Numerical methods for the Navier-Stokes equations

Hans Petter Langtangen[1,2]

[1]Center for Biomedical Computing, Simula Research Laboratory
[2]Department of Informatics, University of Oslo

Nov 11, 2014

Note: **PRELIMINARY VERSION** (expect lots of typos!)

Contents

1 The physical and mathematical problem

Fluid flow is one of the most common physical phenomena in nature and technological devices. Examples include atmospheric flows ("weather"), global ocean currents, air flow around a car, breathing, and circulation of blood, to mention a few. The focus in the forthcoming text is on a subset of flows without turbulence, where the flow can be considered as incompressible, and where the fluid's viscosity is constant. (Actually, the model to be discussed can be used for turbulence, in principle, but the computations are very heavy.)

1.1 The Navier-Stokes equations

For incompressible flow, the key unknowns are the pressure field $p(\boldsymbol{x}, t)$ and the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$. These quantities are governed by the a momentum balance equation,

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\varrho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \tag{1}$$

and a mass balance equation

$$\nabla \cdot \boldsymbol{u} = 0 . \tag{2}$$

Equations (1) and (2) are known as *Navier-Stokes equations* for incompressible flow. The parameter ϱ is the fluid density, ν is the (kinematic) viscosity, and $\boldsymbol{f}$ denotes body forces such as gravity. Geophysical applications often need to incorporate the Coriolis and centrifugal forces in $\boldsymbol{f}$. The Navier-Stokes equations are to be solved in a spatial domain Ω for $t \in (0, T]$.

1.2 Derivation

The derivation of the Navier-Stokes equations contains some equations that are useful for alternative formulations of numerical methods, so we shall briefly recover the steps to arrive at (1) and (2). We start with the general momentum balance equation for a continuum (arising from Newton's second law of motion),

$$\varrho\frac{Du}{dt} = \nabla \cdot \boldsymbol{\sigma} + \varrho\boldsymbol{f}, \tag{3}$$

where $\boldsymbol{\sigma}$ is the stress tensor and the operator D/dt is the material derivative,

$$\frac{D\boldsymbol{u}}{dt} = \boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \tag{4}$$

here denoting acceleration. Therefore, $\varrho D\boldsymbol{u}/dt$ is density ("mass") times acceleration, while the terms on the right-hand side are the forces that induce the motion $\boldsymbol{u}$: the internal stresses $\boldsymbol{\sigma}$ and the external body forces $\boldsymbol{f}$.

The other fundamental equation for a fluid is that of mass conservation, called the continuity equation. It has the general form

$$\varrho_t + \nabla \cdot (\varrho\boldsymbol{u}) = 0, \tag{5}$$

which can be rewritten as

$$\nabla \cdot \boldsymbol{u} = \frac{1}{\varrho}\frac{D\varrho}{dt} .$$

An incompressible flow is defined as a flow where each fluid particle maintains its density. Since $\frac{D\varrho}{dt}$ is the rate of change of ϱ of a fluid particle, incompressible flow means $\frac{D\varrho}{dt} = 0$ and hence $\nabla \cdot \boldsymbol{u} = 0$. The latter is the most useful equation in a PDE system for incompressible flow since it involves the unknown velocity $\boldsymbol{u}$.

Different types of fluids will have different relations between the motion $\boldsymbol{u}$ and the internal stresses $\boldsymbol{\sigma}$. A *Newtonian* fluid has an isotropic, linear relation between $\boldsymbol{u}$ and $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T), \tag{6}$$

where $\boldsymbol{I}$ is the identity tensor, and μ is the dynamic viscosity ($\mu = \varrho\nu$). The relation (6) assumes incompressible flow. Inserting (6) in (3) gives (1) after dividing by ϱ, using $\nabla \cdot (p\boldsymbol{I}) = \nabla p$, and

calculating $\nabla \cdot (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$ as $\nabla^2 \boldsymbol{u} + \nabla(\nabla \cdot \boldsymbol{u}) = \nabla^2 \boldsymbol{u}$. The vector operations involving the nabla operator are easiest performed by using index or dyadic notation, but the derivation of the particular terms is not important for the forthcoming text.

Some numerical methods apply the $\nabla \cdot \boldsymbol{\sigma} = -\nabla p + \nabla \cdot (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$ form in (1):

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \frac{1}{\varrho}\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f}, \tag{7}$$

or

$$\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\varrho}\nabla p + \nu\nabla \cdot (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) + \boldsymbol{f}. \tag{8}$$

Other formulations add a ϱ_t term to the continuity equation, usually by assuming slight compressibility. Then $\varrho = \varrho(p)$ and we have

$$\varrho_t = \frac{\partial \varrho}{\partial p}p_t.$$

It is common to evaluate $\partial \varrho / \partial p$ for some fixed reference value ϱ_0 so that $1/c^2 = \partial \varrho / \partial p$ can be treated as a constant. The parameter c is the speed of sound in the fluid. The equation of continuity is in such cases often written as

$$p_t + c^2\nabla \cdot \boldsymbol{u} = 0, \tag{9}$$

where we have used the simplification $\nabla(\varrho \boldsymbol{u}) = \varrho_0 \nabla \boldsymbol{u}$ for a slightly incompressible fluid and divided the original equation by ϱ_0.

1.3 Boundary conditions

The incompressible Navier-Stokes equations need three scalar conditions on the velocity components or the stress vector at each point on the boundary. The boundary conditions can be classified as follows.

- Dirichlet conditions: components of $\boldsymbol{u}$ are known.

- Neumann conditions:

  – Stress condition: components of $\boldsymbol{\sigma} \cdot \boldsymbol{n}$ are prescribed.
  – Outflow or symmetry condition: $\partial \boldsymbol{u}/\partial n = 0$ (or components of this vector are zero).

We have here introduce the notion of Dirichlet and Neumann conditions using similarities with Laplace and Poisson problems (i.e., whether the condition regards the unknown itself or its derivative).

A combination of velocity and stress boundary conditions at a point is possible. For example, at a symmetry boundary we set the normal velocity to be zero.

2 The classical splitting method

The earliest and still the most widely applied numerical method for the incompressible Navier-Stokes equations is based on splitting the PDE system into simpler components for which we can apply standard discretization methods. Such a strategy is known as *operator splitting*.

2.1 A simple, naive approach

The equation (1) looks similar to a convection-diffusion equation. The simplest possible numerical method for such equations applies an explicit Forward Euler scheme in time. It is therefore tempting to advance (1) in time using a standard Forward Euler discretization:

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^n}{\Delta t} + (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n = -\frac{1}{\varrho}\nabla p^n + \nu\nabla^2 \boldsymbol{u}^n + \boldsymbol{f}^n, \tag{10}$$

which yields an explicit formula for $\boldsymbol{u}^{n+1}$:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2 \boldsymbol{u}^n + \Delta t \boldsymbol{f}^n. \tag{11}$$

There are two fundamental problems with this method:

- the new $\boldsymbol{u}^{n+1}$ will in general not satisfy (2), i.e., $\nabla \cdot \boldsymbol{u}^{n+1} \neq 0$,

- there is no strategy for computing p^{n+1}.

We may say that the incompressible Navier-Stokes equations are difficult to solve numerically because of the incompressibility constraint $\nabla \cdot \boldsymbol{u} = 0$ and the pressure term ∇p.

2.2 A working scheme

Intuitively speaking, the fulfillment $\nabla \cdot \boldsymbol{u}^{n+1}$ requires us to have "more unknowns to play with" when advancing (1). The idea is to basically use the Forward Euler scheme, but evaluate the pressure term at the new time level $n + 1$:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^{n+1} + \Delta t\,\nu\nabla^2 \boldsymbol{u}^n + \Delta t \boldsymbol{f}^n. \tag{12}$$

We must also require

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0. \tag{13}$$

The equations (12)-(13) constitute 3+1 coupled PDEs for the 3+1 unknowns $\boldsymbol{u}^{n+1}$ and p^{n+1}.

The method for solving (12)-(13) is based on a splitting idea where we first propagate the velocity from old values to some intermediate velocity $\boldsymbol{u}^*$, using (12). Then we enforce the incompressibility constraint (13) to compute a correction to $\boldsymbol{u}^*$ and also the new pressure p^{n+1}.

A plain Forward Euler discretization of (1), but with a weight β on the ∇p^n term, reads

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \beta\frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2 \boldsymbol{u}^n + \Delta t \boldsymbol{f}^n \tag{14}$$

The intermediate velocity $\boldsymbol{u}^*$ does not fulfill the incompressibility constraint (13), but we seek a correction $\delta\boldsymbol{u}$,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* + \delta\boldsymbol{u}, \tag{15}$$

such that $\nabla \cdot \boldsymbol{u}^{n+1} = 0$. Since $\delta\boldsymbol{u} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^*$, we can subtract (14) from (12) to find $\delta\boldsymbol{u}$.

$$\delta\boldsymbol{u} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^* = -\frac{\Delta t}{\varrho}\nabla\Phi. \tag{16}$$

The quantity Φ is introduced as a kind of pressure change:

$$\Phi = p^{n+1} - \beta p^n . \tag{17}$$

Inserting $\delta \boldsymbol{u}$ in the incompressibility constraint,

$$\nabla \cdot (\boldsymbol{u}^* + \delta \boldsymbol{u}) = 0,$$

or equivalently,

$$\nabla \cdot \delta \boldsymbol{u} = -\nabla \cdot \boldsymbol{u}^*,$$

results in

$$\nabla^2 \Phi = \frac{\varrho}{\Delta t} \nabla \cdot \boldsymbol{u}^*, \tag{18}$$

since $\nabla \cdot \nabla \Phi = \nabla^2 \Phi$.

As soon as Φ is computed from the Poisson equation (18), we can calculate

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \frac{\Delta t}{\varrho} \nabla \Phi, \tag{19}$$

and

$$p^{n+1} = \Phi + \beta p^n . \tag{20}$$

The solution algorithm at a time level then consists of the following steps:

1. Compute the intermediate velocity $\boldsymbol{u}^*$ from (14).

2. Solve the Poisson equation (18) for Φ.

3. Update the velocity from (19).

4. Update the pressure from (20).

Remarks. The literature is full of papers and books with methods equivalent or almost equivalent to the scheme above. Many schemes apply $\beta = 0$ and replace Φ by p^{n+1}.

2.3   Boundary conditions

What boundary conditions should we assign to $\boldsymbol{u}^*$ when solving (14)? A standard choice is to apply the same boundary conditions as those specified for $\boldsymbol{u}$. It follows that $\delta \boldsymbol{u} = 0$ on the boundary where $\boldsymbol{u}$ is subject to Dirichlet conditions. We let $\partial \Omega_{D,u}$ denote the part of the boundary $\partial \Omega$ with Dirichlet conditions, while $\partial \Omega_{N,u}$ denotes the boundary where Neumann conditions involving $\partial \boldsymbol{u}/\partial n$ apply.

The boundary condition on the pressure in the original incompressible Navier-Stokes equations is simply to prescribe p at a single point, potentially as a function of time. However, when solving the Poisson equation (18) we need Dirichlet or Neumann boundary conditions for Φ (the pressure change) on the whole boundary. Sometimes the pressure is prescribed at an inlet or outlet boundary, which then yields a Dirichlet condition for $\Phi = p^{n+1} - \beta p^n$. At the boundaries where $\boldsymbol{u}$ is subject to Dirichlet conditions, $\boldsymbol{u}^*$ has the same conditions, and $\delta \boldsymbol{u} = 0$, which implies $\nabla \Phi = 0$. In particular, $\partial \Phi/\partial n = 0$, and homogeneous Neumann conditions are therefore used on such boundaries when solving the Poisson equation for Φ. Also, at symmetry boundaries,

$\partial \Phi/\partial n = 0$. At an inlet boundary, a pressure gradient in the flow direction is often known, say as $f(t)$, implying that we can compute $\partial \Phi/\partial n = -(f(t^{n+1}) - \beta f(t^n))$.

We let $\partial \Omega_{D,\Phi}$ be the part of the boundary where Φ is subject to Dirichlet conditions, while $\partial \Omega_{N,p}$ is the remaining part where Neumann conditions involving $\partial \Phi/\partial n$ are assigned.

2.4   Spatial discretization by the finite element method

The equations to be solved, (14), (18), (19), and (20), are of two types: explicit updates (approximations a la $u = f$) and the Poisson equation. We introduce a vector test function $\boldsymbol{v}^{(u)} \in V^{(u)}$ for the vector equations (14) and (19), and a scalar test function $v^{(\Phi)} \in V^{(\Phi)}$ for the Poisson equation and the update (20). Modulo nonzero Dirichlet conditions, we seek $\boldsymbol{u}^*, \boldsymbol{u}^{n+1} \in V^{(u)}$ and $p^{n+1} \in V^{(\Phi)}$.

The variational form of a vector equation like (14) is derived by taking the inner product of the equation and $\boldsymbol{v}^{(u)}$. The Laplace term is integrated by parts, as usual, but this time vectors are involved. The relevant rule takes the form

$$\int_\Omega (\nabla^2 \boldsymbol{u}) \cdot \boldsymbol{v} \, \mathrm{d}x = -\int_\Omega \nabla \boldsymbol{u} : \nabla v \, \mathrm{d}x + \int_{\partial \Omega} \frac{\partial \boldsymbol{u}}{\partial n} \cdot \boldsymbol{v} \, \mathrm{d}s,$$

where $\nabla \boldsymbol{u} : \nabla \boldsymbol{v}$ means the *inner tensor product*: $\boldsymbol{A} : \boldsymbol{B} = \sum_j \sum_j A_{ij} B_{ij}$ (when $\boldsymbol{A}$ has elements $A_{i,j}$ and $\boldsymbol{B}$ has elements $B_{i,j}$. Alternatively, we may say that $\boldsymbol{A} : \boldsymbol{B}$ is simply the scalar product of the tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ when these are viewed as vectors (of length 9 instead of tensors of dimension 3×3 in 3D problems). The normal derivative has the usual definition: $\partial \boldsymbol{u}/\partial n = \boldsymbol{n} \cdot \nabla \boldsymbol{u}$.

The $\int_\Omega \nabla p^n \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}x$ integral can also be a candidate for integrated by parts, if desired. The relevant rule reads

$$\int_\Omega \nabla p \cdot \boldsymbol{v} \, \mathrm{d}x = -\int_\Omega p \nabla \cdot \boldsymbol{v} \, \mathrm{d}x + \int_{\partial \Omega} p \boldsymbol{n} \cdot \boldsymbol{v} \, \mathrm{d}s .$$

We use such an integration by parts below. The advantage is that we get a boundary integral involving $p \boldsymbol{n}$, which is advantageous if we want to set a condition on p, especially at an outflow boundary, but also on an inflow boundary.

For notational simplicity and close correspondence to computer code, we introduce the subscript 1 on quantities from the previous time level n and drop any superscript $n + 1$ for quantities to be computed at the new time level. The resulting variational form can be written as

$$\int_\Omega \left(\boldsymbol{u}^* \cdot \boldsymbol{v}^{(u)} + \Delta t((\boldsymbol{u}_1 \cdot \nabla) \nabla \boldsymbol{u}_1) \cdot \boldsymbol{v}^{(u)} - \frac{\Delta t}{\varrho} p \nabla \cdot \boldsymbol{v}^{(u)} + \right.$$
$$\Delta t \, \nu \nabla \boldsymbol{u}_1 : \nabla \boldsymbol{v}^{(u)} - \Delta t f_1 \Big) \mathrm{d}x = \int_{\partial \Omega_{N,u}} \left(\nu \frac{\partial \boldsymbol{u}}{\partial n} - p \boldsymbol{n} \right) \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}s, \tag{21}$$

$\forall \boldsymbol{v}^{(u)} \in V^{(u)}$. The variational form corresponding to the Poisson equation becomes

$$\int_\Omega \nabla \Phi \cdot \nabla v^{(\Phi)} \, \mathrm{d}x = -\frac{\varrho}{\Delta t} \int_\Omega \nabla \cdot \boldsymbol{u}^* v^{(\Phi)} \, \mathrm{d}x + \int_{\partial \Omega_{N,p}} \frac{\partial \Phi}{\partial n} v^{(\Phi)} \, \mathrm{d}s, \quad \forall v^{(\Phi)} \in V^{(\Phi)} . \tag{22}$$

The variational form for the velocity update (19) is based on taking the inner product of $\boldsymbol{v}^{(u)}$ and (19):

$$\int_\Omega \boldsymbol{u} \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}x = \int_\Omega (\boldsymbol{u}^* - \frac{\Delta t}{\varrho} \nabla \Phi) \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}x, \quad \forall \boldsymbol{v}^{(u)} \in V^{(u)} . \tag{23}$$

Note that this is the same form as in a vector approximation problem: approximate a given vector field $\boldsymbol{f}$ by a $\boldsymbol{u}$, where the components of $\boldsymbol{u}$ are scalar finite element functions. Also note that $\boldsymbol{u}$ in (23) is actually $\boldsymbol{u}^{n+1}$, but that the superscript is dropped since we do not use that in an implementation.

The pressure update has the variational form

$$\int_\Omega p v^{(\Phi)}\,\mathrm{d}x = \int_\Omega (\Phi + \beta p_1) v^{(\Phi)}\,\mathrm{d}x, \quad \forall v^{(\Phi)} \in V^{(\Phi)}\,. \tag{24}$$

(Also here, p denotes p^{n+1} and p_1 is p^n.)

The splitting method presented above allows flexible choices of elements for $\boldsymbol{u}$ and p. In the early days of the finite element method for incompressible flow, fully implicit formulations were used and these require the $\boldsymbol{u}$ element to be of one polynomial degree higher than the p element. This restriction does not apply to the splitting scheme, so one may, e.g., choose P1 elements for the velocity components and the pressure.

Remark on boundary integrals. The boundary integral in (21) comes into play at element faces on the boundary if the nodes on a face are not subject to Dirichlet conditions. As for scalar PDEs, Dirichlet conditions either mean that $\boldsymbol{v} = 0$ on that part of the boundary, or the element matrix and vector (or the global coefficient matrix and right-hand side) are manipulated to enforce known values of the unknown such that any boundary integral is erased and replaced by the boundary value.

The boundary integral most often applies to inflow and outflow boundaries $x = \mathrm{const}$ where we assume unidirectional flow, $\boldsymbol{u} = u\boldsymbol{i}$. Because of $\nabla \cdot \boldsymbol{u} = 0$ we have $\partial \boldsymbol{u}/\partial x = \partial \boldsymbol{u}/\partial n = 0$ and $p = \mathrm{const}$. Very often, the boundary integral in (21) is zero, because we apply it to an outflow boundary where we have $\partial \boldsymbol{u}/\partial n = 0$ and then we fix the pressure at $p = 0$. Note that in the original Navier-Stokes equations, p enters just through ∇p so a boundary condition at one point is necessary to uniquely determine p (otherwise p is known up to a free additive constant). At inflow boundaries, $\boldsymbol{u}$ is either known, which implies that the boundary integral does not apply, or we have $\partial \boldsymbol{u}/\partial n = 0$ and $p = p_0$. In this latter case, the boundary integral involves an integration of $p\boldsymbol{n} \cdot \boldsymbol{v}$.

The boundary integral involving $\partial \Phi/\partial n$ is usually omitted since we apply the condition $\partial \Phi/\partial n = 0$, see Section 2.3.

2.5   Stress formulations

As mentioned in Section 1.2, we may exchange the $\nu \nabla^2 \boldsymbol{u}$ term in (1) with a stress term $\varrho^{-1}\nabla \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ is given by (6). Occasionally, the $\nabla \cdot \boldsymbol{\sigma}$ term is advantageous, because integration by parts of $\int_\Omega \nabla \cdot \boldsymbol{\sigma} \cdot \boldsymbol{v}^{(u)}\,\mathrm{d}x$ gives a boundary integral with the stress vector $\boldsymbol{\sigma} \cdot \boldsymbol{n}$. This is convenient when boundary conditions are formulated in terms of stresses.

2.6   Increasing the implicitness

The explicit scheme (14) resembles the same stability problems as when a Forward Euler scheme is applied to the diffusion equation. However, there is also a convection term $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$ that reduces the time step restrictions. The stability criterion reads

$$\Delta t \le \frac{h^2}{2\nu + Uh}, \tag{25}$$

where h is the minimum element size and U is a characteristic size of the velocity. The term 2ν stems from the viscous (Laplace) term while Uh arises from the convection term in the Navier-Stokes equations. Which of the term that dominates in the denominator therefore depends on whether viscous forces or convection is important in the equation.

Treating the viscosity term $\nu \nabla^2 \boldsymbol{u}$ implicitly helps greatly on the stability properties of the scheme for $\boldsymbol{u}^*$. We may, for example, apply a Backward Euler scheme. Instead of (12) we then have

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^{n+1} \cdot \nabla)\boldsymbol{u}^{n+1} - \frac{\Delta t}{\varrho}\nabla p^{n+1} + \Delta t\,\nu \nabla^2 \boldsymbol{u}^{n+1} + \Delta t \boldsymbol{f}^{n+1}, \tag{26}$$

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0\,. \tag{27}$$

An intermediate velocity can be computed from the first equation if we replace p^{n+1} by βp^n as done earlier:

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^* \cdot \nabla)\boldsymbol{u}^* - \beta\frac{\Delta t}{\varrho}p^{n+1} + \Delta t\,\nu \nabla^2 \boldsymbol{u}^* + \Delta t \boldsymbol{f}^{n+1}$$

To simplify the nonlinearity in $(\boldsymbol{u}^* \cdot \nabla)\boldsymbol{u}^*$ we may use an old value in the convective velocity:

$$(\boldsymbol{u}^* \cdot \nabla)\boldsymbol{u}^* \approx (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^*\,. \tag{28}$$

This approximation is essentially one Picard iteration using u^n as initial guess.

The intermediate velocity $\boldsymbol{u}^*$ is now governed by a linear problem

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^* - \beta\frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu \nabla^2 \boldsymbol{u}^* + \Delta t \boldsymbol{f}^{n+1}$$

The correction $\delta \boldsymbol{u} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^*$ becomes

$$\delta \boldsymbol{u} = \Delta t((\boldsymbol{u}^{n+1} \cdot \nabla)\boldsymbol{u}^{n+1} - (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^*) - \frac{\Delta t}{\varrho}\nabla \Phi + \Delta t\,\nu(\nabla^2(\boldsymbol{u}^{n+1} - \boldsymbol{u}^*))\,.$$

Under the assumption that $\boldsymbol{u}^*$ is close to $\boldsymbol{u}^{n+1}$, we may drop the terms involving $\boldsymbol{u}^{n+1} - \boldsymbol{u}^*$ and just keep the $\nabla \Phi$ term. Then

$$\delta \boldsymbol{u} = \boldsymbol{u}^{n+1} - \boldsymbol{u}^* = -\frac{\Delta t}{\varrho}\nabla \Phi,$$

as before, and the incompressibility constraint $\nabla \cdot \delta \boldsymbol{u} = -\nabla \cdot \boldsymbol{u}^*$ gives $\nabla^2 \Phi = \frac{\varrho}{\Delta t}\nabla \cdot \boldsymbol{u}^*$.

The algorithm becomes the same as for a Forward Euler discretization, except that (14) is replaced by (2.6).

3   Methods based on slight compressibility

By allowing a slight compressibility we can replace the problematic constraint $\nabla \cdot \boldsymbol{u}$ by an evolution equation (9) for p. Essentially, we then have two evolution equations for $\boldsymbol{u}$ and p:

$$\boldsymbol{u}_t = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{1}{\varrho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \tag{29}$$

$$p_t = -c^2 \nabla \cdot \boldsymbol{u}\,. \tag{30}$$

The simplest method is a Forward Euler scheme:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n - \Delta t(\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n - \frac{\Delta t}{\varrho}\nabla p^n + \Delta t\,\nu\nabla^2\boldsymbol{u}^n + \Delta t \boldsymbol{f}^n, \tag{31}$$

$$p^{n+1} = p^n - \Delta t c^2 \nabla \cdot \boldsymbol{u}^n\,. \tag{32}$$

The major problem with this scheme is the stability constraint, which is dictated by the c parameter (velocity of sound): $\Delta t \sim 1/c$. Usually, c is taken as a tuning parameter and values much less than the speed of sound may give solutions with acceptable compressibility.

Any other explicit scheme, say a 2nd- or 4th-order Runge-Kutta method, is easily applied. Implicit schemes are of course also possible, but then one has to solve linear systems, and the original formulation with a true incompressibility constraint $\nabla \cdot \boldsymbol{u} = 0$ is not more complicated and usually preferred. In general, the method based on slight compressibility and explicit time integration becomes computationally very heavy and is not competitive unless one can use a c value much lower than the speed of sound.

4 Fully implicit formulation

Early attempts to use the finite element method to solve the Navier-Stokes equations were based on fully implicit formulations. This is easily derived by applying a Backward Euler scheme to the system (1)-(2):

$$\frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + (\boldsymbol{u}^n \cdot \nabla)\boldsymbol{u}^n = -\frac{1}{\varrho}\nabla p^n + \nu\nabla^2\boldsymbol{u}^n + \boldsymbol{f}^n,$$

$$\nabla \cdot \boldsymbol{u}^n = 0$$

We introduce the test functions $\boldsymbol{v}^{(u)} \in V^{(u)}$ for the momentum balance equation and seek $\boldsymbol{u}^{n+1} \in V^{(u)}$ (or more precisely, the part of $\boldsymbol{u}^{n+1}$ without nonzero Dirichlet conditions). We seek $p \in V^{(p)}$ and use $v^{(p} \in V^{(p}$ as test function for the continuity equation. We may write the system of PDEs as

$$\mathcal{L}_u(\boldsymbol{u}^n, p^n, \boldsymbol{u}^{n-1}) = 0,$$

$$\nabla \cdot \boldsymbol{u}^n = 0\,.$$

A variational formulation can be based on treating the two equations separately,

$$\int_\Omega \mathcal{L}_u(\boldsymbol{u}^n, p^n, \boldsymbol{u}^{n-1}) \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}x = 0,$$

$$\int_\Omega \nabla \cdot \boldsymbol{u}^n v^{(p)} \, \mathrm{d}x = 0,$$

or we may use an inner product of the two equations $(\mathcal{L}_u, \nabla \cdot \boldsymbol{u})$ and the test vector $(\boldsymbol{v}^{(u)}, v^{(p)})$:

$$\int_\Omega \left(\mathcal{L}_u(\boldsymbol{u}^n, p^n, \boldsymbol{u}^{n-1}) \cdot \boldsymbol{v}^{(u)} + \nabla \cdot \boldsymbol{u}^n v^{(p)} \right) \mathrm{d}x = 0\,.$$

To minimize the distance between code and mathematics, we introduce new symbols: $\boldsymbol{u}$ for $\boldsymbol{u}^n$, $\boldsymbol{u}_1$ for $\boldsymbol{u}^{n-1}$, and p for p^n. Integrating the pressure and viscous terms by parts yields

$$\int_\Omega \left(\boldsymbol{u} \cdot \boldsymbol{v}^{(u)} + \Delta t((\boldsymbol{u} \cdot \nabla)\nabla\boldsymbol{u}) \cdot \boldsymbol{v}^{(u)} - \frac{\Delta t}{\varrho}p\nabla \cdot \boldsymbol{v}^{(u)} + \right.$$

$$\left. \Delta t\,\nu\nabla\boldsymbol{u} \cdot \nabla\boldsymbol{v}^{(u)} - \Delta t f\right)\mathrm{d}x + \int_{\partial\Omega_{N,u}} \left(\nu\frac{\partial \boldsymbol{u}}{\partial n} - p\boldsymbol{n}\right) \cdot \boldsymbol{v}^{(u)} \, \mathrm{d}s +$$

$$\int_\Omega \nabla \cdot \boldsymbol{u}\, v^{(p)} \, \mathrm{d}x = 0\,. \tag{33}$$

This is nothing but a coupled, nonlinear equation system for $\boldsymbol{u}$ and p. Inserting finite element expansions for $\boldsymbol{u}$ and p yields discrete equations that can be written in matrix form as

$$Mu + \Delta t C(u)u = -\frac{\Delta t}{\varrho}Lp + \nu Kp + f, \tag{34}$$

$$L^T u = 0, \tag{35}$$

where M is the usual mass matrix, but here for a vector function, u collects all coefficients for the $\boldsymbol{u}$ field, $C(u)$ is a matrix arising from the convection term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$, L is a matrix arising from the $p\nabla \cdot \boldsymbol{v}^{(u)}$ term, K is the matrix corresponding to the Laplace operator (acting on a vector), and f is a vector of the source terms arising from $\boldsymbol{f}$. The nonlinearity is typically handled by a Newton method.

The simplified system arising from dropping the time derivative and the convection term $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ can be analyzed. It turns out that only certain combinations of $V^{(u)}$ and $V^{(p)}$ can guarantee a stable solution. The polynomials in $V^{(u)}$ must be (at least) one degree higher than those in $V^{(p)}$. For example, one may use P2 elements for $\boldsymbol{u}$ and P1 elements for p. This combination is known as the famous Taylor-Hood element. Numerical experimentation indicates that the same stability restriction on the combination of spaces is also important for the fully nonlinear Navier-Stokes equations when solved by a fully implicit method. The splitting into simpler systems, as shown in Section 2.2, introduces further approximations that stabilize the problem such that the same type of element can be used for velocity components and pressure.

The splitting method is much more widely used than the fully implicit formulation. Although the latter is more robust and much better suited for stationary flow, it is also involves much heavier computations. In each Newton iteration, a linear system involving all the coefficients in $\boldsymbol{u}$ and p must be solved, and it is non-trivial to construct efficient iterative solution methods (especially preconditioners).

5 Applications

Figure 1 exemplifies the boundary conditions for flow in a channel between two infinite plates. This flow configuration is assumed to be stationary, $\boldsymbol{u}_t = 0$, and a simple analytical solution can be found in this particular case.

Note that the numerical solution method described above requires a time-dependent problem. Stationary problems must be simulated by starting with some initial condition and letting the flow develop toward the stationary solution as $t \to \infty$.

The velocity field in channel flow is symmetric with respect to the center line. It is therefore sufficient to calculate the flow in half the channel. Figure 2 displays the computational domain and the relevant boundary conditions.

Figure 3 depicts a more complex flow geometry, leading to a more complex velocity field. The boundary conditions are, however, similar to those for channel flow.
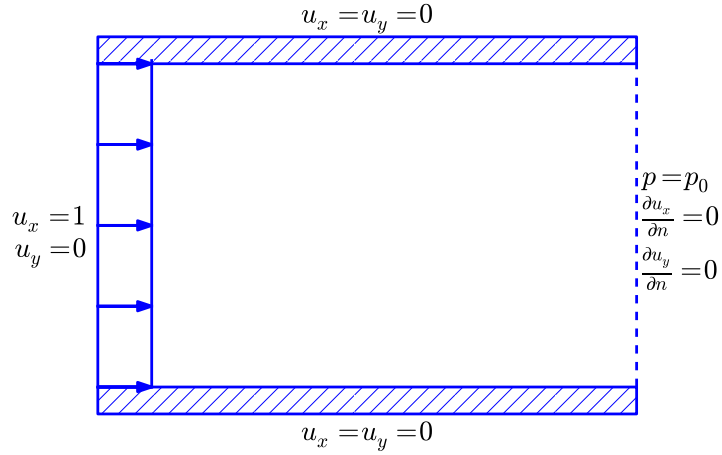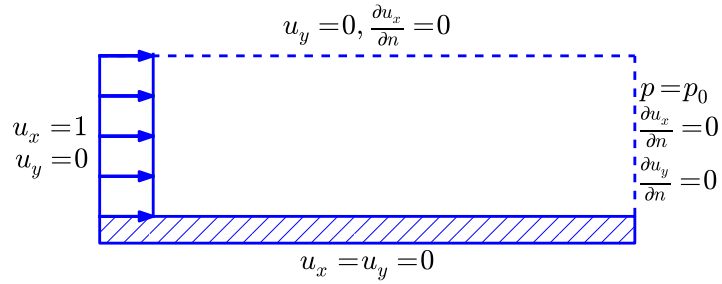
Figure 1:   Flow in a channel.

Figure 2:   Flow in a half a channel with a symmetry line.

The boundary conditions in Figure 4 are not listed in the figure because there are multiple options. The inflow boundary must have a prescribed velocity $\boldsymbol{u}$, and on the cylinder we must have $\boldsymbol{u} = 0$. On the remaining three boundaries we have some freedom in what to assign. At the outflow one typically sets $\partial\boldsymbol{u}/\partial n = 0$ and fix the pressure at one value. Alternatively, one may apply a stress-free condition $\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0$, which implicitly also sets the pressure. On the boundaries AB and DC there is more freedom. The weakest condition is $\partial\boldsymbol{u}/\partial n = 0$, assuming that the boundary is far enough away from the cylinder such that the flow field changes very little. Some prefer to set $\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0$ here instead. A stronger condition is to require $u_y = 0$ and $\partial u_x/\partial n = 0$. However, $u_y = 0$ requires the boundary to be far away from the cylinder.
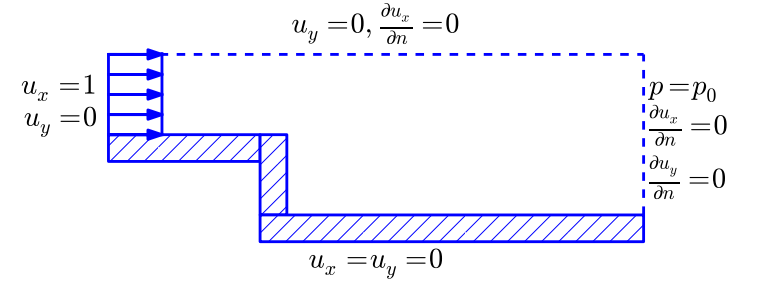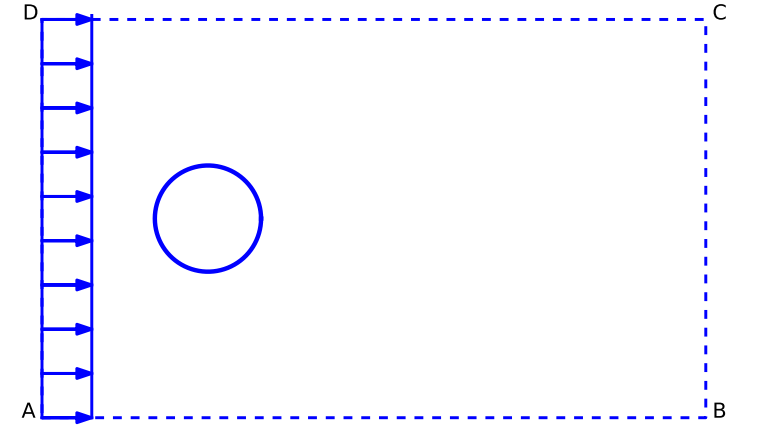
Figure 3:   Flow over a backward facing step.

Figure 4:   Flow around a cylinder.

Index