



HAL
open science

Involutive upgrades of Navier-Stokes solvers

Bijan Mohammadi, Jukka Tuomela

► **To cite this version:**

Bijan Mohammadi, Jukka Tuomela. Involutive upgrades of Navier-Stokes solvers. *International Journal of Computational Fluid Dynamics*, 2009, 23, pp.439-447. 10.1080/10618560903061337. hal-00418174

HAL Id: hal-00418174

<https://hal.science/hal-00418174v1>

Submitted on 19 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INVOLUTIVE UPGRADES OF NAVIER-STOKES SOLVERS

BIJAN MOHAMMADI AND JUKKA TUOMELA

ABSTRACT. We use ideas related to involutive completion of a system of PDEs to formulate computational problems of fluid mechanics. As for the solution of differential algebraic equations the approach requires solution of extra equations for derivative consequences. The extra calculation cost is negligible while the discrete form becomes much simpler to handle. We show that in this way we can quite easily improve the performance of existing solvers. Another interest in this paper is the derivation of special solutions of the Navier-Stokes system under incompressibility constraint in cylindrical domains.

1. INTRODUCTION

The formal theory of PDEs brings in an important concept, the *involutive form* of a given system [18, 16, 19, 17]. This is not very well known in the CFD community and we would like to bring it into light through this paper.

The bottom line with this approach is that numerics for an involutive form is simpler than for the original system because no compatibility relation is necessary between discretization spaces. This permits on one hand the use of generic commercial tools and on the other hand it makes easier to reach higher accuracy by increasing the order of finite elements for a particular variable. And this without being limited by possible compatibility issues for discretization between variables. For instance, one interesting consequence is that inf-sup or LBB condition (Ladyzhenskaya-Babuska-Brezzi [2, 3, 11]) for the Stokes problem disappears: the system stays stable even with higher order discretizations for the pressure than velocity [13].

From a mathematical point of view involutivity is important because it turns out that determining the properties of a given system is in general possible only if the system is involutive. For example some systems may not be elliptic (resp. parabolic) initially, but their involutive forms are elliptic (resp. parabolic) [8, 10]. The construction of the involutive form essentially means that one has to find all the integrability conditions of the given system. This may be easy in some cases, but in general it requires the use of symbolic computation [8, 9, 12, 17]. We will argue that the involutive form is also important from the point of view of numerical computations and illustrate this issue through examples in fluid dynamics computation.

The approach by formal theory is helpful especially in situations where the physical models have constraints or conserved quantities which make the system essentially overdetermined. This is a very frequent situation. However, usually in numerical computations one uses square models (as many equations as unknowns). But then if one “forces” the system to be square by dropping some relevant equations/constraints one may encounter great difficulties in designing appropriate numerical methods because

Key words and phrases. Navier-Stokes and Stokes solvers, involution, ellipticity and uniqueness, centrifugation, level set, LBB condition, Chorin algorithm.

the methods should then take into account the properties of the system which are only implicitly represented in the system. We propose to use the involutive form, i.e. all relevant information is explicitly available.

Our aim here is to show that our approach can also improve existing solvers with light implementation effort. This is illustrated for an existing stabilized Navier-Stokes solver based on the Chorin projection algorithm [4] with boundaries treated by a level set method. We also exhibit some exact solution of the Stokes and Navier-Stokes systems useful for the validation of numerical solvers for rotating flows in cylindrical domains.

2. INVOLUTIVE SYSTEMS

2.1. **Basic idea.** Let us recall what we mean by involutive systems [6, 12, 16, 17, 18, 19, 13]. To make things easier, we consider a simple system

$$\nabla \times y + y = 0$$

Taking the divergence we see that if y is a solution, then it must also satisfy $\nabla \cdot y = 0$. This new equation is called a *differential consequence* or *integrability condition* of the initial system. Hence we have two systems:

$$(2.1) \quad \mathcal{S} \quad : \quad \nabla \times y + y = 0 \quad \quad \mathcal{S}' \quad : \quad \begin{cases} \nabla \times y + y = 0 \\ \nabla \cdot y = 0 \end{cases}$$

We say that \mathcal{S}' is the involutive form of \mathcal{S} because no more new first order differential consequences can be found. Now, \mathcal{S} is not elliptic while \mathcal{S}' is elliptic (or injective, which guarantees uniqueness of solutions). But, the completed system \mathcal{S}' is not *square*, i.e. it has more equations than unknowns. And, general numerical methods are designed for square systems. Below, we described how to transform an involutive system into square finding a compatibility operator in order to keep the augmented system injective.

2.2. **From Involutive to Augmented systems .** This section introduces the mathematical background describing how to obtain an augmented system from an involutive or completed one (where all integrability conditions have been added as in the example above).

Let us consider our problem in a general form

$$(2.2) \quad A_0 y = f$$

and let us suppose that A_0 is already in completed form. For definiteness let us also suppose that A_0 is an elliptic operator. We refer to [6, 8, 9] for more information on overdetermined elliptic operators as well as relevant boundary conditions for them.

Now since A_0 is in general overdetermined, there are typically no solutions for arbitrary f ; hence there are some *compatibility conditions* for f . These conditions are given by an operator A_1 such that $A_1 A_0 = 0$ and (2.2) has a solution only if $A_1 f = 0$. Such an operator A_1 is called the *compatibility operator*; for more technical definition we refer to [6].

Let us now introduce some function spaces V_i such that $A_i : V_i \rightarrow V_{i+1}$. It is convenient to represent these spaces and operators with help of some diagrams. Let us

consider the sequence of such operators:

$$\cdots \longrightarrow V_i \xrightarrow{A_i} V_{i+1} \xrightarrow{A_{i+1}} V_{i+2} \longrightarrow \cdots$$

Such a sequence is a *complex*, if $A_{i+1}A_i = 0$ for all i . The complex is *exact at* V_{i+1} , if $\text{image}(A_i) = \ker(A_{i+1})$. It is *exact*, if it is exact at all V_i . For example the exactness of the complexes

$$0 \longrightarrow V_A \xrightarrow{A} W_A \quad , \quad V_B \xrightarrow{B} W_B \longrightarrow 0 \quad ,$$

mean that A is injective and B is surjective.

Let us now suppose that the following complex is exact¹:

$$0 \longrightarrow V_0 \xrightarrow{A_0} V_1 \xrightarrow{A_1} V_2 \longrightarrow 0$$

This suggests that we can decompose V_1 as follows:

$$\text{image}(A_0) \oplus \text{image}(A_1^T) \simeq V_1$$

where A_1^T is the formal transpose of A_1 . Of course to be able to write equality instead of \simeq we should specify carefully the relevant vector spaces. However, this decomposition is obviously valid if V_i are finite dimensional vector spaces and A_i are linear maps. Anyway, proceeding formally, this decomposition suggests that it is indeed possible to find some functional framework such that the combined operator (A_0, A_1^T) would be bijective or Fredholm. Hence reasonable discretizations of these operators should yield a well-posed numerical problem.

So instead of trying to solve the original system (2.2) in some least square sense, we introduce an auxiliary variable z and solve

$$(2.3) \quad A_0 y + A_1^T z = f$$

We call this system the *augmented system*. This formulation is reasonable because the augmented system is square, hence standard software is readily available. Also all the relevant information about the original system is contained in the completed operator A_0 which means that the results will be reliable. The drawback is that we have introduced an extra variable z which increases the computational cost. However, we can use z in error indicator, also useful for mesh adaptation, and the work done for computing z will not be in vain.

Applying the above ideas to our example (2.1) we obtain the following augmented system:

$$(2.4) \quad \mathcal{S}'' \quad : \quad \begin{cases} \nabla \times y + y - \nabla z = 0 \\ \nabla \cdot y - z = 0 \end{cases}$$

In this case $A_1 = (\nabla \cdot, -1)$. Now we have as many equations than variables. This system is elliptic, and could be solved in a straightforward manner. On the other hand a proper discretization of the original system \mathcal{S} would be difficult because the principal

¹From our application point of view, spaces V_i are functional spaces for the solution and right-hand-side in the augmented Stokes or Navier-Stokes systems 5.2. Space V_0 (resp. V_1 and V_2) is the function space for u and p (resp. f and z). In the same way, in discrete form with any finite element discretization for variables (u, p, z) the space V_0 (resp. V_1 and V_2) will be generated by the corresponding finite element basis for u and p (resp. f and z) on the chosen mesh.

part of the operator has an infinite dimensional kernel. Hence in the numerical solution there may appear components which are approximately in this kernel; these are called spurious solutions [7].

So in this example we first pass to the completed form which eliminates the infinite dimensional kernel of the principal part, and then pass to the augmented system which provides a numerically convenient way to do the computations. Precisely the same idea applies in the general case: a completed system has better properties than the original system, and augmented system is used in the computations.

3. ROTATING FLOW IN CYLINDER

The domain of application we consider is centrifugation for which we will exhibit some non trivial analytical solution for the Stokes and Navier-Stokes systems. These are useful to validate general numerical simulation tools for such flows.

3.1. Geometry and boundary conditions. Let us now first consider finite and infinite cylinder with radius one:

$$\begin{aligned}\Omega &= \{(r, \theta, z) \mid 0 \leq r < 1\} \\ \Omega_L &= \{(r, \theta, z) \mid 0 \leq r < 1, |z| < L\}\end{aligned}$$

Their boundaries are denoted by

$$\begin{aligned}\Gamma &= \Gamma^1 \cup \Gamma^2 \quad \text{where} \\ \Gamma^1 &= \{(r, \theta, z) \mid r = 1, z > 0\} \\ \Gamma^2 &= \{(r, \theta, z) \mid r = 1, z < 0\} \quad \text{and} \\ \Gamma_L &= \Gamma_L^1 \cup \Gamma_L^2 \cup \Gamma_L^3 \cup \Gamma_L^4 \quad \text{where} \\ \Gamma_L^1 &= \{(r, \theta, z) \mid r = 1, 0 < z \leq L\} \\ \Gamma_L^2 &= \{(r, \theta, z) \mid r = 1, -L \leq z < 0\} \\ \Gamma_L^3 &= \{(r, \theta, z) \mid 0 \leq r \leq 1, z = L\} \\ \Gamma_L^4 &= \{(r, \theta, z) \mid 0 \leq r \leq 1, z = -L\}\end{aligned}$$

We consider the fluid flow in the cylinder and express the velocity field in cylindrical coordinates: $u = (u^r, u^\theta, u^z)$. Let us consider the following boundary conditions on the finite cylinder Ω_L :

$$(3.1) \quad \begin{aligned}u^r = u^z = 0 & \quad \text{on all } \Gamma_L \\ u^\theta = 1 & \quad \text{on } \Gamma_L^1 \\ u^\theta = 0 & \quad \text{on } \Gamma_L^2 \quad \text{and } \Gamma_L^4 \\ u^\theta = r & \quad \text{on } \Gamma_L^3\end{aligned}$$

In other words we rotate the upper part of the cylinder while the lower part remains fixed.

3.2. Special solutions. Now, we will use some elementary properties of Bessel functions [1] to exhibit analytical solutions for the Stokes and Navier-Stokes systems for the geometry and boundary conditions above. The Bessel functions (resp. modified Bessel functions) of order n which are bounded at origin are denoted by J_n (resp. I_n).

Writing the Stokes system in \mathbb{R}^3 in cylindrical coordinates we obtain

$$(3.2) \quad \begin{cases} -u_{rr}^r - \frac{1}{r^2} u_{\theta\theta}^r - u_{zz}^r - \frac{1}{r} u_r^r + \frac{2}{r^2} u_\theta^r + \frac{1}{r^2} u^r + p_r = 0 \\ -u_{rr}^\theta - \frac{1}{r^2} u_{\theta\theta}^\theta - u_{zz}^\theta - \frac{2}{r^2} u_\theta^r - \frac{1}{r} u_r^\theta + \frac{1}{r^2} u^\theta + \frac{1}{r} p_\theta = 0 \\ -u_{rr}^z - \frac{1}{r^2} u_{\theta\theta}^z - u_{zz}^z - \frac{1}{r} u_r^z + p_z = 0 \\ u_r^r + \frac{1}{r} u_\theta^\theta + u_z^z + \frac{1}{r} u^r = 0 \\ p_{rr} + \frac{1}{r^2} p_{\theta\theta} + p_{zz} + \frac{1}{r} p_r = 0 \end{cases}$$

Looking for some solutions with separation of variables we find two cases.

3.2.1. first case. We consider first the case where only the u^θ component is nontrivial, and assume that this component does not depend on θ . Hence if we put $u^\theta = b_r(r)b_z(z)$ we find two families of solutions:

$$(3.3) \quad \begin{cases} u^r = 0 \\ u^\theta = (a_0 + b_0 z)r + \sum_{k=1}^{\infty} (a_k e^{\mu_k z} + b_k e^{-\mu_k z}) J_1(\mu_k r) \\ u^z = 0 \\ p = \text{constant} \end{cases}$$

$$(3.4) \quad \begin{cases} u^r = 0 \\ u^\theta = (a_0 + b_0 z)r + \sum_{k=1}^{\infty} (a_k \cos(\gamma_k z) + b_k \sin(\gamma_k z)) I_1(\gamma_k r) \\ u^z = 0 \\ p = \text{constant} \end{cases}$$

3.2.2. pressure. For the pressure we always have $\Delta p = 0$. Looking again for the solutions which do not depend on θ we get two cases

$$(3.5) \quad p = (c_1 e^{\mu z} + c_2 e^{-\mu z}) J_0(\mu r)$$

$$(3.6) \quad p = (c_1 \cos(\mu z) + c_2 \sin(\mu z)) I_0(\mu r)$$

When we allow the dependence on θ we get

$$(3.7) \quad p = (c_1 e^{\mu z} + c_2 e^{-\mu z}) (c_3 \cos(n\theta) + c_4 \sin(n\theta)) J_n(\mu r)$$

and

$$(3.8) \quad p = (c_1 \cos(\mu z) + c_2 \sin(\mu z)) (c_3 \cos(n\theta) + c_4 \sin(n\theta)) I_n(\mu r)$$

where n is an integer.

3.2.3. *2nd case.* Let us now try solutions of the form

$$u(r, z) = (a_r(r)a_z(z), 0, c_r(r)c_z(z))$$

Let the pressure be given by (3.5).

$$(3.9) \quad \begin{cases} u^r = J_1(\mu r) \left((-c_1 + c_4 - 2c_1\mu z)e^{\mu z} + (-c_2 - c_3 + 2c_2\mu z)e^{-\mu z} \right) \\ u^\theta = 0 \\ u^z = J_0(\mu r) \left((-c_1 - c_4 + 2c_1\mu z)e^{\mu z} + (c_2 - c_3 + 2c_2\mu z)e^{-\mu z} \right) \\ p = 4\mu(c_1e^{\mu z} + c_2e^{-\mu z})J_0(\mu r) \end{cases}$$

Then if the pressure is given by (3.6):

$$(3.10) \quad \begin{cases} u^r = I_1(\mu r) \left((-c_1 + c_4 - c_2\mu z) \cos(\mu z) + (-c_3 + c_1\mu z) \sin(\mu z) \right) \\ u^\theta = 0 \\ u^z = I_0(\mu r) \left((c_2 - c_3 + c_1\mu z) \cos(\mu z) + (-c_4 + c_2\mu z) \sin(\mu z) \right) \\ p = 2\mu(c_1 \cos(\mu z) + c_2 \sin(\mu z))I_0(\mu r) \end{cases}$$

Hence we have four families of solutions.

Of course linear combinations of 1st and 2nd cases are also solutions.

3.3. Application to (3.1). We can solve the problem in finite cylinder with boundary condition (3.1) by combining solutions of the form (3.3) and (3.4). Let us denote by μ_k the zeros of J_1 and let $\gamma_k = k\pi/L$. Let us consider functions

$$f_I(r, z) = a_0 r + \sum_{k=0}^{\infty} a_{2k+1} \sin(\gamma_{2k+1} z) I_1(\gamma_{2k+1} r)$$

$$f_J(r, z) = b_0 r + \sum_{k=1}^{\infty} b_k \sinh(\mu_k z) J_1(\mu_k r)$$

Now let us choose $a_0 = b_0 = 1/2$ and

$$a_{2k+1} = \frac{2}{L\gamma_{2k+1} I_1(\gamma_{2k+1})}$$

$$b_k = -\frac{1}{\mu_k \sinh(\mu_k L) J_0(\mu_k)}$$

Then one can verify that

$$(3.11) \quad u = (u^r, u^\theta, u^z) = (0, f_I + f_J, 0), \quad p = \text{constant}$$

is a solution of the problem (3.2) with boundary conditions (3.1) as shown in figures 5.1 and 5.2.

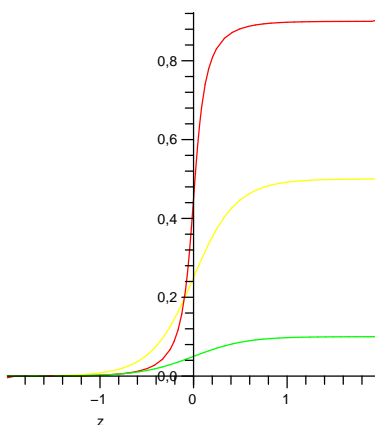


FIGURE 3.1. The solution component u^θ in (3.11) with $L = 2$ as a function of x_3 for $r = 0.1$, $r = 0.5$ and $r = 0.9$.

4. LEVEL SET

In our solver boundaries have been accounted for through a level set representation which is an established technique to represent fixed or moving interfaces on cartesian grids. Immersed boundary, fictitious domain methods as well as penalizing techniques are methods to impose boundary conditions on surfaces which are not unions of edges and faces of elements of the (non-body fitted) computational mesh [5, 14, 15].

A parametrization of a boundary Γ by the level set method is based on the zero-level curve of a function ψ :

$$\Gamma = \{x \in \Omega : \psi(x) = 0\}$$

The function ψ could be the signed Euclidean distance to Γ :

$$\psi(x) = \pm \inf_{y \in \Gamma} |x - y|$$

with the convention of a plus sign if $x \in \Omega$ and minus sign otherwise. Hence

$$(4.1) \quad \psi|_{\Gamma} = 0, \quad \psi|_{\mathbb{R}^d \setminus \Omega} < 0, \quad \psi|_{\Omega} > 0$$

The definition can be extended to open shapes by using Γ^\pm instead of Ω .

For a given shape given by (4.1) the normal to Γ is $n = \nabla\psi/|\nabla\psi|$ at $\psi = 0$.

A relaxed signed characteristic function of Ω and its set-complement is

$$(4.2) \quad \chi = \psi / (|\psi| + \varepsilon_{opt}(h))$$

where $\varepsilon_{opt}(h)$ is a “relaxation function” strictly positive tending to zero with the background mesh size h . It is defined solving minimization problems for a sampling in h :

$$\varepsilon_{opt}(h) = \operatorname{argmin}_{\varepsilon(h) > 0} \|u_h(\chi(\psi_h(\varepsilon(h)))) - \Pi_h u_{ref}\|$$

where Π_h is the restriction operator to mesh h and u_h the discrete state. The numerical results given below have been computed with $\varepsilon_{opt}(h) = ch$ for some constant $c > 0$. This choice guarantees the consistency of the scheme. The coefficient c is fitted, once for all, in order to minimize the error $\|u_{ref} - u_h^\varepsilon\|$ for a reference solution u_{ref} which can

be either a solution obtained with a body fitted mesh or, when available, an analytical solution.

Once ψ_h is known, we take into account the boundary conditions for a generic state equation $F_h(u_h) = 0$ using the equation

$$F_h(u_h)\chi(\psi_h) + F_{\Gamma_h}(u_h)\delta_{\psi_h} = 0$$

Here F_{Γ_h} is the extension of the boundary condition for F_h on Γ_h over the domain and δ_{ψ} is a relaxed Dirac measure which is constructed using $\chi(\psi_h)$ and whose support approximates the boundary.

5. INVOLUTIVE AND AUGMENTED FORMS OF NAVIER-STOKES SYSTEM

In the past we have shown how involutive completion improves numerical solution of constrained PDE systems [13]. Our aim is to show that the approach can improve existing solvers with very low implementation effort. Hence, we would like to test our formulation on an existing stabilized Navier-Stokes solver based on the Chorin projection algorithm [4] and using level set function to account for complex geometries on cartesian meshes.

Consider the time dependent Stokes system

$$(5.1) \quad \begin{cases} u_t - \Delta u + \nabla p = f \\ \nabla \cdot u = 0 \end{cases}$$

Putting $y = (u, p)$ we can write it as $A_0 y = 0$. The compatibility operator for A_0 is

$$A_1 = (\nabla \cdot, 1, \partial_t - \Delta)$$

and hence the augmented system is

$$(5.2) \quad A_0 y + A_1^T z = \begin{cases} u_t - \Delta u + \nabla p - \nabla z = 0 \\ -\Delta p + z = 0 \\ z_t - \nabla \cdot u - \Delta z = 0 \end{cases}$$

Semidiscretizing in time we see that at each step we have to solve

$$(5.3) \quad A_0 y^n + A_1^T z^n = \begin{cases} \alpha u^n - \Delta u^n + \nabla p^n - \nabla z^n = \alpha u^{n-1} \\ -\Delta p^n + z^n = 0 \\ -\nabla \cdot u^n + (\alpha I - \Delta) z^n = 0 \end{cases}$$

Here $\alpha = 1/\delta t$ (δt is the time step). The involutive approach can be extended to Navier-Stokes equations using the method of characteristics. Let us introduce the Lagrangian derivative of the velocity: $Du/Dt = \partial u/\partial t + u\nabla u$ and denote the characteristic vector field by \mathcal{X} . Then we semidiscretize along the characteristics by

$$\frac{Du}{Dt}(x) = \alpha(u^n(x) - u^{n-1}(\mathcal{X}^{n-1}(x)))$$

where we consider the characteristic issued from x , i.e. the solution of the ODE below with boundary conditions at the end of the interval:

$$\frac{d}{d\tau} \mathcal{X}(\tau) = u^{n-1}(\tau, \mathcal{X}(\tau)), \quad \mathcal{X}(t^n) = x, \quad \tau \in [t^{n-1}, t^n],$$

and call $\mathcal{X}^{n-1}(x) = \mathcal{X}(t^{n-1})$. The augmented system becomes then

$$(5.4) \quad A_0 y + A_1^T z = \begin{cases} \frac{Du}{Dt} - \frac{1}{\text{Re}} \Delta u + \nabla p - \nabla z = 0 \\ -\Delta p + z = -\nabla \cdot \left(\frac{Du}{Dt} \right) \\ -\nabla \cdot u + \left(\frac{D}{Dt} - \frac{1}{\text{Re}} \Delta \right) z = 0 \end{cases}$$

One sees that the approach does not remove the need for upwinding as a total derivative formulation needs to be considered. In practice, any upwinding technique can be applied.

5.1. Numerical implementation. The Navier-Stokes solver we consider is based on the classical widely used Chorin projection iterations [4] (u_h^n, p_h^n given):

$$(5.5) \quad \begin{cases} \frac{D_h u_h^n}{Dt} - \frac{1}{\text{Re}} \Delta_h u_h^n = 0 & \rightarrow & u_h^{n+1/2} \\ -\Delta_h p_h^{n+1} + \frac{1}{\delta t} \nabla_h \cdot u_h^{n+1/2} = 0 & \rightarrow & p_h^{n+1} \\ u_h^{n+1} = u_h^{n+1/2} - \delta t \nabla_h p_h^{n+1} & \rightarrow & u_h^{n+1} \end{cases}$$

This algorithm treats separately the pressure and the advection/diffusion terms in the Navier-Stokes equations. We use an intermediate velocity field $u_h^{n+1/2}$ (in general $\nabla_h \cdot u_h^{n+1/2} \neq 0$) to take into account advection and diffusion, and then use the Poisson equation for pressure to enforce $\nabla_h \cdot u_h^{n+1} = 0$.

Of course, the new augmented system 5.4 can be solved simultaneously and does not need a projection step. But, here we would like to update this solver in order to keep its structure and its other features. This is of practical importance as often solvers have special ingredients to account for various physical phenomenon (turbulence, combustion, ...) and it is interesting to be able to keep those.

u_h^n, p_h^n, z_h^n given:

$$(5.6) \quad \begin{cases} \frac{D_h u_h^n}{Dt} - \frac{1}{\text{Re}} \Delta_h u_h^n - \nabla_h p_h^n = 0 & \rightarrow & u_h^{n+1/2} \\ -\Delta_h z_h^{n+1} = -\Delta_h p_h^n + \frac{1}{\delta t} \nabla_h \cdot u_h^{n+1/2} = 0 & \rightarrow & z_h^{n+1} \\ -\Delta_h p_h^{n+1} = -\Delta_h z_h^{n+1} & \rightarrow & p_h^{n+1} \\ u_h^{n+1} = u_h^{n+1/2} - \delta t \nabla_h z_h^{n+1} & \rightarrow & u_h^{n+1} \end{cases}$$

z_h^0 is initialized as uniformly zero. The boundary conditions for z need be compatible with the target solution which is with z vanishing. Then either homogeneous Dirichlet or Neuman conditions are valid. We imposed the former on the cylinder and the later on the two extremities. In this algorithm, the projection step uses z and not p anymore. In term of programming this does not need development of new coding ingredients.

It is interesting to notice that the constant pressure solution is also a solution for the Navier-Stokes system and is suitable for the validation of such flow solvers. Figure 5.3 shows the effect of the algorithm on the results for the constant pressure solution case. One sees that the calculated solution is closer to the exact solution and this also on coarse meshes. As expected solving Navier-Stokes directly results in some problems with pressure at $z = 0$, especially near $r = 1$. While our method gives a much smoother approximation to constant pressure. Note that the above problem is essentially a scalar problem because of the special coordinate system used in the analysis. However, numerical computations ignore that information and are done in

cartesian coordinates (on cartesian meshes with boundaries expressed through level sets).

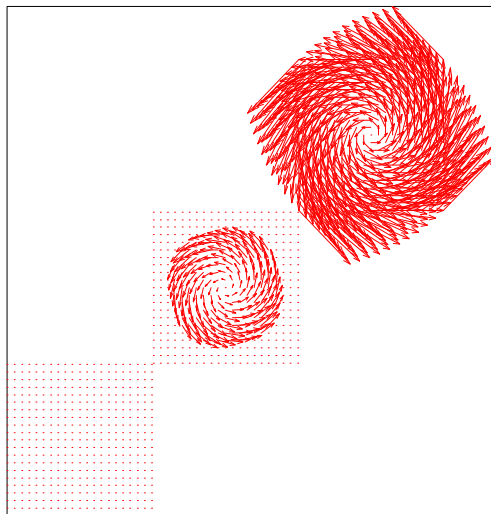


FIGURE 5.1. Analytical velocity solution of (3.2) with boundary conditions (3.1) (cross-section at $z = -L, 0, L$). Pressure is constant.

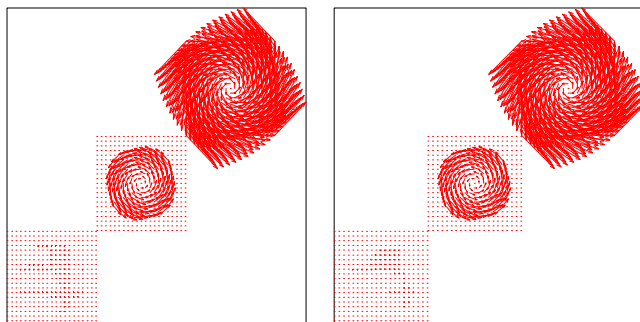


FIGURE 5.2. Velocity fields (cross-section at $z = -L, 0, L$). Left: initial Chorin 5.5, right: augmented 5.6. As expected, not much differences can be seen on the primal variable.

It is interesting to see how much could have been gained if considering the full augmented system 5.4 in discrete form instead of the update 5.6:

$u_h^n, p_h^n, z_h^n, (z_h^0 = 0)$ given:

$$(5.7) \quad \begin{cases} \frac{D_h u_h^{n+1}}{Dt} - \frac{1}{\text{Re}} \Delta_h u_h^{n+1} + \nabla_h p_h^{n+1} - \nabla_h z_h^{n+1} = 0 \\ -\Delta_h p_h^{n+1} + z_h^{n+1} + \nabla \cdot \left(\frac{D_h u_h^{n+1}}{Dt} \right) = 0 \\ \left(\frac{D_h}{Dt} - \frac{1}{\text{Re}} \Delta_h \right) z_h^{n+1} - \nabla_h \cdot u_h^{n+1} = 0 \end{cases}$$

Figure 5.4 indicates that the augmented system preserves the exact solution and over-perform the involutive upgrade which did improve the original Chorin projection prediction shown in figure 5.3. But the full implicit treatment required in 5.4 is costly.

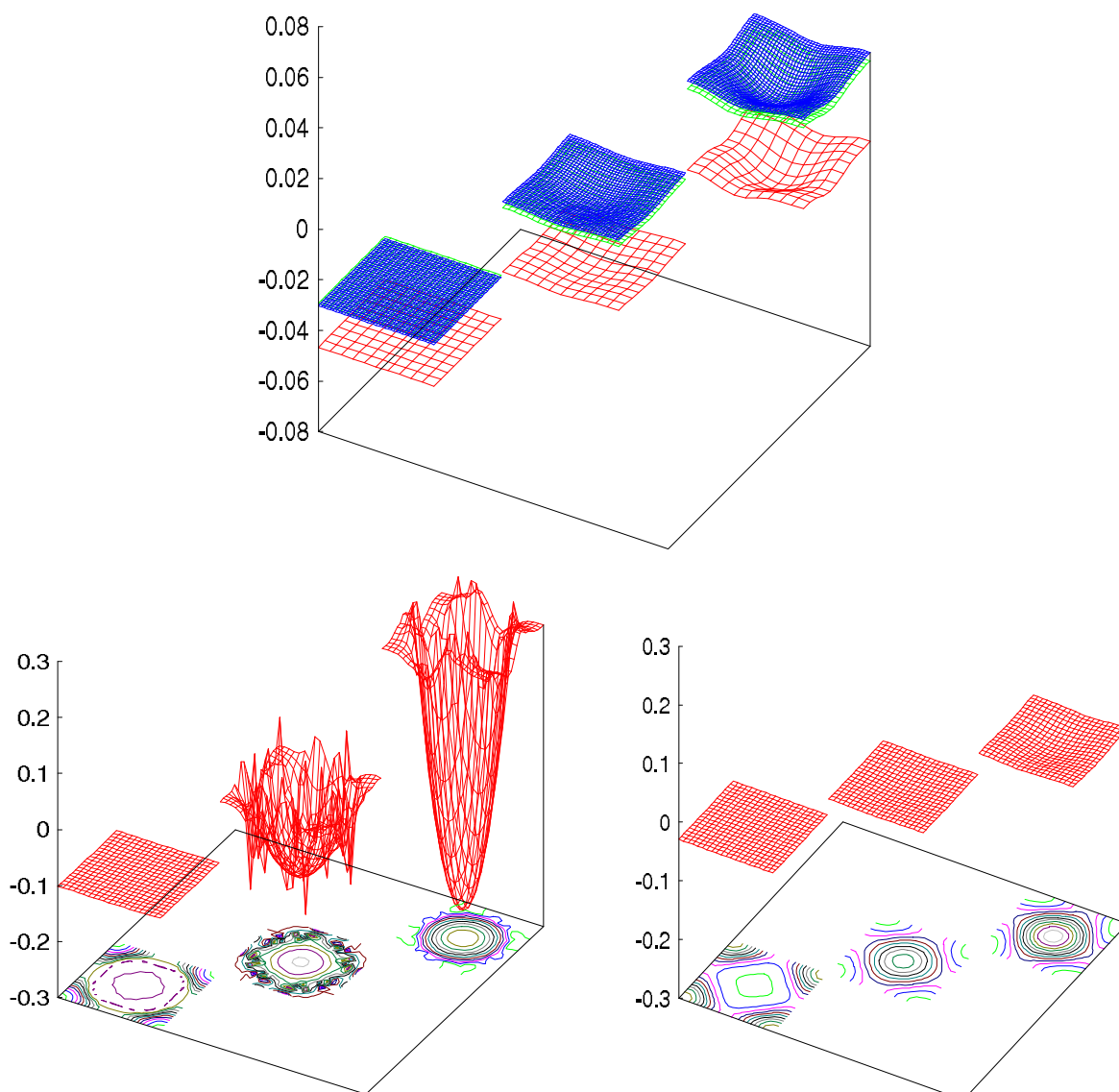


FIGURE 5.3. Impact of upgrading Navier-Stokes solvers to involutive on the pressure. Top: pressure on three meshes (11*11*21, 21*21*41, 41*41*81). Exact pressure is constant. Mesh independency is almost achieved for the intermediate mesh. Lower-left: initial Chorin 5.5, lower-right: augmented 5.6 (21*21*41 mesh).

Note that the equation for z in (5.4) is not used in (5.6). This suggests that if we set

$$\varepsilon_h^n = -\nabla_h \cdot u_h^n + \left(\frac{D_h}{Dt} - \frac{1}{\text{Re}} \Delta_h \right) z_h^n$$

then we could use ε_h^n as an error indicator. Using this indicator we can see the effect of inadequate coordinate system on the numerical solution in figure 5.5. Hence most of the error in our method is due to the treatment of boundary conditions by the level set method. This is a common situation in practice. The error would have been lower, if curvilinear elements had been used to represent the boundary, but this is rarely the case in simulation solvers.

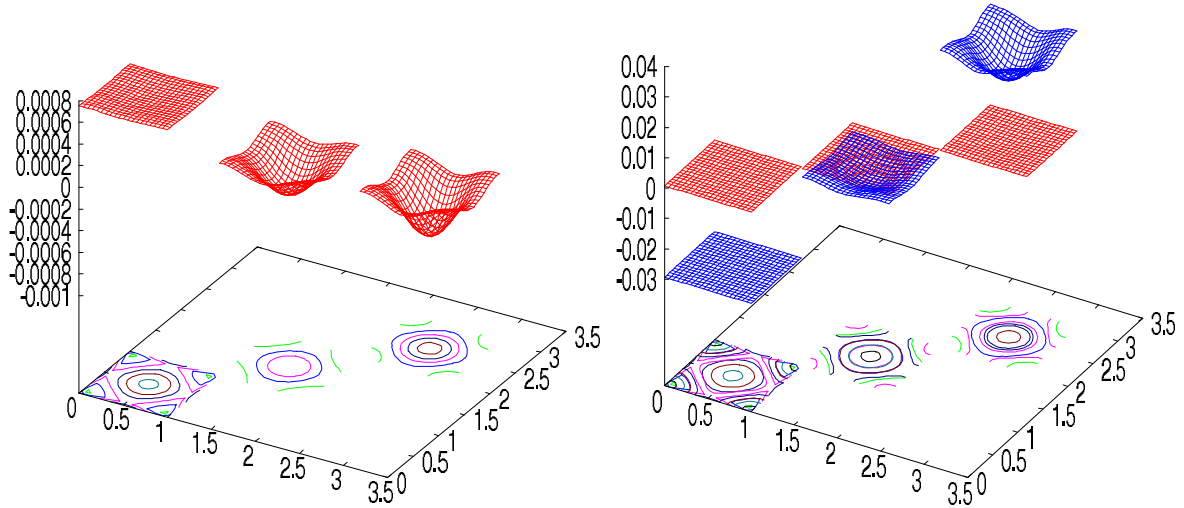


FIGURE 5.4. Left: pressure with 5.4 at $x_3 = -L, 0, L$. Right: pressure comparison with and 5.6.

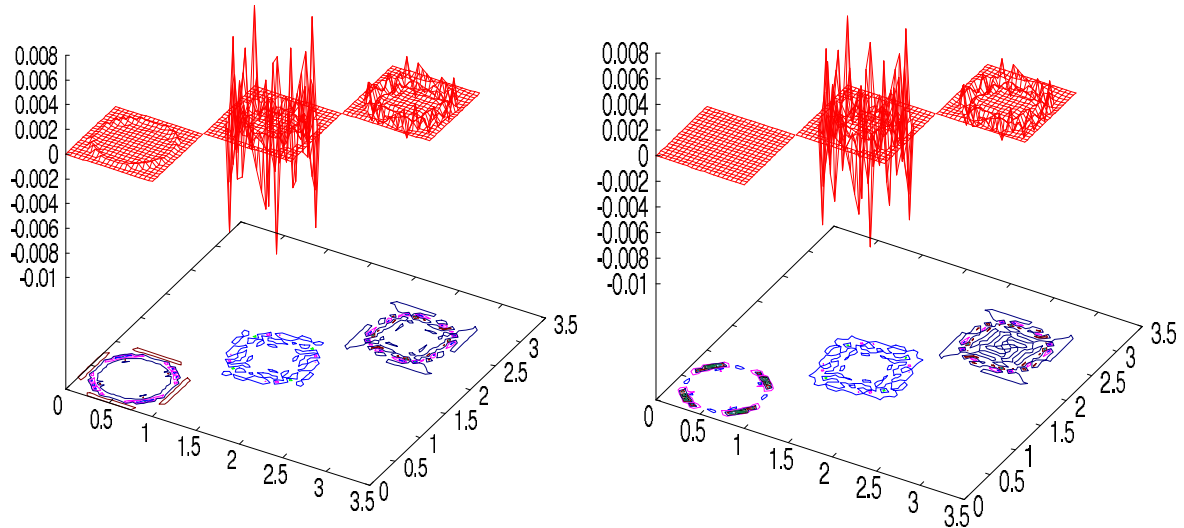


FIGURE 5.5. Left: indicator ε on the $21 \times 21 \times 41$ mesh at $x_3 = -L, 0, L$. Right: same with pressure and third velocity component set to their exact values.

Of course, another error indicator is z itself. Figure 5.6 shows iso-surfaces for z for calculations with the Chorin algorithm for the involutive system 5.6 and with the full augmented system 5.4. The results are clearly superior when considering the full system.

6. CONCLUDING REMARKS

We have applied the involutive formulation of overdetermined systems of PDEs to the numerical solution of Navier-Stokes equations in 3D. After the derivation of some analytic solutions featuring discontinuities on the boundary, these have been used to evaluate the performance of a classical Navier-Stokes solver compared to the same solver upgraded with some of the features of involutive formulation of the equations. The results show that the updated version, staying close to the original formulation

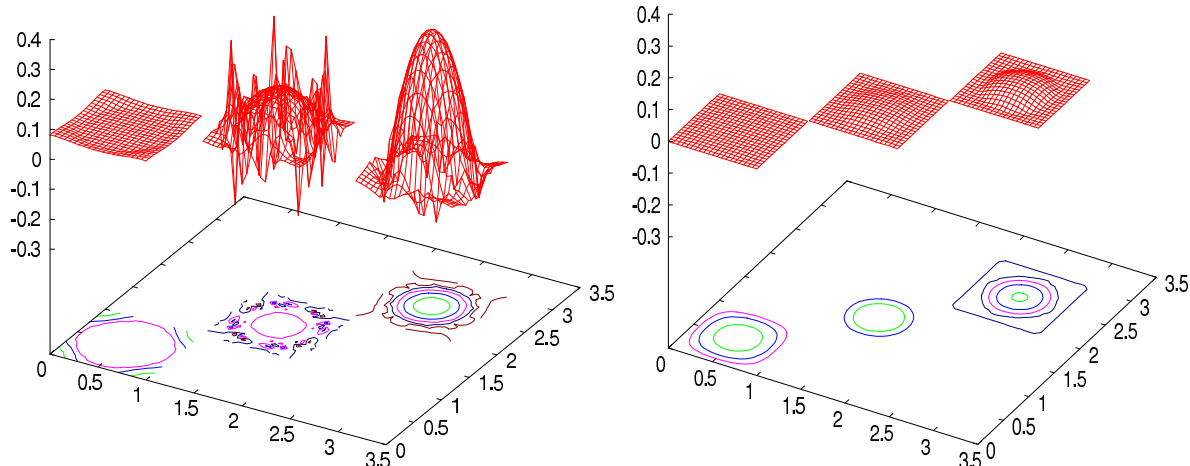


FIGURE 5.6. z on the $21 \times 21 \times 41$ mesh at $x_3 = -L, 0, L$ with system 5.6 (left) and the full augmented system 5.4.

produces superior results at virtually no extra computational cost. Finally using full involutive formulation we get still better results, but this time the extra computational cost is not negligible.

REFERENCES

- [1] G. Andrews, R. Askey, and R. Roy, *Special functions*, Encyclopedia of Mathematics and its Applications, vol. 71, Cambridge University Press, 1999.
- [2] I. Babuska, *The finite element method with lagrangian multipliers*, Numer. Math. **20-179** (1973).
- [3] F. Brezzi, *On the existence uniqueness and approximation of saddle-point problems arising from lagrangian multipliers*, Revue Francaise d'Automatique et de recherche opérationnelle **2-129** (1974).
- [4] J.A. Chorin, *A numerical method for solving incompressible viscous flow problems*, J. Comput. Phys. **2** (1967), 12–26.
- [5] A. Dervieux and F. Thomasset, *Multifluid incompressible flows by a finite element method*, Lecture Notes in Physics **11** (1981), 158–163.
- [6] P.I. Dudnikov and S.N. Samborski, *Linear overdetermined systems of partial differential equations. Initial and initial-boundary value problems*, Partial Differential Equations VIII (M.A. Shubin, ed.), Encyclopaedia of Mathematical Sciences 65, Springer-Verlag, Berlin/Heidelberg, 1996, pp. 1–86.
- [7] B. Jiang, J. Wu, and L. Povinelli, *The origin of spurious solutions in computational electromagnetics*, J. Comput. Phys. **7** (1996), 104–123.
- [8] K. Krupchyk, W. Seiler, and J. Tuomela, *Overdetermined elliptic systems*, Found. Comp. Math. **6** (2006), no. 3, 309–351.
- [9] K. Krupchyk and J. Tuomela, *Shapiro–Lopatinskij condition for elliptic boundary value problems*, LMS J. Comp. Math. **9** (2006), 287–329.
- [10] ———, *Completion of overdetermined parabolic PDEs*, J. Symb. Comput. **43** (2008), no. 3, 153–167.
- [11] O. A. Ladyshenskaya, *The mathematical theory of viscous incompressible flow*, 1969.
- [12] E. L. Mansfield, *A simple criterion for involutivity*, J. London Math. Soc. (2) **54** (1996), no. 2, 323–345.
- [13] B. Mohammadi and J. Tuomela, *Simplifying numerical solution of constrained PDE systems through involutive completion*, M2AN **39** (2005), no. 5, 909–929.
- [14] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations*, JCP **79(1)** (1988), 12–49.

- [15] C. Peskine, *The fluid dynamics of heart valves: experimental, theoretical and computational methods*, Ann. Rev. Fluid Mech. **14** (1981), 235–259.
- [16] J. F. Pommaret, *Systems of partial differential equations and Lie pseudogroups*, Mathematics and its applications, vol. 14, Gordon and Breach Science Publishers, 1978.
- [17] W.M. Seiler, *Involution — the formal theory of differential equations and its applications in computer algebra*, Springer, to appear.
- [18] D. Spencer, *Overdetermined systems of linear partial differential equations*, Bull. Am. Math. Soc **75** (1969), 179–239.
- [19] N. N. Tarkhanov, *Complexes of differential operators*, Mathematics and its Applications, vol. 340, Kluwer Academic Publishers Group, 1995.

MATHEMATICS AND MODELING INSTITUTE, MONTPELLIER UNIVERSITY, FRANCE AND DEPARTMENT OF MATHEMATICS, UNIVERSITY OF JOENSUU, P.O. BOX 111, FIN-80101 JOENSUU, FINLAND
E-mail address: `bijan.mohammadi@univ-montp2.fr` and `jukka.tuomela@joensuu.fi`