH1, 2601 nodes, 5000 elements.

Re	100	400	1000	5000	10000
Ghia et al. [58]	$x = 0.6172$ $y = 0.7344$	$x = 0.5547$ $y = 0.6055$	$x = 0.5313$ $y = 0.562$	$x = 0.5117$ $y = 0.5352$	$x = 0.5117$ $y = 0.5333$
NSIKE, MESH1	$x = 0.61$ $y = 0.75$	$x = 0.58$ $y = 0.615$	$x = 0.545$ $y = 0.56$	$x = 0.53$ $y = 0.53$	$x = 0.525$ $y = 0.53$

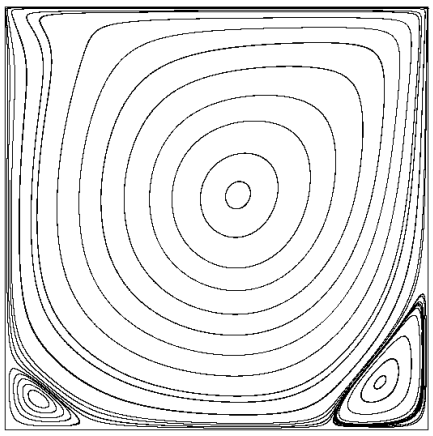
Table 1. Primary vortex center position.



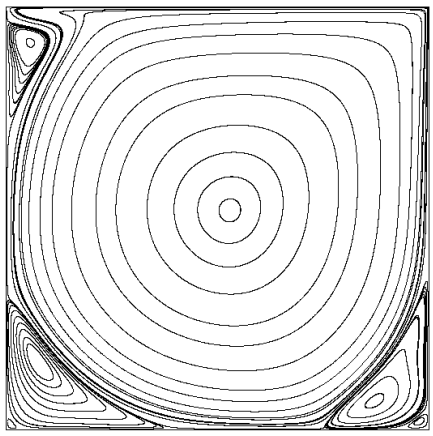
(a) Re = 1



(b) Re = 400



(c) Re = 1000



(d) Re = 10000

Figure 11: Streamlines for different Reynolds numbers.

4.3 2D flow over a step

The flow over a step has been analyzed by a variety of authors [57], and for this configuration (see Figure 12) there exists also a detailed experimental study [56].

At the entrance boundary ($u = u_P$), for a horizontal velocity, u_1, a profile is prescribed (with the mean value $\bar{u}_1 = 2/3$, and the vertical velocity u_2 is set to zero. Reynolds number for this test case is defined as $Re = \frac{\bar{u}_1(h_1+h_2)}{\nu}$.

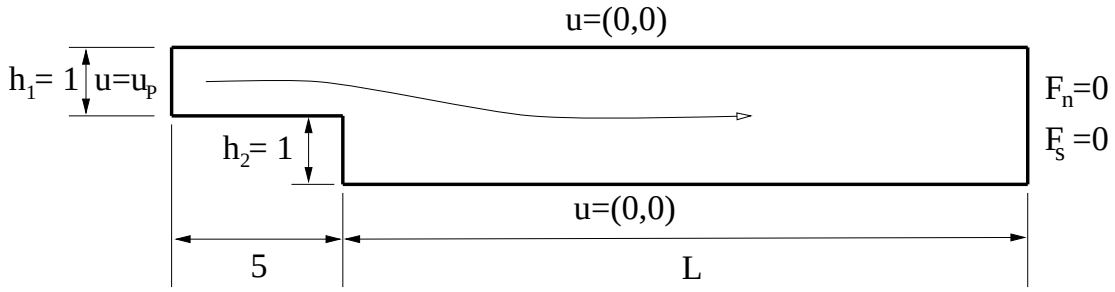


Figure 12: Problem description.

We analyzed the flow over a backward facing step for different values of Reynolds number Re: 100,150,200,400,500,600,800.

In our computations we have tested several meshes, ranging from MESH1, uniform mesh, with 1291 nodes and 2400 elements (used for Reynolds number less than 200), to MESH4, nonuniform mesh, refined in the direction normal to the solid wall, with 5616 nodes and 10900 elements (which was basically employed for the computations for high Reynolds number).

In Figures 13 and 14 we present the streamlines and isolines of pressure for $Re = 800$. We summarize the computed length of the primary recirculation vs. Reynolds number and compare our results with experimental results [56] and the results from some other authors [57] in Figure 15. Generally, the computations have a tendency to underestimate the length of the recirculations (both primary and secondary on the upper wall), and that was already remarked by several other authors (the cause are probably the threedimensional effects).

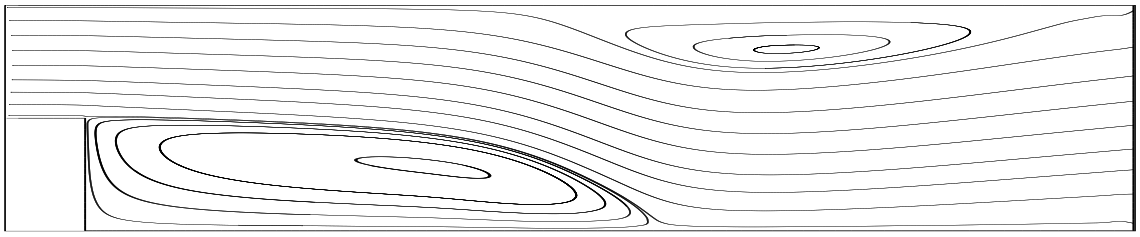


Figure 13: Streamlines, $Re = 800$.

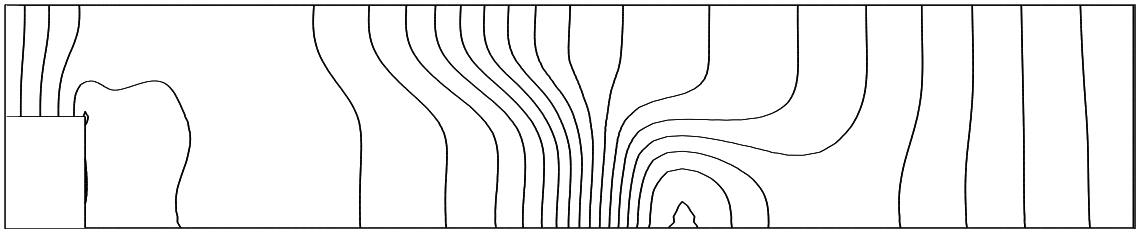


Figure 14: Isolines of pressure p.

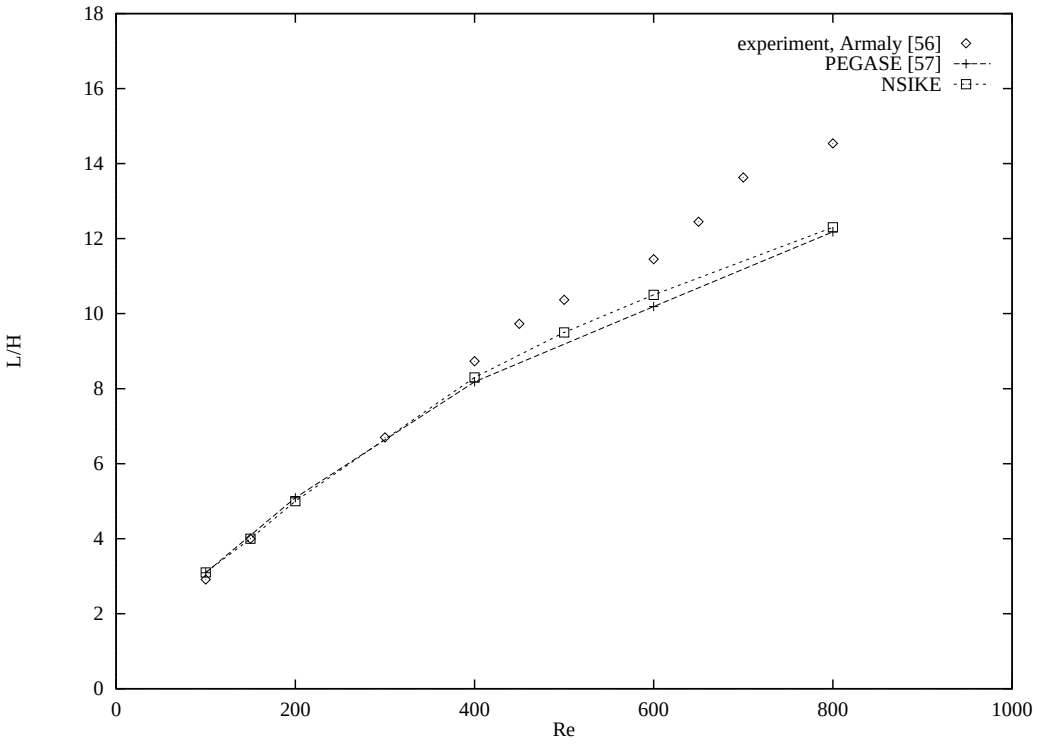


Figure 15: Primary recirculation length L for different values of Re.

4.4 2D flow over a circular cylinder

Due to a variety of circular cylinder domain definitions, in the numerical experiments carried out with the aim of computing vortex shedding past a cilinder, Rannacher proposed the standardization of the flow over a circular cylinder test case (the nonstationary flow was often achieved by defining the asymmetric domain, where this asymmetry was different from one computation to another).

In Figure 16 the domain used in the computation of Rannacher's test cases 2D-1 (steady) and 2D-2 (unsteady) is presented. The profiles used at the entrance boundary is

$$u_1(y) = 4u_m y(H-y)/H^2, \qquad u_2 = 0, \tag{112}$$

where $H = 0.41m$, and u_m is set for the test case 2D-1: $u_m = 0.3m/s$, and for the test case 2D-2 $u_m = 1.5m/s$.

The Reynolds number is defined as $Re = \frac{\bar{u}_1 d}{\nu}$, where $d = 0.1m$ is the cylinder diameter , and $\bar{u}_1$ is the mean entrance horizontal velocity $\bar{u}_1 = \frac{2}{3}u_1(H/2)$.

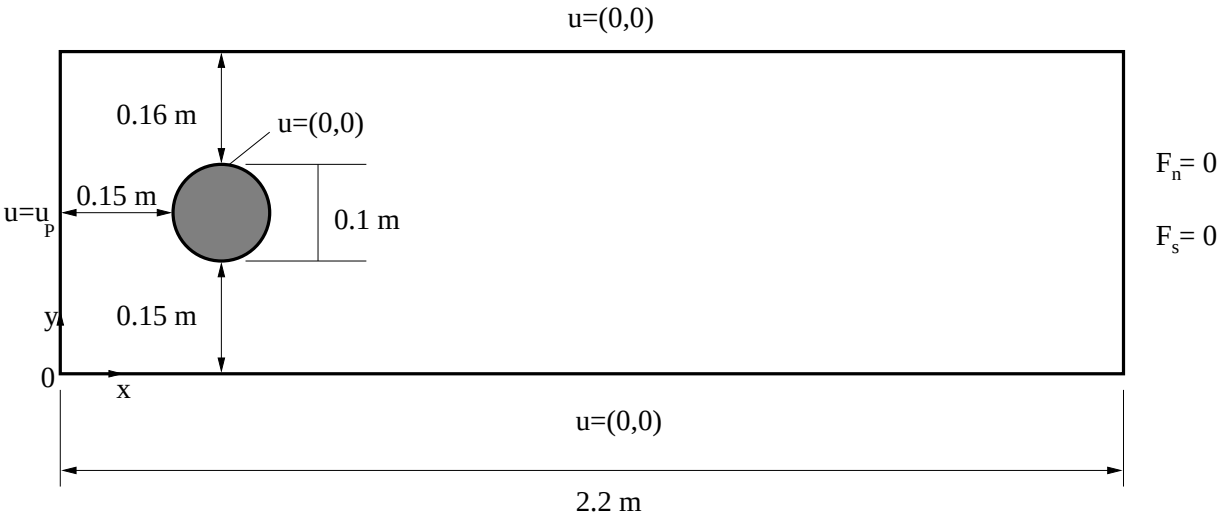


Figure 16: Problem description.

Here, we present the results for the steady test case (2D-1) computed on the computational mesh with 4190 nodes and 8110 elements, which are summarized in Table 2.

In the computation for unsteady test case (2D-2) we were obliged to use much finer mesh with 10643 nodes and 21842 elements and summary results are presented in Table 3.

	C_D	C_L	Δp	L_r
benchmark [54]	5.58	0.011	0.085	0.1175
NSIKE	5.65	0.012	0.082	0.121

Table 2. Summary of the results - test case 2D-1.

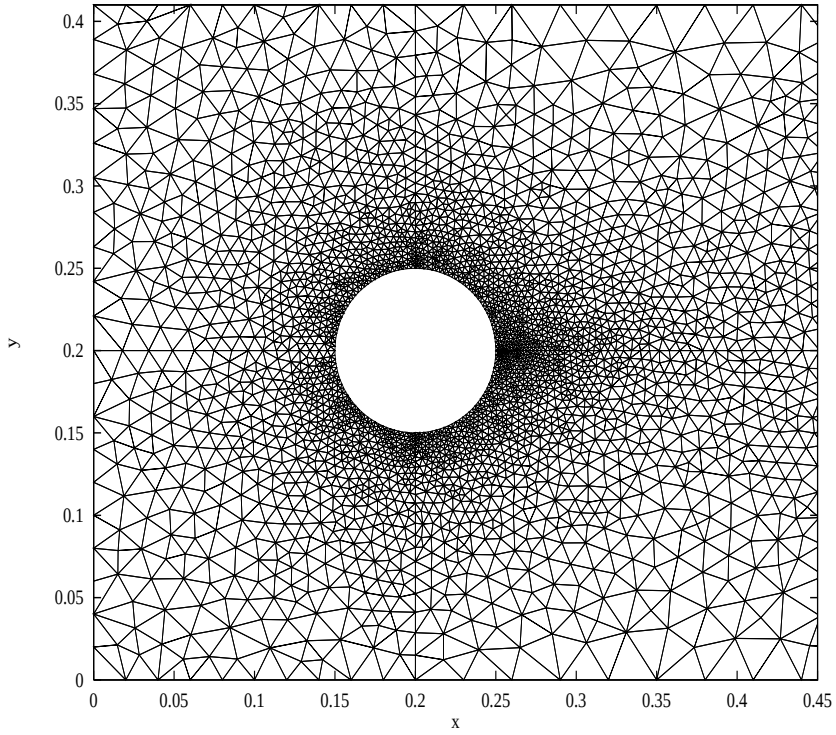


Figure 17: Computational mesh, MESH1, 4190 nodes, 8110 elements (partial view).

	C_{Dmax}	C_{Lmax}	St	Δp
benchmark [54]	3.23	1.00	0.30	2.48
NSIKE	3.2	0.92	0.034	2.4

Table 3. Summary of the results - test case 2D-2.

4.5 3D cavity

The cavity flow was the first 3D flow analyzed in order to verify the 3D code. Although, the 3D domain is obtained from the 2D domain extending the domain in the z-direction for the width $b = 0.1$, and imposing the slip boundary conditions: $u \cdot n = 0$ and $S \cdot n = 0$ in the planes $z = 0$ and $z = b$.

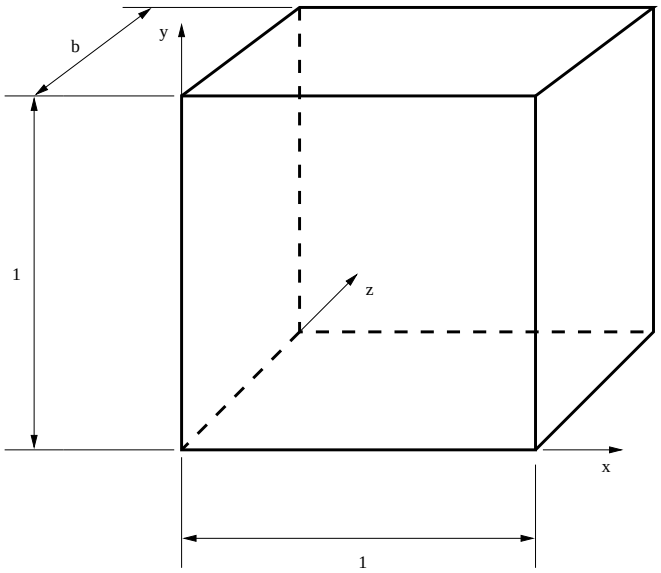


Figure 18: 3D domain.

We have chosen to do the computations for $Re = 400$ where our 2D computations implied rather stable flow which bares some complex features (the existance of several recirculations in both lower corners).

The mesh employed in the simulation consisted in 13005 nodes, 60000 elements and 11600 boundary faces and is presented in Figure 19.

As expected the computations resulted in the flow that is twodimensional (coincides in each plane $z = const$). Qualitatively, the streamlines (in plane $z = 0.$) shown in Figure 20 correspond very well to those from 2D computations (see Figure 11). Besides, the positions of the centers of three principal recirculations from two computations (2D and 3D) coincide in $(0.58, 0.615)$ for the primary recirculation, $(0.905, 0.126)$ and $(0.044, 0.044)$ for the secondary recirculations in bottom right and left corner.

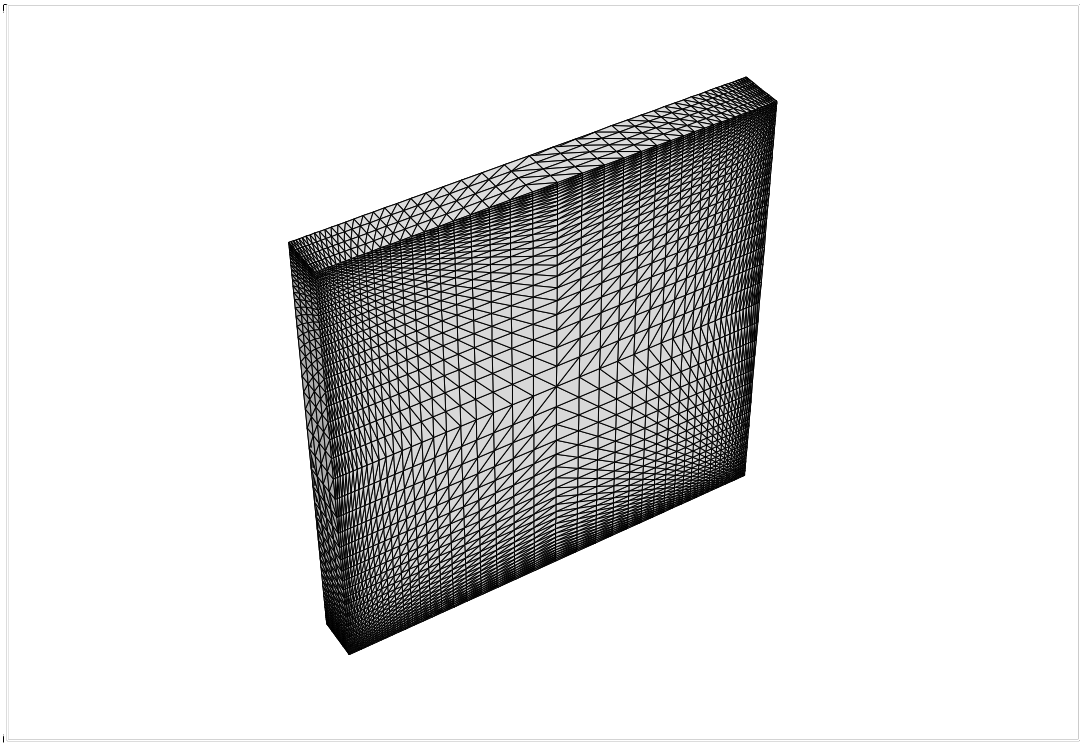


Figure 19: Computational mesh, 13005 nodes, 60000 elements.

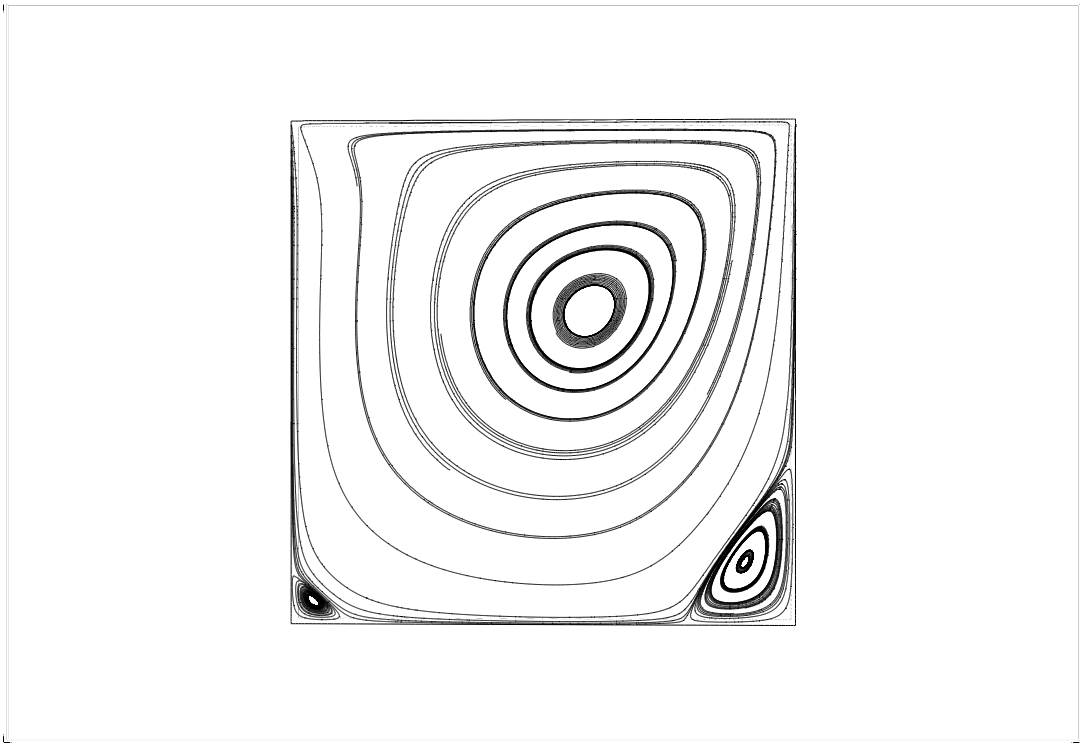


Figure 20: Streamlines in the plane $z = 0.$, $Re = 400.$

4.6 3D flow over a step

Three-dimensional flow over a step was computed to verify the 3D code (like in the case of cavity flow, also). Like the 3D cavity test case, it is not really a 3D problem, because the domain is obtained by extending the 2D domain into z-direction for a width $b = 0.5h_2$, further on defining the boundary conditions in the planes $z = 0$ and $z = b$ as slip boundary conditions (or symmetry boundary conditions): $u \cdot n = 0$ and $S \cdot n = 0$.

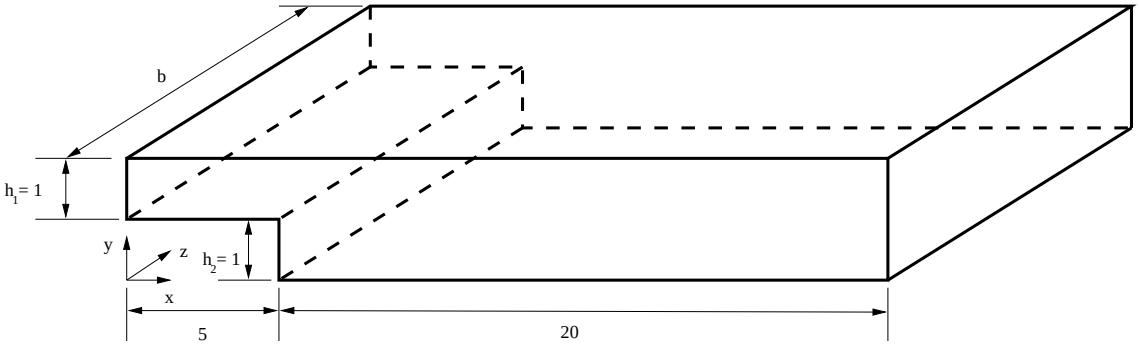


Figure 21: 3D domain.

The Reynolds number chosen for the verification of code is chosen to have a regime where the flow is two-dimensional (according to experiments).

We have used the same basic two-dimensional mesh, as used in 2D computations, that gave our 3D computational mesh with 6455 nodes, 28800 elements and 6240 boundary faces, presented in Figure 22.

The comparison of 3D computations with the 2D computations for the Reynolds number $Re = 150$ are presented further on. Basically, we were interested in the main recirculation as an integral parameter that is characterizing the flow.

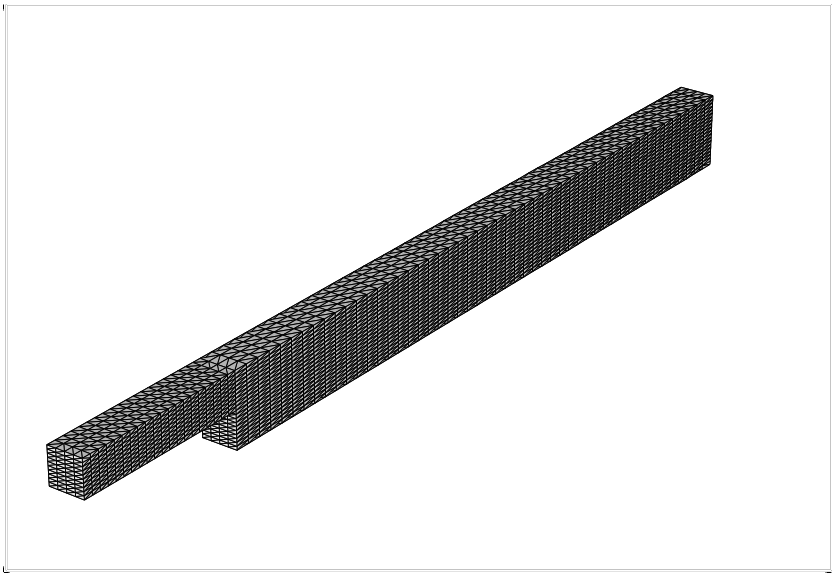


Figure 22: Computational mesh, 6455 nodes, 28800 elements.

As in the case of 3D cavity flow we obtain perfectly two-dimensional flow, where the recirculation length corresponds to the one computed in two dimensions.

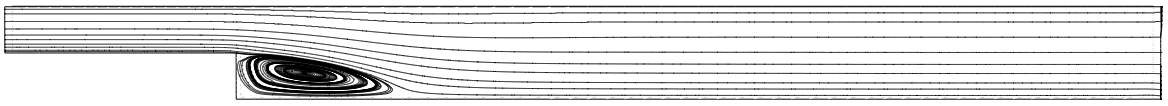


Figure 23: Streamlines in the plane $z = 0.$, $Re = 150$.

The coordinates of the computed reattachement point (or line) are (3.97,0) that coincide almost exactly with the ones from 2D computations (3.99,0).

5 NUMERICAL TESTS IN TURBULENT REGIME

In turbulent regime, we have carried out several test case computations. In 2D, we analyzed: flat plate flow and flow over a backward facing step. In 3D, we analyzed the flow over a backward facing step. As in the case of laminar 3D test cases, the 2D domain was extended in $z-$direction with imposing the symmetry boundary conditions:

$$u \cdot n = 0, \qquad S \cdot n = 0. \tag{113}$$

The results were compared with available experimental results and, in some cases, with the results obtained with the compressible code NSC2KE [28].

For the flat plate flow the friction coefficient along the plate was verified against the experimental results, as well as different variables profiles in the boundary layer. The computations were carried out for $Re = 2.86 \cdot 10^6$.

The standard test case ([27], [30]), flow over a backward facing step at $Re = 44000$ is analyzed further on, comparing the obtained recirculation length with the results available in literature, as well as other variable fields.

Finally, we computed the 3D flow over a backward facing step at $Re = 44000$. The resulting flow is two-dimensional, and we compared the obtained results with the ones from 2D computations.

5.1 2D flat plate

The configuration of flat plate test case is exactly the same as for 2D flat plate in laminar regime shown in Figure 3, except that on the plate we impose the wall-laws instead of Dirichlet boundary conditions. We have chosen small inflow values for k and ε meaning that there is no important level of turbulence at the entrance, while at the exit (and at the symmetry boundary) we impose homogeneous Neumann boundary conditions for k and ε. We computed the flow for $Re = \frac{u_\infty L}{v} = 2.86 \cdot 10^6$.

In our computations we have used several meshes: MESH1 (3801 nodes, 7200 elements, normal distance of the first layer of nodes from the boundary where we impose wall-laws equal to $10^{-3}\, L$), MESH2 (3801 nodes, 7200 elements, normal distance $5\, 10^{-4}\, L$) and MESH3 (4465 nodes, 8633 elements, normal distance $10^{-4}\, L$). We applied different values of parameter δ in wall-laws varying from $10^{-3}\, L$ to $10^{-4}\, L$, resulting in $y^+ = \frac{u_\tau \delta}{\nu}$ values around 100 and 10, respectively.

The flow pattern (isolines of $|u|$, p and k) computed on MESH3 is shown in Figures 25, 26 and 27.

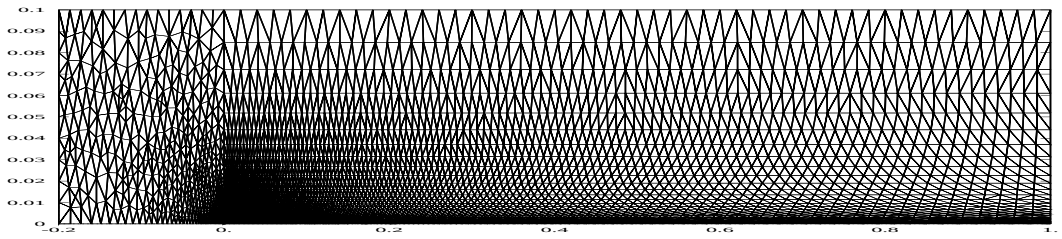


Figure 24: Computational mesh MESH3, 4465 nodes, 8633 elements.

Friction coefficient c_f distribution along the plate computed with different meshes is shown in Figure 28. Further on, we present the nondimensionalized profiles of horizontal velocity U, kinetic energy of turbulence k and turbulent viscosity ν_t for $x = 0.9L$ (the vertical coordinate y in the profile is nondimensionalized using the boundary layer thickness δ_{99}, i.e. the value of y for which the horizontal velocity equals 99% of inflow velocity).

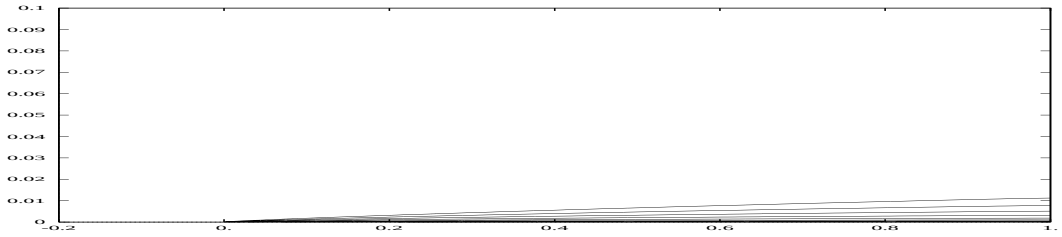


Figure 25: Isolines of $|u|$, MESH3.

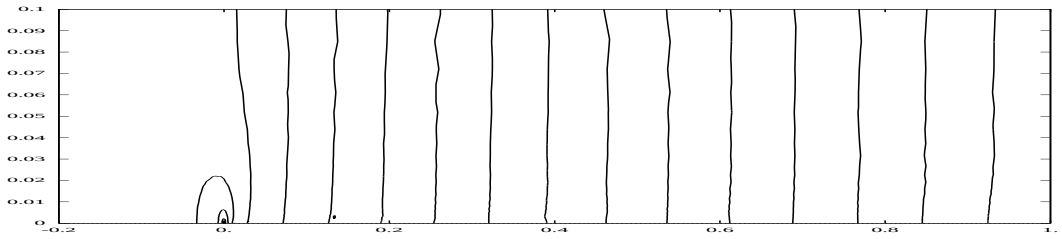


Figure 26: Isolines of p, MESH3.

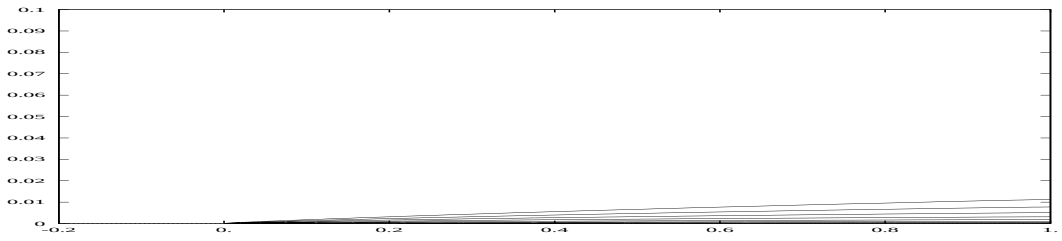


Figure 27: Isolines of k, MESH3.

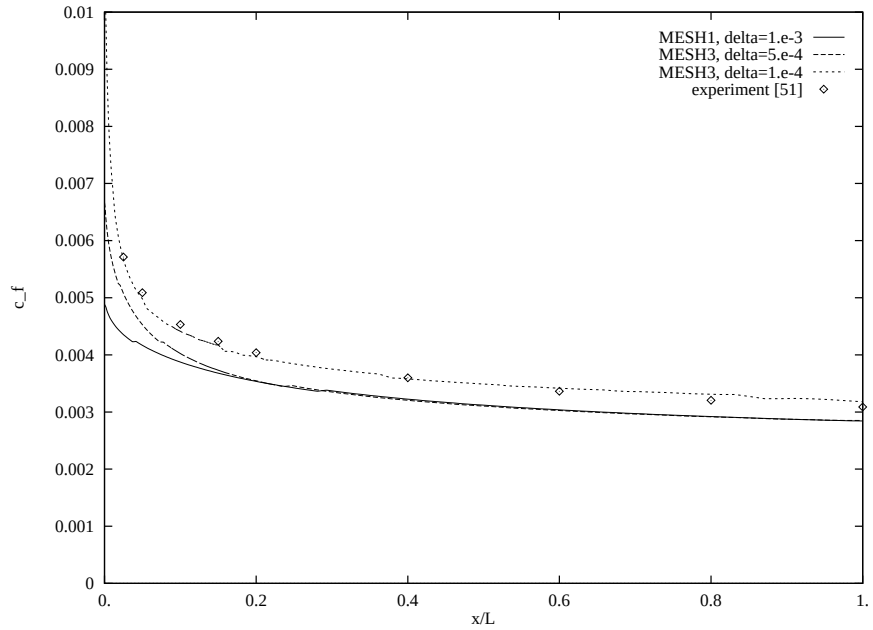


Figure 28: Friction coefficient c_f along the plate computed with different meshes.

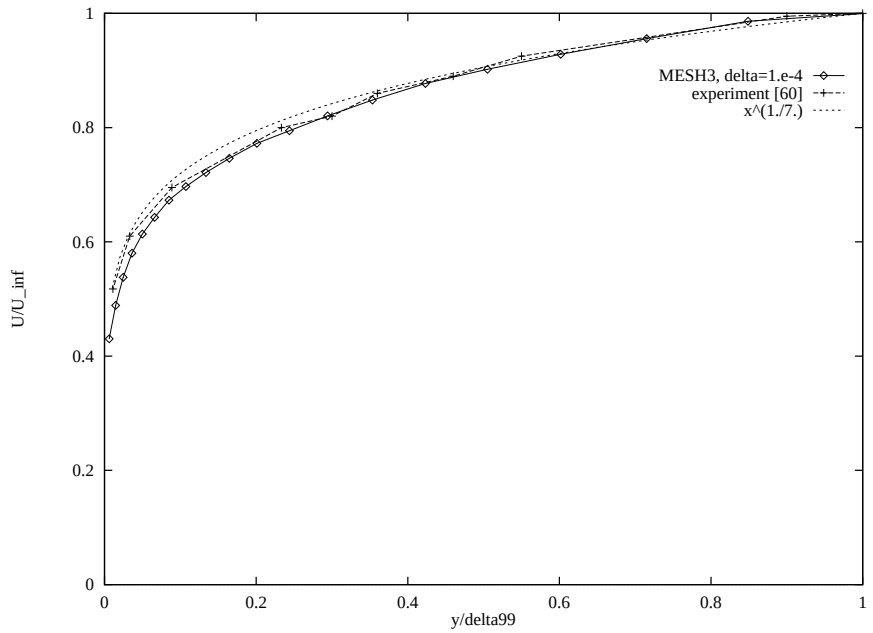


Figure 29: Velocity profile at $x = 0.9L$, MESH3.

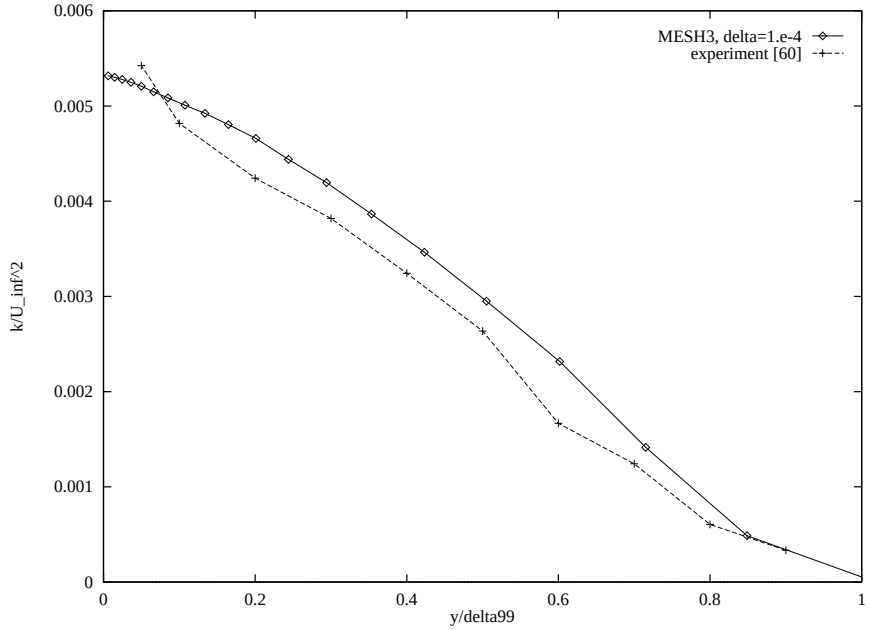


Figure 30: Kinetic energy of turbulence k at $x = 0.9L$, MESH3.

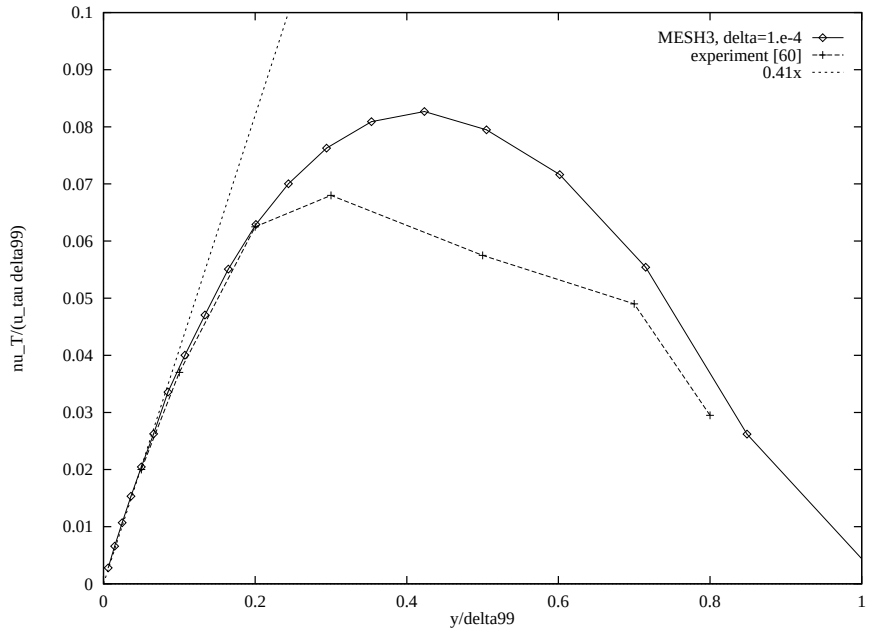


Figure 31: Turbulent viscosity ν_t at $x = 0.9L$, MESH3.

5.2 2D flow over a step

The configuration for the flow over a step is similar to the test case in laminar regime. The important difference is that here on the solid walls we impose the wall-laws instead of Dirichlet boundary conditions. Besides, at the inflow small values for k and ε are chosen meaning that there is no important level of turbulence at the entrance, while at the exit (and at the symmetry boundary) we impose homogeneous Neumann boundary conditions for k and ε. We computed the flow for $Re_H = \frac{u_\infty H}{\nu} = 44000$.

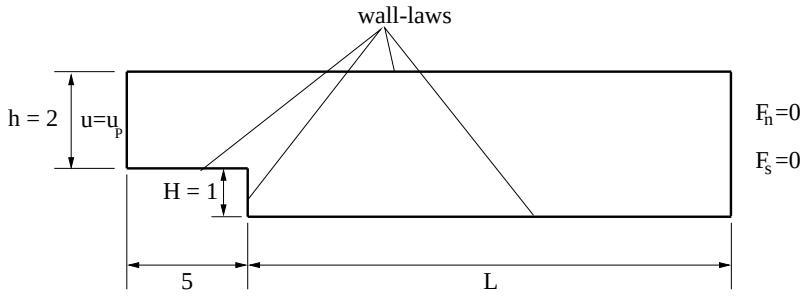


Figure 32: Problem description.

In our computations we have used several meshes, but here we present the results obtained with rather coarse mesh: 2748 nodes and 5192 elements (see Figure 33). The normal distance of the first layer of nodes from the boundary where we impose wall-laws varies; in the region just downstream of the step it is less than $0.01\,H$, and elsewhere we have used coarser mesh. The parameter δ in wall-laws is set to value $0.01\,H$, yielding y^+ values of less than 40.

The computed flow pattern is shown if Figures 34, 35 and 36, while the computed friction coefficient c_f distribution is shown in Figure 37, together with y^+ distribution in Figure 38 (note the difference in c_f and y^+ values on two walls - due to a different boundary layer mesh). In Figure 37, friction coefficient is multiplied with the sign of horizontal velocity to emphasize the position of the recirculation.

The predicted main recirculation length is $6.8\,H$, while we have computed a very weak secondary recirculation in the bottom of the step (even using this very coarse mesh).

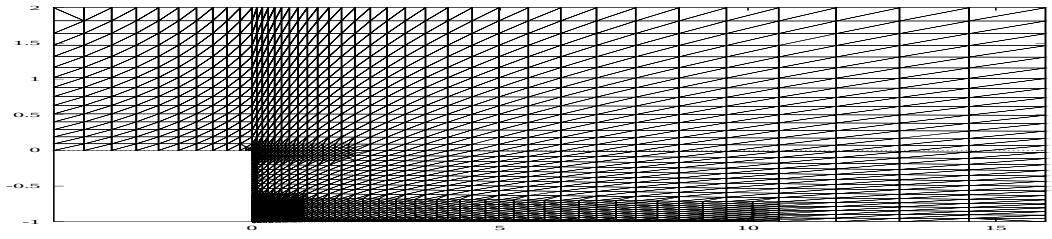


Figure 33: Computational mesh, 2748 nodes, 5192 elements.

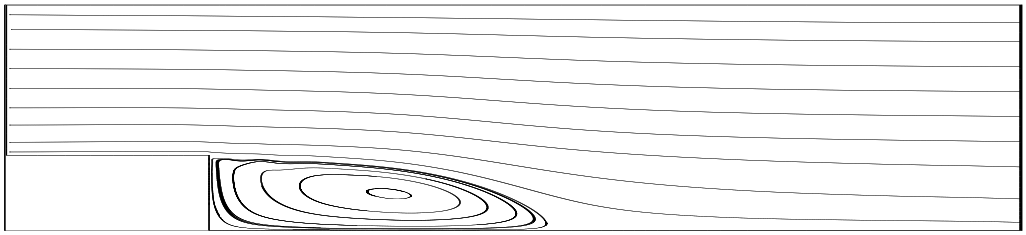


Figure 34: Streamlines, $Re_H = 44000$.

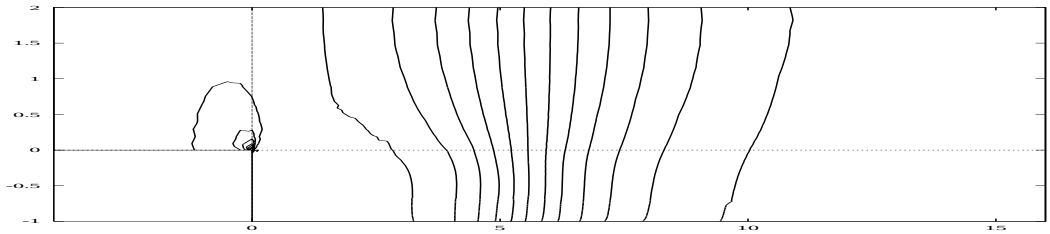


Figure 35: Isolines of pressure, $Re_H = 44000$.

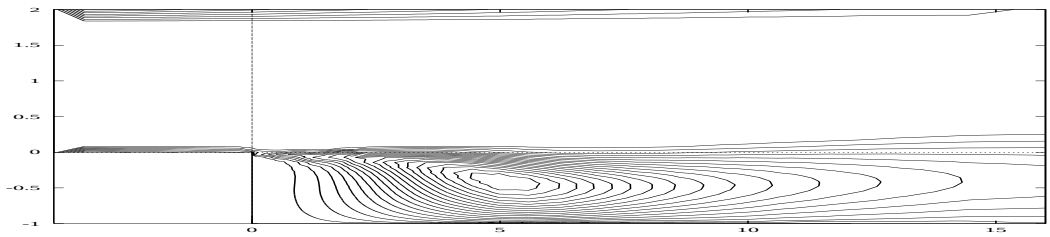


Figure 36: Isolines of k, $Re_H = 44000$.

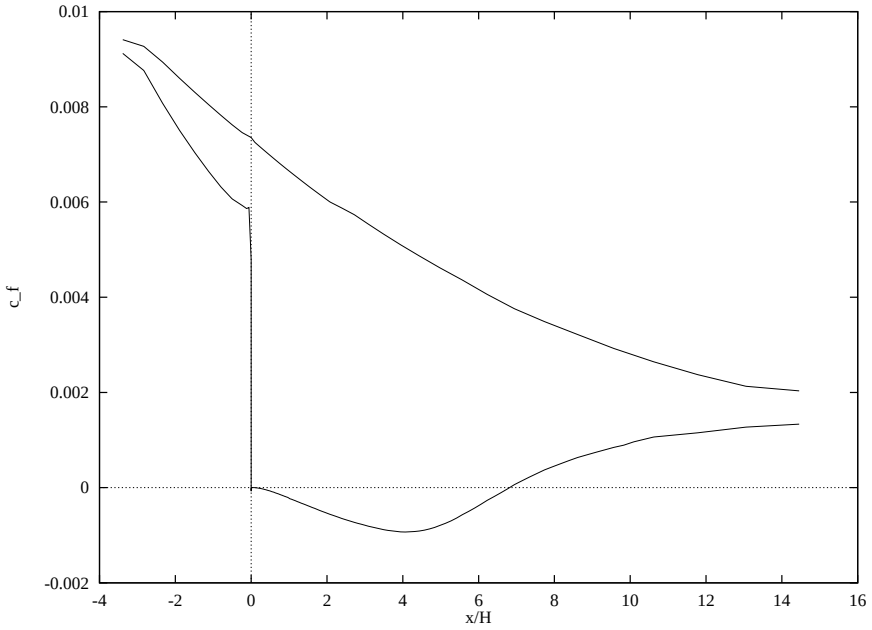


Figure 37: Friction coefficient c_f distribution, $Re_H = 44000$.

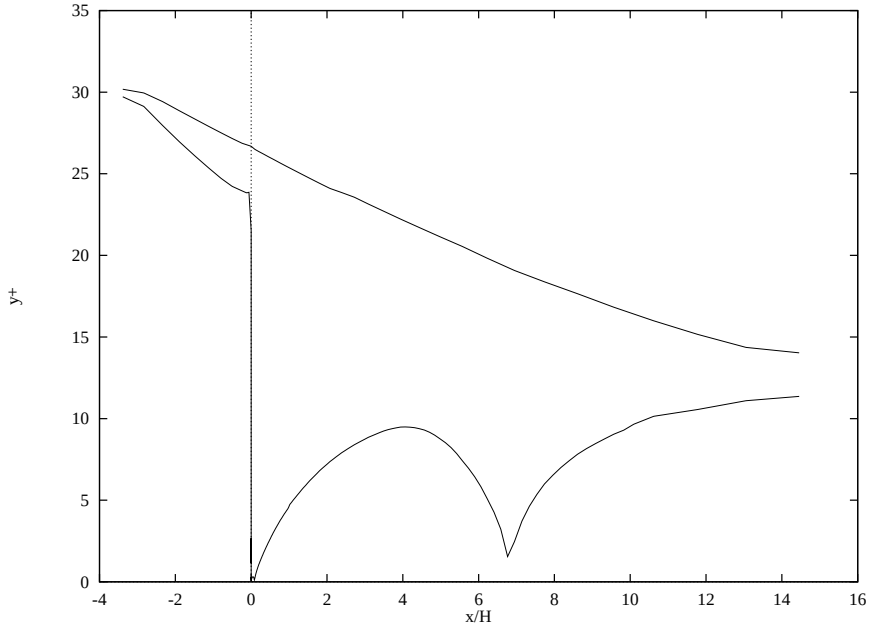


Figure 38: $y^+ = \frac{u_\tau \delta}{\nu}$ distribution, $Re_H = 44000$.

5.3 3D flow over a step

As in laminar regime, the three-dimensional flow over a step was computed to verify the 3D code, now in turbulent regime. Like the other 3D test cases, the domain is obtained by extending the 2D domain into z-direction for a width $b = 0.5H$, further on defining the boundary conditions in the planes $z = 0$ and $z = b$ as slip boundary conditions (or symmetry boundary conditions): $u \cdot n = 0$ and $S \cdot n = 0$.

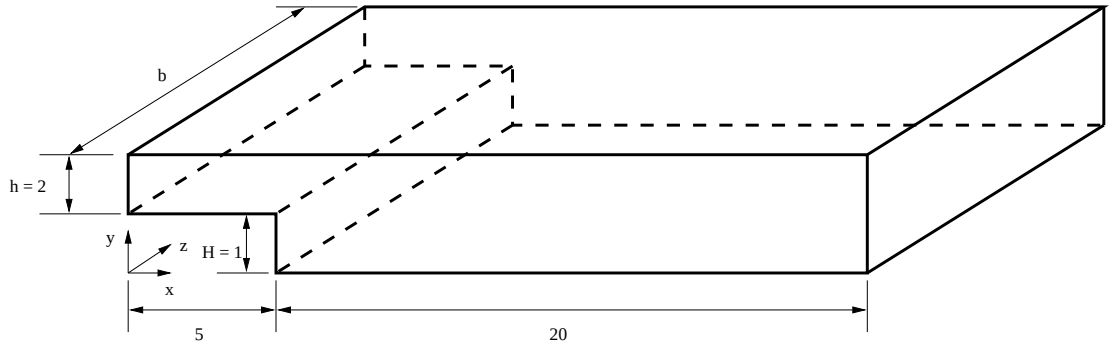


Figure 39: 3D domain.

We have chosen the same values of Reynolds number as in 2D computations ($Re_H = 44000$) as well as the value of the parameter δ in wall-laws ($\delta = 0.01\,H$).

The same basic two-dimensional mesh used in 2D computations gave our 3D computational mesh with 13740 nodes, 62304 elements and 12800 boundary faces, presented in Figure 40.

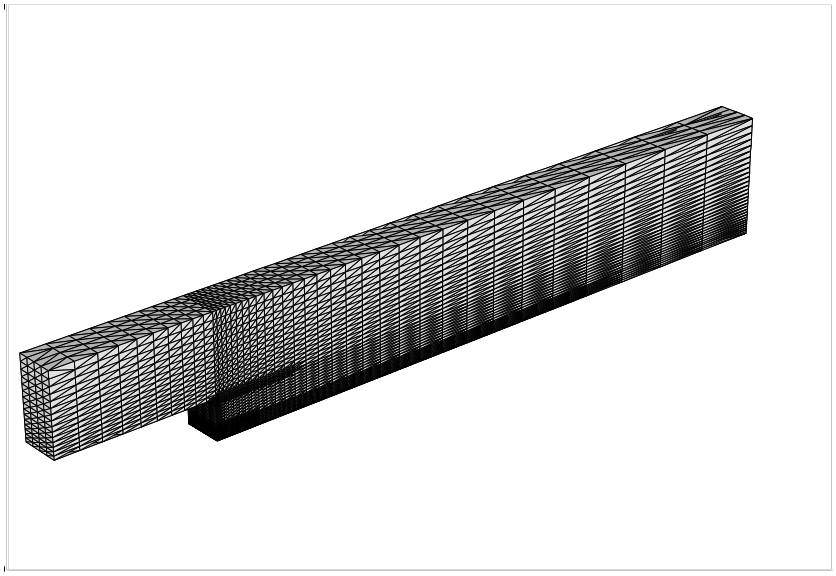


Figure 40: Computational mesh, 13740 nodes, 62304 elements.

As for the other 3D test cases (using this thin domains, with symmetry boundary conditions) the resulting flow is perfectly two-dimensional.

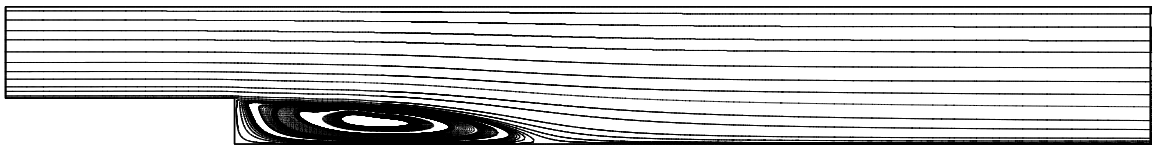


Figure 41: Streamlines in the plane $z = 0.$, $Re = 44000$.

The main recirculation length is $6.8H$, corresponding to the one from 2D computations. The distribution of the friction coefficient values on the lower wall behind the step computed with 2D and 3D computations (in the plane $z = 0.$) is presented in Figure 42.

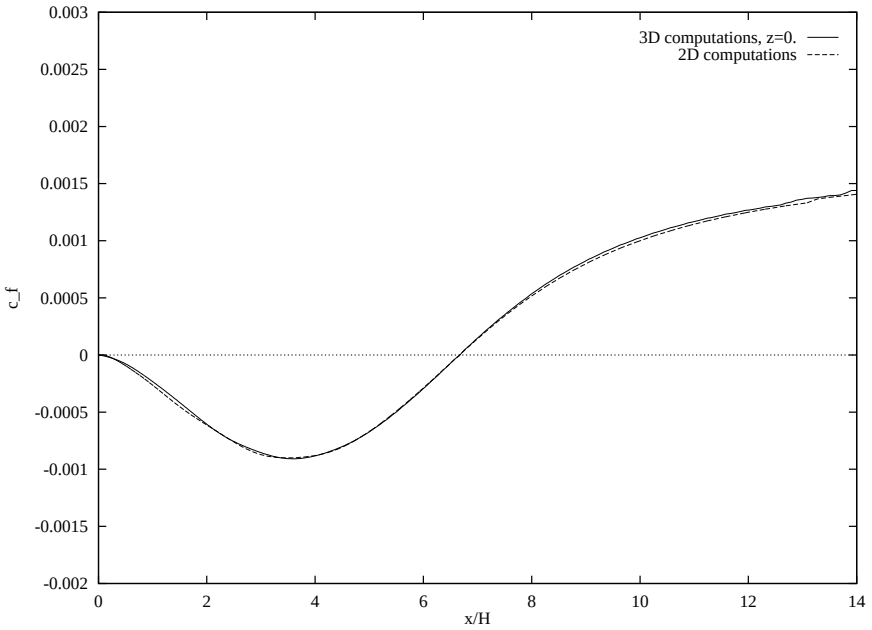


Figure 42: Friction coefficient c_f distribution on the bottom wall behind the step, $Re_H = 44000$.

6 CONCLUSION

A new solver for 3D incompressible Navier-Stokes equations based on the projection scheme ([2], [3]) is developed. The standard $k-\varepsilon$ turbulence model is implemented for turbulent flow simulation. The resolution of the problem for the intermediate velocity as well as for k and ε is ecompletely explicit. The stabilization of the convection terms is via PSI (Positive Streamwise Invariant) residual distribution scheme ([12], [20]). The Poisson problem for pressure is solved using conjugate gradient technique. In the turbulent flow simulations, the wall-laws are used for the solid boundaries, with the requirement $u \cdot n = 0$ treated in the strong sense.

The results presented for typical test configurations are very satisfactory compared with other results in the literature, indicating that this approach can be used competetively in the 3D $k-\varepsilon$ turbulent simulations. Besides, the computational time stays reasonable - an overnight computation for 3D turbulent flow problem with an unstructured mesh with approximately 50000 nodes on a workstation doing 10 MFlops.

Extensions of the solver for domains with moving boundaries using both injection-suction boundary conditions and ALE (Arbitrary Lagrangian-Eulerian) formulation and for flows including rotational effects are being realized.

In that sense, this solver has already been applied to different complex problems including turbulent flow analysis, such as optimization (engine cooling fan blade design, [61]) and fluid-structure interaction problems (stability analysis of viaduct cross-sections, [62]).

7 APPENDIX 1: NSIKE code description and the instructions for utilization

7.1 The code description

The code is programmed using FORTRAN77 and it consists of 12 fortran files, containing 31 subroutines.

- aires.f: - subroutines: aires2, aires3;
- caldtl.f: - subroutine: caldtl;
- clhaut.f: - subroutines: clhaut2, clhaut3;
- config.f:- subroutine: config;
- grad.f: - subroutine: grad;
- init.f: - subroutine: init;
- loglaw.f: - subroutines: lawlaminar, loglaw2d, reichlaw2d, loglaw3d;
- mailla.f: - subroutines: mailla2, mailla3, cvnox2, cvnox3;
- nsike.f: - main;
- result.f: - subroutines:result, printiso2, isovat;
- solve.f: - subroutines: res-exp, flux-p2, flux-p3, flux, rodec2, rodec3, smooth, unwall2d, unwall3d;
- waldat.f: - subroutine: waldat2d, waldat3d;

The parallelization of code is previsedfor the next version, and the programation technique is largely addapted to the future parallelization. The structure of the code is mainly the following:

→ main:
  → config,mailla,init,caldtl (initialization)
  → resexp (algorithm):
    → flux (rodec2 or rodec3, wall-law subroutines)
    → grad (flux-p2 or flux-p3)
    → caldtl (time step updating)
    → result,waldat (results output)

The code uses and generates several files: DATA, MESH.amdba or MESH.amdba3, INIT.BB, SOL.BB, GNU.* or *.ascii2d, *.ascii3d, WALL.DATA, cdl.data, RESIDUAL. To start a simulation only two files are needed: DATA file defining the problem parameters and MESH.* file defining the problem domain discretization.

All input and output files used and generated by the code are now described in details:

- MESH.amdba, MESH.amdba3 - *.amdba, *.amdba3 ASCII formats are used for mesh definition (nodes coordinates, elemental conectivity, node references (frontier triangles - in 3D case). In 2D, the mesh format is the standard *.amdba format, already used in the code NSC2KE, and the routine reading the 2D mesh is the following:

```
open(1,file="MESH.amdba",status="old")
read(1, * ) ns,nt !number of nodes,number of triangles
  if(ns.gt.nn) stop 'nn'
  if(nt.gt.nnt) stop 'nnt'
 do is=1,ns  !loop on nodes
 read(1,*) i,coor(1,is),coor(2,is),logfr(is)
 enddo
 do jt=1,nt  !loop on triangles
 read(1,*) k,(nu(i,jt),i=1,nen)
 enddo
close(1)
```

  In 3D situation is slightly more complicated, because of the fact that the logic of the frontier is defined through faces, but in fact the code uses also the logic of the nodes. We are suggesting the user to verify the logic of the nodes for each particular case, especially when using wall-laws and symmetry boundaries, through a pre-processor. In this sense, we use the MESH.amdba3 format that is equivalent to the standard *.amdba3 format except that at the end of the file we read the array containing the logic of all nodes:

```
open(1,file="MESH.amdba3",status="old")
rewind(1)
read(1,*) ns,nt,nfr
  if(ns.gt.nn) stop 'nn'
  if(nt.gt.nnt) stop 'nnt'
  if(nfr.gt.nn) stop 'nfr'
do is=1,ns
read(1,*)  ( coor(i,is), i= 1, 3 )
enddo
```

```
read(1,*) ( ( nu(k,jt), k= 1, 4 ), jt= 1, nt )
read(1,*) ( logf(ifac), ifac= 1, nfr )
read(1,*) ( ( nuf(i,ifac), i= 1, 3 ), ifac= 1, nfr )
read(1,*) ( logfrs(is), is= 1, ns)
close(1)
```

The mesh has to be defined in meters, in 2D we don't accept the triangles with more than one side belonging to the frontier Γ_h. Besides, the orientation of nodes in each triangle is important for the computation of the normal in 2D, so the nodes in a triangle connectivity table have to be ordered counter-clockwise.

- DATA - simulation dependent parameters are adjusted, namely: CFL number, time integration procedure, initialization type, choice of output files, Reynolds number, Stokes or Navier-Stokes simulation, turbulence model application, wall-law choice.

  An example of the DATA file is given below: Navier-Stokes + $k-\varepsilon$ model + Reichardt wall-law for the simulation of the backward facing step test-case, $Re = 44000$:

```
2          --> isave (=0 nothing, =1 SOL+WALL, =2 GNU,=3 vigie)
1          --> ins (=0 stokes, =1 ns)
44000      --> Reynolds
0.0 0.0    --> teta1,teta2
0          --> axisymmetric =1
cccc time scheme ccccccccccccccccccccccccccccccccccccccccccccccccccc
1.         --> cfl
30000     --> number of time step
50         --> save every
100.        --> maximum time for run
-5         --> order of magnitude for residual decreasing
0          --> =0 uniform initial, =1 - start from INIT.BB (2-turb)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
1          --> =0 no turbulence model/ =1 k-e model
2          --> WL type =1 loglaw =2 reichardt, =5 euler, =0 two-layer
5.e-3      --> delta
```

- INIT.BB - if used, defines the initial data for simulation. The most general case is the turbulent flow computation and the routine which reads the file in that case is:

```
open(20,file="INIT.BB",status="old")
rewind(20)
read(20,*)
```

```
      do is=1,ns
      read(20,*)(un(iv,is),iv=1,nsd),pres(is)
     1  ,(un(it,is),it=nsd+1,nsd+ntu)
      enddo
      read(20,*) kt,t
      close(20)
```

- SOL.BB - the solution is saved in this file, in the same format as INIT.BB, so it can be used directly as an initial condition for the next run
- GNU.* - files with the output ready for postprocesing in GNUPLOT
- WALL.DATA - file with values of pressure and friction coefficients for all nodes on the solid wall (having reference 3). The file is ready for the postprocessing in GNUPLOT, for example. It consists of the following 5 columns: x, y, $c_p = \frac{p-p_\infty}{\frac{1}{2}\rho_\infty |u_\infty|^2}$, $c_f = \frac{\tau_w}{\frac{1}{2}\rho_\infty |u_\infty|^2} = \frac{S \cdot n \cdot s}{\frac{1}{2}\rho_\infty |u_\infty|^2}$, $y^+ = \frac{u_\tau \delta}{\nu}$.
- cdl.data - file with the drag, lift and momentum coefficient evolution. The file is ready for the postprocessing in GNUPLOT. It consists of the following 5 columns: *iteration number* $\equiv kt$, t, $C_L = \frac{\int_{\Gamma_w}[T\cdot n]_2}{\frac{1}{2}\rho_\infty |u_\infty|^2 cb}$, $C_D = \frac{\int_{\Gamma_w}[T\cdot n]_1}{\frac{1}{2}\rho_\infty |u_\infty|^2 cb}$, $C_M = \frac{\int_{\Gamma_w}(x_2[T\cdot n]_1 + x_1[T\cdot n]_2)}{\frac{1}{2}\rho_\infty |u_\infty|^2 c^2 b}$ where c and b are the characteristic length scales, the directions 1 and 2 represent the inflow direction and the direction perpendicular to the inflow direction and T is the strain tensor defined as $T = -pI + S$.
- nsike.desc, nsike.ascii2d, nsike.ascii3d - files with the output ready for postprocesing in VIGIE
- RESIDUAL - file with the evolution of the residual quantities (used for convergence monitoring). This file is ready for the postprocessing in GNUPLOT. It consists of the following 6 columns: *iteration number* $\equiv kt$, t, $|R^i_{div}|$, $|R^i_{NS}|$, $|R^i_k|$, $|R^i_\varepsilon|$, where R denotes the vector of residuals of the dicrete formulation for continuity equation, Navier-Stokes equations, k equation and ε equation, respectively, normalized by the value of appropriate $|R_i|$ from the first iteration $kt = 0$.

7.2 Directions for the flow simulation

Nondimensionalisation

The code is nondimensionalized and we are working with nondimensionalized variables u^*, p^*, k^* and ε^*. In this sense, we should always scale the domain size to have the characteristic length of a problem equal to 1. In that way, the only value that defines the flow is the Reynolds number of the actual physical flow we want to compute and it has to be given in the file DATA. The actual physical values can be obtained from the nondimensionalized variables using the following expressions:

$$t = t^* \frac{L_{ref}}{U_{ref}}, \qquad L = L^* L_{ref},$$

$$u = u^* U_{ref}, \quad p = p^* U_{ref}^2, \quad k = k^* U_{ref}^2, \quad \varepsilon = \varepsilon^* \frac{U_{ref}^3}{L_{ref}},$$

where p (and p^*) is kinematic pressure (divided by density ρ). All the other output values are nondimensionalized (e.g. C_L, C_D, c_p, c_f etc.)

Physical model

Regarding the physical model to be used, the choice has to be made whether the simulation will use a turbulent model, or not, and further on, in the case of laminar flow simulations there is the choice between the Stokes and the Navier-Stokes equations. Besides, the user has to define the value of Reynolds number. The Euler flow can be simulated using laminar Navier-Stokes model choosing 5 for the wall-law type (corresponding to the homogeneous Neumann condition for the tangential velocity).

Boundary conditions

After providing a mesh of the problem domain Ω, satisfying the constraints mentioned above, one can define the problem dependent parameters. The definition of boundary conditions is done through the logic of the nodes in the file defining the mesh in the following way:

- logic=0 : internal node,
- logic=2 : slip (or symmetry) boundary condition $u \cdot n = 0$, $S \cdot n = 0$
- logic=3 : solid boundary - in the laminar case implies $u = 0$, in the turbulent case, the wall-law imposed depends on the choice of the wall-law in the file DATA (0 - two-layer simulation, 1 - loglaw, 2 - Reichardt's law, 3 - Reichardt's law with pressure gradient and convection corrections, 5 - slipping boundary condition)

- logic=4 : outflow boundary condition: $p - S \cdot n \cdot n = 0$, $S \cdot n \cdot s = 0$,
- logic=5 : inflow boundary condition - in fact one imposes the Dirichlet boundary condition on velocity defined with θ_1 in 2D: $u^* = (\cos(\theta_1), \sin(\theta_1))$, and with θ_1 and θ_2 in 3D: $u^* = (\cos(\theta_1)\cos(\theta_2), \sin(\theta_2), \sin(\theta_1)\cos(\theta_2))$, where θ_1 and θ_2 are given in the file DATA. For turbulence quantities, the inflow condition is defined as $k^*_{in} = 10^{-5}$ and $\varepsilon^*_{in} = 10^{-5}$ If necessary, the values given for the nodes with the logic=5 can be easily changed in the subroutine *init*, in the file *init.f*.
- logic=7,8 : periodic boundary conditions in 2D (periodicity is programmed in y direction, so the code searches the correspondance of the nodes which have the logic equal to 7 and 8 with the smallest distance in x), and in 3D (also in y direction, but here the code searches the correspondance of the nodes which have the logic 7 and 8 and the smallest distance in x and z).

Nevertheless, the specification of boundary conditions is rather delicate, for example in 3D, especially when defining boundary conditions on the edges of the three-dimensional domain that correspond to symmetry or wall-laws boundary conditions.

Initialization

Now, the user can proceed with the definition of the problem parameters in the file DATA. The next point is the initialization. If the file INIT.BB is not present, the initialization will be done with the uniform solution corresponding to the default values at the nodes with the logic=5, with the satisfaction of the boundary conditions for the solid boundary (in the laminar case $u = 0$, in the turbulent case $U \cdot n = 0$. Otherwise, the values are read from the file INIT.BB, according to the parameter value in the file DATA (1 - for laminar flows, 2 - for turbulent flows).

Numerical integration

Finally, the parameters regarding the numerical integration have to be chosen in file DATA. The parameter defining the size of the time step cfl, can be taken as 1., this works for all cases presented, except for the Stokes flow simulations, where it has to be set smaller. Regarding the simulation stopping criterium three choices can be made, the maximum number of iterations, the maximum global time and the logarithm of the minimal normalized residual value.

8 APPENDIX 2: Navier-Stokes equations in the axisymmetric coordinate system

The incompressible Navier-Stokes equations are

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla p + \nabla \cdot S + f, \tag{114}$$

$$\nabla \cdot u = 0, \tag{115}$$

where $S = \nu(\nabla u + \nabla u^T)$, u is the velocity vector, p the kinematic pressure (divided by density), ν the kinematic viscosity, f is a given body force and $(\nabla u)_{ij} = u_{i,j}$ the gradient of u.

The gradient and divergence operators are in the axisymmetric coordinate system (z, r) defined as

$$\nabla \psi = (\frac{\partial \psi}{\partial z}, \frac{\partial \psi}{\partial r}), \qquad \nabla \phi = \begin{bmatrix} \frac{\partial \phi_z}{\partial z} & \frac{\partial \phi_r}{\partial z} \\ \frac{\partial \phi_z}{\partial r} & \frac{\partial \phi_r}{\partial r} \end{bmatrix}, \tag{116}$$

$$\nabla \cdot \phi = \frac{\partial \phi_z}{\partial z} + \frac{\partial \phi_r}{\partial r} + \frac{\phi_r}{r}, \tag{117}$$

leading to the following form of the Navier-Stokes equations in the axisymmetric coordinate system (z, r)

$$\begin{aligned} \frac{\partial u_z}{\partial t} + u \cdot \nabla u_z = &-\frac{\partial p}{\partial z} + \frac{\partial}{\partial z}(2\nu \frac{\partial u_z}{\partial z}) \\ &+ \frac{\partial}{\partial r}(\nu(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z})) + \frac{1}{r}\nu(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}) + f_z, \end{aligned} \tag{118}$$

$$\begin{aligned} \frac{\partial u_r}{\partial t} + u \cdot \nabla u_r = &-\frac{\partial p}{\partial r} + \frac{\partial}{\partial z}(\nu(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z})) \\ &+ \frac{\partial}{\partial r}(2\nu \frac{\partial u_r}{\partial r}) + \frac{1}{r} 2\nu \frac{\partial u_r}{\partial r} + f_r, \end{aligned} \tag{119}$$

$$\frac{\partial u_z}{\partial z} + \frac{\partial u_r}{\partial r} + \frac{u_r}{r} = 0. \tag{120}$$

Remark

The variatonal formulation for the Navier-Stokes equations:

Find $u \in H^1(\Omega)^d$, $p \in L^2(\Omega)$

$$\int_\Omega \frac{\partial u}{\partial t} v + \int_\Omega u \cdot \nabla u v + \int_\Omega \nabla p v + \int_{\partial\Omega} S \nabla v - \int_\Omega f v = 0, \forall v \in H^1(\Omega)^d \tag{121}$$

$$\int_\Omega \nabla \cdot u q = 0, \forall q \in L^2(\Omega) \tag{122}$$

changes in the axisymmetric case only in the computation of the integral over the domain Ω, and the way $\nabla \cdot u$ is computed.

9 APPENDIX 3: Navier-Stokes equations in cylindrical coordinate system

The formulation of Navier-Stokes equations in cylindrical coordinate system could be interesting when analyzing the flows, when the rotation around an axis is imposed (for example, the turbomachinery flows).

The gradient and divergence operators are in the cylindrical coordinate system (z, r, φ) defined as

$$\nabla\psi = (\frac{\partial\psi}{\partial z}, \frac{\partial\psi}{\partial r}, \frac{1}{r}\frac{\partial\psi}{\partial\varphi}), \tag{123}$$

$$\nabla\cdot\phi = \frac{\partial\phi_z}{\partial z} + \frac{\partial\phi_r}{\partial r} + \frac{\phi_r}{r} + \frac{1}{r}\frac{\partial\phi_\varphi}{\partial\varphi}, \tag{124}$$

and the tensor S is defined as

$$S = \nu\begin{bmatrix} 2\frac{\partial u_z}{\partial z} & \frac{\partial u_r}{\partial z}+\frac{\partial u_r}{\partial z} & \frac{\partial u_\varphi}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial\varphi} \\ \frac{\partial u_z}{\partial r}+\frac{\partial u_r}{\partial z} & 2\frac{\partial u_r}{\partial r} & \frac{1}{r}\frac{\partial u_r}{\partial\varphi}+\frac{\partial u_\varphi}{\partial r}-\frac{u_\varphi}{r} \\ \frac{\partial u_\varphi}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial\varphi} & \frac{1}{r}\frac{\partial u_r}{\partial\varphi}+\frac{\partial u_\varphi}{\partial r}-\frac{u_\varphi}{r} & \frac{1}{r}\frac{\partial u_\varphi}{\partial\varphi}+\frac{u_r}{r} \end{bmatrix}, \tag{125}$$

leading to the following form of the Navier-Stokes equations in the cylindric coordinate system (z, r, φ)

$$\frac{\partial u_z}{\partial t} + u\cdot\nabla u_z = -\frac{\partial p}{\partial z} + (\nabla\cdot S)_z + f_z, \tag{126}$$

$$\frac{\partial u_r}{\partial t} + u\cdot\nabla u_r - \frac{u_\varphi^2}{r} = -\frac{\partial p}{\partial r} + (\nabla\cdot S)_r + f_r, \tag{127}$$

$$\frac{\partial u_\varphi}{\partial t} + u\cdot\nabla u_\varphi - \frac{u_\varphi u_r}{r} = -\frac{1}{r}\frac{\partial p}{\partial\varphi} + (\nabla\cdot S)_\varphi + f_\varphi, \tag{128}$$

where the last terms in the left hand side of the equations for u_r and u_φ come from the transformation of the computation of acceleration in the cylindrical coordinate system.

Remarks

1. In the case where we have constant value of u_φ, for the flows with swirl, the only term which has to be added to the axisymmetric formulation is the last term on the left hand side of the equation for u_r.

2. In the variational formulation for the Navier-Stokes equations, the formulation stays almost the same as the one in the axisymmetric case, apart of the addition of two more acceleration terms in the equations for u_r and u_φ, but now the operator of gradient is slightly different (with respect to the same operator in (x, y, z) coordinate system), and the same is valid for the divergence operator.

10 APPENDIX 4: Time-step computations

The time-step that can be used in the integration is calculated applying the results of the stability analysis for 1D convection-diffusion equation

$$u_t + au_x = \nu u_{xx}. \tag{129}$$

In the case of explicit time integration procedure and the application of the simple, first order in space accurate, upwind scheme for the discretization of the equation, Fourier analysis gives the limitation on the time step:

$$\Delta t \leq \frac{\Delta x}{|a| + \frac{2\nu}{\Delta x}}. \tag{130}$$

The generalization of relation (130) for multidimensional problems consists in replacing Δx, the element diameter in 1D, with the minimal triangle or tetrahedra height in 2D and 3D respectively. Furthermore, the application of (130) to the Navier-Stokes equations leads to

$$\Delta t \leq \frac{\Delta x}{|u| + \frac{2(\nu+\nu_t)}{\Delta x}}, \tag{131}$$

where ν_t is used only for turbulent flow computations.

In the formulation (81), finally, when computing steady flows, the application of local time stepping technique, leads to

$$\Delta t_i = CFL \min_{K, q_i \in K} \frac{\Delta x_K}{|u_K| + \frac{2(\nu+\nu_t)_K}{\Delta x_K}}. \tag{132}$$

When computing unsteady flows, the time step T is the minimum of the local time steps.

Remark

It is interesting to note that in the case of the application of an explicit compressible solver, with the time step restriction

$$\Delta t \leq \frac{\Delta x}{|u|(1 + \frac{1}{M}) + \frac{2(\nu+\nu_t)}{Pr\Delta x}}, \tag{133}$$

where M and Pr are Mach and Prandtl number respectively, to the simulation of incompressible flows ($M \ll 1$, e.g. $M = 0.1$) leads to an order of magnitude smaller Δt (for the flows where the viscous effects are small).

11 APPENDIX 5: Error estimations for projection methods

The derivation will be given only for the Algorithm 2, where we have only the Dirichlet boundary condition $u = w$ on Γ. The first step consists in comparing $\tilde{u}(T)$ with the Navier-Stokes solution $u(T)$. We will use Taylor series expansion (with the notation $\dot{f} = \frac{\partial f}{\partial t}$ and $\ddot{f} = \frac{\partial^2 f}{\partial t^2}$)

$$u(t) = u_0 + t\dot{u}_0 + \frac{t^2}{2}\ddot{u}_0 + O(t^3), \tag{134}$$

$$\tilde{u}(t) = \tilde{u}_0 + t\dot{\tilde{u}}_0 + \frac{t^2}{2}\ddot{\tilde{u}}_0 + O(t^3), \tag{135}$$

with $\tilde{u}_0 = u_0$ (by definition).

Using the partial differential equations (assuming sufficient smoothness) we have

$$\dot{u}_0 = \nabla \cdot S_0 + f_0' - \nabla p_0, \tag{136}$$

and

$$\dot{\tilde{u}}_0 = \nabla \cdot S_0 + f_0' - \nabla p_0, \tag{137}$$

where $f' = f - u \cdot \nabla u$. Further on, we have

$$\ddot{u}_0 = \frac{\partial}{\partial t}[\nabla \cdot S + f - \nabla p]_{t=0} = \nabla \cdot \nu(\nabla \dot{u}_0 + \nabla \dot{u}_0^T) + \dot{f}_0 - \nabla \dot{p}_0, \tag{138}$$

$$\ddot{\tilde{u}}_0 = \frac{\partial}{\partial t}[\nabla \cdot \tilde{S} + \tilde{f} - \nabla p_0]_{t=0} = \nabla \cdot \nu(\nabla \dot{u}_0 + \nabla \dot{u}_0^T) + \dot{\tilde{f}}_0, \tag{139}$$

where we have used the fact that $\dot{u}_0 = \dot{\tilde{u}}_0$. Since

$$\dot{f}_0 = \frac{\partial f_0}{\partial u_0}\dot{u}_0, \tag{140}$$

and further on $\dot{f}_0 = \dot{\tilde{f}}_0$, we have in Ω:

$$\tilde{u}(t) - u(t) = \frac{t^2}{2}\nabla \dot{p}_0 + O(t^3). \tag{141}$$

For φ we have (at time $t = T$)

$$\Delta\varphi = \nabla \cdot \tilde{u}(T) = \Delta(\frac{T^2}{2}\dot{p}_0 + O(T^3)) \text{ in } \Omega, \tag{142}$$

$$\frac{\partial \varphi}{\partial n} = n \cdot (\tilde{u}(T) - w(T))$$

$$= T\frac{\partial}{\partial n}(\beta_1 P_0 + \beta_2 T\dot{P}_0)$$

$$= \frac{\partial}{\partial n}(\frac{T^2}{2}\dot{P}_0 + O(T^3)) \text{ on } \Gamma, \tag{143}$$

where the second line in the above equation comes from the assumption that we are searching for "optimal" boundary conditions for $\tilde{u}$ in the form

$$\tilde{u}(t) = w(t) + t\nabla(\beta_1 P_0 + \beta_2 T\dot{P}_0) \text{ on } \Gamma, \tag{144}$$

and the final line implies that

$$\frac{\partial}{\partial n}(\beta_1 T P_0 + (\beta_2 - \frac{1}{2})T^2\dot{P}_0) = O(T^3) \tag{145}$$

should be satisfied, leading to the selection $\beta_1 = 0$ and $\beta_2 = \frac{1}{2}$ in the expression for "optimal" boundary conditions for $\tilde{u}$:

$$\tilde{u}(t) = w(t) + \frac{t^2}{2}\nabla\dot{P}_0 \text{ on } \Gamma. \tag{146}$$

Remark

However, in the actually programmed Algorithm 2, this is not respected, and we take $\tilde{u} = w$ on Γ.

Finally, it follows that the solution of the Poisson problem for φ is

$$\varphi = \frac{T^2}{2}\dot{p}_0 + O(T^3) \text{ in } \bar{\Omega}. \tag{147}$$

Examining, now, $v = \tilde{u}(T) - \nabla\varphi$ using (141) and (147) we have

$$v = u(T) + O(T^3) \text{ in } \Omega. \tag{148}$$

Lately, several analysis ([3] and [9])were carried out to justify that even the violation of compatible boundary conditions for projection method, in algorithms of the Algorithm 1 and 2 type, destroys the following error estimates only in the near-wall boundary layer,i.e for $x > \delta$, where $\delta \equiv \sqrt{(\nu T)}$.

References

[1] J.A. Chorin, *A numerical method for solving incompressible viscous flow problems,* J. Comput. Phys., Vol. 2, pp. 12-26, 1967.

[2] J.A. Chorin, *Numerical solution of the Navier-Stokes equations,* Math. Comput., Vol.22, pp. 745-762, 1968.

[3] P.M. Gresho, *On the theory of semi-implicit projection methods for viscouc incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix I: theory,* Int. Journ. Num. Meth. Fluids, Vol. 11, pp. 587-620, 1990.

[4] P.M. Gresho, *Some current CFD issues relevant to the incompressible Navier-Stokes equations,* Comput. Methods Appl. Mech. Eng., Vol. 87, pp. 201-252, 1991.

[5] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations,* J. Comput. Phys., Vol. 59., pp. 308.-, 1985.

[6] J. Van Kan, *A second-order pressure correction scheme for viscous incompressible flow,* SIAM J. Sci. Stat. Comput., Vol. 7., pp. 870-, 1986.

[7] T. Zang and Y. Hussaini, *On spectral multigrid-methods for the time-dependent Navier-Stokes equations,* Appl. Math. Comput., Vol. 19.,pp. 359-, 1986.

[8] R. Temam, *Navier-Stokes equations,* North-Holland (Elsevier), Amsterdam, 1984.

[9] C. Schwab, *Remarks on pressure approximation in projection methods for viscous incompressible flow,* Proc. of the Intern. Conf. on Finite Elements in Fluids - New trends and applications, Venezia, 15-21 October, 1995.

[10] J.-L. Guermond and L. Quartapelle, *Unconditionally stable finite-element method for the unsteady Navier-Stokes equations*, Proceedings of the Ninth International Conference on 'Finite Elements in Fluids', pp:367-376, Venezia, Italy, 15-21 October, 1995.

[11] J.C. Simo and F. Armero, *Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations,* Comput. Methods Appl. Mech. Eng., Vol. 111, pp. 111-154, 1994.

[12] H. Deconinck, R. Struijs, G. Bourgois and P.L. Roe, *Compact advection schemes on unstructured grids*, VKI Lecture Series 1993-04.

[13] H. Paillere, J.-C. Carette and H. Deconinck, *Multidimensional upwind and SUPG methods for the solution of the compressible flow equations on unstructured grids*, VKI Lecture Series 1994-05.

[14] A.N. Brooks and T.J.R. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, CMAME, vol. 32, pp:199-259, 1982.

[15] P. Hansbo and A. Szepessy, *A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations*, CMAME, vol. 84, pp:175-192, 1990.

[16] L.P. Franca, S.L. Frey and T.J.R. Hughes, *Stabilized finite element methods: I. Application to the advective-diffusive model*, CMAME, vol. 95, pp:253-276, 1992.

[17] L.P. Franca and S.L. Frey, *Stabilized finite element methods: II The incompressible Navier-Stokes equations*, CMAME, vol. 99, pp:209-233, 1992.

[18] E. van der Weide and H. Deconinck, *Aerodynamic simulation using compact upwind schemes on unstructured grids*, Proceedings of ECCOMAS 98, Athenes, Greece, pp:369-376, 1998.

[19] S. Bogaerts, G. Degrez and B. Nitrosso, *Multidimensional upwind schemes for the solution of the incompressible Navier-Stokes equations*, Proceedings of ECCOMAS 96, Paris, France, pp:57-63, 1996.

[20] S. Bogaerts, G. Degrez and E. Razafindrakoto, *Upwind residual distribution schemes for incompressible flows*, Proceedings of ECCOMAS 98, Athenes, Greece, pp:742-747, 1998.

[21] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer Series in Computational Mathematics 5, Springer-Verlag, 1980.

[22] D. Silvester, *Stabilised Mixed Finite Element Methods*, UMIST, Numerical Analysis Report No. 262, August 1995.

[23] R. Becker, *An adaptive finite element method for the incompressible Navier-Stokes equations on time-dependent domains*, PhD Thesis, Institut fur Angewandte Mathematik, Universitat Heidelberg, 1995.

[24] C. Pares Madronal, *Etude Mathematique et Approximation Numerique de Quleques Problemes aux Limites de la Mecanique des Fluides Incompressilbes*, These de Doctorat, Universite Paris VI, 1992.

[25] B.E. Launder and D.B. Spalding,*Mathematical Models of Turbulence,* Academic Press (1972).

[26] B. Mohammadi and O. Pironneau,*Analysis of the K-Epsilon Turbulence Model*, WILEY, 1994 (Book).

[27] B. Mohammadi and O. Pironneau,*Unsteady Separated Turbulent Flows Computation with Wall-Laws and $k-\varepsilon$ Model,* CMAME, vol. 148, pp:393-405, 1997.

[28] B. Mohammadi,*NSC2KE : an User Guide*, Technical report INRIA No.164, 1994.

[29] B. Mohammadi and G. Medić, *A Critical Evolution of the $k-\varepsilon$ Model and Wall-Laws for Separated Unsteady Flows*, IJCFD (accepted for publication), 1997.

[30] G. Medić, B. Mohammadi and O. Pironneau, *Algorithmic Issues for the $k-\varepsilon$ model on unstructured meshes*, (to be published), 1998.

[31] G. Comte-Bellot, S. Corrsin,*Simple Eulerian Time-Correlation of Full an d Narrow-Band Velocity Signals in Grid-Generated Isotropic Turbulence,* JFM, vol.48, pp:273-33 7, 1971.

[32] S. Thangam,*Analysis of Two-Equation Turbulence Models for Recirculating Flows*, ICASE report No. 91-61, 1991.

[33] P.L.Roe, *Approximate Riemann Solvers, Parameters Vectors and Difference Schemes,* J.C.P. Vol.43, 1981.

[34] G.D.Van Albada, B. Van Leer,*Flux Vector Splitting and Runge-Kutta Methods for the Euler Equations*, ICASE 84-27, June 1984.

[35] A. Dervieux,*Steady Euler Simulations using Unstructured Meshes*, VKI lecture series, 1884-04, (1985).

[36] R. Struijs, H. Deconinck, P. de Palma, P. Roe, G.G.Powel, *Progree on Multidimensional Upwind Euler Solvers for Unstructured Grids,* AIAA paper 91-1550, (1991).

[37] J. Steger, R.F. Warming, *Flux Vector Splitting for the Inviscid gas dynamic with Applications to Finite-Difference Methods*, J. Comp. Phys. 40, pp:263-293. (1983).

[38] M.H.Lallemand,*Schemas Decentrés Multigrilles pour la Résolution des Equations D'Euler en Eléments Finis*, Thesis, Univ. of Provence-Saint Charles, 1988.

[39] V. Haroutunian and M.S. Engelman,*Two-Equation Simulations of Turbulent Flows: A Commentary on Physical and Numerical Aspects,* ASME 1993 Winter Annual Meeting, New Orleans, USA, 1993.

[40] W. Rodi, *On the Simulation of Turbulent Flow Past Bluff Bodies,* Journal of Wind Engineering and Industrial Aerodynamics, Vol. 46/47, pp:3-19, 1993.

[41] W.Rodi, *LES versus RANS Calculations of Complex Flows past Bluff Bodies,* Proceedings of Basel World CFD Used Days 1996, Third World Conference in Applied CFD, Freiburg, Germany, May 19-23, 1996.

[42] R. Franke and W. Rodi, *Calculation of vortex shedding past a square cylinder with various turbulence models,*, Turbulent Shear Flows 8, Dust et al. (eds.), Springer Verlag, 1993.

[43] G. Bosch, *Experimentelle und theoretische Untersuchung der instationaren Stromungen um zylindrische Strukturen,* Ph.D. Thesis, University of Karlsruhe, 1995.

[44] B. Cantwell and D. Coles, *An experimental study of entrainement and transport in the turbulent near-wake of a circular cylinder,* J. Fluid Mech., Vol. 136, pp.321-374, 1983.

[45] R. Franke, *Numerische berechnung der instationaren Wirbeleblosung hinter zylindrischen Korpern,* Ph.D. Thesis, University of Karlsruhe, 1991.

[46] T. Tamura, I. Ohta and K. Kuwahara, *On the reliability of two-dimensional simulation for unsteady flows around a cylinder-type structure*, J. Wind Eng. and Ind. Aerodyn., Vol. 35, pp. 275-298, 1990.

[47] C.C. Song and M. Yuan, *Simulation of vortex shedding flow about a circular cylinder at high Reynolds numbers,* J. Fluids Eng., Vol. 112, pp. 155-163, 1990.

[48] H. Borouchaki, P.L. George, F. Hecht, P. Laug and E. Saltel, Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie I: Algorithmes, RR 2741, INRIA, (1995).

[49] H. Borouchaki, P.L. George, F. Hecht, P. Laug, B. Mohammadi and E. Saltel, Mailleur bidimensionnel de Delaunay gouverné par une carte de métriques. Partie II: Applications, RR 2760, INRIA, (1995).

[50] Borouchaki H and Laug P, Le mailleur adaptatif bidimensionnel BL2D: manuel d'utilisation et documentation, RT 0185, INRIA, (1995).

[51] H. Schlichting, *Boundary layer theory*, 7th edition, McGraw-Hill, 1978.

[52] T.-H. Shih, J. Zhu and J.L. Lumley, *Calculation of wall-bounded complex flows and free shear flows*, Int. Journ. Num. Meth. Fluids, Vol. 23, pp. 1133-1144, 1996.

[53] D.J. Monson, H.L. Seegmiller, P.K. McConnaughey and Y.S. Chen, *Comparison of experiment with calculations using curvature-corrected zero and two equation turbulence models for a two-dimensional U-duct*, AIAA Paper 90-1484, 1990.

[54] M. Schafer and S. Turek (with support by F. Durst, E. Krause and R. Rannacher), *Benchmark computations of laminar flow around a cylinder*, internal pubblication, Universitat Heidelberg.

[55] W.J. Kim and V.C. Patel, *An experimental study of boundary layer flow in a curved rectangular duct*, ASME Paper, FED, Vol. 146, pp. 13-28, 1993.

[56] B.F. Armaly, F. Durst, J.C.F. Pereira and B. Schonung, *Experimental and theoretical investigation of backward-facing step flow*, J. Fluid Mech., Vol. 127, pp. 473-496, 1983.

[57] T.H. Le, B. Troff, P. Sagaut, K. Dang-Tran, L. Ta Phuoc, *PEGASE: A Navier-Stokes solver for direct numerical simulation of incompressible flows*, Int. Journ. Num. Meth. Fluids, Vol. 24, pp. 833-861, 1997.

[58] U. Ghia, K.N. Ghia and C.T. Shin, *High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method*, Journ. Comp. Phys., Vol. 48, pp. 387-411, 1982.

[59] J. Shen, *Hopf Bifurcation of the Unsteady Reguralized Driven Cavity Flow*, Journ. Comp. Phys., Vol. 95, pp. 228-245, 1991.

[60] P. S. Klebanoff, *Characteristics of turbulence in a boundary layer with zero pressure gradient*, NACA report 1247, 1955.

[61] G. Medić, B. Mohammadi, N. Petruzzelli and M. Stanciu, *3D Optimal Shape Design for Complex Flows : Application to turbomachinery*, 37th AIAA Aerospace Sciences, January 11-14, Reno, USA, AIAA-99-0833 Paper, 1999.

[62] G. Medic, B. Mohammadi, M. Stanciu, *Prediction and aeroelastic simulation of turbulent flows in civil engineering applications*, Proceedings of ECCOMAS 98, Athenes, J.Wiley & Sons, 1062-1068, 1998.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399