An Accurate Finite Element Method for the Numerical Solution of Isothermal and Incompressible Flow of Viscous Fluid

Bilen Emek Abali 🆔

Institute of Mechanics, Technische Universität Berlin, 10587 Berlin, Germany; bilenemek@abali.org

Abstract: Despite its numerical challenges, finite element method is used to compute viscous fluid flow. A consensus on the cause of numerical problems has been reached; however, general algorithms—allowing a robust and accurate simulation for any process—are still missing. Either a very high computational cost is necessary for a direct numerical solution (DNS) or some limiting procedure is used by adding artificial dissipation to the system. These stabilization methods are useful; however, they are often applied relative to the element size such that a local monotonous convergence is challenging to acquire. We need a computational strategy for solving viscous fluid flow using solely the balance equations. In this work, we present a general procedure solving fluid mechanics problems without use of any stabilization or splitting schemes. Hence, its generalization to multiphysics applications is straightforward. We discuss emerging numerical problems and present the methodology rigorously. Implementation is achieved by using open-source packages and the accuracy as well as the robustness is demonstrated by comparing results to the closed-form solutions and also by solving well-known benchmarking problems.

Keywords: finite element method; fluid dynamics; computation; viscous fluid flow

1. Introduction

Isothermal flow of viscous fluid is modeled in Cartesian coordinates by using the balance equations of mass and linear momentum:

$$\frac{\partial \rho}{\partial t} + \frac{\partial v_i \rho}{\partial x_i} = 0 \,, \tag{1a}$$

$$\frac{\partial \rho v_j}{\partial t} + \frac{\partial}{\partial x_i}\left(v_i \rho v_j - \sigma_{ij}\right) = \rho g_j \,, \tag{1b}$$

respectively, where ρ denotes the mass density, v_i the velocity, σ_{ij} the non-convective flux term (Cauchy's stress), g_i the specific supply (gravitational forces); here and henceforth we apply Einstein's summation convention to repeated indices. In the case of Newtonian fluids such as water, oil, or alcohol, a linear relation for stress furnishes the governing equations with sufficient accuracy. This linear relation is sometimes called the Navier–Stokes equation:

$$\sigma_{ij} = (-p + \lambda d_{kk})\delta_{ij} + 2\mu d_{ij} \,, \quad d_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \,, \tag{2}$$

with the material constants λ, μ; and a new parameter called (hydrostatic) pressure, p. In the literature, often the law of motion, i.e., the material equation inserted into the balance equation is called Navier–Stokes equation. Navier did use Lagrangean method for derivation of the law of

motion; however, Stokes used—as we do it herein as well—the method introduced by Cauchy for separating material equation from the balance equation, see [1] for historical remarks. Consider an incompressible flow in a control volume initially filled with homogeneous water. Thus, the mass density remains constant in space and time; from the mass balance in Equation (1a), we obtain

$$\frac{\partial v_i}{\partial x_i} = 0 \, , \tag{3}$$

which is used for computing the pressure p. For an incompressible flow—as seen from Equations (2) and (3)—the mechanical pressure $-\frac{1}{3}\sigma_{ii}$ becomes identical to the hydrostatic pressure p such that we handle p as the pressure generated in a pump. Velocity and pressure fields have to satisfy Equations (1b) and (3).

In analytical mechanics, the aforementioned equations are fulfilled locally (in every infinitesimal volume element or simply point in space). For a computation we discretize the space, herein by using the finite element method (FEM). Within each element, the analytical functions for the unknowns v_i and p are represented by form (shape) functions with a local support, i.e., by means of a discrete element. The shape functions are not smooth, they belong to C^n with a finite n. In other words, the unknowns are finitely differentiable and depending on the governing equations and constitutive relations—from the mathematical analysis in [2]—we know that the correct choice of the form functions for velocity and pressure is of paramount importance for a robust computation. This so-called Ladyzhenskaya–Babuska–Brezzi (inf-sup compatibility) condition (LBB condition) tells us how to adjust the shape functions of velocity and pressure in the case of an isothermal and incompressible flow. Special elements like the Taylor–Hood element [3] or a mixed element with bubble functions [4] are often used for the isothermal and incompressible fluid flow problems. If one wants to include temperature deviation and electromagnetism into the computation, we fail to know the corresponding LBB condition for all shape functions (velocity, pressure, temperature, and electromagnetic fields). For practical purposes, a robust computation without exploiting the LBB condition is useful for a straightforward extension to multiphysics applications. This aspect is the main motivation of this work.

1.1. Computational Fluid Dynamics (CFD)

There are several methods for solving the aforementioned equations numerically. The finite element method is very well suited for problems in solid mechanics such that a robust finite element method in fluid dynamics allows us to implement a fluid structure interaction quite easily. Moreover, the finite element method also delivers results in thermodynamics and electromagnetism by using standard form functions as demonstrated in [5] such that a generalization to multiphysics problem attracts attention. Therefore, we concentrate on the finite element method for fluid flow problems. Within a finite element, the governing equations are satisfied globally (over the domain of the element). There exists a general assumption that we can use the same local governing equations holding globally in finite elements; however, this strategy leads to several numerical problems and to various proposals in [6–14], for a review of such suggestions see [15]. These so-called stabilization methods introduce a numerical parameter depending on the underlying mesh. There are several successful implementations as in [16–23]. From a practical point of view, such a stabilization method is very useful; however, introducing numerical parameters may be challenging as they need to be tuned depending on the application, see [24] for a discussion about this issue leading to methods being conditionally stable; see references in [25] about accuracy problems of different techniques. Indeed, numerical strategies exist for performing accurate simulations without numerical parameters. By using small elements—smaller than the characteristic length scale, for example Kolmogorov scale—we can overcome the numerical problems. This direct numerical simulation (DNS) is accurate and robust as demonstrated in [26–32]; however, it is often not feasible without access to super-computers. There is another class of so-called splitting or projection methods such as Chorin's method or its derivatives as in [33–39]. The finite volume method (FVM) is often used in the

computational fluid dynamics (CFD) as it is accurate and stable [40–43]. These methods are reliable; but it is challenging to adopt a splitting method or FVM in multiphysics. By using the FEM for viscous fluid flow, we intend to design a computation strategy by employing only physical (measurable) parameters in such a way that the strategy shows a local monotonous convergence as expected from the FEM.

The briefly mentioned problems are well known in the literature such that various new computation methods are suggested for viscous flow problems. The (numerical) parameter free approach in [44] shows 2D and 3D results for stationary viscous flows without the nonlinear convection term (often called Stokes's problem). Based on this idea an under-integrated mass matrix is used in [45,46] to perform simulations without stabilization terms in 2D. Several mesh-dependent stabilization terms and their connections to mesh-independent stabilization methods are investigated in [47]. In [48] vorticity is used instead of velocity such that a new kind of splitting scheme is proposed for solving 3D problems. In [49] balance equations of mass, momentum, angular momentum, and energy are used for performing 3D computations without numerical parameters; however, the method already uses the energy equation such that generalization for the non-isothermal case seems to be quite difficult. In [50] vorticity is introduced as an independent term ensuring that the balance of moment of momentum is satisfied, in 2D numerical solutions are performed without necessitating any (numerical) parameters. In [51] different strategies are performed for establishing 3D simulations. They are all based on writing the nonlinear convection term in a different (mathematically equivalent) form. In [52] a gradient-velocity-pressure formulation is suggested to solve 2D numerical experiments.

1.2. Scope of This Work

In this work, we discuss a special yet general case, namely an isothermal and incompressible flow. For understanding the numerical problems, often, the pressure related numerical problems and velocity related numerical problems are studied separately. We use one balance equation for calculating the pressure and another balance equation for calculating the velocity. It is important to remark that both balance equations are coupled such that it is not uniquely defined which one to use for computing pressure and which one to utilize for determining velocity. More robust numerical strategies use both balance equations for both of the unknowns, this approach has already been undertaken in Chorin's method and then extensively exploited by the pressure stabilized Petrov–Galerkin method (PSPG). We use essentially the same strategy herein by motivating this approach from a different perspective. Numerical problems are surpassed by incorporating the balance of angular momentum delivering the necessary smoothness for the pressure. Conventionally, the balance of angular momentum is neglected since it is already fulfilled locally by the balance of linear momentum (for the case of non-polar fluids). Furthermore, we discuss the integration by parts and suggest another approach than usually seen in the literature. We explain in detail how to generate the weak form. Additionally, we emphasize that the weak form can be extended to fluid structure interaction or multiphysics problems very easily. We use open-source packages developed under the FEniCS project (The FEniCS computing platform, https://fenicsproject.org/) and solve some academic examples in order to present the accuracy, local monotonous convergence, and robustness of the proposed methodology. All codes are made public on the web site in [53] to be used under the GNU Public license as in [54] for promoting an efficient scientific exchange as well as further studies.

2. Variational Formulation

Consider the following general balance equation:

$$\left(\int_{\Omega} \psi \, \mathrm{dv} \right)^{\bullet} = \int_{\Omega} z \, \mathrm{dv} + \int_{\partial \Omega} (f + \phi) \, \mathrm{d}a \, , \tag{4}$$

where the rate of the variable ψ is balanced with the supply term, z, acting volumetrically and with the flux terms—convective f and non-convective ϕ—applying on the surface, $\partial\Omega$, of a domain (control volume), Ω. By using Table 1, we can obtain the balance equations of mass, linear momentum, and angular momentum.

Table 1. Volume densities, their supply and flux terms in the balance equations.

ψ	z	f	ϕ
ρ	0	$n_i(w_i - v_i)\rho$	0
ρv_j	ρg_j	$n_i(w_i - v_i)\rho v_j$	σ_{ij}
$\rho(s_j + \epsilon_{jkl}x_k v_l)$	$\rho(\ell_j + \epsilon_{jkl}x_k g_l)$	$n_i(w_i - v_i)\rho(s_j + \epsilon_{jkl}x_k v_l)$	$m_{ij} + \epsilon_{jkl}x_k\sigma_{il}$

The domain may have its own velocity, $x_i^{\bullet} = w_i$, independent on the velocity of the fluid particle, v_i. For a discussion of the balance equations in a control volume with the domain velocity, we refer to [55,56]. This domain velocity can be chosen arbitrarily without affecting the underlying physics. Herein we fix the domain by setting $w_i = 0$. We assume that initially the fluid rests, $v_i(\boldsymbol{x}, t = 0) = 0$, and it is a homogeneous material, $\rho(\boldsymbol{x}, t = 0) =$ const. Moreover, we assume that the flow is incompressible, i.e., the mass density remains constant in time, $\partial\rho/\partial t = 0$. After utilizing the Gauss–Ostrogradskiy theorem, we obtain

$$\int_\Omega \rho \frac{\partial v_i}{\partial x_i} \, \mathrm{dv} = 0 \,, \tag{5a}$$

$$\int_\Omega \left(\rho \frac{\partial v_j}{\partial t} - \rho g_j + \rho \frac{\partial v_i v_j}{\partial x_i} - \frac{\partial \sigma_{ij}}{\partial x_i} \right) \mathrm{dv} = 0 \,, \tag{5b}$$

$$\int_\Omega \left(\rho \frac{\partial s_j}{\partial t} + \rho \epsilon_{jkl} x_k \frac{\partial v_l}{\partial t} - \rho \ell_j - \rho \epsilon_{jki} x_k g_i - \frac{\partial}{\partial x_i} \left(-v_i \rho (s_j + \epsilon_{jkl} x_k v_l) + m_{ij} + \epsilon_{jkl} x_k \sigma_{il} \right) \right) \mathrm{dv} = 0 \,, \tag{5c}$$

where the spin density, s_i, its flux term (couple stress), m_{ij}, and its supply term, ℓ_i, they all vanish for a non-polar medium like water (furnishing a symmetric stress). Then the angular momentum is identical to the moment of (linear) momentum,

$$\int_\Omega \epsilon_{jkl} x_k \left(\rho \frac{\partial v_l}{\partial t} - \rho g_l + \rho \frac{\partial v_i v_l}{\partial x_i} - \frac{\partial \sigma_{il}}{\partial x_i} \right) \mathrm{dv} = 0 \,, \tag{6a}$$

$$\int_\Omega \left(\rho \frac{\partial v_l}{\partial t} - \rho g_l + \rho \frac{\partial v_i v_l}{\partial x_i} - \frac{\partial \sigma_{il}}{\partial x_i} \right) \mathrm{dv} = 0 \,, \tag{6b}$$

hence, in analytical mechanics, we may neglect it. However, we observe this term as being important for resolving numerical challenges.

We multiply the latter equations by an arbitrary test function with the same rank of the integrand, i.e., Equation (5a) by a scalar, Equation (5b) by a vector, and Equation (6b) by a vector. Having an arbitrary test function ensures that the global condition holds locally as well. According to the Galerkin approach, we will choose the test functions from the same space as the unknowns, v_i and p. Therefore, it is natural to use δp and δv_i as possible test functions. As we need another vector as well, we may choose δv_i or even construct a vector by using gradient of δp. By utilizing the latter, we affect the velocity distribution by the pressure gradient leading to an additional enforcement of the correct pressure distribution. We know that several numerical problems start by computing the pressure distribution wrongly and we circumvent such numerical instabilities by choosing $\partial\delta p/\partial x_l$ as the test function for Equation (6b)—this particular choice is one of the key contributions of this work providing stability and robustness to the computational method. A possible justification of this observation relies on the restriction about the gradient of pressure, which may be interpreted as the missing condition for the necessary numerical smoothness for pressure related to the LBB condition; however, we enforce it by using an additional integral form instead of changing the order of shape functions. An analogous

term multiplied by the mesh size is added in PSPG for the sake of a pressure stabilization. Herein we use it in equal manner for every finite element independent of their size. Therefore, this formulation is not called a stabilization since the same weak form is evaluated in each node with no dependence on the mesh size.

We begin utilizing the discrete representations of continuous fields; but we omit a clear distinction in the notation since we never use continuous and discrete functions together. We underline that the choice of the scalar test function is critical and we suggest to use

$$\int_\Omega \rho \frac{\partial v_i}{\partial x_i} \delta p \, \mathrm{dv} = 0 \,, \tag{7a}$$

$$\int_\Omega \left(\rho \frac{\partial v_j}{\partial t} - \rho g_j + \rho \frac{\partial v_i v_j}{\partial x_i} - \frac{\partial \sigma_{ij}}{\partial x_i} \right) \delta v_j \, \mathrm{dv} = 0 \,, \tag{7b}$$

$$\int_\Omega \left(\rho \frac{\partial v_j}{\partial t} - \rho g_j + \rho \frac{\partial v_i v_j}{\partial x_i} - \frac{\partial \sigma_{ij}}{\partial x_i} \right) \frac{\partial \delta p}{\partial x_j} \, \mathrm{dv} = 0 \,. \tag{7c}$$

We refrain ourselves from inserting the mass balance, $\partial v_i / \partial x_i = 0$, into the latter formulation. In many implementations in the literature, as velocity divergence has to vanish (for incompressibility), the term $\partial v_i / \partial x_i = 0$ is inserted in Equation (7b). Since this condition is tested by δp, we cannot expect that it is fulfilled for the velocity distribution, thus, we choose to enforce it by testing with δv_j in Equation (7b). We stress that this particular choice improves the robustness of the numerical implementation.

We solve the transient integral forms in discrete time slices with a time step Δt, this discretization in time is established by using Euler backwards method resulting in

$$\frac{\partial v_i}{\partial t} = \frac{v_i - v_i^0}{\Delta t} \,, \tag{8}$$

where v_i^0 indicates the value from the last time step. This implicit method is stable (for real valued problems) and easy to implement. We emphasize that the implementation is an implicit method since we evaluate the derivative at the current time. Simply by using a Taylor series around the current time,

$$v_i(t - \Delta t) = v_i(t) - \Delta t \frac{\partial v_i}{\partial t} + O(\Delta t^2) \,, \tag{9}$$

and truncating after the linear term in Δt, we immediately obtain Equation (8). As we neglect quadratic time steps, we have to choose small time steps in the simulation in order to increase the accuracy of the time discretization.

3. Generating the Weak Form

Integral forms in Equation (7) will be rewritten in the same unit such that we can sum them up. First, we divide Equation (7a) by the mass density ρ and bring it to the unit of power. Second, we undertake no changes in Equation (7b) as it is already in the unit of power. Third, we divide Equation (7c) by the mass density and multiply it by the time step Δt and bring it also to the unit of power. The choice of the unit, herein the unit of power, is arbitrary. Often the formulation is employed in a dimensionless form generating the well-known Reynolds number.

We rewrite the constitutive equation, $\sigma_{ij} = -p \delta_{ij} + \tau_{ij}$, by combining all terms with d_{ij} into τ_{ij}. The symmetric part of the velocity gradient, d_{ij}, thus the term τ_{ij} already includes velocity's first derivative in space. As a consequence of the first derivative, we need to have a representation of the velocity function, which is at least C^1 continuous. In the integral forms, we observe another space derivative in τ_{ij} leading to the restriction that a velocity approximation belonging to the class C^2 has to be implemented. This condition will be weakened by integrating by parts. We emphasize that the integration by parts is applied only on the terms having (at least) second derivative of the

unknown in order to "shift" one differentiation to the test function. This strategy is not conventional. Often, integration by parts is used to all flux terms. We observe numerical problems by using an integration by parts employed to all flux terms. We suggest to integrate by parts only if the term consists of a second derivative of the unknown. This approach is another key contribution of the work. After bringing to the same unit and integrating by parts where necessary, by using the usual comma notation for space derivative $()_{,i} = \partial()/\partial x_i$, we obtain the following weak forms:

$$F_1 = \int_\Omega v_{i,i}\delta p \,\mathrm{dv}\,,$$

$$F_2 = \int_\Omega \left(\rho \frac{v_j - v_j^0}{\Delta t}\delta v_j - \rho g_j\delta v_j + \rho(v_iv_j)_{,i}\delta v_j + p_{,i}\delta v_i + \tau_{ij}\delta v_{j,i} \right) \mathrm{dv} - \int_{\partial\Omega} n_i\tau_{ij}\delta v_j \,\mathrm{d}a\,, \tag{10}$$

$$F_3 = \int_\Omega \left((v_j - v_j^0) - \Delta t g_j + \Delta t(v_iv_j)_{,i} - \frac{\Delta t}{\rho}(-p\delta_{ij} + \tau_{ij})_{,i} \right)\delta p_{,j} \,\mathrm{dv}\,.$$

Summing them reads the weak form to be solved,

$$F\big|_\Omega = F_1 + F_2 + F_3\,, \tag{11}$$

for one finite element. It becomes zero by inserting the correct pressure and velocity distribution. A control volume is decomposed into several elements, by assembling the weak forms of each finite element, Form $= \sum_e F\big|_\Omega$, the weak form for the whole control volume is acquired. This assembling generate on the element boundaries the following term:

$$[\![n_i\tau_{ij}]\!] = [\![n_i(\sigma_{ij} + p\delta_{ij})]\!]\,, \tag{12}$$

with jump brackets $[\![(\cdot)]\!]$ indicating the difference between the values of the quantity computed in adjacent elements. We use continuous form elements for pressure and velocity such that $[\![n_ip]\!] = 0$, moreover, we enforce $[\![n_i\sigma_{ij}]\!] = 0$ relying on the balance of linear momentum on singular surfaces. Therefore, the integral term along the element boundaries vanish within the control volume. On the boundaries of the control volume either velocity or pressure is given. For the parts where velocity is given, we choose $\delta v_i = 0$ such that the boundary term vanishes. For the boundary parts, for example where fluid enters or leaves the domain with a prescribed pressure, p, against the plane outward direction, $t_i = -pn_i$, we obtain $n_i\tau_{ij} = n_i(\sigma_{ij} + p\delta_{ij}) = t_j + n_jp = 0$. Therefore, all boundary terms vanish in the weak form for an incompressible flow.

4. Algorithm and Computation

Continuous finite elements are used for all simulations. In three-dimensional space (3D) we use tetrahedrons as elements and in two-dimensional space (2D) we use triangles as elements—both with linear form functions, i.e., form functions of degree $n = 1$. The primitive variables are pressure and velocity; they are represented with corresponding nodal values interpolated using the form functions. Concretely, 4 primitive variables $\boldsymbol{P} = \{p, v_1, v_2, v_3\}$ in three-dimensional space belong to

$$\mathcal{V} = \left\{ \boldsymbol{P} \in [\mathcal{H}^n(\Omega)]^4 : \boldsymbol{P}|_{\partial\Omega} = \text{given} \right\}\,, \tag{13}$$

where $[\mathcal{H}^n]^4$ is a 4-dimensional Hilbert space of class C^n as defined in [57] with additional differentiability properties such that it is called a Sobolev space. The test functions, $\delta\boldsymbol{P} = \{\delta p, \delta v_1, \delta v_2, \delta v_3\}$, stem from the same space

$$\hat{\mathcal{V}} = \left\{ \delta\boldsymbol{P} \in [\mathcal{H}^n(\Omega)]^4 : \delta\boldsymbol{P}|_{\partial\Omega} = \text{given} \right\}\,, \tag{14}$$

which is the Galerkin approach. We use $n = 1$ for all simulations, in other words, we use the same linear form functions for pressure as well as for velocity. This choice is risky since we fail to fulfill the aforementioned LBB conditions. Despite this fact, at least for the demonstrated applications,

no spurious oscillations occur owing to the additional governing equation restricting the pressure gradient as well as the careful choice of terms for integrating by parts. This outcome may be justified by the additional integral term assuring that the pressure gradient is correctly computed. We intend to construct a numerical scheme with linear form functions for both pressure and velocity such that a generalization to multiphysics becomes straightforward.

The weak form is nonlinear and coupled such that we need to linearize and solve it monolithically. For the linearization, we follow the ideas in ([58], Part I, Section 2.2.3) and perform an abstract linearization using Newton's method at the partial differential level. The functional Form= $F(\boldsymbol{P}, \delta \boldsymbol{P})$ is an integral of the function depending on the primitive variables $\boldsymbol{P}$ and their variations (test functions) $\delta \boldsymbol{P}$. We know the correct values of $\boldsymbol{P}$ at $t = 0$. The weak form is initially zero—we obtained it by subtracting left-hand sides from right-hand sides in the balance equations. We search for $\boldsymbol{P}(t + \Delta t)$ at the next time step, $\Delta t + t$, by using the known values $\boldsymbol{P}(t)$. By rewriting the unknowns $\boldsymbol{P}(t + \Delta t)$ in terms of the known values

$$\boldsymbol{P}(t + \Delta t) = \boldsymbol{P}(t) + \Delta \boldsymbol{P}(t) \,, \tag{15}$$

we redefine the objective to searching for $\Delta \boldsymbol{P}(t)$ instead of $\boldsymbol{P}(t + \Delta t)$. If Δt is chosen sufficiently small, then the solution is near to the known solution such that $\Delta \boldsymbol{P}(t)$ is small. This condition leads to a Taylor expansion around the known values, $\boldsymbol{P}(t)$, up to the (polynomial) order one

$$F(\boldsymbol{P} + \Delta \boldsymbol{P}, \delta \boldsymbol{P}) = F(\boldsymbol{P}, \delta \boldsymbol{P}) + \nabla_{\boldsymbol{P}} F(\boldsymbol{P}, \delta \boldsymbol{P}) \cdot \Delta \boldsymbol{P} \,, \tag{16}$$

where we omit the time argument for the sake of clarity in notation. The expansion is linear in $\Delta \boldsymbol{P}$, hence we need to construct a linear in $\Delta \boldsymbol{P}$ differentiation operator, $\nabla_{\boldsymbol{P}}$, which is established by the so-called Gateaux derivative:

$$\nabla_{\boldsymbol{P}} F(\boldsymbol{P}, \delta \boldsymbol{P}) \cdot \Delta \boldsymbol{P} = \lim_{\epsilon \to 0} \frac{\mathrm{d}}{\mathrm{d}\epsilon} F(\boldsymbol{P} + \epsilon \Delta \boldsymbol{P}, \delta \boldsymbol{P}) \,, \tag{17}$$

where ϵ is an arbitrary parameter. Since we first differentiate in ϵ and then set the parameter ϵ equal to zero, only terms of order one in $\Delta \boldsymbol{P}$ remain in the solution. Hence, we can solve in $\Delta \boldsymbol{P}$ and update the solution in an iterative manner, $\boldsymbol{P} := \boldsymbol{P} + \Delta \boldsymbol{P}$, where ":=" is an assign operator in computational algebra. Here is the ultimate algorithm:

$$\begin{aligned} &\text{while } |\Delta \boldsymbol{P}| > \text{TOL.} \\ &\quad \text{solve } \Delta \boldsymbol{P}, \text{ where } F(\boldsymbol{P}, \delta \boldsymbol{P}) + J(\boldsymbol{P}, \delta \boldsymbol{P}) \cdot \Delta \boldsymbol{P} = 0 \\ &\quad \boldsymbol{P} := \boldsymbol{P} + \Delta \boldsymbol{P} \end{aligned} \tag{18}$$

The so-called Jacobian, $J = \nabla_{\boldsymbol{P}} F$ is computed automatically by means of symbolic differentiation implemented under the name SyFi within the FEniCS project, see [59,60]. This automatic linearization procedure allows us to use any nonlinear constitutive equation in the code. Herein we use a linear constitutive equation in order to achieve a comparison with closed-form solutions. For higher Reynolds numbers, the procedure may lead to a slow or even non-convergence. This numerical problem is caused by the linearization itself as discussed in ([61], Section 7.2). As we have expanded with a linear Taylor series, the initial guess of the Newton–Raphson algorithm affects the convergence greatly. A better option is to use the Picard (fix point) iteration with the same weak formulation as introduced herein.

The geometry is constructed in Salome (Salome, the open-source integration platform for numerical simulation, http://www.salome-platform.org) by using NetGen algorithms (Netgen Mesh Generator, https://sourceforge.net/projects/netgen-mesher/) for the triangulation. Then the mesh is transformed as explained in ([62], Appendix A.3) and implemented in a Python code using packages developed by the FEniCS project, which is wrapped in C++ and solved in a Linux machine running Ubuntu (Ubuntu, open source software operating system, https://www.ubuntu.com/).

5. Comparative Analysis

The suggested weak form is implemented and solved for various problems. First, we examine the accuracy and convergence behavior of 3D computation by comparing numerical results to semi-analytical closed-form solutions for simple geometries, we call them analytical solutions. They are all well-known and can be found in different textbooks, for example see [63]. This analysis is of importance to present the local monotonous convergence that is the important and prominent feature of the FEM. For a flow problem, we increase the accuracy at every point by decreasing the mesh size. Second, we present benchmarking problems in 2D in order to verify the robustness of the method.

5.1. Steady State Hagen–Poiseuille Flow

Consider a laminar flow in an infinite pipe as a result of the given pressure difference. This configuration is called the Hagen–Poiseuille flow and it has a steady-state solution obtained in cylindrical coordinates, r, θ, z, under the assumption that the flow is only along the pipe. The pipe is oriented along z, which is set as the axis of the pipe. No-slip condition, $v_i = 0$, is applied on the outer walls, $r = a$. The steady-state solution reads

$$v_i = \begin{pmatrix} 0 \\ 0 \\ -\dfrac{\mathrm{d}p}{\mathrm{d}z}\dfrac{a^2}{4\mu}\left(1 - \dfrac{r^2}{a^2}\right) \end{pmatrix} , \tag{19}$$

for an incompressible flow of a linear viscous fluid like water of viscosity μ. We use this solution for an analysis of convergence and accuracy. A three-dimensional pipe of length ℓ is constructed and on the inlet and outlet, the pressure is given as Dirichlet boundary conditions, $p(z = 0) = p_{\text{in}}$ and $p(z = \ell) = p_{\text{out}}$. The flow is driven by the pressure difference such that the gravity is neglected. Moreover, we are interested in the steady-state where the inertial term vanishes. In order to mimic the infinite pipe, we set the radial and circumferential velocities zero on the inlet and outlet. FEM computation is realized in Cartesian coordinates, so we basically allow v_z on inflow plane, $x = 0$, and outflow plane, $x = \ell$, by setting $v_x = v_y = 0$ as Dirichlet conditions.

The pressure distribution is expected to be linear along z, hence, for the analytical solution $\mathrm{d}p/\mathrm{d}z = (p_{\text{out}} - p_{\text{in}})/\ell$. For a better comparison we use the diameter D as the characteristic length and half of the maximum velocity, $v_z^{\text{M}} = v_z(r = 0)$, as the characteristic velocity for calculating the Reynolds number:

$$Re = \frac{v_z^{\text{M}} D \rho}{2\mu} . \tag{20}$$

For a small pipe of an inch long and a quarter inch wide, we construct the mesh by using a global element length h. We use SI units such that $\ell = 25.4\,\text{mm}$ and $D = 6.35\,\text{mm}$ and water as the fluid with

$$\rho = 998.2 \times 10^{-6}\,\text{g/mm}^3 , \quad \mu = 1001.6 \times 10^{-6}\,\text{Pa\,s} , \quad \lambda = 0.6\,\text{Pa\,s} . \tag{21}$$

Especially, the choice of λ is of importance since this parameter is not measured directly. Although λ is a physical and measurable quantity, for a conventional pressure amount, the incompressibility makes the measurement of λ very challenging. In the Navier–Stokes Equation (2), the term λd_{kk} vanishes for the correct velocity solution. Therefore, for the numerical sense, λ has to be great enough that d_{kk} is enforced to vanish. Hence, we choose λ multiple times greater than μ, in reality (for compressible flows) λ and μ are independent parameters. Interestingly, we have observed that choosing λ greater than suggested value slows down the convergence. A remedy to this relies on the aforementioned fixed point iteration; but we use herein the Newton–Raphson iteration. By doubling λ, the same accuracy of the solution is obtained with more degrees of freedom (DOF). Its value does not change the convergence behavior, as long as it is great enough. In order to determine the value of λ,

we simply decreased until the maximum *Re* was achieved. Less than the used value leads to numerical problems in the Newton–Raphson iterations.

By using a standard convergence analysis, we compare three different meshes starting with a global edge length of tetrahedrons, $h = 0.6$ mm, then reducing it by half. For every simulation, a new mesh is generated such that the number of nodes fail to increase exactly by 2^3 times in 3D. We use an unstructured mesh and the mesh quality is nearly identical because of using the same algorithm on the relatively simple geometry. The expected monotonic convergence has been attained as seen in Figure 1 for 2 different Reynolds numbers.

(**a**)

(**b**)
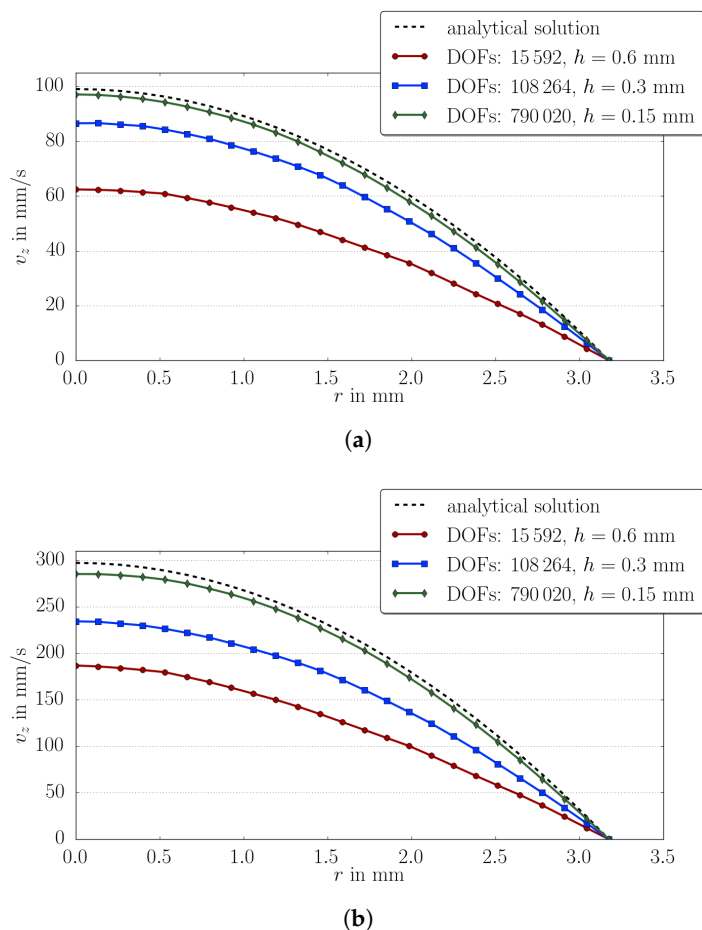
Figure 1. Three-dimensional (3D) computation of steady state solution and its comparison to the analytical solution in a pipe for two different Reynolds numbers. (**a**) $Re = 313$; (**b**) $Re = 940$.

Two important facts need to be underlined. First, the parabolic distribution of the velocity along the diameter is achieved even with a coarse mesh. Second, the relative error is 1.1% at $r = 0$ for a low Reynolds number and 4.1% at $r = 0$ for a high Reynolds number by using mumps direct solver.

5.2. Starting Hagen–Poiseuille Flow

In order to test the accuracy in the transient simulation, we use the same configuration and solve in every time step. Since we have obtained the expected convergence in space discretization for the steady-state solution, we expect to have a monotonic convergence in time discretization, too. Therefore, we solve the same example with different time steps from $t = 0$ to $t = 10$ s with the initially

applied pressure difference. For this case there is a closed-form solution under the same assumptions as before,

$$
v_i = \begin{pmatrix} 0 \\ 0 \\ -\dfrac{\mathrm{d}p}{\mathrm{d}z}\dfrac{a^2}{4\mu}\left(1 - \dfrac{r^2}{a^2} - \sum_{n=1}^{N} 8\dfrac{J_0(\Lambda_n \frac{r}{a})}{\Lambda_n^3 J_1(\Lambda_n)}\exp(-\Lambda_n^2 \tau)\right) \end{pmatrix},
\tag{22}
$$

with $\tau = t\mu/(\rho a^2)$ and Bessel functions (of the first kind) J_0 and J_1 with Λ_n being the roots of J_0. We compute the (semi-)analytical solution by using SciPy packages for Bessel functions as well as its roots and choose $N = 50$. By choosing the best mesh obtained from the convergence analysis in the steady-state case, namely the mesh with $h = 0.15\,\mathrm{mm}$, we compute the solution in time by using different time steps. We present in Figure 2 the maximum value $v_z(r = 0, t)$ over time for different time steps showing the expected convergence with decreasing time steps as well as the distribution at different time instants for the smallest time step.
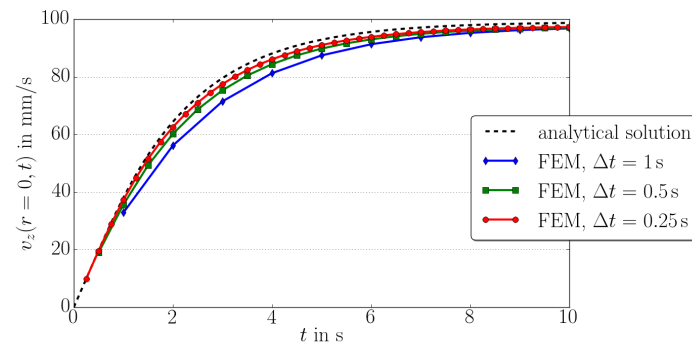
(**a**)
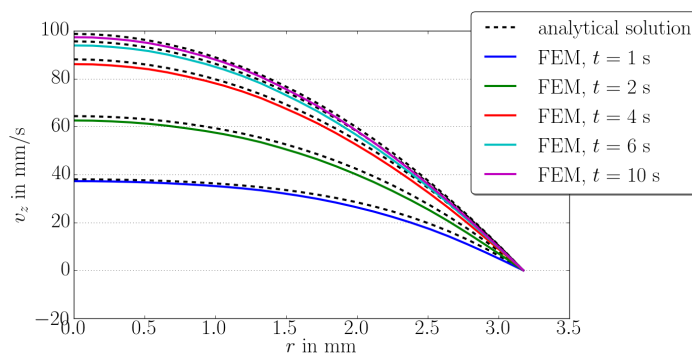
(**b**)

Figure 2. Three-dimensional computation of transient solution and its comparison to the analytical solution in a pipe for $Re = 313$ in (**a**) $\Delta t = \{1, 0.5, 0.25\}$ s; (**b**) $\Delta t = 0.25$ s.

In addition to the convergence in time, the relative error remains unchanged over time. We conclude that the suggested formalism is capable of simulating a simple, laminar, three-dimensional flow of a linear viscous fluid accurately under a monotonic loading. The velocity and pressure distributions show no artifacts or mesh dependency as presented in Figure 3.

We have used mumps direct solver for achieving the highest accuracy in the numerical solution. We emphasize that the pressure difference is applied instantaneously, which is numerically challenging. For transient loading scenarios, there are various assumptions used for obtaining a closed-form solution such that we omit to examine further cases. Based on the presented examples, we conclude that the approach delivers an accurate solution.
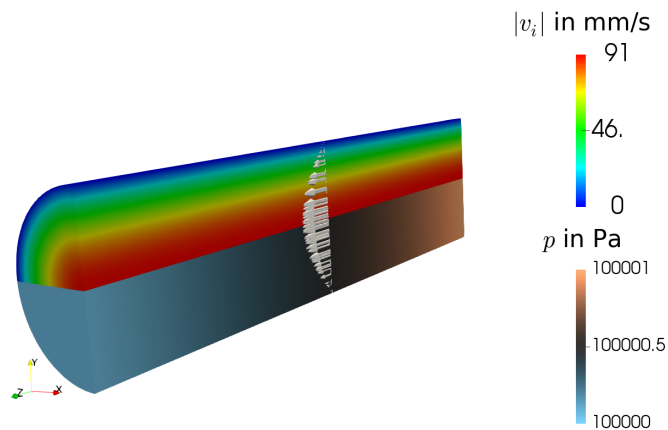
Figure 3. Velocity solution (direction as scaled arrows and magnitude as colors) and pressure solution (as colors) of the transient pipe flow at $t = 5\,\text{s}$, shown on the half of the pipe (upper part: velocity and lower part: pressure) for $Re = 313$.

5.3. Lid Driven Cavity

Especially in higher Reynolds numbers, the stability of the numerical method becomes critical. Hence, we examine a prominent benchmark problem by following [64], where the numerical solution is obtained by using a different numerical solution strategy. We model a rectangle with a given shearing velocity on top, called lid driven cavity. As fluid we use water with parameters as in Equation (21); the solution is shown in normalized units for the sake of a direct comparison to [64] that we use as the reference solution. A transient computation has been realized, where the shear velocity on the lid is slowly increased; so we may assume that the solution in each time step is tantamount to the steady state solution, for which the results are compiled in ([64], Tables I and II). As an example we demonstrate in Figure 4 the solution for $Re = 5000$.

As is seen in Figure 4, the velocity distributions along horizontal and vertical axes may be used for comparing results adequately as presented in Figure 5.
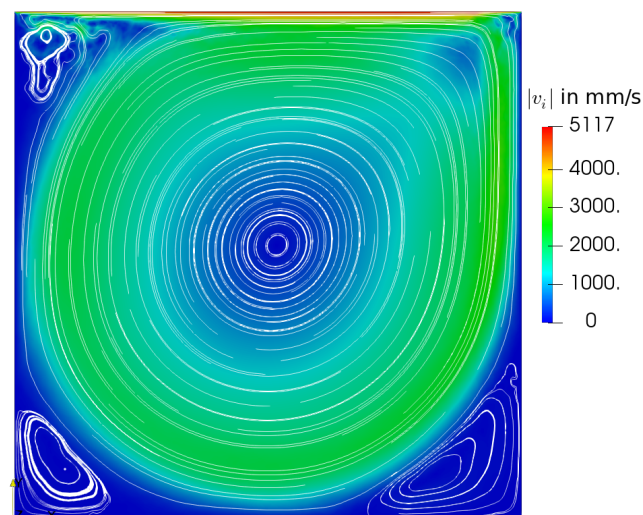

Figure 4. Velocity distribution and the corresponding streamline for $Re = 5000$ simulated with parameters of water from Equation (21).

Results from the reference solution and results obtained by the proposed method show no significant difference up to $Re = 5000$. This good agreement fails to be the case in the higher Reynolds number, we skip an analysis about the strength and weak points of the method used for the reference solution; there is no consensus on the correct numerical results for larger Reynolds number, for an elaborate discussion about different results in the literature, we refer to [65] and reference therein. We emphasize that the proposed method is delivering results for this application showing instabilities generating vortices at the edges. No numerical problems emerge for the transient solution even in high Reynolds numbers. Typical numerical stability problems occurring in the finite element method are circumvented by using the aforementioned weak form as well as keeping the term with the volume viscosity in the formulation. For each application, the volume viscosity affects the numerical stability. We have chosen $\lambda = 10\,\mathrm{Pa\,s}$. Considering the analysis before, we emphasize that results indeed show the same local and monotonous convergence properties. By choosing λ bigger, we slow down the convergence such that a dense mesh is used for acquiring reliable results.
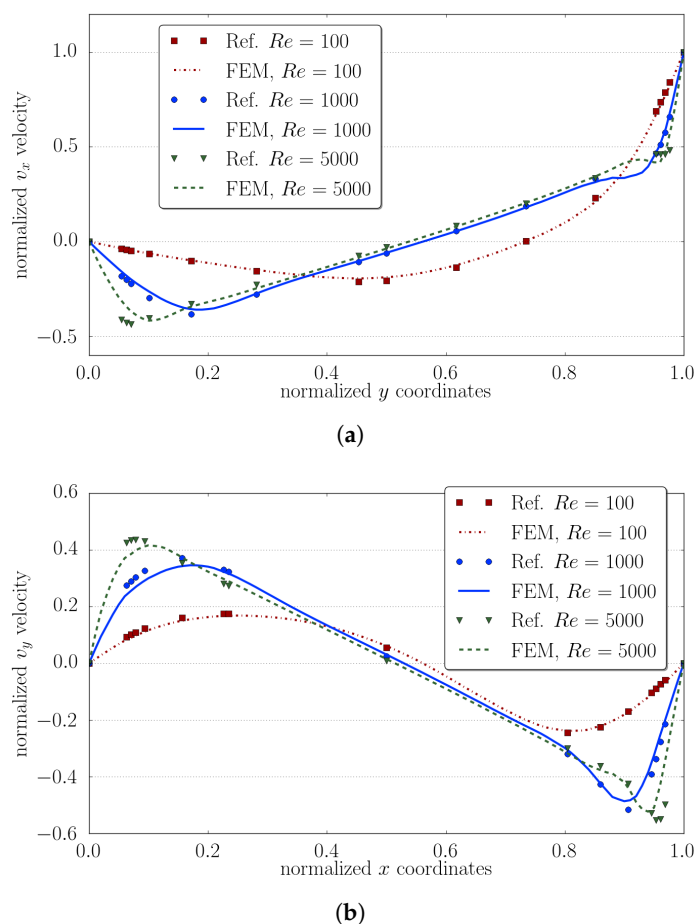

(a)

(b)

Figure 5. Velocity distribution in the lid driven cavity benchmark problem. (**a**) Horizontal velocities along the vertical axis through the center are divided by the (given) lid velocity. (**b**) Vertical velocities along horizontal axis through the center are normalized by the lid velocity. For three different Reynolds numbers, the results are compared to the reference solution (denoted as "Ref.") taken from ([64], Tables I and II).

5.4. Karman Vortex Street

Flow past an obstacle is one of the heavily studied phenomena in the literature. Especially the instabilities in wakes behind an obstacle—realized as a bluff in the middle of the flow—has obtained much attention, we refer to [66] for a detailed historical review. By precisely setting up an arrangement as analyzed in [67], it is possible to obtain a vortex street configuration depending on the force

dragging and lifting the obstacle. For example, such a problem is solved in ([68], Section 3.4) by using the so-called incremental pressure correction scheme (IPCS). This splitting method is powerful for isothermal case, but difficult to apply for non-isothermal cases. In the approach presented herein, we solve pressure and velocity at once by using the same order of form functions. As suggested in [69], we implement a benchmarking problem for creating a laminar Karman vortex street. This benchmark problem is used to test a new method or code, it is accurately computed in [70], by using Taylor–Hood elements as quadratic form functions for velocity and linear functions for pressure, without stabilization, which we use as the reference solution. For a qualitative comparison with presented solutions in ([70], Figure 2), at the same time instants, we visualize velocity distributions in Figure 6.

A rectangle of $\ell = 2.2\,\text{m} \times H = 0.41\,\text{m}$ has a circular obstacle of diameter $0.1\,\text{m}$ with its center located at $(0.2, 0.2)$. On the upper and lower walls as well as on the obstacle, fluid adheres to the fixed walls. Fluid is pumped in from the left hand side and flows around the obstacle, which is placed vertically not at the center such that fluid flow gets perturbed differently on the upper and lower sides of the obstacle. Hence, vortex shedding behind the obstacle generates wakes. On the right end, the Dirichlet boundary condition is set for the pressure $p = p_{\text{ref.}}$. On the left, where fluid enters the domain, the velocity profile:

$$v_i = \begin{pmatrix} v_{\max} \sin\left(\pi\dfrac{t}{8}\right) \dfrac{4y(H - y)}{H^2} \\ 0 \end{pmatrix} ,$$ (23)

is applied with $v_{\max} = 1.5\,\text{m/s}$ leading to the Reynolds number $Re = 100$ for fluid of $\rho = 1\,\text{kg/m}^3$ and $\mu = 0.001\,\text{Pa}\,\text{s}$. Of course, such a fluid performing an incompressible flow is difficult to find in reality; however, the benchmark problem needs to be seen as a computationally challenging problem demonstrating the strength and robustness of the proposed code since a computation in such a low kinematic viscosity, $\nu = \mu/\rho$, may cause numerical problems. By using $\lambda = 0.01\,\text{Pa}$ and $\Delta t = 1/1600\,\text{s}$, we have successfully computed up to $8\,\text{s}$ such that the velocity increases up to $4\,\text{s}$ and then decreases (sinusoidally, in a half period).

The vortex shedding behind the obstacle occurs because of the boundary layer separation. This separation is due to the changing pressure gradients on upper and lower parts of the obstacle. There are two stagnation points visible in front of and behind the obstacle. Along the boundary of the obstacle, pressure gradient changes its sign at the top point. This change in pressure gradient resolves a wake behind the obstacle, as the pressure is lower it leads to a flow separation. We have modeled this phenomenon with the same element type and size around the boundary, in other words, without any special boundary layer modeling. This benchmark problem is chosen for proving the robustness of the proposed method. In Figure 7, a quantitative analysis is shown between the reference solution and the FEM computation by using the drag and lift coefficients:

$$\begin{aligned} c_D &= \frac{-2}{v_{\text{mean}}^2 \ell} \int_\Omega \left(\frac{\mu}{\rho} v_{i,j} v_{i,j}^D + v_i v_{j,i} v_j^D - p v_{i,i}^D \right) \mathrm{d}v \, , \\ c_L &= \frac{-2}{v_{\text{mean}}^2 \ell} \int_\Omega \left(\frac{\mu}{\rho} v_{i,j} v_{i,j}^L + v_i v_{j,i} v_j^L - p v_{i,i}^L \right) \mathrm{d}v \end{aligned}$$ (24)

where $\boldsymbol{v}^D = (1,0)$ and $\boldsymbol{v}^L = (0,1)$ on the obstacle leading to the forces along the flow direction (drag) and perpendicular to the flow direction (lift). The forces are normalized by the mean velocity, $v_{\text{mean}} = 1\,\text{mm/s}$, and characteristic length (diameter of the obstacle), $\ell = 0.1\,\text{mm}$. The reference solution and the herein proposed method, both solve the same equations by the finite element method. As we have shown the convergence in the last section, we demonstrate not the quantitative agreement but the reliability of the proposed method. Even with a low amount of degrees of freedom (DOFs), a numerical solution is possible by using the same type of form functions for pressure and velocity without stabilization. In order to examine the capability of the method, we choose the model with DOFs = 67,035 and change the viscosity and inlet velocity to obtain a flow with $Re = 1000$ reached in

1 s and held 100 s long. Again we managed to compute without numerical problems the analogous Karman vortex street with a different formation as depicted in Figure 8.
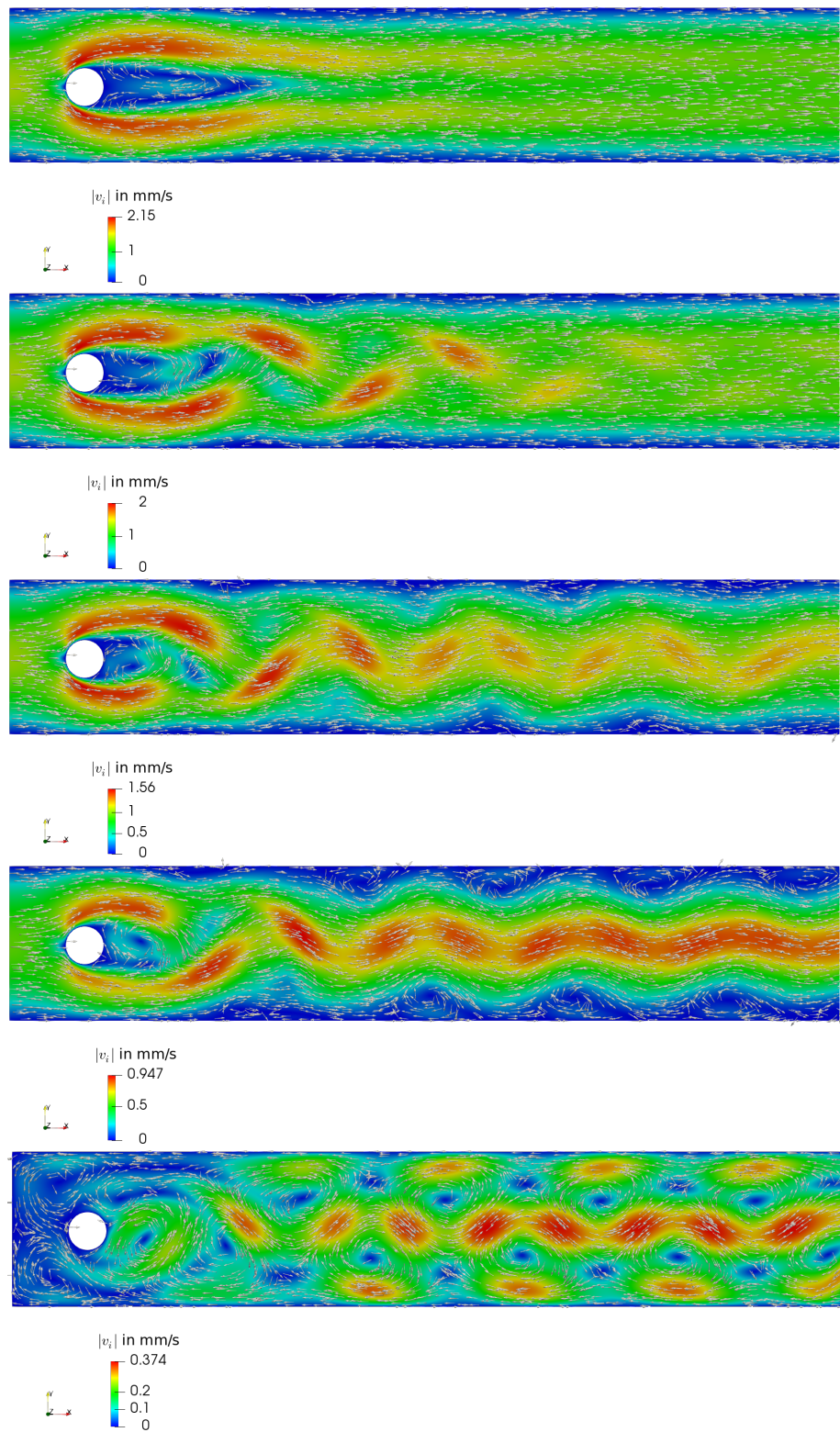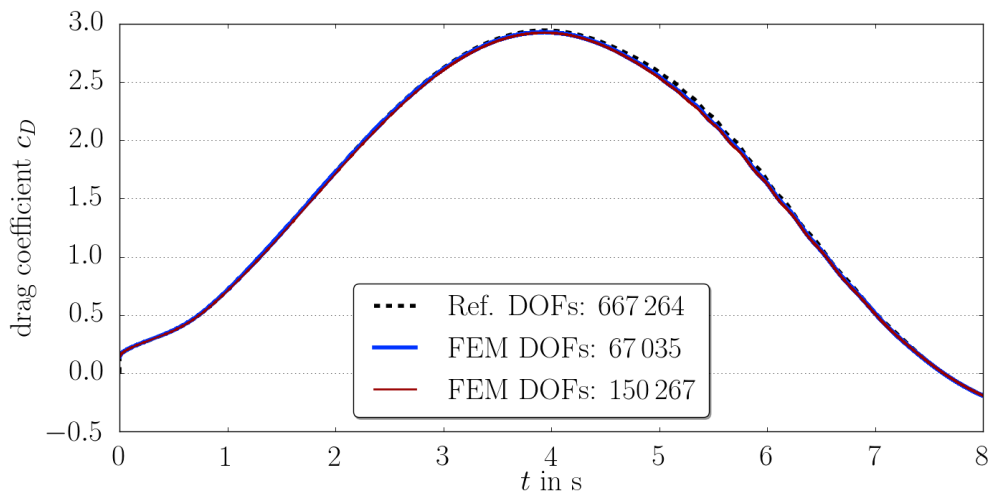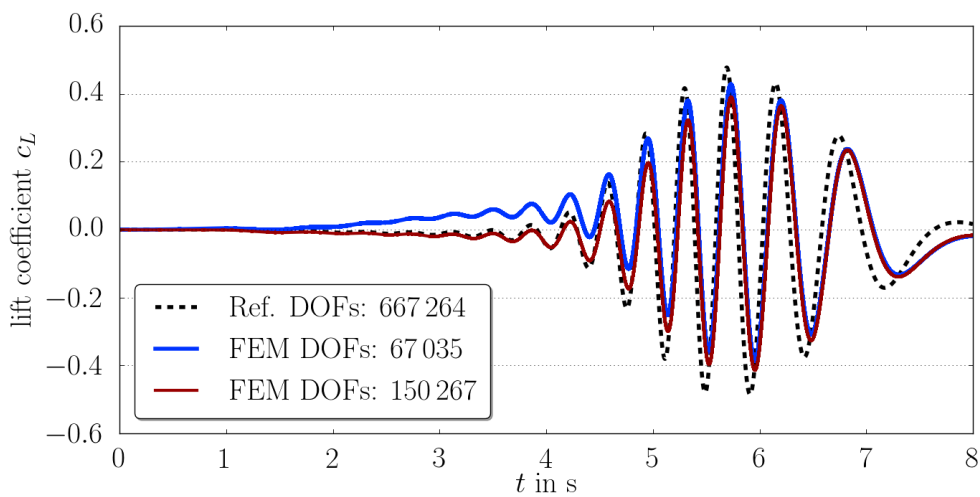

Figure 6. Two-dimensional (2D) computation of the vortex shedding with the configuration as in [70], velocity distribution is presented with colors denoting to magnitude and with arrows showing their direction at times 4, 5, 6, 7, 8 s.

(a)

(b)

Figure 7. Drag (**a**) and lift (**b**) coefficients caused by the force applied on the cylinder by flowing viscous fluid. Results from two meshes are shown as continuous lines labeled as finite element method (FEM) denoting the computation presented herein and the reference solution is demonstrated as dashed lines labeled as Ref. obtained from http://www.featflow.de/en/benchmarks/cfdbenchmarking/flow/dfg_benchmark3_re100.html
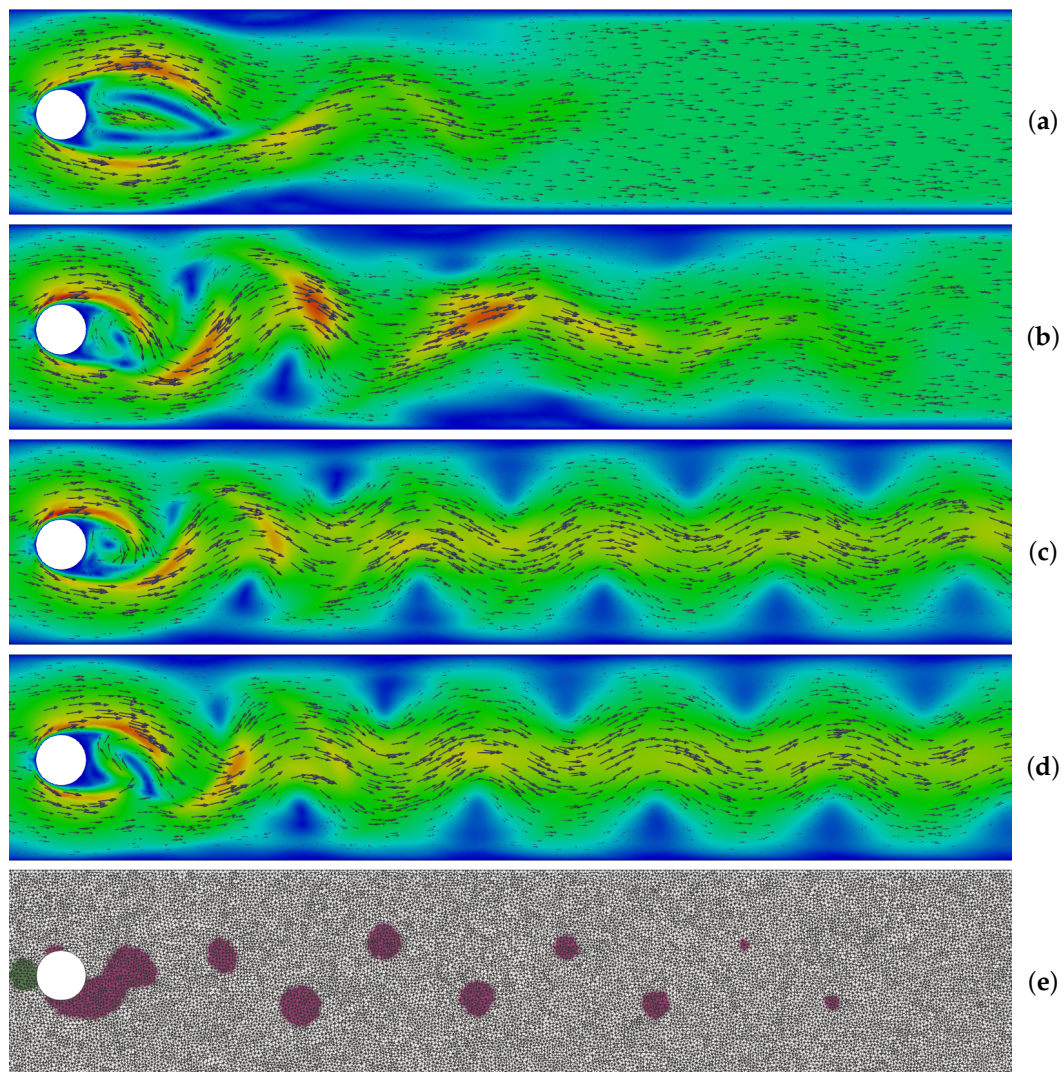
Figure 8. Two-dimensional computation of the vortex shedding with $Re = 1000$ leading to the Karman vortex street, velocity distribution is presented as arrows and their magnitude as colors. At $t = 15\,\mathrm{s}$ in (**a**) the perturbation starts invoking the first vortex, at $t = 25\,\mathrm{s}$ in (**b**) the vortex street is forming, at $t = 65\,\mathrm{s}$ in (**c**) the vortex shedding is affecting the whole domain, and at $t = 80\,\mathrm{s}$ (**d**) the periodic street is visible. Pressure distribution is shown at $t = 80\,\mathrm{s}$ in (**e**) on top of the finite elements discretization with an increased pressure before and moving decreased pressure "islands" after the cylinder.

6. Conclusions

A new method is proposed to compute fluid dynamics of isothermal and incompressible flows by means of the FEM. We have rigorously investigated the convergence and the accuracy of the proposed approach by using closed-form solutions. Convergence in space and time is difficult to achieve in FEM for flow problems. The accuracy is only possible by using extremely high degrees of freedom. With the proposed method, we have attained a good accuracy with relatively coarse meshes. Local monotonic convergence in space and time is a remarkable quality of the FEM and our method exploits this feature. We stress that this property is of paramount importance for problems where the accurate solution is not known. Based on the local monotonic convergence, a posteriori error analysis is possible. Hence, the proposed method is reliable. Moreover, we have demonstrated the robustness of the implementation in open-source packages called FEniCS by solving benchmark problems. All used codes are publicly available on the web site in [53] to be used under the GNU Public license [54].

Further research is being conducted for computing real-life problems and verification of the proposed method with the aid of experimental results.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Darrigol, O. Between hydrodynamics and elasticity theory: The first five births of the Navier-Stokes equation. *Arch. Hist. Exact Sci.* **2002**, *56*, 95–150. [CrossRef]
2. Babuska, I.; Rheinboldt, W. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.* **1978**, *15*, 736–754. [CrossRef]
3. Taylor, C.; Hood, P. A numerical solution of the Navier-Stokes equations using the finite element technique. *Comput. Fluids* **1973**, *1*, 73–100. [CrossRef]
4. Arnold, D.N.; Brezzi, F.; Fortin, M. A stable finite element for the Stokes equations. *Calcolo* **1984**, *21*, 337–344. [CrossRef]
5. Abali, B.E.; Reich, F.A. Thermodynamically consistent derivation and computation of electro-thermo-mechanical systems for solid bodies. *Comput. Methods Appl. Mech. Eng.* **2017**, *319*, 567–595. [CrossRef]
6. Book, D.L.; Boris, J.P.; Hain, K. Flux-corrected transport II: Generalizations of the method. *J. Comput. Phys.* **1975**, *18*, 248–283. [CrossRef]
7. Hughes, T.J.R.; Brooks, A. A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions: Application to the streamline-upwind procedure. *Finite Elem. Fluids* **1982**, *4*, 47–65.
8. Hughes, T.J.R.; Franca, L.P.; Balestra, M. A new finite element formulation for computational fluid dynamics: circumventing the Babuska-Brezzi condition: A stable Petrov- Galerkin formulation of the Stokes problem accommodating equal-order Interpolation. *Comput. Methods Appl. Mech. Eng.* **1986**, *59*, 85–99. [CrossRef]
9. Tezduyar, T.E.; Mittal, S.; Ray, S.; Shih, R. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity pressure elements. *Comput. Methods Appl. Mech. Eng.* **1992**, *95*, 221–242. [CrossRef]
10. Silvester, D.J.; Kechkar, N. Stabilised bilinear-constant velocity-pressure finite elements for the conjugate gradient solution of the Stokes problem. *Comput. Methods Appl. Mech. Eng.* **1990**, *79*, 71–86. [CrossRef]
11. Franca, L.P.; Farhat, C. Bubble functions prompt unusual stabilized finite element methods. *Comput. Methods Appl. Mech. Eng.* **1995**, *123*, 299–308. [CrossRef]
12. Blank, H.; Rudgyard, M.; Wathen, A. Stabilised finite element methods for steady incompressible flow. *Comput. Methods Appl. Mech. Eng.* **1999**, *174*, 91–105. [CrossRef]
13. Brezzi, F.; Fortin, M. A minimal stabilisation procedure for mixed finite element methods. *Numerische Mathematik* **2001**, *89*, 457–491. [CrossRef]
14. Gravemeier, V. The variational multiscale method for laminar and turbulent flow. *Arch. Comput. Methods Eng.* **2006**, *13*, 249. [CrossRef]
15. Morton, K.W. Finite element methods for non-self-adjoint problems. In *Topics in Numerical Analysis*; Turner, P., Ed.; Springer: Berlin/Heidelberg, Germany, 1982; Volume 965, pp. 113–148.
16. Kuzmin, D.; Turek, S. Flux correction tools for finite elements. *J. Comput. Phys.* **2002**, *175*, 525–558. [CrossRef]
17. Rannacher, R. Chapter 6. Incompressible viscous flows. In *Encyclopedia of Computational Mechanics*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2004; Volume 3, pp. 155–181.
18. Hughes, T.J.R.; Scovazzi, G.; Franca, L.P. Chapter 2. Multiscale and stabilized methods. In *Encyclopedia of Computational Mechanics*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2004; Volume 3, pp. 5–59.
19. Donea, J.; Huerta, A. *Finite Element Methods for Flow Problems*; John Wiley & Sons: Ltd.: Chichester, UK, 2005.
20. Hoffman, J.; Johnson, C. *Computational Turbulent Incompressible Flow, Applied Mathematics: Body and Soul 4*; Springer: Berlin/Heidelberg, Germany, 2007.
21. Löhner, R. *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*; John Wiley & Sons: Hoboken, NJ, USA, 2008.

22. Förster, C.; Wall, W.; Ramm, E. Stabilized finite element formulation for incompressible flow on distorted meshes. *Int. J. Numer. Methods Fluids* **2009**, *60*, 1103–1126. [CrossRef]

23. Arzani, A.; Shadden, S.C. Characterizations and correlations of wall shear stress in aneurysmal flow. *J. Biomech. Eng.* **2016**, *138*, 014503. [CrossRef] [PubMed]

24. Barth, T.; Bochev, P.; Gunzburger, M.; Shadid, J. A taxonomy of consistently stabilized finite element methods for the Stokes problem. *SIAM J. Sci. Comput.* **2004**, *25*, 1585–1607. [CrossRef]

25. Bowers, A.L.; Cousins, B.R.; Linke, A.; Rebholz, L.G. New connections between finite element formulations of the Navier–Stokes equations. *J. Comput. Phys.* **2010**, *229*, 9020–9025. [CrossRef]

26. Eggels, J.; Unger, F.; Weiss, M.; Westerweel, J.; Adrian, R.; Friedrich, R.; Nieuwstadt, F. Fully developed turbulent pipe flow: A comparison between direct numerical simulation and experiment. *J. Fluid Mech.* **1994**, *268*, 175–210. [CrossRef]

27. Hirsch, C. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*; Elsevier: New York, NY, USA, 2007.

28. Varghese, S.S.; Frankel, S.H.; Fischer, P.F. Direct numerical simulation of stenotic flows. Part 1. Steady flow. *J. Fluid Mech.* **2007**, *582*, 253–280. [CrossRef]

29. Lee, M.; Moser, R.D. Direct numerical simulation of turbulent channel flow up to Re = 5200. *J. Fluid Mech.* **2015**, *774*, 395–415. [CrossRef]

30. Tryggvason, G. 2.2 Direct Numerical Simulations of Gas–Liquid Flows. In *Multiphase Flow Handbook*; CRC Press: Boca Raton, FL, USA, 2016; p. 95.

31. Hoffman, J.; Jansson, J.; Jansson, N. FEniCS-HPC: Automated predictive high-performance finite element computing with applications in aerodynamics. In Proceedings of the International Conference on Parallel Processing and Applied Mathematics, Krakow, Poland, 6–9 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 356–365.

32. Nguyen, V.D. High-Performance Finite Element Methods: With Application to Simulation of Diffusion MRI and Vertical Axis Wind Turbine. Ph.D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2018.

33. Chorin, A.J. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comput.* **1969**, *23*, 341–353. [CrossRef]

34. Temam, R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I). *Arch. Ration. Mech. Anal.* **1969**, *32*, 135–153. [CrossRef]

35. Temam, R. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II). *Arch. Ration. Mech. Anal.* **1969**, *33*, 377–385. [CrossRef]

36. Goda, K. A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows. *J. Comput. Phys.* **1979**, *30*, 76–95. [CrossRef]

37. Bell, J.B.; Colella, P.; Glaz, H.M. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **1989**, *85*, 257–283. [CrossRef]

38. Zienkiewicz, O.C.; Codina, R. A general algorithm for compressible and incompressible flow—Part I. The split, characteristic-based scheme. *Int. J. Numer. Methods Fluids* **1995**, *20*, 869–885. [CrossRef]

39. Brown, D.L.; Cortez, R.; Minion, M.L. Accurate projection methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **2001**, *168*, 464–499. [CrossRef]

40. Anderson, J.D.; Wendt, J. *Computational Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 1995; Volume 206.

41. Jasak, H. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. PhD Thesis, University of London and Imperial College, London, UK, 1996.

42. Versteeg, H.K.; Malalasekera, W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*; Pearson Education: London, UK, 2007.

43. Wang, Z.J.; Fidkowski, K.; Abgrall, R.; Bassi, F.; Caraeni, D.; Cary, A.; Deconinck, H.; Hartmann, R.; Hillewaert, K.; Huynh, H.T.; et al. High-order CFD methods: Current status and perspective. *Int. J. Numer. Methods Fluids* **2013**, *72*, 811–845. [CrossRef]

44. Dohrmann, C.R.; Bochev, P.B. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *Int. J. Numer. Methods Fluids* **2004**, *46*, 183–201. [CrossRef]

45. He, Y.; Li, J. A stabilized finite element method based on local polynomial pressure projection for the stationary Navier–Stokes equations. *Appl. Numer. Math.* **2008**, *58*, 1503–1514. [CrossRef]

46. Li, J.; He, Y. A stabilized finite element method based on two local Gauss integrations for the Stokes equations. *J. Comput. Appl. Math.* **2008**, *214*, 58–65. [CrossRef]

47. Burman, E. Pressure projection stabilizations for Galerkin approximations of Stokes' and Darcy's problem. *Numer. Methods Partial Differ. Equ.* **2008**, *24*, 127–143. [CrossRef]

48. Olshanskii, M.A.; Rebholz, L.G. Velocity–vorticity–helicity formulation and a solver for the Navier–Stokes equations. *J. Comput. Phys.* **2010**, *229*, 4291–4303. [CrossRef]

49. Abali, B.E.; Müller, W.H.; Georgievskii, D.V. A discrete-mechanical approach for computation of three-dimensional flows. *ZAMM J. Appl. Math. Mech.* **2013**, *93*, 868–881. [CrossRef]

50. Palha, A.; Gerritsma, M. A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2D incompressible Navier–Stokes equations. *J. Comput. Phys.* **2017**, *328*, 200–220. [CrossRef]

51. Charnyi, S.; Heister, T.; Olshanskii, M.A.; Rebholz, L.G. On conservation laws of Navier–Stokes Galerkin discretizations. *J. Comput. Phys.* **2017**, *337*, 289–308. [CrossRef]

52. Fu, G.; Jin, Y.; Qiu, W. Parameter-free superconvergent H(div)-conforming HDG methods for the Brinkman equations. *IMA J. Numer. Anal.* **2018**. [CrossRef]

53. Abali, B.E. Technical University of Berlin, Institute of Mechanics, Chair of Continuum Mechanics and Material Theory, Computational Reality. 2017. Available online: http://www.lkm.tu-berlin.de/ComputationalReality/ (accessed on 7 January 2019).

54. GNU Public. Gnu General Public License. 2007. Available online: http://www.gnu.org/copyleft/gpl.html (accessed on 7 January 2019).

55. Müller, W.H.; Muschik, W. Bilanzgleichungen offener mehrkomponentiger Systeme I. Massen- und Impulsbilanzen. *J. Non-Equilib. Thermodyn.* **1983**, *8*, 29–46. [CrossRef]

56. Muschik, W.; Müller, W.H. Bilanzgleichungen offener mehrkomponentiger Systeme II. Energie-und Entropiebilanz. *J. Non-Equilib. Thermodyn.* **1983**, *8*, 47–66. [CrossRef]

57. Hilbert, D. *The Foundations of Geometry*; Translated by Townsend, E.J.; The Open Court Publishing Co.: Chicago, IL, USA, 1902.

58. Logg, A.; Mardal, K.A.; Wells, G. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 84.

59. Alnaes, M.S.; Mardal, K.A. On the Efficiency of Symbolic Computations Combined with Code Generation for Finite Element Methods. *ACM Trans. Math. Softw.* **2010**, *37*, 6. [CrossRef]

60. Alnaes, M.S.; Mardal, K.A. SyFi and SFC: Symbolic Finite Elements and Form Compilation. In *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*; Logg, A., Mardal, K.A., Wells, G.N., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Chapter 15.

61. Elman, H.; Silvester, D.; Wathen, A. Finite Element and Fast Iterative Solvers. *Numerical Mathematics and Scientific Computation*; Oxford University Press: Oxford, UK, 2005.

62. Abali, B.E. *Computational Reality, Solving Nonlinear and Coupled Problems in Continuum Mechanics*; Advanced Structured Materials; Springer: Berlin/Heidelberg, Germany, 2017.

63. Savaş, Ö. *Advanced Fluid Mechanics, ME-260 A/B Lecture Notes*; University of California at Berkeley, CA, USA, 2017.

64. Ghia, U.; Ghia, K.N.; Shin, C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.* **1982**, *48*, 387–411. [CrossRef]

65. Erturk, E.; Corke, T.C.; Gökçöl, C. Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers. *Int. J. Numer. Methods Fluids* **2005**, *48*, 747–774. [CrossRef]

66. Williamson, C.H. Vortex dynamics in the cylinder wake. *Annu. Rev. Fluid Mech.* **1996**, *28*, 477–539. [CrossRef]

67. Karman, T.V. Über den Mechanismus des Wiederstandes, den ein bewegter Körper in einer Flüssigkeit erfährt. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1912**, *1912*, 547–556.

68. Langtangen, H.P.; Logg, A. *Solving PDEs in Python: The FEniCS Tutorial I*; Springer: Berlin/Heidelberg, Germany, 2016.

69. Schäfer, M.; Turek, S.; Durst, F.; Krause, E.; Rannacher, R. Benchmark computations of laminar flow around a cylinder. In *Flow Simulation with High-Performance Computers II*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 547–566.

70. John, V. Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder. *Int. J. Numer. Methods Fluids* **2004**, *44*, 777–788. [CrossRef]